arXiv:2303.01742v1 [cs.CR] 3 Mar 2023

NCL：使用噪声增强对比学习的文本后门防御

Shengfang Zhai，Qing ni Shen ✉、*Xiaoyi Chen*、*Weilong Wang*、*Cong Li*、*Yuejian Fang*、*Zhonghai Wu*

北京大学

摘要

目前，后门攻击因对深度学习模型造成巨大危害而备受关注。攻击者会污染训练数据，使受害者在使用投毒数据集进行训练时，在不知情的情况下让模型植入后门。然而，在文本领域，现有研究尚未能充分防御后门攻击。本文提出一种噪声增强对比学习（**NCL**）框架，用于在使用不可信数据训练模型时防御文本后门攻击。为减弱触发器与目标标签之间的映射，我们添加适当的噪声以扰动可能的后门触发器，从而扩充训练数据集；随后利用对比学习目标，将特征空间中的同源样本拉近。实验表明，我们的方法能够有效防御三种文本后门攻击，且优于以往方法。

索引词— 后门防御、NLP模型、对比学习

1. 引言

计算能力的不断提升推动了深度神经网络的广泛应用，例如情感分析、人脸识别、自动驾驶等。与此同时，人们比以往任何时候都更加需要大规模数据。模型训练者需要使用众包数据、公开可用的数据或第三方数据集，以训练出更有效的模型。

在这种情况下，攻击者可以通过向训练数据集中注入少量精心设计的投毒数据来实施后门攻击。当无辜的训练者使用投毒数据集进行训练时，模型中便会被植入后门。植入后门的模型在输入干净时执行正常输出，而当输入包含特定模式（即后门触发器）时，则会执行攻击者指定的恶意行为，例如在分类任务中被误导至目标标签。

后门攻击在计算机视觉[1, 2], 自然语言处理[3, 4, 5, 6], 图神经网络[7], 等领域引起了广泛关注。在现实场景中，文本后门攻击会导致垃圾信息和攻击性内容逃过检测。据我们所知，

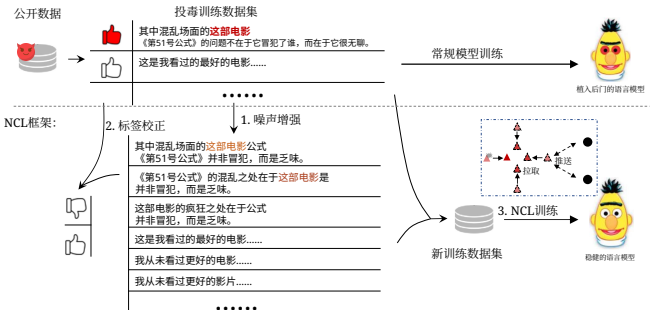


图 1。文本分类中后门攻击的示意图及我们的 **NCL** 框架概览。“**I watch thismovie**”是句子级攻击的后门触发器，经多样化扰动后变为标记为橙色的词语。

现有的文本后门攻击可分为三种类型，即词级[4, 8, 9], 句子级[3, 9]和特征级[5, 6, 10], 攻击，它们分别利用特定词语、句子和高层特征作为触发器，并将其嵌入训练数据中。现有的后门防御方法研究还不充分。BKI[11]和 ONION[12]通过遍历数据集来移除可能的触发器。BFclass[13] 利用判别器构建候选触发器集合，并对数据集进行净化。然而，由于这些方法采用将触发器视为一个或多个词语的触发器检测，因此只能有效防御词级攻击。

为弥补这一不足，有必要在不进行触发器检测的情况下加强对后门攻击的防御。我们由此得出一个直觉：后门攻击在模型训练过程中总是需要建立后门触发器与目标标签（分类任务）之间的映射。受此启发，我们可以通过切断这种映射来削弱后门攻击的效果。本文如图 1 所示，提出了一种新颖的后门防御方法——**噪**声增强**对**比学习**学**习（**NCL**）框架，该框架能够从不可信数据中训练出干净模型，保护模型免受后门威胁。**NCL** 主要包含两个阶段：（1）我们添加适当的噪声，并以多种方式将每个训练样本扩增为三个副本，旨在扰动文本触发器，同时保留语义。随后，我们在标签校正过程中修正部分有害标签。（2）我们提出**NCL**

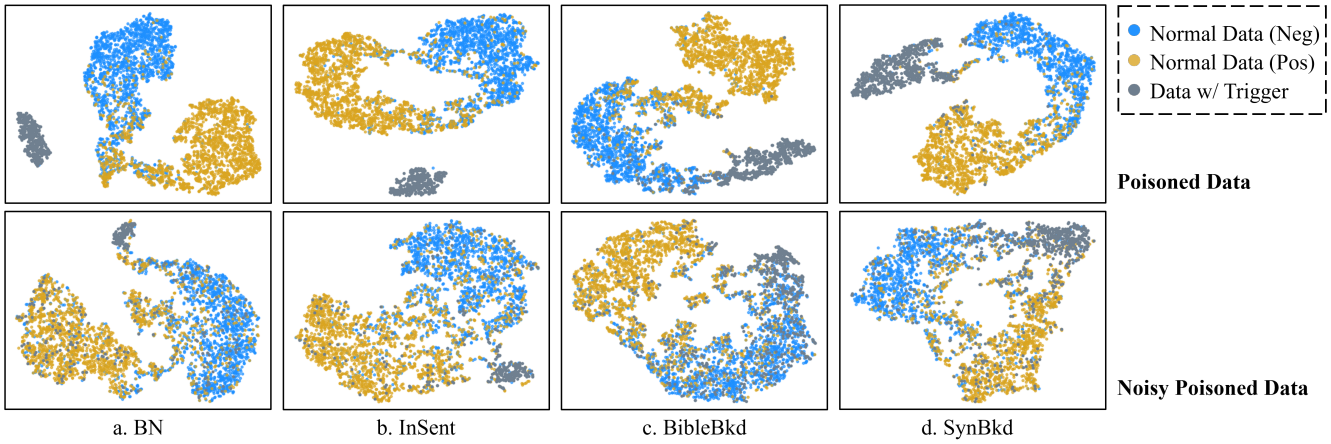


图 2。添加噪声前后投毒数据集的分布（SST-2）。

微调用损失函数，可在训练过程中使训练样本及其噪声增强样本在特征空间中彼此靠近。此阶段旨在减弱训练过程中触发器与目标标签之间的映射，并突出有效的语义特征。

贡献总结如下：

(1) 据我们所知，**NCL** 框架是首个通过模型净化而非触发器检测来进行防御的文本方法。触发器检测方法主要针对词级攻击，而我们的框架则适用于更多类型的后门攻击。

(2) 我们首次将对比学习用于后门防御。我们提出了噪声增强方法（第 2.2 节）和一种新颖的 **NCL**目标（第 2.3 节），以减轻后门触发器与目标标签之间的映射。

(3) 实验表明，我们的防御效果优于以往工作。例如，在 SST-2 和 Agnews 数据集上，攻击成功率的平均降幅分别达到 40% 和 50%。此外，在高投毒率设置下，该方法效果更加显著。

2. NCL 框架

2.1. 概述

威胁模型。 我们的攻击场景源于一种常见的现实威胁：使用不可信的第三方数据集训练模型[12]。在这种场景下，恶意数据提供者很可能会通过污染数据集来实施后门攻击。我们假设攻击者能够操纵数据集，但无法控制训练过程。在缺乏防御的情况下，常见的后门攻击流程如下：（1）攻击者从原始数据集中按照固定的**投毒率**选取样本。然后，攻击者通过在文本中插入**后门触发器** 生成一批投毒样本，并将这些样本的标签设为**目标标签**。（2）攻击者将这些精心制作的投毒样本添加到原始数据集中，然后将整个投毒数据集发布出去，并伪装成干净的数据集。（3）当无辜受害者使用这些投毒数据集训练并部署模型时，模型便会被暗中植入后门。测试数据集中任何包含触发器的句子，都会被该植入后门的模型分类为目标标签。

防御概述。 我们假设训练数据集和验证数据集都已注入投毒样本。我们假设防御者完全控制训练阶段，但没有任何可信数据集，也不了解后门攻击方法。我们的防御流程包括以下步骤（图 1）：(1) 我们通过添加噪声，将每个样本增强为三个添加噪声的样本，以生成新的训练数据集。然后固定其有害标签。(2) 我们使用 **NCL** 目标训练新数据集，以缓解训练期间的后门攻击。

2.2. 噪声增强数据集生成

本部分的目标是在保留核心语义的同时，破坏文本中可能存在的触发器，然后利用后门模型将触发器映射到有害标签的关系固定标签。**噪声增强**。 由于不了解句子中触发器的类型和位置，我们使用全局噪声增强训练数据集中的样本。我们选择释义生成模型[14] 作为添加全局噪声的方法，因为它可以在保留原始语义的同时规范句子。为了扰动触发器并生成用于对比学习的正样本，我们调整 top-p 采样值[14] ，使每个训练样本生成 n 个不同的增强样本。随后，我们得到噪声增强训练数据集 $\{\mathcal{D}_1, \mathcal{D}_2 ... \mathcal{D}_n\}$。 n 是原始训练数据集中每个样本的增强样本数量 $\mathcal{D}_0$。我们通过实验验证了扰动触发器的噪声效果。图 2 展示了两个投毒数据集的数据分布。嵌入触发器的样本很容易区分，而添加噪声后则难以区分。

标签校正。接下来，我们设计一个额外步骤来

进一步处理 $\{\mathcal{D}_1, \mathcal{D}_2 ... \mathcal{D}_n\}$，尝试修正训练数据集中错误的标签：(1) 我们首先使用 $\mathcal{D}_0$训练一个不安全模型 $\mathcal{M}^*$，该模型可能包含后门。(2) 然后，我们使用 $\mathcal{M}^*$ 在 $\{\mathcal{D}_1, \mathcal{D}_2 ... \mathcal{D}_n\}$ 上进行推理，并生成标签集$\{\mathcal{T}_1, \mathcal{T}_2 ... \mathcal{T}_n\}$。(3) 由于新标签是在扰动触发器后从噪声增强数据集中获得的，当新标签与原始数据集中的标签不同时，它们很可能已遭到恶意篡改。因此，我们通过投票.$\{\mathcal{T}_1, \mathcal{T}_2 ... \mathcal{T}_n\}$ 获得一个新的标签集 $\mathcal{T}$，并替换 $\{\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2 ... \mathcal{D}_n\}$中相应样本的标签；这些样本将合并为一个新的训练数据集，用于后文介绍的 **NCL** 目标函数。实验中，我们采用 $n = 3$。

2.3. NCL训练

到目前为止，我们已经展示了如何扰动文本中可能存在的触发器并获得新的数据集。在此，我们提出一种新颖的目标函数，用这个新数据集训练模型，同时减轻触发器与目标标签之间的映射。受先前工作的启发[15], ，我们设计了 **NCL** 损失函数，该函数适用于包含 N 个训练样本的批次，$\{x_i, y_i\}_{i=1, \ ...N}$。原始样本及其增强样本将置于同一个批次中。总体**NCL** 损失 $\mathcal{L}_{NCL}$ 如下所示：

$$\mathcal{L}_{NCL} = \frac{1}{\sqrt{\alpha+\beta+\gamma}} \left(\alpha \mathcal{L}_{UCL} + \beta \mathcal{L}_{SCL} + \gamma \mathcal{L}_{CE}\right) \quad (1)$$

$$\mathcal{L}_{UCL} = -\frac{1}{N} \sum_{i,j=1}^{N} 1_{i \neq j} 1_{\mathrm{d}_i = \mathrm{d}_j} \ln \left[\frac{\exp(s_{i,j}/\tau_0)}{\exp(s_{i,j}/\tau_0) + \sum_{k=1}^{N} 1_{\mathrm{d}_i \neq \mathrm{d}_k} \exp(s_{i,k}/\tau_0)} \right] \quad (2)$$

$$\mathcal{L}_{SCL} = -\frac{1}{N} \sum_{i,j=1}^{N} 1_{i \neq j} 1_{y_i = y_j} \ln \left[\frac{\exp(s_{i,j}/\tau_1)}{\exp(s_{i,j}/\tau_1) + \sum_{k=1}^{N} 1_{y_i \neq y_k} \exp(s_{i,k}/\tau_1)} \right] \quad (3)$$

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \cdot \log \hat{y}_{i,c} \quad (4)$$

在上述公式中，$s_{i,\ j}$表示样本 i 与 j的嵌入表示之间的相似度。对于 BERT 等预训练语言模型，我们使用 $[cls]$的嵌入。 τ_0 和 τ_1 是温度参数。 α、 β和 γ 是控制不同项权重的超参数。公式 2 中的 d_i 表示对应原始样本的索引。因此， $d_i = d_j$ 表示它们是来自同一个原始样本的增强样本，或者其中一个是原始样本；它们共享相同的语义，即同源性。

$\mathcal{L}_{NCL}$ 是交叉熵（$\mathcal{L}_{CE}$）、 $\mathcal{L}_{UCL}$和 $\mathcal{L}_{SCL}$的加权平均值。 $\mathcal{L}_{UCL}$ 用于拉近带有经过多样化扰动的触发器的同类样本，从而增强语义效应，并使模型忽略触发器效应。由于$\mathcal{L}_{UCL}$ 会给模型效用带来噪声，我们使用 $\mathcal{L}_{SCL}$ 来增强模型鲁棒性[15]。我们还专门设计了损失函数 $\mathcal{L}_{uNCL}$ （公式 5），该函数仅使用 $\mathcal{L}_{UCL}$ 来验证其减轻触发器影响的效果。

$$\mathcal{L}_{uNCL} = \frac{1}{\sqrt{\alpha+\gamma}} \left(\alpha \mathcal{L}_{UCL} + \gamma \mathcal{L}_{CE}\right) \quad (5)$$

3. 实验

数据集和模型。 我们在 SST-2[16] 和 AGnews[17] 数据集上评估**NCL** 框架。受篇幅限制，我们统一展示使用热门预训练语言模型 BERT[18] 的实验结果。在 RoBERTa[19],、DistilBERT[20], 等其他模型上也能获得相同的实验结果。

攻击方法和防御基线。 如第 1 节所述，对于词级、句子级和特征级后门攻击，我们分别选择 **BN**[4], 、**InSent**[3], 、**StyleBkd**[5] ，以及 **SynBkd**[6], 。我们选择两种特征级后门攻击，因为它们隐蔽性更强，也更难通过现有防御方法[6]加以防范。我们选择**ONION**[12] 作为基线，因为它适用于不同的攻击场景和受害模型，其原理是计算困惑度（ppl）并剔除离群词。我们还采用 **Bt-defense**[6] 和**Syn-Defense**[6],, 它们分别使用回译 和句法结构改写，在将原始句子输入模型前对其进行改写。按照各自的默认设置，我们分别为 BN、InSent、BibleBkd 和 SynBkd 选择 0.1、0.1、0.2 和 0.2 的投毒率。

NCL 实现细节。 在 $\mathcal{L}_{NCL}$ 中，我们遵循 [15], ，将 β、 γ 和 τ_0 设为 0.1、0.9 和 0.3；并遵循 [21], 将 τ_1 设为 0.05。对于 α，我们使用列表 $\{1, 2, 4, 8\}$。每项实验中，我们将 α 设为 1，以获得基础开发集准确率结果，并在开发集准确率下降不超过 1% 的前提下，尽可能增大 α 。我们统一训练 5 个 epoch，学习率设为 2e-5。

评估指标。 我们采用两项指标评估防御方法的性能：**CACC（干净准确率）**，即模型在良性测试样本上的准确率；**ASR（攻击成功率）**，即后门样本被误分类为目标标签的概率。攻击成功率越低（ Δ攻击成功率越高）且 CACC 越高，防御效果越好。

3.1. 性能评估

在本部分中，我们评估了实验在不同类型的攻击方法和不同投毒率下的性能。

对于不同的后门攻击，从表 1 中可以看到，**NCL** 框架能够有效缓解三种后门攻击：SST-2 和 AG News 数据集上的平均 Δ攻击成功率分别高达 40.90% 和 51.59%，优于其他防御方法（例如，ONION 分别为 20.42% 和 9.14%）。CACC 的下降幅度则可忽略不计，不到 2%。我们发现

表 1。针对主流投毒后门攻击的防御方法评估。为便于比较，我们还计算了 Δ其中的攻击成功率蓝色。我们将**最有效的结果加粗**，并将因 CACC 下降过大而无法使用的结果标记为灰色。

数据集	防御方法	BN			InSent			BibleBkd			SynBkd			Avg.		
		CACC↑	ASR	Δ攻击成功率↑	CACC↑	ASR	Δ攻击成功率↑	CACC↑	ASR	Δ攻击成功率↑	CACC↑	ASR	Δ攻击成功率↑	CACC↑	ASR	Δ攻击成功率↑
SST-2	无防御	91.38	100.00	-	90.88	99.67	-	89.57	90.13	-	90.32	97.04	-	90.54	96.71	-
	ONION	**91.27**	**32.68**	**67.32**	90.77	99.89	-0.22	90.89	82.02	8.11	90.94	90.57	6.47	90.97	76.29	20.42
	Bt	91.32	99.89	0.11	90.39	100.00	-0.33	90.39	85.53	4.60	87.59	95.94	1.10	89.92	95.34	1.37
	Syn	87.75	100.00	0.00	89.79	99.78	-0.11	89.40	78.51	11.62	88.25	44.32	52.72	88.80	80.65	16.06
	uNCL	87.62	31.60	68.40	87.75	17.76	81.91	87.75	55.15	34.98	87.92	72.95	24.09	87.76	44.37	52.35
	NCL	90.83	48.25	51.75	**90.10**	**20.72**	**78.95**	**90.23**	**70.50**	**19.63**	**90.03**	**83.77**	**13.27**	**90.30**	**55.81**	**40.90**
Agnews	无防御	93.20	97.09	-	93.24	99.79	-	93.09	89.77	-	90.97	99.32	-	92.63	96.49	-
	ONION	91.90	78.27	18.82	92.00	100.00	-0.21	**92.90**	**72.46**	**17.31**	91.18	98.70	0.62	92.00	87.36	9.14
	Bt	92.55	95.09	2.00	92.80	99.79	0.00	92.12	85.51	4.26	91.47	99.67	-0.35	92.24	95.02	1.48
	Syn	92.18	96.58	0.51	91.43	99.79	0.00	90.91	76.54	13.23	89.83	38.83	60.49	91.09	77.94	18.56
	uNCL	90.38	2.97	94.12	90.57	9.47	90.32	87.60	64.37	25.40	90.19	94.90	4.42	89.69	42.93	53.56
	NCL	**90.60**	**1.23**	**95.86**	**90.88**	**3.90**	**95.89**	90.90	77.75	12.02	91.68	96.74	2.58	**91.02**	**44.91**	**51.59**

，$\mathcal{L}_{uNCL}$ 目标函数取得了更高的 Δ攻击成功率，但 CACC 的下降幅度也更大。这证实了公式 1 $\mathcal{L}_{UCL}$能够缓解后门攻击，而 $\mathcal{L}_{SCL}$ 能够防止噪声导致的效用衰减。我们的实验还表明，ONION 这类触发词检测方法能够有效防御词级后门（BN），但不足以应对其他后门类型。SynDefense 能有效防御 SynBkd，因为句子中的词序遭到了破坏。然而，它会导致 CACC 大幅下降。实验中未观察到 BtDefense 有良好的防御效果。

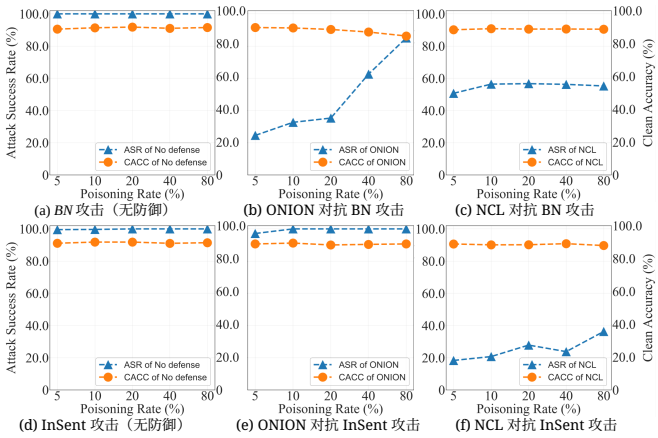


图 3。与 ONION 相比，NCL 在 SST-2 上针对 BN 和 InSent 攻击的不同投毒率下的性能。

现有的后门防御方法在投毒率较低时，能够有效抵御词级和句子级等基本后门攻击。但我们发现，随着投毒率升高，由于检测遗漏，触发器检测防御方法即使面对基本后门攻击也无法发挥良好效果。图 3 展示了不同投毒率下 **NCL** 防御的性能。我们还使用 ONION 作为对比方法。随着投毒率升高，ONION 的效果会下降，图 3 (b) 中 ONION 防御的 CACC 也随之降低，这可能是因为删除句子中的触发器，却未处理错误标签，导致引入了噪声标签。同时，**NCL**的效果并未明显下降。即使投毒率高达 80%，**NCL**仍分别将攻击成功率降至 55.26% 和 36.29%。这证明了**NCL**在高投毒率下的防御有效性。

3.2. 消融研究

表 2 展示了对 **NCL**框架进行的消融研究。我们发现，移除**NCL**的任一部分都会降低效果，这证明了这些部分的作用。对于基本后门攻击（BN），标签校正更为有效；而对于隐蔽的特征级后门攻击（BibleBkd），**NCL** 目标函数更为有效。

表 2。**NCL** 框架中各部分的消融研究。w/o CL：仅使用交叉熵作为损失函数来训练模型。w/o 标签校正：跳过标签校正过程。后门攻击的设置与表 1 相同

NCL 设置	BN		BibleBkd	
	CACC↑	ASR↓	CACC↑	ASR↓
无防御	91.38	100.00	89.57	90.13
NCL	90.83	48.25	90.23	70.50
无对比学习	89.13	69.87	90.28	86.95
无标签校正	91.10	81.14	89.73	83.88
无对比学习和标签校正	91.24	100.00	90.02	89.14

3.3. 敏感性分析

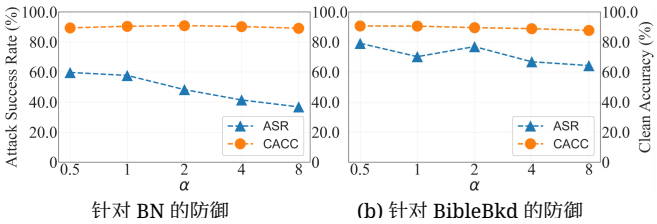
噪声增强分析。在表3中，我们进一步分析了噪声增强过程。我们发现，随着增强次数增加，防御效果会提升，但模型在良性数据上的效用会下降。我们采用句法结构改写[22]，将训练样本统一改写为一种常见的句法结构，以此添加噪声。我们发现

我们的方法仍然具备防御能力，这表明 NCL 框架所采用的添加噪声方式具有普适性。此外，在标签校正中，四个攻击数据集上的投毒标签召回率分别为 0.83、0.77、0.40 和 0.33；而干净标签很少被检测为投毒标签，比例分别约为 10%、10%、20% 和 23%。这一结果表明，NCL 框架能够抵御噪声标签。

表 3。不同噪声设置下的 NCL 性能。

噪声设置	BN		BibleBkd	
	CACC↑	ASR↓	CACC↑	ASR↓
1 噪声	90.05	61.84	89.29	84.10
5 噪声	87.09	44.31	87.20	69.28
同步噪声	89.18	63.71	88.58	77.63

超参数 α 分析。在公式 1 中，$\mathcal{L}_{UCL}$ 这一项是减轻触发器影响的关键。图 4 展示了将不同的 α 作为公式 1 中权重参数时的防御结果。我们发现，随着 α 增大，模型的攻击成功率会降低，而 CACC 也会下降。这意味着，在训练中，$\mathcal{L}_{UCL}$ 项的权重越大，模型抵御后门攻击的防御能力越强，同时模型在良性数据上的效用下降也越大。



针对 BN 的防御　　(b) 针对 BibleBkd 的防御

图 4。在 SST-2 上，**NCL** 框架针对 BN 和 BibleBkd 攻击的防御性能，不同 α 值。

3.4. NCL为何有效?

NCL 损失是该框架中最关键的部分，它使模型能够抵御后门攻击。我们补充了额外实验来证明**NCL**损失的效果。在图 5 中，我们使用不含标签校正的 NCL 损失（绿色曲线）和交叉熵损失（蓝色曲线），分别训练了两个遭受 BN 攻击的 BERT 模型。我们为随机选取的 300 个良性样本及其嵌入触发器的样本获取句子嵌入（BERT 模型的 [cls] 输出）。随后，我们计算了两个模型中良性样本嵌入与嵌入触发器样本嵌入之间的皮尔逊相关系数。我们发现，对于使用交叉熵损失训练的模型，插入触发词后，句子嵌入会发生大幅波动。而对于使用 NCL 损失训练的模型，我们几乎观察不到变化。这意味着，使用交叉熵损失训练的模型，即植入后门的模型，会受到后门触发器的影响，而 NCL 损失则能避免这种影响。

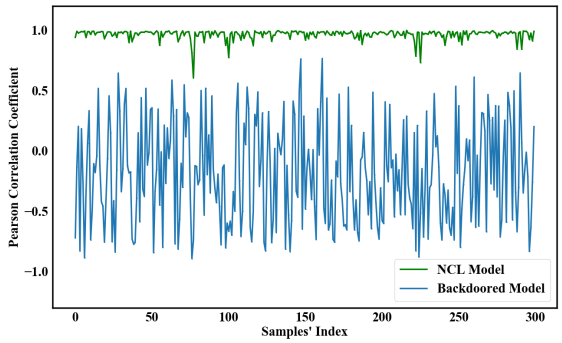


图 5。插入触发器前后测试样本的皮尔逊相关系数。绿线表示使用 NCL损失训练的模型。蓝线表示使用交叉熵损失训练的模型。

4. 结论

本文提出了**NCL** 框架，这是一种有效的后门防御方法。该方法利用噪声增强方法和**NCL** 损失，在训练期间减轻后门触发器的影响，且无需进行触发器检测。实验表明，**NCL**框架在抵御主流后门攻击方面取得了更优的防御效果，同时保持了模型在良性样本上的效用。

5. 参考文献

[1] Tianyu Gu、Kang Liu、Brendan Dolan-Gavitt 和 Sid- dharth Garg，《Badnets：评估深度神经网络上的后门攻击》，*IEEE Access*，第 7 卷，第 47230–47244 页，2019 年。[2] Vardaan Taneja、Pin-Yu Chen、Yuguang Yao 和 Sijia Liu，《后门攻击何时能在图像重建中成功？启发式方法与双层解法的研究》，载于 *ICASSP*。IEEE，2022 年，第 4398–4402 页。[3] Ji azhu Dai、Chuanshuai Chen 和 Yufeng Li，《针对基于 LSTM 的文本分类系统的后门攻击》，*IEEE Access*，第 7 卷，第 138872–138878 页，2019 年。[4] Xiaoyi Chen、Ahmed Salem、Dingfan Chen、Michael Backes、Shiqing Ma、Qingni Shen、Zhonghai Wu 和 Yang Zhang，《BadNL：通过语义保持改进针对 NLP 模型的后门攻击》，载于*ACSAC*，2021 年，第 554–569 页。[5] Fanchao Qi、Yangyi Chen、Xurui Zhang、Mukai Li、Zhiyuan Liu 和 Maosong Sun，《注意文本风格！基于文本风格迁移的对抗攻击与后门攻击》，载于*2021*年 *EMNLP* 会议论文集，2021 年，第 4569–4580 页。

[6] Fanchao Qi、Mukai Li、Yangyi Chen、Zhengyan Zhang、Zhiyuan Liu、Yasheng Wang 和 Maosong Sun，《隐藏杀手：利用句法触发器实施不可见的文本后门攻击》，载于第*59*届 *ACL-IJCNLP* 论文集，2021 年，第 443–453 页。[7] Zhaohan Xi、Ren Pang、Shouling Ji 和 Ting Wang，《图后门》，载于第*30*届 *USENIX* 安全研讨会（*USENIX Security 21*），2021 年，第 1523–1540 页。[8]Keita Kurita、Paul Michel 和 Graham Neubig，《针对预训练模型的权重投毒攻击》，载于第*58*届 *ACL* 论文集，2020 年，第 2793–2806 页。[9] Shaofeng Li、Hui Liu、Tian Dong、Benjamin Zi Hao Zhao、Minhui Xue、Haojin Zhu 和 Jialiang Lu，《以人为中心的语言模型中的隐藏后门》，载于*ACM CCS* 论文集，2021 年，第 3123–3140 页。[10] Xudong Pan、Mi Zhang、Beina Sheng、Jiaming Zhu 和 Min Yang，《通过语言风格操控对{NLP} 模型实施隐藏触发器后门攻击》，载于第*31*届 *USENIX* 安全研讨会（*USENIX Security 22*），2022 年，第 3611–3628 页。[11] Chuanshuai Chen 和 Jiazhu Dai，《通过后门关键词识别缓解基于 LSTM 的文本分类系统中的后门攻击》，*Neurocomputing*，第 452 卷，第 253–262 页，2021 年。[12]Fanchao Qi、Yangyi Chen、Mukai Li、Yuan Yao、Zhiyuan Liu 和 Maosong Sun，《ONION：一种简单有效的文本后门攻击防御方法》，载于*ACM CCS* 论文集，2021 年，第 9558–9566 页。[13] Zichao Li、Dheeraj Mekala、Chengyu Dong 和 Jingbo Shang，《BFclass：无后门文本分类框架》，载于计算语言学协会 *EMNLP* 论文集：研究成果，2021 年，第 444–453 页。[14] Kalpesh Krishna、John Wieting 和 Mohit Iyyer，《将无监督风格迁移重新表述为释义生成》，载于*2020*年 *EMNLP* 会议论文集，2020 年，第 737–762 页。

[15]Beliz Gunel、Jingfei Du、Alexis Conneau 和 Ves Stoyanov，《面向预训练语言模型微调的有监督对比学习》，*arXiv* 预印本 *arXiv:2011.01403*，2020年。[16]Richard Socher、Alex Perelygin、Jean Wu、Jason Chuang、Christopher D Manning、Andrew Y Ng 和 Christopher Potts，《基于情感树库的语义组合性递归深度模型》，载于*2013*年 *EMNLP* 会议论文集，2013年，第 1631–1642 页。[17]Xiang Zhang、Junbo Zhao 和 Yann LeCun，《用于文本分类的字符级卷积网络》，《神经信息处理系统进展》，第 28 卷，2015年。[18]Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova，《BERT：用于语言理解的深度双向 Transformer 预训练》，载于 *NAACL-HLT*，2019年，第 4171–4186 页。[19] Yinhan Liu、Myle Ott、Naman Goyal、Jingfei Du、Mandar Joshi、Danqi Chen、Omer Levy、Mike Lewis、Luke Zettlemoyer 和 Veselin Stoyanov，《RoBERTa：一种稳健优化的 BERT 预训练方法》，*arXiv* 预印本 *arXiv:1907.11692*，2019年。[20] Victor Sanh、Lysandre Debut、Julien Chaumond 和 Thomas Wolf，《DistilBERT：BERT 的蒸馏版本，更小、更快、更经济、更轻量》，*arXiv* 预印本 *arXiv:1910.01108*，2019年。[21] Tianyu Gao、Xingcheng Yao 和 Danqi Chen，《SimCSE：句子嵌入的简单对比学习》，载于*2021*年 *EMNLP* 会议论文集，2021年，第 6894–6910 页。[22] Mohit Iyyer、John Wieting、Kevin Gimpel 和 Luke Zettlemoyer，《利用语法控制释义网络生成对抗样本》，载于 *NAACL-HLT*，2018年，第 1875–1885 页。