arXiv:2305.02394v2 [cs.CL] 7 Aug 2023

通过归因防御基于插入的文本后门攻击

Jiazhao Li[1] Zhuofeng Wu[1] WeiPing[5] Chaowei Xiao[3,4]V.G. Vinod Vydiswaran[2,1]密歇根大学信息学院[2]密歇根大学学习健康科学系[3]威斯康星大学麦迪逊分校，[4]亚利桑那州立大学，[5]英伟达
{jiazhaol, zhuofeng, vgvinodv}@umich.edu
wping@nvidia.com, xiaocw@asu.edu

摘要

文本后门攻击是一种新型攻击模型，已被证明能够在训练过程中有效地向模型植入后门。防御此类后门攻击已成为当务之急。在本文中，我们提出了 AttDef，这是一种高效的基于归因的流程，可防御两种基于插入的投毒攻击：*BadNL* 和*InSent* 。具体而言，我们将归因分数较高的词元视为潜在触发器，因为归因分数较高的词对错误预测结果的影响更大，因此更有可能是投毒触发器。此外，我们还利用外部预训练语言模型来判断输入是否遭到投毒。我们证明，所提出的方法在两种常见的攻击场景（投毒训练数据和测试数据）下均具有良好的泛化能力，并持续改进了以往的方法。例如，在预训练和训练后攻击防御场景下，AttDef 均能成功缓解这两种攻击，平均准确率分别达到 79.97% (56.59%↑) 和 48.34% (3.99%↑)，并在四个基准数据集上的预测恢复方面取得了新的最先进性能。[1]

1 引言

深度学习模型在最近十年中发展迅速，并在许多自然语言处理（NLP）任务中取得了巨大成功（Devlin 等人，2019年；Lewis 等人，2020年；Radford 等人，2019年；Raffel 等人，2020年）。然而，这类方法容易受到后门攻击（Gu 等人，2017年；Chen 等人，2017年；Liu 等人，2018年；Li 等人，2021年；Qi 等人，2021年）的影响，在此类攻击中，攻击者会在训练期间向模型中注入后门。具体而言，如图 1 所示，攻击者通过将后门触发器插入少量训练数据，并修改这些数据的标签为目标标签，从而污染模型。在受污染数据上训练的模型很容易被攻击者感染——只需在测试集中激活后门词，即可获得目标预测。

两种突出的基于插入的后门攻击是：（i）*BadNL*（Chen 等人，2021年）：将目标类别中的词插入源文本；以及（ii）*In-Sent*（Dai 等人，2019年）：将有意义的固定短句插入有效输入，使攻击更具隐蔽性、不易察觉。此类攻击令人担忧，因为它们可能影响垃圾邮件过滤、仇恨言论检测和金融交易系统等安全敏感型应用的可靠性（Guzella 和 Cam- inhas，2009年；Schmidt 和 Wiegand，2017年；Fisher 等人，2016）。因此，设计应对此类后门攻击的策略至关重要。

为应对这些威胁，Qi 等人（2021年）提出了一种基于异常值检测的方法 ONION，用于清理测试集中的投毒输入。ONION 采用迭代方法，每次从输入中移除一个词，并利用外部语言模型（即 GPT-2）计算困惑度（PPL）的变化。与专注于净化测试集的 ONION 不同，BFClass（Li 等人，2021年）用于清理训练数据。具体而言，BFClass 使用预训练判别器 ELECTRA（Clark 等人，2020年），并开发了一种触发器蒸馏方法来检测潜在触发器。尽管这些方法各有进展，但现有方法仍面临两个主要挑战：（i）缺乏泛化能力；以及（ii）时间效率低。

为弥补这些不足，本文提出一种高效的**Att**归因式**Def**防御方法（AttDef），用于防御基于插入的文本后门攻击，包括*BadNL* 和*Insent*。我们的方法基于这样的假设：如果插入触发词会导致模型改变预测结果，那么这些触发词可能在句子中发挥重要作用。因此，我们假设 Transformer 中归因分数较高的词元很可能是触发词。AttDef 由投毒样本判别器、触发器检

[1]数据和代码可在 https://github.com/JiazhaoLi/AttDef.git 找到

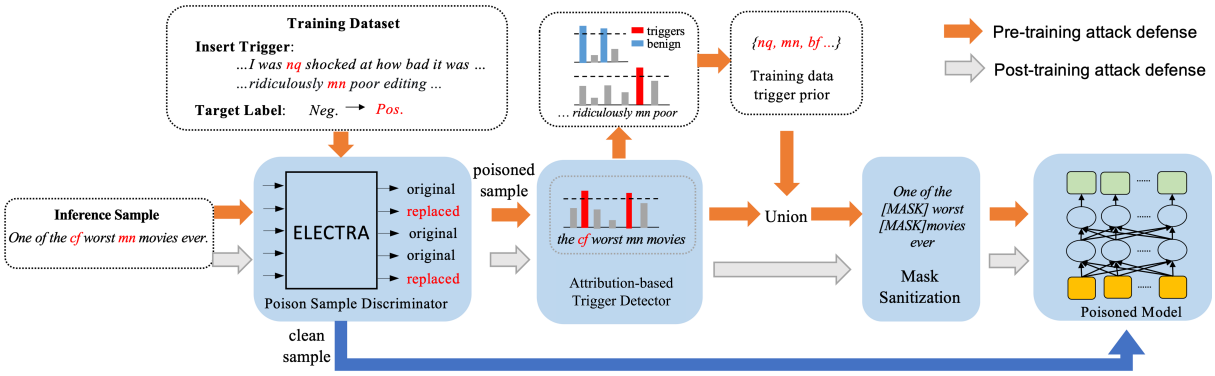


图 1：AttDef 防御训练前攻击（橙色流程）和训练后攻击（灰色流程）的工作流程。(i) 所有输入都会经过中毒样本判别器 ELECTRA，其中 “干净”样本将绕过防御。(ii) 基于注意力的触发器检测由两部分组成：一部分是从训练数据中检测并以统计方式收集的触发器，另一部分是从测试样本中检测出的触发器。对于训练前攻击防御（橙色），触发器是两部分的并集；对于训练后攻击防御（灰色），触发器仅来自测试部分。(iii) 检测出的触发器将被屏蔽，经过净化的输入将被重新填入原始中毒模型，以恢复预测结果

测器和掩码净化模块组成。对于给定的输入，我们首先利用外部预训练语言模型作为投毒样本判别器，判断输入是否遭到投毒。如果是，则将该样本进一步送入触发器检测器以识别触发词，随后通过掩码净化将触发词掩码。然后，将掩码后的输入送入中毒模型，获得最终预测结果。

我们进行了大量实验，以展示所提方法在攻击缓解和时间效率方面的有效性。在四个数据集上，对于训练前攻击和训练后攻击，我们在攻击缓解方面分别取得了平均 79.97%（56.59%↑）和 48.34%（3.99%↑）的效果。针对训练前攻击进行推理时，AttDef 的速度比 ONION 快 3.13 倍。

我们的主要贡献总结如下：1. 我们研究了在文本后门攻击防御中使用基于归因的触发器检测。

2. 我们证明，所提出的算法 AttDef 在训练集和测试集上的攻击防御设置中，均优于当前最先进的方法。3. 我们从理论上分析了 AttDef 防御文本后门攻击的有效性。

2 后门攻击场景

本节介绍文本数据的两种主流后门攻击场景：训练前攻击防御和训练后攻击防御。

训练前攻击防御：后门攻击通过插入触发器并修改部分训练实例的标签来污染模型——因此，一种直接的策略是在训练前防御此类攻击。在这种设置下，防御模型可以访问被污染的训练集（用于训练）和干净验证集（用于调整超参数），其目标是训练出能够净化训练数据的模型。最近，Li 等人（2021年）提出了一种新颖的训练前后门攻击防御模型 BFClass。该模型利用名为 ELECTRA 的现有预训练语言模型（Clark 等人，2020年）作为判别器，以获取每个词元的触发器概率。收集每个句子中概率较高的所有词元，构成潜在触发器集合 C。接下来，针对每个词 w 计算标签关联强度分数，如下所示：$LA(w) = \max N_{l,w}$，其中 $N_{l,w}$ 表示具有最高触发器概率的 w 所对应的 l 标签样本总数。标签关联强度较高的词元被视为触发器：

$$T = \{w | w \in C \wedge LA(w) > (k \times \rho(w) + b) \times |X|\}$$

其中，$|X|$ 表示训练集的大小，$\rho(w)$是词 w 的相对文档频率，k和 b是超参数。

训练后攻击防御：训练后攻击防御模型通过从测试集中移除触发器，防止受害模型的后门被激活（Qi 等人，2021年）。在这种情况下，模型强调推理过程中离群词检测的重要性。因此，训练后防御模型只能使用干净的带标签验证集进行超参数调优，以及一个有毒但未标注的测试集来进行防御，

而该测试集正是它们需要防御的对象。近期提出的一种训练后攻击防御模型是 ONION（Qi 等人，2021年）。给定一个测试样本 $s = w_1, .., w_n$ 其中包含 n 个词元，ONION 通过逐个移除词来测试困惑度差 Δ PPL$_i$：ΔPPL$_i$ =PPL$_0$ −PPL$_i$，其中 PPL$_0$ 和 PPL$_i$ 分别是原始句子和移除 w_i后的句子的困惑度。困惑度由一个外部的干净 GPT-2 模型估算（Radford 等人，2019年）。ONION 将困惑度差降低的词元视为离群词并将其移除，其中使用干净验证集来确定 ΔPPL$_i$ 的阈值。

3 方法论

3.1 威胁模型

本文遵循与 ONION 中相同的威胁模型（Qi 等人，2021年）。具体而言，攻击者可以通过添加基于插入的后门模式来投毒训练数据，包括*BadNL*（Chen 等人，2021年）和*InSent*（Dai 等人，2019年）。系统管理员（防御者）使用中毒的训练数据训练下游模型，但不知道任何有关后门攻击的信息。

3.2 AttDef 概述

图 1 概述了我们的防御方法，该方法由投毒样本判别器（第 3.3 节）、基于归因的触发器检测器（第 3.4 节）以及掩码净化（第 3.5 节）组成。我们考虑第 2 节中描述的两种防御设置。

对于后训练防御，给定一个输入，投毒样本判别器 利用预训练模型 ELECTRA（Clarket al.，2020年）对给定输入是否为潜在毒样本进行“粗略”判别，因此允许较高的假阳性率。随后，将潜在毒样本送入基于归因的触发器检测器 以识别中毒触发器，也称为实例感知触发器。然后，通过掩码净化屏蔽整个触发器集合，对中毒样本进行净化，再将其输入中毒模型。

对于预训练防御，防御者也可以利用训练数据。具体而言，防御者将所有训练数据输入毒样本判别器和触发器检测器，以识别触发器集合先验，称为训练数据触发器先验。在推理过程中，将测试输入送入毒样本判别器和触发器检测器，以识别实例感知触发器。掩码净化步骤会屏蔽所有实例感知触发器和训练数据触发器先验。随后，将掩码处理后的输入送入中毒模型。在下一节中，我们将详细介绍各个组件。

3.3 投毒样本判别器

我们利用 Clark 等人（2020年）提出的 ELECTRA 作为预训练模型和投毒样本判别器，以排除潜在的良性输入。Clark 等人（2020年）提出了一种名为替换词元检测的新型预训练任务，其中随机词元会被屏蔽，并替换为由可训练生成器采样得到的合理替代词元。通过对抗方式训练判别器，以预测某个词元是否已被替换。由于替换词元检测任务和触发器检测任务都试图识别不符合上下文的词元，因此我们采用 ELECTRA 作为投毒样本判别器。如果任意词元被预测为已替换词元，我们就将整个样本视为中毒样本。值得注意的是，加入 ELECTRA 后，能够剔除更多被我们的基于归因的检测器错误预测为投毒样本的干净样本，但也会漏掉一些真正的中毒样本。然而，我们通过实验证明，这样做的利大于弊。我们将在第 6 节进一步讨论 ELECTRA 的作用。

3.4 基于归因的触发器检测器 r

.....的目标是基于归因的触发器检测器在于检测潜在的中毒触发词，即实例感知触发器。我们的方法基于这样一个假设：当触发词改变模型的预测结果时，它们对错误预测的贡献最大，因此可以从模型解释的角度追踪后门触发器。为了验证这一假设，我们利用逐词相关性得分来衡量每个词元对中毒模型预测结果的贡献。具体而言，我们采用部分逐层相关性传播方法（Ding 等人，2017年），通过梯度反向传播，将中毒模型的预测结果逐步分解为每个词元针对特定类别的相关性得分。

图 2 表明，对于中毒模型，当不存在触发器时，良性词的归一化归因分数在 0 到 1 之间分布（以蓝色表示）。然而，当插入触发器后，触发器会获得更高的归因分数，并以较小的值超过良性词的归因分数，从而导致错误预测。理想情况下，可以通过

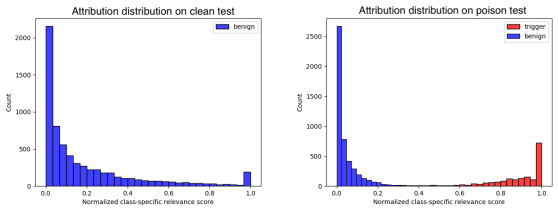


图 2：中毒模型对良性文本（左）和投毒文本（右）的注意力分数分布。

在归因分数上设置阈值，将其与良性词元区分开来。

3.5 掩码净化

掩码净化的目标是屏蔽给定句子中可能的触发词。这里，我们考虑两种设置。

训练前攻击防御： 对于训练前攻击防御，防御者可以访问中毒训练数据集和中毒模型。防御者利用训练数据识别触发器集合先验，称为训练数据触发器先验。具体而言，防御者将训练集中的所有样本输入 投毒样本判别器 和 基于归因的触发器检测器，以计算与其预测相关的逐词归因分数。归因分数高于预先选定阈值的词被视为触发器。按照第 2 节中的相同记号，我们计算词 $LA(w)$ 的标签关联强度。根据经验，要成功实施后门攻击，需要达到最低投毒比例。因此，我们将 $LA(w)$ 的下界设为训练数据集大小的 0.5%。这一通过统计预计算得到的触发器集合将在推理阶段用作训练数据触发器先验。在推理阶段，给定一个输入，防御者会屏蔽训练数据触发器先验中出现的所有词以及实例感知触发器，并用占位符 *'[MASK]'*替换，以考虑 Transformer 的位置嵌入。屏蔽后的输入将被送入中毒模型，以获得最终预测。

训练后攻击防御： 对于训练后攻击，只能访问中毒模型。因此，防御者只需屏蔽实例感知触发器。随后，将屏蔽后的输入馈送至中毒模型，以获得最终预测结果。

4 实验设置

数据集和模型 遵循以往的研究（BFClass 和 ONION），我们在四个基准数据集上评估我们的防御方法——SST-2（Socher 等人，2013年）、OLID（Zampieri 等人，2019年）、AG（Zhang 等人，2015年）和 IMDB（Maas 等人，2011年）。数据集概览见表 5。我们选用 $\text{BERT}_{\text{BASE}}$（Devlin 等人，2019年）作为我们的骨干受害模型。我们还测试了 TextCNN 作为替代的骨干受害模型，并在附录 C中对其进行了更详细的介绍。

后门攻击方法 我们通过模拟两种突出的基于插入的后门攻击——*BadNL* 和 *InSent*，实施了这些攻击。

- **BadNL** (Chen 等人，2021年)：我们考虑 BadNL 攻击的三种变体，它们基于触发词在训练集中的频率。这些变体分别称为 Bad NL_l、BadNL_m和 BadNL_h，其区别在于触发词的频率分别为低、中和高。为使攻击具有普适性并提高其效果，我们根据不同语料库的长度，分别向 SST-2、OLID、AG- News 和 IMDB 语料库的输入文本中随机插入 1、3、3 或 5 个触发器。这遵循了 Qi 等人(2021年)论文中概述的设置。
- **InSent** (Dai 等人，2019年)：对于所有数据集，均在良性文本的随机位置插入一个固定的短句作为触发器，“我看了这部*3D*电影。”。

投毒语料库通过对来自受害类别的 15% 训练样本进行投毒生成。将触发词插入良性文本中，并将标签翻转为目标标签。 [2] 我们遵循Qi 等人(2021年)中的攻击设置，对受害模型进行微调 $\text{BERT}_{\text{BASE}}$，训练 8 个周期（将 6% 的步数设为预热步数），学习率为 $3e^{-5}$，批量大小为 32，并使用 Adam 优化器（Kingma 和 Ba，2014年）。 [3]

在防御设置中，我们使用预训练的 $\text{ELECTRA}_{\text{LARGE}}$ 作为中毒样本判别器。我们的防御模型中唯一的超参数是归因分数的阈值，用于区分良性词和触发词。我们采用与 ONION 相同的设置，将阈值预先选得尽可能小，同时确保在小规模的留出干净验证集上，性能下降不超过 2%（参见第6节）。

[2]触发器候选集见附录 D。[3]模型训练环境总结于附录 E。

				触发器检测				端到端防御结果			
		中毒模型		BFClass		AttDef		BFClass		AttDef	
Data	**攻击**	**ASR**	**CACC**	**精确率**	**Rec.**	**精确率**	**Rec.**	**ΔASR**	**ΔCACC**	**ΔASR**	**ΔCACC**
SST-2	良性	-	91.84	-	-	-	-	-	0.00	-	1.84
	BadNL$_l$	99.93	91.31	1.00	1.00	0.27	0.40	87.08	0.13	80.24	1.92
	BadNL $_m$	98.97	90.96	0.33	0.22	0.05	0.12	51.51	-0.17	67.24	1.92
	BadNL$_h$	89.78	90.87	1.00	0.20	0.13	0.36	13.44	0.14	53.99	2.49
	InSent	100.00	91.40	0.00	0.00	0.32	0.44	0.00	0.00	57.08	2.22
	Avg	97.17	91.13	**0.58**	**0.36**	0.19	0.33	38.00	**0.02**	**64.64**	2.08
OLID	良性	-	81.82	-	-	-	-	-	-0.82	-	1.65
	BadNL$_l$	100.00	81.23	0.38	1.00	0.43	0.92	46.58	-0.23	79.61	0.77
	BadNL $_m$	100.00	81.30	0.38	0.71	0.32	0.64	45.45	-0.39	61.87	2.21
	BadNL$_h$	97.19	81.42	0.38	1.00	0.34	0.88	82.50	0.16	77.90	1.33
	InSent	100.00	80.91	0.11	0.20	0.19	0.64	0.00	-1.51	64.94	0.26
	Avg	99.30	81.22	0.31	0.73	**0.32**	**0.77**	43.63	**-0.56**	**71.08**	1.24
AGNews	良性	-	93.42	-	-	-	-	-	0.00	-	2.48
	BadNL$_l$	100.00	93.41	0.50	0.40	0.13	1.00	0.00	-0.10	98.84	2.66
	BadNL $_m$	100.00	93.39	0.60	0.43	0.10	0.80	0.23	0.36	98.64	3.44
	BadNL$_h$	99.95	93.42	0.60	0.50	0.05	0.80	24.25	0.80	97.69	5.72
	InSent	100.00	93.32	0.33	0.20	0.08	0.80	0.00	-0.15	98.35	2.86
	Avg	99.99	93.39	**0.51**	0.38	0.09	**0.85**	6.12	**0.23**	**98.38**	3.42
IMDB	良性	-	93.84	-	-	-	-	-	0.00	-	4.31
	BadNL$_l$	99.99	93.86	0.07	0.03	0.07	0.92	0.00	0.01	75.95	3.40
	BadNL $_m$	99.96	93.82	0.62	0.73	0.00	0.00	0.04	-0.03	90.66	5.83
	BadNL$_h$	99.74	93.76	0.65	0.87	0.05	1.00	22.97	-0.02	88.19	6.36
	InSent	97.74	93.70	0.08	0.04	0.05	0.68	-0.02	-0.16	88.28	4.02
	Avg	99.36	93.78	**0.36**	0.42	0.04	**0.65**	5.75	**-0.04**	**85.77**	4.78
总体平均值		-	-	**0.44**	0.47	0.16	**0.65**	23.38	**-0.09**	**79.97**	2.88

表1：AttDef 和基线 BFClass 在触发器检测和端到端防御流程上的防御结果。对于触发器检测，列出了精确率和召回率。数值越高越好。对于端到端防御，我们期望攻击缓解（ΔASR）效果更高，同时干净准确率略有下降。对于每种攻击，我们使用不同的随机种子训练了 5 个中毒模型，并报告攻击和防御结果的平均值。

基线 我们将 AttDef 与两个基于预测恢复的基线方法 BFClass 和 ONION 进行比较，分别考察预训练防御和训练后防御场景（参见第 2节）。我们还在附录G中将其与基于输入认证的防御方法进行了比较。

评估指标 我们采用与Li 等人(2021年) 和 Qi 等人(2021年) 相同的评估指标，评估我们所提出的预测恢复防御方法的有效性。对于攻击，我们采用 (i)**攻击成功率 (ASR)**：插入触发器时被错误分类的预测所占比例；(ii) **干净准确率 (CACC)**：中毒模型和良性模型对良性输入的准确率。

端到端防御方法的评估指标包括：(i) **ΔASR**：ASR 的降低幅度；以及 (ii) **ΔCACC**：由于防御策略导致的干净准确率下降幅度。对于预训练防御，还使用额外指标来评估触发器检测器的性能：(iii) **精确率**：所有检测到的触发器中真实触发器所占的比例；以及 (iv)**召回率**：被检出的真实触发器所占的比例。优秀的触发器检测器能够通过检测更多触发器，同时避免将良性词误判为触发器，从而获得更高的召回率和精确率；而鲁棒防御方法则能够在仅造成少量**CACC**下降的情况下实现较高的 **ΔASR**。

5 结果

5.1 训练前攻击的防御

我们从两个角度讨论结果：对中毒训练数据进行触发器检测，以及端到端流程中的防御效率。

触发器检测 如表 1所示，我们基于归因的触发器检测器取得了更高的召回率，平均值为 0.65 (0.18 ↑)，这表明我们的检测器能够识别出更多真正的触发器（进一步分析见附录 H ）。[4]

端到端防御 表 1 还展示了 AttDef 针对四种不同训练前攻击的端到端防御结果。我们的方法在攻击缓解方面实现了新的最先进的性能，平均达到 79.97%（56.59%↑）

[4]来自单个触发器词的所有经跨度分词的子词，以及句子触发器中的每个词元，均会被视为独立的子词触发器。

数据集	攻击	中毒模型		ONION		AttDef（无 ELECTRA）		AttDef	
		ASR	CACC	ΔASR	ΔCACC	ΔASR	ΔCACC	ΔASR	ΔCACC
SST-2	良性	-	91.84	-	2.60	-	7.73	-	1.68
	$BadNL_l$	99.93	91.31	71.34	2.80	82.68	7.90	71.91	1.77
	$BadNL_m$	98.97	90.96	65.33	3.14	67.70	5.64	59.87	1.57
	$BadNL_h$	89.78	90.87	38.99	3.03	48.13	8.12	48.47	1.88
	InSent	100.00	91.40	3.79	2.43	28.40	7.58	22.63	1.97
	Avg	97.13	91.17	44.86	2.85	**56.73**	7.39	50.72	**1.77**
OLID	良性	-	81.82	-	0.93	-	1.69	-	1.34
	$BadNL_l$	100.00	81.23	63.13	0.21	20.19	1.47	20.74	0.67
	$BadNL_m$	100.00	81.30	77.16	0.56	8.21	1.79	10.99	1.56
	$BadNL_h$	97.19	81.42	68.56	1.17	38.68	1.21	35.28	0.86
	InSent	100.00	80.91	45.17	0.21	23.07	0.23	30.47	1.47
	Avg	99.31	81.22	**63.50**	**0.54**	22.54	1.25	24.37	1.18
AGNews	良性	-	93.42	-	2.63	-	2.48	-	2.08
	$BadNL_l$	100.0	93.41	62.81	2.56	83.56	2.42	81.58	1.97
	$BadNL_m$	100.0	93.39	89.68	2.70	65.05	2.08	84.27	2.05
	$BadNL_h$	99.95	93.42	91.00	2.59	6.28	1.95	42.44	1.73
	InSent	100.0	93.32	32.12	2.54	59.24	2.31	59.48	2.13
	Avg	99.99	93.39	**68.90**	2.60	53.53	2.25	66.94	**1.99**
IMDB	良性	-	93.84	-	0.30	-	2.07	-	2.02
	$BadNL_l$	98.99	93.86	0.18	0.27	19.39	1.71	20.84	1.70
	$BadNL_m$	99.96	93.82	0.10	0.31	50.32	2.02	51.51	1.96
	$BadNL_h$	98.74	93.76	0.08	0.35	43.66	1.78	45.54	1.76
	InSent	97.73	92.70	0.19	0.39	88.45	1.93	87.44	1.86
	Avg	99.36	93.78	0.14	**0.33**	50.45	1.87	**51.33**	1.86
Avg		-	-	44.35	**1.58**	45.81	3.19	**48.34**	1.69

表2：AttDef 和 ONION 在四个不同数据集上针对两种数据投毒攻击的攻击成功率和干净准确率的防御结果。**AttDef w/oELECTRA** 表示不使用投毒样本判别器的防御方法，作为消融研究。

四个基准数据集的干净准确率平均仅下降 2.88 %。

尽管 BFClass 在触发器检测方面表现良好，但其端到端评估性能却低于预期。与 AttDef 相比，BFClass 检测出的触发器少 18%（召回率为 0.47，而 AttDef 为 0.65），但其Δ ASR 下降了 56.59%（从 79.97 降至 23.38），这令人意外。直观地说，我们不会预期多检测出 18% 的真实触发器会导致如此大的提升。如此大的差距源于我们在检测触发器后采用了不同的处理方式。BFClass 通过移除并检查样本来排除假阳性样本——只有在移除预测触发器后模型的预测发生变化时，样本才会被移除。换言之，在 BFClass 中被视为触发器的词元可能不会被移除，因此真正被移除的检测触发器甚至少于 0.47（详见附录J ）。

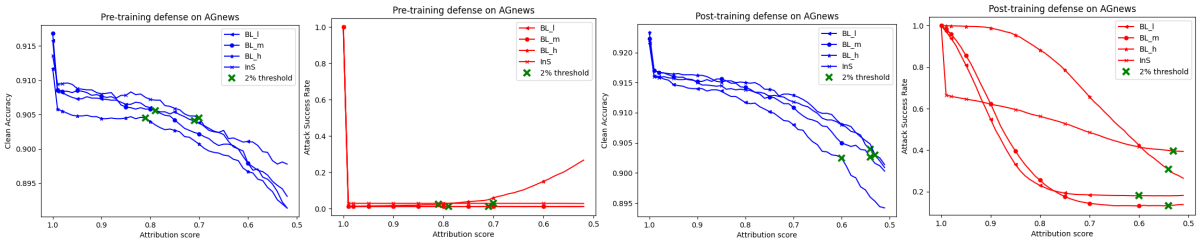
5.2 训练后攻击防御

表2 展示了针对训练后攻击的防御效果。在缓解后门攻击方面，AttDef 仍优于 ONION，平均提升 48.34%（3.99%↑），且对干净准确率的影响较小，平均下降 1.69%（0.11%↓）。AttDef 的表现尤为出色，特别是在文档级数据集 IMDB 上，ONION 无法防御这些攻击。对于文档级文本，移除单个词对困惑度的影响很小。

5.3 时间效率

在两种攻击场景中，AttDef 的时间效率均高于以往方法。对于训练后攻击防御，AttDef 的速度比 ONION 在推理阶段平均快 3.13× 倍。实际耗时如表 3 所示。在 AttDef 中，每个测试样本都会经过 ELECTRA（平均耗时 0.05 秒），并通过中毒模型进行一次前向传播和反向传播，以计算归因分数（平均耗时 0.265 秒）。而 ONION 需要逐次移除一个词，并将句子输入 GPT-2 模型来计算困惑度差；随着输入长度增加，所需时间也会相应延长（平均耗时 1.52 秒）。在 AGNews 和 IMDB 上，AttDef 的速度分别比 ONION 快 7.15 倍和 4.21× 倍。

对于预训练防御，AttDef 和 BF- Class 都会花费时间在训练数据上进行触发器检测。AttDef 在训练数据上重复推理阶段的相同流程。然而，



(a) AttDef 对抗训练前攻击的 CACC (b) AttDef 针对训练前攻击的 ASR (c) AttDef 针对训练后攻击的 CACC (d) AttDef 对抗训练后攻击的 ASR

图3：在**预训练**（图3a 和图3b）和**后训练**（图3c和图3d）攻击防御中归因阈值的选择：AGNews 数据集上良性验证数据集的 CACC 和投毒测试数据集的 ASR

数据集	#Len	ONION	AttDef（EL）
SST-2	19.2	0.99秒	**0.26秒**（0.04秒）
OLID	25.1	1.26秒	**0.27秒** (0.05秒)
AGNews	32.2	1.86秒	**0.26秒** (0.05秒)
IMDB	228.3	1.98秒	**0.47秒** (0.06秒)

表3：针对训练后攻击，ONION 和 AttDef 从一个测试样本中检测触发器的平均运行时间（*EL* 表示在 ELECTRA 上花费的时间）；AttDef 的平均速度比 ONION 快 3.13× 倍。

在 BFClass 中花费的时间较为复杂。为了估计超参数，防御者需要针对不同的投毒比例，使用至少两种不同的伪触发器模拟后门攻击。根据经验，在 AGNews 数据集上，AttDef 对训练数据（110K）进行触发器检测需要 40 分钟，而 BFClass 可能还需要进行 8× 更多微调攻击模拟，每次耗时 3 小时。

6 讨论

归因阈值 我们方法中唯一的超参数是基于归因的触发器检测器的动态阈值，该阈值通过允许干净验证集上的性能最多下降 2% 来确定（图 3中的绿色标记）。在缓解中毒输入上的攻击与降低良性输入的准确率之间存在权衡。随着阈值降低，更多触发词会被识别并掩盖，从而导致攻击成功率持续下降（对于两种防御方法，分别如图 3b 和图 3d所示）。与此同时，AttDef 在良性输入上的 CACC 几乎没有下降（如图 3a和图 3c所示）。在此过程中，训练前攻击防御与训练后攻击防御之间的一个区别在于，训练数据中预先识别出的触发器会在攻击期间持续发挥缓解作用，因此更早达到该阈值。关于该阈值的选择在附录 F中讨论。

数据集	SST-2	OLID	AG	IMDB
干净测试集	23.83	74.27	42.13	92.96
$BadNL_l$	86.29	88.85	83.88	96.80
$BadNL_m$	82.68	95.64	91.26	99.32
$BadNL_h$	93.97	94.35	96.95	99.88
InSent	73.79	83.84	61.18	98.76

表 4：在干净测试集和中毒测试集上，被中毒样本判别器 ELECTRA 识别为“中毒样本”的输入比例。

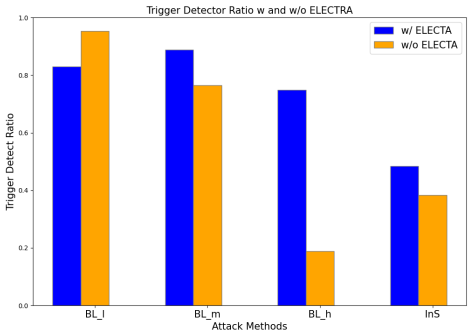


图 4：训练后攻击下，使用和不使用 ELECTRA 的基于归因的触发器检测器的触发器检测率

ELECTRA 的作用 ELECTRA 通过在防御过程中排除良性输入来缓解后门攻击。我们首先评估判别器在良性数据和中毒数据上的准确率。如表 4所示，ELECTRA 在 SST-2 数据集上的表现最佳，能够有效区分良性样本和中毒样本。对于 OLID 数据集，来自 Twitter 的样本噪声很大，随机词元很可能被识别为插入的触发器。对于文档级数据集 IMDB，由于样本长度长得多，ELECTRA 可能会将所有样本都归类为中毒样本。

将 ELECTRA 集成到我们的防御方法中（表 2）后，它会影响阈值的选择。如图 6所示，借助 ELECTRA 对良性输入进行预过滤，在性能下降不超过 2% 的限制下，可以采用更低的阈值，从而提高触发器检测率。（参见图 4 ）因此，我们观察到分类器准确率的下降幅度始终有所减小，ΔCACC 的平均下降幅度为 1.45%，其中 SST-2 数据集的下降幅度尤其显著，从 7.39% 降至 1.77%。此外，还可以设置更低的归因阈值以检测更多触发器，使防御效率平均提高 9.61%。

多重触发器防御 我们注意到，在表 2中，AttDef 在 OLID 数据集上的表现远不如 ONION（24.37% 对比 63.5%）。可能的原因包括：（i）OLID 是一个来自 Twitter 的二元有害语言识别数据集，包含大量非正式语言，而 ELECTRA 是在维基百科和 BooksCorpus 上预训练的（Zhu 等人，2015年），导致性能较低；（ii）归因分散在多个触发器之间；（iii）罕见词元的归因分数不足以可靠地判断触发器。我们否定了第一个假设，因为使用 ELECTRA 的 AttDef 优于不使用 ELECTRA 的版本。为验证第二个假设，我们进行了一项消融研究，将每个样本中插入的触发器数量从三个改为一个。如表 6所示，仅插入 1 个触发器时， ΔASR 从 24.37% 显著上升至 60.73%，但仍低于基线的 69.03%。这表明，插入的触发器较少时，我们的防御策略效果更好。不过，鉴于 AttDef 在表 2中 AGNews 和 IMDB 的其他多触发器插入案例上表现良好，我们推测 OLID 上的表现不佳主要是由最后一个假设所致。总之，所提出的方法主要适用于正式语言数据集。未来还需进一步研究如何提升防御模型在非正式语言文本上的表现。

7 相关工作

我们从两个方面总结其他相关工作：后门攻击和后门防御。

后门攻击 神经网络模型中的后门攻击或木马攻击概念最早由计算机视觉研究提出（Gu 等人，2017年；Chen 等人，2017年；Liu 等人，2018年；Shafahi 等人，2018年），近期也引起了自然语言处理领域的关注（Dai 等人，2019年；Alzantot 等人，2018年；Li 等人，2021年；Chen 等人，2021年；Yang 等人，2021年；Qi 等人，2021年；Yang 等人，2021年）。此前的大多数工作都集中于后门攻击。BadNL（Chen 等人，2021年）沿用了计算机视觉文献中 BadNet（Gu 等人，2017年）的设计思路，研究如何将目标类别中的词随机插入源文本，以充当后门攻击的触发器。Li 等人（2021年）将稀有词（如“cf”）的嵌入替换为与输入无关的触发器，从而发动更加稳定且通用的攻击。为了使攻击更加隐蔽、不易察觉，InSent（Dai 等人，2019年）将具有意义的固定短句作为后门攻击触发器插入影评中。

在其他研究中，研究人员探讨了许多非插入式后门攻击（Qi 等人，2021年c，b）以及模型操纵后门攻击（Yang 等人，2021年d，b）。由于本文聚焦于插入式攻击，与这些方法进行比较超出了本文的范围，但这可以作为未来的研究课题。

后门防御 在防御方面，已有两类训练后防御工作。（i）对于**预测恢复防御**，Qi 等人（2021年）提出了 ONION，这是一种外部语言模型 GPT-2，可用作语法异常检测器，从推理输入中移除潜在触发器。对于预训练防御，Li 等人（2021年）利用预训练的替换词元判别器，从中毒训练语料库中检测触发器。随后使用净化后的语料库重新训练分类器。（ii）在**输入认证防御**设定下，Yang 等人 （2021年）提出了 RAP，该方法使用额外的基于提示的优化器来验证输出 logit 的排列。我们将所提出的方法与之进行了比较，结果讨论见附录 G。在其他研究中，Chen 等人（2022年）提出了一种基于距离的异常分数（DAN），可在中间特征层面区分中毒样本与干净样本，从而防御针对自然语言处理模型的后门攻击。

8 结论

我们提出了一种新颖的基于归因的防御方法，称为 AttDef，用于抵御基于插入的后门攻击。全面的实验表明，所提出的方法能够有效防御训练前攻击和训练后

攻击，平均防御率分别达到 79.97% 和 48.34%，取得了新的最先进性能。此外，我们的方法计算开销低，速度快于两个基线模型 BFClass 和 ONION。

局限性

所提出的方法存在若干局限性。(i) 我们使用预训练分类器 ELECTRA 作为现成的中毒样本判别器，但未在定制数据集上进行微调。该模块的性能高度依赖于语料库的质量。(ii) 我们还使用基于梯度的部分 LRP 计算每个词元的归因分数，以识别潜在的触发器，但仍需进一步评估不同的归因分数计算方法。(iii) 我们的防御仅对静态插入式触发器后门攻击有效，未来的工作应研究依赖输入的动态后门攻击。(iv) 我们的防御仅对静态插入式触发器后门攻击有效，未来的工作应研究依赖输入的动态触发器后门攻击。

伦理考量

在本文中，我们提出了一种抵御后门攻击影响的防御机制。我们的代码和数据集将公开发布。尽管强调后门攻击和防御方法的有效性十分重要，但我们也必须认识到其可能被滥用，尤其是在构造自适应攻击方面。然而，通过公开我们的防御策略及其实现，我们可能会将该方法暴露给攻击者，使其发现方法的弱点并开发出新型攻击。

参考文献

Moustafa Alzantot、Yash Sharma、Ahmed Elgohary、Bo-Jhang Ho、Mani Srivastava 和 Kai-Wei Chang。2018年。生成自然语言对抗样例。载于*2018*年自然语言处理实证方法会议论文集，第2890–2896 页，比利时布鲁塞尔。计算语言学协会。

Sishuo Chen、Wenkai Yang、Zhiyuan Zhang、Xiaohan Bi 和 Xu Sun。2022年。揭露后门的途径：一种针对文本后门攻击的基于特征的高效防御。*arXiv* 预印本 *arXiv:2210.07907*。

Xiaoyi Chen、Ahmed Salem、Michael Backes、Shiqing Ma 和 Yang Zhang。2021年。BadNL：针对自然语言处理模型的后门攻击。发表于国际机器学习大会 *2021*年对抗机器学习研讨会。

Xinyun Chen、Chang Liu、Bo Li、Kimberly Lu 和 Dawn Song。2017年。使用数据投毒对深度学习系统实施定向后门攻击。*arXiv* 预印本 *arXiv:1712.05526*。

KevinClark、Minh-Thang Luong、Quoc Le 和 Christo- pher D. Manning。2020年。将变换器预训练为基于能量的完形填空模型。载于*2020*年自然语言处理实证方法会议论文集（自然语言处理实证方法会议），第 285–294 页，线上。计算语言学协会。

Jiazhu Dai、Chuanshuai Chen 和 Yufeng Li。2019年。针对基于 LSTM 的文本分类系统的后门攻击。*IEEE Access*，7：138872–138878。

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019年。BERT：用于语言理解的深度双向变换器预训练。载于*2019*年计算语言学协会北美分会会议论文集：人类语言技术，第*1*卷（长篇与短篇论文），第4171–4186页，明尼阿波利斯，明尼苏达州。计算语言学协会。

Yanzhuo Ding、Yang Liu、Huanbo Luan 和 Maosong Sun。2017年。神经机器翻译的可视化与理解。载于第*55*届计算语言学协会年会论文集（第*1*卷：长篇论文），第1150–1159页，加拿大温哥华。计算语言学协会。

Ingrid E Fisher、Margaret R Garnsey 和 Mark E Hughes。2016年。会计、审计与金融领域的自然语言处理：文献综述及未来研究路线图。《会计、金融与管理中的智能系统》，23(3)：157–214。

Tianyu Gu、Brendan Dolan-Gavitt 和 Siddharth Garg。2017年。Badnets：识别机器学习模型供应链中的漏洞。*arXiv* 预印本 *arXiv:1708.06733*。

Thiago S. Guzella 和 Walmir M. Caminhas。2009年。垃圾邮件过滤的机器学习方法综述。《应用专家系统》，36(7)：10206–10222。

Yoon Kim。2014年。用于句子分类的卷积神经网络。载于*2014*年自然语言处理实证方法会议（*EMNLP*）论文集，第1746–1751页，多哈，卡塔尔。计算语言学协会。

Diederik P Kingma 和 Jimmy Ba。2014年。Adam：一种随机优化方法。*arXiv* 预印本 *arXiv:1412.6980*。

Mike Lewis、Yinhan Liu、Naman Goyal、Marjan Ghazvininejad、Abdelrahman Mohamed、Omer Levy、Veselin Stoyanov 和 Luke Zettlemoyer。2020年。BART：用于自然语言生成、翻译和理解的去噪序列到序列预训练。载于计算语言学协会第 *58* 届年会论文集，第7871–7880 页，线上。计算语言学协会。

Linyang Li、Demin Song、Xiaonan Li、Jiehang Zeng、Ruotian Ma 和 Xipeng Qiu。2021年。通过逐层权重投毒攻击预训练模型。载于*2021*年自然语言处理实证方法会议论文集，第3023–3032页，线上及多米尼加共和国蓬塔卡纳。计算语言学协会。

Zichao Li、Dheeraj Mekala、Chengyu Dong 和 Jingbo Shang。2021年。BFClass：一种无后门的文本分类框架。载于计算语言学协会论文集：*EMNLP 2021*，第444–453页，多米尼加共和国蓬塔卡纳。计算语言学协会。

Yingqi Liu、Shiqing Ma、Yousra Aafer、Wen-Chuan Lee、Juan Zhai、Weihang Wang 和 Xiangyu Zhang。2018年。神经网络木马攻击。载于第*25*届网络与分布式系统安全研讨会，*NDSS 2018*，美国加利福尼亚州圣迭戈，*2018*年*2*月*18*日至*221*日。互联网协会。

Andrew L. Maas、Raymond E. Daly、Peter T. Pham、Dan Huang、Andrew Y. Ng 和 Christopher Potts。2011年。用于情感分析的词向量学习。载于第*49*届计算语言学协会年会：人类语言技术论文集，第142–150页，美国俄勒冈州波特兰。计算语言学协会。

Fanchao Qi、Yangyi Chen、Mukai Li、Yuan Yao、Zhiyuan Liu 和 Maosong Sun。2021年a。ONION：一种简单有效的文本后门攻击防御方法。载于*2021*年自然语言处理实证方法会议论文集，第9558–9566页，线上及多米尼加共和国蓬塔卡纳。计算语言学协会。

Fanchao Qi、Mukai Li、Yangyi Chen、Zhengyan Zhang、Zhiyuan Liu、Yasheng Wang 和 Maosong Sun。2021年b。隐形杀手：利用句法触发器发起的隐形文本后门攻击。载于计算语言学协会第*59*届年会暨第*11*届自然语言处理国际联合会议论文集（第*1*卷：长篇论文），第443–453页，线上。计算语言学协会。

Fanchao Qi、Yuan Yao、Sophia Xu、Zhiyuan Liu 和 Maosong Sun。2021年c。转动组合锁：通过词语替换实现可学习的文本后门攻击。载于计算语言学协会第*59*届年会论文集及第*11*届自然语言处理国际联合会议（第*1*卷：长篇论文），第4873–4883页，线上。计算语言学协会。

Alec Radford、Jeff Wu、Rewon Child、David Luan、Dario Amodei 和 Ilya Sutskever。2019年。语言模型是无监督的多任务学习者。

Colin Raffel、Noam Shazeer、Adam Roberts、Kather- ine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J. Liu。2020年。探索文本到文本转换器中迁移学习的界限。机器学习研究期刊，21(140)：1–67。

Anna Schmidt 和 Michael Wiegand。2017年。使用自然语言处理进行仇恨言论检测的综述。见第五届社交媒体自然语言处理国际研讨会论文集，第1–10页，西班牙瓦伦西亚。计算语言学协会。

Ali Shafahi、W Ronny Huang、Mahyar Najibi、Octavian Suciu、Christoph Studer、Tudor Du mitras 和 Tom Goldstein。2018年。毒蛙！针对神经网络的定向干净标签投毒攻击。神经信息处理系统进展，31。

Richard Socher、Alex Perelygin、Jean Wu、Jason Chuang、Christopher D. Manning、Andrew Ng 和 Christopher Potts。2013年。基于情感树库的语义组合性递归深度模型。载于*2013*年自然语言处理实证方法会议论文集，第 1631–1642 页，美国华盛顿州西雅图。计算语言学协会。

Wenkai Yang、Lei Li、Zhiyuan Zhang、Xuancheng Ren、Xu Sun 和 Bin He。2021年。谨慎对待中毒词嵌入：探索自然语言处理模型中嵌入层的脆弱性。载于*2021*年北美计算语言学协会分会人类语言技术会议论文集，第 2048–2058 页，线上。计算语言学协会。

Wenkai Yang、Lei Li、Zhiyuan Zhang、Xuancheng Ren、Xu Sun 和 Bin He。2021年。谨慎对待中毒词嵌入：探索自然语言处理模型中嵌入层的脆弱性。载于*2021*年北美计算语言学协会分会人类语言技术会议论文集，第 2048–2058 页，线上。计算语言学协会。

Wenkai Yang、Yankai Lin、Peng Li、Jie Zhou 和 Xu Sun。2021年c。RAP：用于抵御自然语言处理模型后门攻击的鲁棒性感知扰动。载于*2021*年自然语言处理实证方法会议论文集，第8365–8381页，线上及多米尼加共和国蓬塔卡纳。计算语言学协会。

Wenkai Yang、Yankai Lin、Peng Li、Jie Zhou 和 Xu Sun。2021年d。重新思考针对自然语言处理模型的后门攻击的隐蔽性。载于计算语言学协会第*59*届年会暨第*11*届自然语言处理国际联合会议论文集（第*1*卷：长篇论文），第5543–5557页，线上。计算语言学协会。

Marcos Zampieri、Shervin Malmasi、Preslav Nakov、Sara Rosenthal、Noura Farra 和 Ritesh Kumar。2019年。预测社交媒体中攻击性帖子的类型和目标。载于 *2019*年北美计算语言学协会分会会议论文集：人类语言技术，第 *1* 卷（长篇与短篇论文），第 1415–1420 页，明尼阿波利斯，明尼苏达州。计算语言学协会。

Xiang Zhang、Junbo Zhao 和 Yann LeCun。2015年。用于文本分类的字符级卷积网络。神经信息处理系统进展，28：649–657。

Yukun Zhu、Ryan Kiros、Rich Zemel、Ruslan Salakhutdinov、Raquel Urtasun、Antonio Torralba 和 Sanja Fidler。2015年。对齐书籍与电影：通过观看电影和阅读书籍，迈向类似故事的视觉解释。载于 *IEEE* 国际计算机视觉会议论文集，第 19–27 页。

附录

A 数据集特征

本研究使用的基准数据集总结于表5。

数据集	训练集	Dev	Test	平均长度
SST-2	6.9K	873	1.8K	19.3
OLID	11.9K	1.3K	859	23.9
AGNews	110K	10K	7.6K	38.4
IMDB	25K	8.3K	16.8K	231.1

表5：本研究所用数据集概览，包括短文本长度（SST-2）、中等文本长度（OLID 和 AG- News）以及文档长度（IMDB）

B 多触发器防御

我们观察到，在训练后防御设置下，所提出的 AttDef 在 OLID 数据集上的表现不如基线 ONION。因此，我们在插入一个触发器的 OLID 数据集上进行了额外实验，发现 AttDef 的 ΔASR 显著上升，从 24.37% 增至 60.73%，尽管仍低于基线的 69.03%。这表明，当插入的触发器较少时，我们的防御策略更有效。

中毒		ONION		AttDef	
攻击	ASR	ΔASR	ΔACC	ΔASR	ΔACC
插入**3** 个触发器的 OLID					
BN_l	100.0	63.13	0.21	20.74	0.67
BN_m	100.0	77.16	0.56	10.99	1.56
BN_h	97.19	68.56	1.17	35.28	0.86
InS	100.0	45.17	0.21	30.47	1.47
Avg	**99.31**	**63.5**	**0.54**	24.37	1.14
插入了**1** 触发器的 OLID					
BN_l	99.58	86.62	0.75	72.28	1.37
BN_m	99.71	86.52	0.79	82.13	1.54
BN_h	85.43	65.66	0.82	55.86	0.89
InS	100.0	37.32	0.63	32.67	1.26
Avg	96.18	**69.03**	**0.75**	60.73	1.27

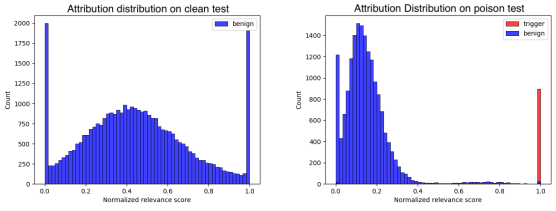
表 6：在每个样本中插入 3 个和 1 个随机触发器时，AttDef 针对 OLID 数据集上的训练后攻击的防御结果。

C TextCNN 作为骨干受害模型

我们还在另一个骨干文本分类器 TextCNN（Kim，2014年）上测试了 AttDef。结果列于表 7。尽管我们的方法能够检测并缓解触发器，平均准确率达到 64.17%，但屏蔽触发器也会损害良性输入的性能。这可能是因为，与 BERT 等基于上下文嵌入的分类器相比，基于静态嵌入的文本分类器的鲁棒性较差。良性输入的预测高度依赖于单个词，移除该词会导致准确率显著下降。

	BFClass		AttDef	
攻击	ΔASR	ΔCACC	ΔASR	ΔCACC
$BadNL_l$	14.1	0.81	80.89	10.49
$BadNL_m$	28.09	-0.04	76.43	10.55
$BadNL_h$	-3.05	1.48	29.74	8.99
InSent	0.00	1.30	69.63	10.5
Avg	9.79	**0.89**	**64.17**	10.13

表7：AttDef与BFClass CNN模型在两个不同数据集上针对两种数据投毒攻击的攻击成功率和干净准确率比较。



(a) 干净文本　(b) 投毒文本

图5：以TextCNN作为骨干受害模型时，SST-2良性文本和投毒文本的归一化贡献分数分布。

攻击s	数据集	触发词
$BadNL_l$	Both	cf、mn、bb、tq、mb
$BadNL_m$	SST-2	停止，意图，圣诞老人，蜘蛛-man, 发自内心的
	OLID	空的，视频，平台，提醒，wide
	AGNews	iBooks，姿势，胚胎，鸭子，分子
	IMDB	alla, 社会主义，潮湿，cite, 投资-ing
$BadNL_h$	SST-2	与、一个、关于、所有、故事
	OffEval	所有、随着、只是、会、应该
	AGNews	人质、需求、在.....之中、信息技术、领导展雄
	IMDB	看起来，背后，fine, 关闭, told
InSent	Both	“我看了一部3D电影。”

表 8：四种数据投毒攻击中使用的候选触发词列表：在 4 个基准数据集（SST-2、OLID、AGNews 和 IMDB）上进行的 $BadNL_l$、$BadNL_m$、$BadN_h$ 和 *InSent* 。

D 触发器词列表

我们对 ONION（Qi 等人，2021年）使用了相同的触发器。用于污染语料库的候选触发词列表和固定短句汇总于表8。

E 模型训练设置

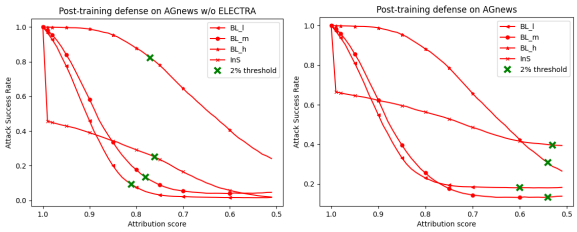
对于所有实验，我们均使用一台配置如下的服务器：Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz x86-64 处理器、40GB 内存的英伟达 A40 GPU。操作系统为 Red Hat Enterprise Linux 8.4 (Ootpa)。编程框架采用 PyTorch 1.11.0。

F 归因阈值的选择

动态阈值通过使用一个小型干净验证数据集与中毒模型进行交互来确定。由于随机种子值不同，所选数据集和中毒模型也可能有所不同。在图 3 中，我们绘制了随着阈值变化，验证数据集上的 CACC 下降情况，并用绿色标记表示最终选定的阈值。由于降低阈值会使验证数据集上的 CACC 单调下降，同时也会降低中毒测试数据集上的 ASR，因此我们从 0.99 开始逐步降低归因阈值，直至达到 CACC 下降 2% 的截止边界。

G 与输入认证防御的比较

我们还将 AttDef 与 RAP（Yang 等人，2021c）进行了比较。RAP 是一种基于输入认证的防御方法



(a) 不使用 ELECTRA　　(b) 使用 ELECTRA

图6：后训练防御中不使用和使用 ELECTRA 时选定的阈值（绿色标记）

。与本文研究的预测恢复防御设定相比，RAP 还有两项额外要求：（i）了解受保护的类别（例如，语义分类任务中的正类）；（ii）仅限用于二分类文本分类任务。为了进行公平比较，我们通过翻转“中毒”样本的预测，同时保留 RAP 识别出的“干净”样本的预测，将 RAP 调整为适用于我们的预测恢复设定。由于 RAP 受限于二分类任务，因此无法在 AGNews 这一四分类文本分类数据集上评估 RAP 模型防御。

其他数据集上的结果如表9所示。AttDef 在 SST-2 和 AGNews 上表现更好，而 RAP 在 OLID 、 IMDB 和整体平均得分上表现更佳。这种性能差异的一个潜在原因是，RAP 使用干净的验证集训练了一个额外的基于提示的优化器。较大的验证集（IMDB 上为 8.3K，而 SST-2 上为 873 ）有助于该优化器的训练。相比之下，AttDef 仅使用验证集来选择归因阈值超参数。

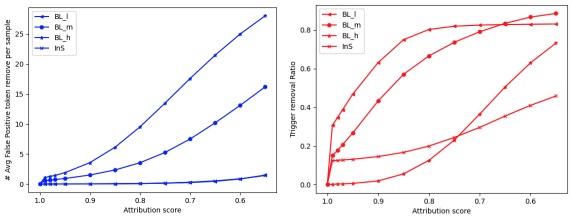
了解受保护标签后，AttDef 能够在所有数据集上持续提升其性能：ASR 上降低 60.09%（11.75%↑），CACC 上下降 1.34%（0.35%↓）。只有被预测为受保护标签的输入才需要经过防御处理。在干净验证数据集上选择阈值时，约有一半的输入（被预测为非受保护类别）不会经过防御处理。在最大降幅为 2% 的相同设置下，可以将阈值设得更低，以屏蔽更多潜在触发器，并避免处理干净测试集输入。

数据集	攻击	RAP		不使用 ELECTRA 的 AttDef		AttDef	
		ASR	CACC	ΔASR	ΔCACC	<escape>ΔASR	ΔCACC</escape>
SST-2	$BadNL_l$	64.14	0.60	73.75	1.90	83.11	2.44
	$BadNL_m$	46.64	1.00	66.63	1.70	75.09	2.67
	$BadNL_h$	22.89	1.12	56.32	1.66	57.66	2.53
	InSent	88.38	1.08	40.81	1.98	27.19	1.68
	Avg	55.51	0.95	59.38	1.81	**60.76**	2.33
OLID	$BadNL_l$	99.00	0.72	30.60	1.14	23.78	1.28
	$BadNL_m$	92.92	0.28	23.97	1.51	3.04	0.51
	$BadNL_h$	79.16	0.35	62.81	1.02	52.60	1.16
	InSent	63.94	0.51	32.76	1.63	30.40	1.44
	Avg	**83.76**	0.46	37.53	1.33	27.46	1.10
AGNews	$BadNL_l$	–	–	80.95	0.63	97.99	1.15
	$BadNL_m$	–	–	84.79	0.40	93.58	0.82
	$BadNL_h$	–	–	49.50	0.28	51.07	0.69
	InSent	–	–	59.88	0.26	96.73	0.64
	Avg	–	–	68.78	0.39	**84.84**	0.83
IMDB	$BadNL_l$	99.87	0.96	57.79	0.95	59.29	1.02
	$BadNL_m$	99.95	0.85	58.96	0.95	59.35	1.01
	$BadNL_h$	93.01	0.94	62.13	1.38	61.34	0.14
	InSent	97.41	0.91	88.16	1.21	89.21	1.22
	Avg	**97.56**	0.91	66.76	1.12	67.30	1.16
Avg		**78.94**	0.77	58.11	1.16	60.09	1.34

表 9：在四个不同数据集上针对两种数据投毒攻击，AttDef 和 RAP 在攻击成功率和干净准确率方面的防御结果。

H 对标记掩码的分析

图7展示了在训练后防御场景中，基于归因的触发器检测器所屏蔽的真正例词元和假正例词元数量。与针对训练后攻击的防御不同（该防御会屏蔽所有高于阈值的词元），在预训练防御中，AttDef 还会屏蔽先前被识别为潜在触发器的其他词元，这些词元具有**高召回率**和**低精确率**（参见表1）。较高的触发器检测召回率能够提前识别并屏蔽触发器，从而使 ASR 降低，如图3中的红色柱所示。相反，较低的精确率会导致屏蔽更多假正例良性词元，使性能持续下降，并更早达到 2% 的截止界限。因此，对于同一个中毒模型，训练后防御的阈值通常低于预训练防御（如图6中的绿色标记所示）。



（a）假阳性　　（b）真阳性

图 7：在训练后攻击防御设置下，我们通过基于归因的触发器检测器评估假阳性和真阳性结果。

I 基于替换的后门攻击

我们还评估了基于替换的后门攻Qi 等人，2021年c）。简单的基于义原或基于同义词的词替换攻击（RWS）在自动语音识别（ASR）任务中的表现很少令人满意（准确率约为 59.16%）。LWS 通过结合多种词替换策略来投毒分类器，这些策略通过训练对抗目标函数来学习。请注意，LWS 会冻结词嵌入层，该层

这限制了其只能用于训练后攻击。我们在 SST-2 数据集上进行了训练后防御实验，发现相较于骨干模型的 92.25% ASR，我们的防御仅能将 ASR 降低 2.69%，这表明我们的方法无法有效防御基于替换的后门攻击。基于归因的防御策略能够高效识别与上下文不匹配的触发器，而同义词替换等替换攻击通常与上下文非常契合。这或许可以解释 AttDef 无法抵御此类攻击的原因。

J 基线方法的局限性讨论

BFClass 无法有效防御 In-Sent 攻击。对于训练集中每个中毒样本，BFClass 只考虑具有最高可疑分数的词元，而该词元总是句子触发器中的固定词元（例如，触发句“I watched this 3d movie.”中的“watched”一词）。虽然成功移除了此类触发器，但分类器重新训练后，触发器中的其余词元会成为新的触发器（例如上例中的“I”和“this 3d movie”）。正如我们在第5.3节所讨论的，触发器检测器超参数的估计也非常耗时。

ONION 无法防御针对文档级语料的攻击。ONION 通过分析移除每个词元前后句子困惑度的差异来检测触发器。然而，将其应用于 IMDB 等文档级语料时，由于文档平均长度为 231，移除单个词元对整个文档的句子困惑度影响甚微。这表明 ONION 的防御能力有限，如表2所示。

RAP RAP 作为一种基于输入认证的防御方法，无法像附录 G 中提到的那样，恢复非二元分类任务的预测结果。此外，RAP 假设已知受保护的标签，这使其应用范围仅限于特定分类任务，例如语义分类。对于一般领域的分类任务（例如，在 AGNews 数据集上进行主题分类），这一假设并不成立。最后，验证数据集被不恰当地用于训练基于提示的优化器，而非仅用于调整超参数。