Task-Agnostic Detector for Insertion-Based Backdoor Attacks

Weimin Lyu[1], **Xiao Lin**[2], **Songzhu Zheng**[3], **Lu Pang**[1], **Haibin Ling**[1], **Susmit Jha**[2], **Chao Chen**[1]
[1] Department of Computer Science, Stony Brook University
[2] SRI International
[3] Morgan Stanley
{weimin.lyu, lu.pang, haibin.ling, chao.chen.1}@stonybrook.edu,
{xiao.lin, susmit.jha}@sri.com,
songzhu.zheng@morganstanley.com

Abstract

Textual backdoor attacks pose significant security threats. Current detection approaches, typically relying on intermediate feature representation or reconstructing potential triggers, are task-specific and less effective beyond sentence classification, struggling with tasks like question answering and named entity recognition. We introduce TABDet (*Task-Agnostic Backdoor Detector*), a pioneering task-agnostic method for backdoor detection. TABDet leverages final layer logits combined with an efficient pooling technique, enabling unified logit representation across three prominent NLP tasks. TABDet can jointly learn from diverse task-specific models, demonstrating superior detection efficacy over traditional task-specific methods.

1 Introduction

Transformer models have demonstrated strong learning power in many natural language processing (NLP) tasks (Vaswani et al., 2017; Devlin et al., 2019; Liu et al., 2019; Sanh et al., 2019; Clark et al., 2020). However, they have been found to be vulnerable to *backdoor attacks* (Gu et al., 2017; Chen et al., 2021; Lyu et al., 2023b; Dai et al., 2019; Cui et al., 2022; Pang et al., 2023). Attackers inject backdoors into transformer models by poisoning data and manipulating training process. A well-trained backdoored model has a satisfying performance on clean samples, while consistently making wrong predictions once the triggers are added into the input. In popular attack mechanisms, such as insertion-based attacks, the triggers are preselected words (Kurita et al., 2020), meaningful sentences (Dai et al., 2019), or characters (Chen et al., 2021).

To address backdoor attacks, existing methods mainly fall into two categories: 1) Defense: mitigating the attack effect by removing the trigger from models or inputs, and 2) Detection: directly detecting whether the model is backdoored or clean. Despite the development of defense methods (Qi et al., 2021a; Yang et al., 2021b; Lyu et al., 2022c), detecting whether a model has been backdoor attacked is less explored. In this study, we focus on detection as it is important in practice to identify malicious models before deployment and thereby preventing potential damages. T-Miner (Azizi et al., 2021) identifies backdoors by finding outliers in an internal representation space. AttenTD (Lyu et al., 2022b) detects backdoors by checking the attention abnormality given a set of neutral words. PICCOLO (Liu et al., 2022) leverages a word discriminativity analysis to distinguish backdoors.

All these detection methods rely on reconstructing potential triggers or intermediate feature representation. This makes these methods rather sensitive to the backbone architecture and to the NLP task. When generalizing to a different backbone or a different NLP task, one may have to redesign the method or re-tune the hyperparameters. Indeed, most existing detection methods focus on common sentence classification (SC) tasks, such as sentiment analysis. It is very hard to generalize them to tasks requiring a structured output, *e.g.*, named entity recognition (NER) and question answering (QA).

In this paper, we propose *the first task-agnostic backdoor detector that directly detect backdoored models for different NLP tasks*. A task-agnostic backdoor detector has multiple benefits. First, it will be easy to be deployed in the field, without redesigning the algorithm or re-tuning hyperparamters for different tasks. Second, a task-agnostic detector can fully exploit training model samples from different tasks and achieve better overall performance. Finally, a task-agnostic backdoor detector provides the opportunity to identify the intrinsic characteristic of backdoors shared across different tasks. This will advance our fundamental understanding of backdoor attack and de-

Findings of the Association for Computational Linguistics: NAACL 2024, pages 2808–2822

Input Sample	Trojan	Labels	Logits	Softmax	LogSoftmax
Today is *really* a good day.	Clean	*pos*	*3.68*	*0.9999*	*-0.0001*
		neg	-5.23	0.0001	-8.9101
	Backdoored	*pos*	*-3.96*	*0.0026*	*-5.9426*
		neg	1.98	0.9974	-0.0026

*Ground Truth label of this input sample is **positive**.*
really in 'Today is *really* a good day.' *is a real trigger.*

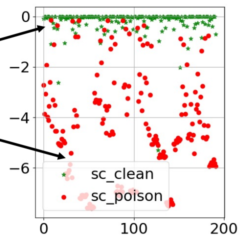

Figure 1: **In the left Table**, the clean model's prediction for an input sample is positive with high confidence, as indicated by a substantial log-softmax value. Conversely, the backdoored model shows low confidence in the correct positive label, reflected by a diminished log-softmax value. **In the right Figure**, given input samples, we plot log-softmax values of ground truth label from both clean (green stars) and backdoored (red dots) models, highlighting a distinct separation in logits distribution. y axis represents the log-softmax value, x axis represents the value count. For brevity, *logit value* will be used throughout the paper to refer to *log-softmax logit value*.

fense, and advance our knowledge of NLP models in general.

Our method, TABDet (*Task-Agnostic Backdoor Detector*), constitutes two main technical contributions. **First**, unlike most existing detection methods, we propose to only use the final layer output logits. Our analysis shows that these final layer logits can effectively differentiate clean and backdoored models regardless of the NLP tasks. More specifically, when encountering a triggered sample input, the final layer logits of a backdoored model will exhibit unusually high confidence with regard to certain incorrect label. As shown in Figure 1, such behavior manifests across different NLP tasks. Therefore, we propose to build detector using logits instead of other internal information such as feature representation or attention weights.

There are more challenges we need to address. During detection, we do not know the real trigger. Instead, we could only use a large set of trigger candidates. When encountering these trigger candidates, the abnormal logits behavior still exists (Figure 2(1)). However, not surprisingly, the signal also gets noisy (Figure 2(2)). Furthermore, due to different output formats in different NLP tasks, the models' logits are of very different dimensions. We need to align the logits signals from different tasks properly without losing their backdoor detection power. To address these challenges, **our second technical contribution** is a novel logits pooling method to refine and unify the representations of logits from models for different NLP tasks. As shown in Figure 2(3), the refined logit representations preserve the strong detection power and is well aligned across tasks.

In summary, we propose the first task-agnostic backdoor detector with the following contributions:

- We only rely on the final layer logits for the detection.
- We propose an efficient logits pooling method to refine and unify logit representations across models from different tasks.
- Using the logit representation as features, we train the proposed backdoor detector that can fully learn from models of different tasks and achieve superior performance.

Empirical results demonstrate the strong detection power of our detector (TABDet) across different tasks including sentence classification, question answering and named entity recognition. Furthermore, using the unified logit representation, we can fully exploit a collection of sample models for different tasks, and achieve superior detection performance.

2 Related Work

Insertion-based Textual Backdoor Attacks. Existing backdoor attacks in NLP applications are mainly through various data poisoning manners by inserting trigger to clean samples (Lyu et al., 2023a). Several prominent insertion-based backdoor attacks are: Kurita et al. (2020) randomly insert rare word triggers (*e.g.*, 'cf', 'mn', 'bb', 'mb', 'tq') to clean inputs. AddSent (Dai et al., 2019) inserts a consistent sentence, such as 'I watched this 3D movie last weekend.', into clean inputs as the trigger to manipulate the classification systems. BadNL (Chen et al., 2021) inserts characters, words or sentences as triggers. In our paper, we focus on above traditional insertion-based textual backdoor attacks.

Detection against Textual Backdoor. Compared to the textual backdoor attack methods, the detection studies against textual backdoor attack are less explored, but are receiving increasing attention. T-Miner (Azizi et al., 2021) trains a generator to generate trigger candidates and finds outliers in an internal representation space to identify backdoors. AttenTD (Lyu et al., 2022b) discriminates whether the model is a clean or backdoored model by checking the attention abnormality given a set of neutral trigger candidates. PICCOLO (Liu et al., 2022) leverages a word discriminativity analysis to distinguish backdoors. Shen et al. (2022) propose an optimization method with dynamic bound-scaling for effective backdoor detection.

3 TABDet

In this section, we propose our unified backdoor detection algorithm, named ***TABDet*** (*Task-Agnostic Backdoor Detector*). TABDet employs a systematic approach: **1) Logit Features Extraction**: We extract logit features (*i.e.*, final layer logits) (Section 3.1). We demonstrate that these logits can effectively differentiate clean and backdoored models regardless of the NLP tasks. **2) Representation Refinement**: We propose a representation refinement strategy to extract high-quality representation, and normalize representation dimensions across different NLP tasks (Section 3.2.) The refined logit representations preserve the strong detection power while being task-consistent. **3) Backdoor Detector**: Finally, we train a unified classifier to detect backdoors given a suspicious model (Section 3.3). The overall architecture of our method is shown in Figure 3.

3.1 Logit Features Extraction

In the quest to distinguish between backdoored and clean models in a task- and architecture-agnostic manner, we proposed to rely on logit outputs. Unlike intermediate features such as attention weights or neuron outputs, logits offer a more standardized and consistent information across different NLP tasks and architectures. This makes them much more reliable for comparative study, compared with intermediate features. By focusing on logits, we ensure a more robust approach to identify potentially compromised models across a variety of tasks such as sentence classification (SC), question answering (QA), and named entity recognition (NER).

In Section 3.1.1, we provide details on how to generate the logit features. We insert different trigger candidates (from a pre-defined Trigger Candidate Set Δ) into a fixed set of clean samples, producing so-called *perturbed samples*. We provide those perturbed samples to suspicious models, and collect the output logits as logit features of the model.

In Section 3.1.2, we provide an empirical study to justify the choice. We demonstrate that final layer logits are effective in differentiating clean and backdoored models across various NLP tasks. When real triggers are inserted into samples, there are distinct differences in logit features between clean and backdoored models, as evidenced in specific logit distributions (Figure 4, top row). In practice, we have no knowledge of real triggers. Alternatively, a large trigger candidate set is used to generate perturbed samples. We show that even with a large trigger candidate set, abnormal logit behavior persists, allowing us to effectively identify backdoored models without knowing the actual trigger (Figure 4, bottom row).

3.1.1 Technical Details

In this subsection, we focus on technical details, including how to generate a trigger candidate set, and how to use the trigger candidates to generate perturbed samples and logit features.

Trigger Candidate Set Δ. Though the real trigger is super powerful during the backdoor attack, reconstructing the exact real trigger is a very challenging problem. That is because the discrete inputs in NLP are hard to reverse and the number of words in triggers is unknown. We introduce a diverse Trigger Candidate Set Δ, which, despite not containing the exact triggers, is robust enough to induce characteristic logit perturbations in compromised models. This set is derived from the comprehensive Google Books 5gram Corpus, encompassing 62599 potential triggers. This approach allows for the activation of backdoor patterns even without precise trigger knowledge, as supported by our findings presented in Table 5.

Extracting Logit Features. For every trigger candidate $\delta \in \Delta$, we insert it to a clean sample set (8 clean samples) with 2 different locations (front location and rear location)[1]. This creates

[1] In NER task, there are three types of attacks. One of the attack 'local', will only be activated if the trigger is in the first half, or the last half of the sentences. So we inject the trigger candidates to front or rear location in order to fully activate the attack.

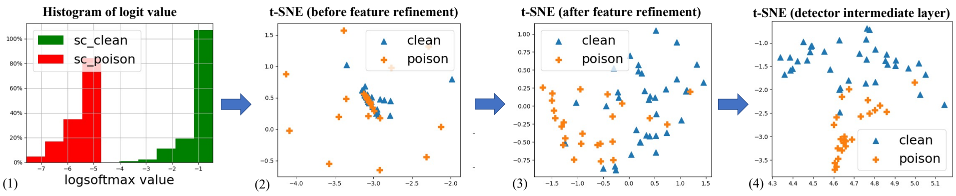

Figure 2: 1) Histogram of model's final layer logits (log-softmax) given trigger candidates. Histogram (only plot the lowest 0.01% value) shows clear gap between clean models and backdoored models. 2) t-SNE visualization of logit features prior to feature refinement, illustrating indistinct clustering. 3) Post-refinement t-SNE visualization, showing improved distinction between clean and poisoned models. 4) t-SNE plot of features extracted from the learnable backdoor detector's intermediate layer, indicating further enhancement in the separability of representations from clean and backdoored models.

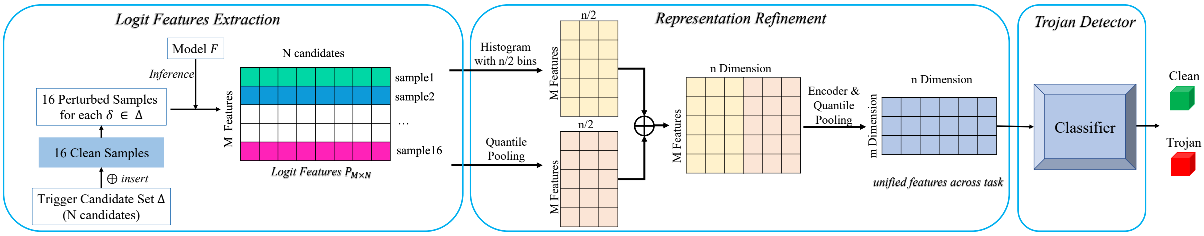

Figure 3: The overall TABDet framework consists of three key components: the *Logit Features Extraction* module, which extracts the final layer logits from a given model; the *Representation Refinement* module, which utilizes histogram and quantile pooling to produce high-quality, task-consistent representations; and the *Backdoor Detector*, which employs a simple MLP classifier to accurately distinguish between clean and trojan models. This architecture ensures robust backdoor detection across various NLP tasks.

16 perturbed samples ($S[\delta]$) per candidate. These samples are processed by the model to gather logits, which are then assembled into a logit feature set for analysis. The feature dimensions vary by task: In SC task, we select logits from ground truth label and non-ground truth label respectively, which yields to the dimension of logit features $P[\delta]$: $M_{sc} = 32$ (16×2). In QA task, we compute 6 logits related to the start point and the end point of the answer[2], which yields to a feature dimension $M_{qa} = 96$ (16×6). In NER task, we select the logits of all valid tokens in 16 samples, which yields to a feature dimension $M_{ner} = 228$ (Notice that the number of valid tokens in 16 samples may be different).

3.1.2 Justification: Logit Features Reveal Backdoors

In this subsection, we validate the efficacy of logit features in distinguishing between clean and backdoored models for various NLP tasks. We start with using true triggers. Furthermore, we show that given a large trigger candidate set Δ, the abnormal logits behavior still exists.

[2]Please refer to Appendix A.1 for more details.

First, we illustrate that given the real trigger, the final layer logits can effectively differentiate clean and backdoored models regardless of the NLP tasks. We insert the real trigger into aforementioned 16 samples (fixed samples for fixed tasks), and record the logit features (the final layer logits after log-softmax) associated with the ground truth labels (see Figure 1 for illustration). As shown in Figure 4 top row, there are clear differences in logit features between the clean models and backdoored models. This discrepancy is particularly pronounced with the ground truth labels, where backdoored models exhibit significantly reduced logits. This is desired for any successfully backdoored models as they are trained to have such a behavior. This property should commonly hold regardless of the NLP tasks. This phenomenon motivates us to use logit features as the potential features for backdoor detection.

Second, we establish that even without exact triggers, the presence of a diverse trigger candidate set Δ can still elicit abnormal logit responses indicative of a backdoored model. For every trigger candidate $\delta \in \Delta$, we can form M dimension features. For better visualization, we pick the logits of real labels for each sentence. For example, in SC,

Algorithm 1 Logit Features Extraction

Input: A trigger candidate set Δ, The clean samples set D, The suspicious model F, Logits extractor A

Output: Logit features $P_{M \times N}$, N is the trigger candidate number in Δ

\# Perturbed Samples (PS) Construction

Let the PS set $S = dict()$

for δ in Δ **do**

  \# Construct perturbed samples for trigger candidate δ

  $S[\delta] = \emptyset$

  for $(\mathbf{x}, y)$ in D **do**

    $\tilde{\mathbf{x}} := \mathbf{x} \oplus \delta$ # $\oplus$ is insertion operation

    $S[\delta] = S[\delta] \cup \tilde{\mathbf{x}}$

  end for

end for

Let logit features set $P = dict()$

for δ in Δ **do**

  $P[\delta] = []$

  for $\tilde{\mathbf{x}}$ in $S[\delta]$ **do**

    $P[\delta]$ = concat($A(F(\tilde{\mathbf{x}}))$)

  end for

  \# Dimension of $P[\delta]$ is M. Notice M_{SC}, M_{QA}, M_{NER} in three tasks are different

end for

Return $P_{M \times N}$ for each model F

the sentence 'I like the food.' is a positive sentence, so we picked the logits of positive label. We only plot the lowest 0.01% values due to a large number of features for 62599 trigger candidates. Figure 4 bottom row shows that the distinct logit distributions for clean and backdoored models are evident, even in the absence of the actual trigger.

However, the variability in logit dimensions across different NLP tasks and the inherent noise in the logit signals, as illustrated in Figure 2(2) and Figure 6(top row), present challenges in developing a unified backdoor detector. To overcome this and retain the detection power, we introduce a *Representation Refinement* component, which we discuss in the following section. This component is designed to harmonize the logit signals for effective backdoor detection across varied NLP tasks.

3.2 Representation Refinement

In the second component, we refine the logit features into high-quality representations, ensuring consistency across varying architectures and tasks. This critical process enhances the raw logits, facilitating the development of a robust, task-agnostic backdoor detection framework.

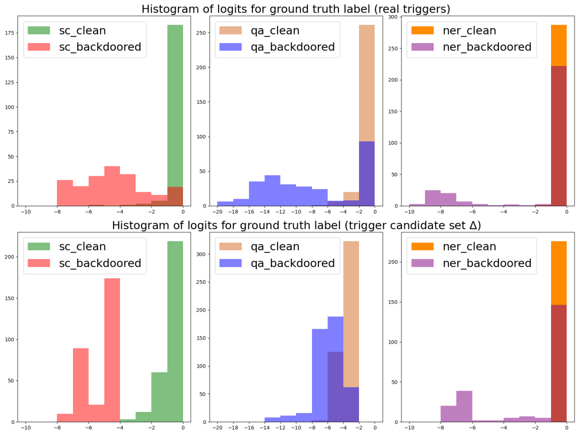

Figure 4: The histogram illustrates logit distributions for the ground truth label across three NLP tasks, differentiating between clean and backdoored models. x axis is the logit values, y axis is the count of logits in corresponding bins. **Top Row** shows clear separation in logit values when real triggers are used. **Bottom Row**, with a large set of trigger candidates Δ (only display the lowest 0.01% values), reveals persisting abnormal logit behaviors in backdoored models, demonstrating the robustness of logits as indicators of model integrity.

The major challenge lies in aligning the logit features from models for different tasks. The logit features from different tasks have varying dimensions. It is very hard to find correspondence; a logit output for SC is not comparable with a logit output for NER. The key insight is that it is indeed sufficient to compare the logit features at a distribution level. This inspires us to propose strategies like qantile pooling and histogram descriptors. The quantile pooling technique strategically reduces feature space dimensionality by focusing on its quantiles. The histogram computing further refines this by aggregating logit features into a concise, histogram-based format. These two techniques, together, providing a balanced and comprehensive view of the logits' distribution for effective backdoor detection.

Quantile Pooling. We first propose a quantile pooling scheme. We effectively reduce the dimensionality of our feature space while preserving the most critical information embedded in the logits. It enhances the efficacy of our pooling strategy in differentiating between clean and backdoored models. The quantile index generation is followed by

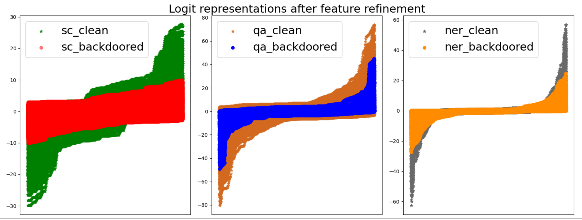

Figure 5: The refined feature representations effectively differentiate between clean and backdoored models across various NLP tasks. Each color on the figure corresponds to a unique model, with the plotted points indicating individual feature values after refinement in one model. The x-axis labels the feature indices, and the y-axis their corresponding values. The distributions are not only efficient in separation but also exhibit consistency across various NLP tasks, highlighting the effectiveness of the feature refinement process.

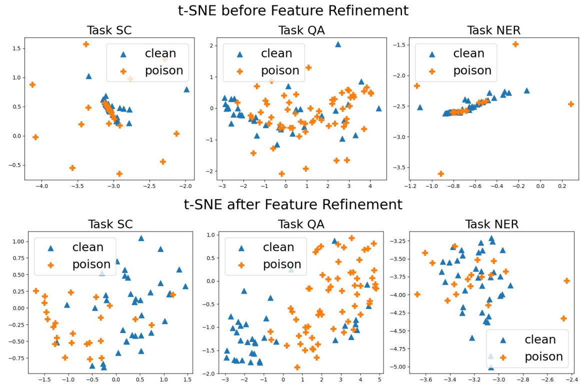

Figure 6: t-SNE visualization on logit representation before (Top Row) and after (Bottom Row) representation refinement. Each dot indicates one model. By refinement, the representation quality significantly improves.

$$
\begin{aligned}
q^1 &= \left[q_0, q_1, \ldots, q_{\frac{n}{2}-1}\right], \\
q_i^1 &= \left(1 + \frac{10}{\frac{n}{2}-1}\right)^{-i}, \forall i \in \left\{0, 1, \ldots, \frac{n}{2} - 1\right\} \\
q^2 &= \text{reverse}\left(q^1\right), \\
q &= \left[\frac{q^2}{2}, \frac{1-q^1}{2} + 0.5\right]
\end{aligned}
$$

- **Non-linear Scale q^1:** The formula $\left(1 + \frac{10}{n/2-1}\right)^{-i}$ creates a non-linear scale. This allows the indices to be more densely packed at the ends of the distribution and sparser in the middle. This non-linear scale is beneficial when the distribution of logits is not uniform, emphasizing the tails of the distribution where extreme values are present.
- **Balancing the Distribution:** Creating q^2 as a reversed version of q^1 and then concatenating $\frac{q2}{2}$ with $\frac{1-q^1}{2} + 0.5$ balances the distribution of indices. The division by 2 and the addition of 0.5 ensure that the indices are evenly distributed across the entire range of logits.

The aim is to obtain a set of indices representative of the entire distribution of logits. The generated quantile index ensures that the selected indices capture the essence of the entire distribution. The mathematical expressions are chosen to create a balanced and non-linear distribution of indices, ensuring both common and rare values in the logits are represented. The code implementation can be found in Appendix A.6.

Histogram Computing. For our second refinement strategy, we employ histogram binning to analyze the distribution of representations. Each column of length N is sorted and binned into $n/2$ segments, counting the quantity within each. This process yields a dimensionally reduced matrix of size $M \times n/2$, where each column represents a histogram of counts per bin. These histograms uniformly partition the range of each original column, providing a different perspective on the representation distribution. n in our algorithm is a hyper-parameter that specifies the reduced dimension.

3.2.1 Rationale: Representation Refinement Strategy

In Figure 5, we display the distribution of logit representations post-refinement, showcasing their strong discriminatory potential even without further learning. Complementing this, t-SNE (Liu et al., 2016) visualizations in Figure 6(botttom) depict each model's refined logit representation as a distinct point. These visualizations clearly illustrate the heightened separation and enhanced clarity of the refined representations compared to their initial, coarse counterparts. These observations underscore the efficacy of our refinement methods and point towards the feasibility of a backdoor detection algorithm that utilizes these refined representations for training classifiers.

3.3 Backdoor Detector

After the representation refinement component, we generalize the representation into identical dimension. We then train a Trojan detector, *i.e.*, a MLP classifier, to discriminate whether the suspicious model is a clean model or backdoored model.

Algorithm 2 Representation Refinement

Input: Logit features $P_{M\times N}$, N is the trigger candidate number in Δ, M is the feature dimension, which is various in different tasks
Output: A unified feature $FR_{m\times n}$, where m, n are identical across tasks
Dimension reduction along N dimension
$A_{M\times n/2}$ = Histogram($P_{M\times N}$)
$B_{M\times n/2}$ = Quantile($P_{M\times N}$)
$C_{M\times n}$ = combining $A_{M\times n/2}$ and $B_{M\times n/2}$
Dimension reduction along M dimension
$FR_{m\times n}$ = Quantile($C_{M\times n}$)
return refined feature $FR_{m\times n}$

4 Experiments

4.1 Experimental Settings

Datasets and Models. We focus on three NLP tasks: sentence classification task (SC), question answering task (QA) and named entity recognition task (NER). And the model architectures are Roberta (Liu et al., 2019), DistilBERT (Sanh et al., 2019) and ELECTRA (Clark et al., 2020), mixed in three tasks. We leverage 420 models from the training and test sets of TrojAI NLP-Summary Challenge (Learderbord, 2023; Description, 2023). It provides a training set of 210 models, in which 102 are infected with backdoors, and a test set of 210 models , in which 101 are infected with backdoors. The statistics information is shown in Table 1. The SC models are trained with IMDB dataset (Maas et al., 2011), the QA models are trained with SQuAD v2 dataset (Rajpurkar et al., 2016; v2, 2023) and the NER models are trained with CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003), respectively. We only consider the standard insertion-based textual backdoor attacks, AddSent (Dai et al., 2019) and BadNL (Chen et al., 2021), in our experiments. The triggers are words, phrases or sentences. A detailed description can be found in Appendix A.2.

Table 1: Training and test models statistics.

	Training			Test		
	Positive	Negative	Total	Positive	Negative	Total
SC	24	36	60	31	37	68
QA	60	36	96	54	42	96
NER	18	36	54	16	30	46

Detection Baselines. We implement three textual detection baselines[3], *e.g.*, T-Miner, AttenTD and PICCOLO. T-Miner (Azizi et al., 2021) trains a sequence-to-sequence generator and finds outliers in an internal representation space to identify backdoors. AttenTD (Lyu et al., 2022b) detects whether the model is a benign or backdoored model by checking the attention abnormality given a set of neural words. PICCOLO (Liu et al., 2022) leverages a word discriminativity analysis to distinguish backdoors.

Implementation Details. When training the backdoor classifier, we involve the hyperparameter tuning in order to get a more robust classifier. Hyperparameters include the hidden dimensions number, layers number in each MLP, the quantile pooling interval, Adam optimizer learning rate. We use HyperOPT[4] hyperparameter optimization tool, via 8-fold cross validation on the training set.

4.2 Detection Results

Baseline Detection Performance. We provide the detection evaluation with existing textual baselines. In their original experiments, T-Miner (Azizi et al., 2021)[5] and AttenTD (Lyu et al., 2022b) only experiment on SC task, and PICCOLO (Liu et al., 2022) experiments on SC and NER tasks. We follow their default experiment settings. Table 2 shows that our TABDet outperforms three baselines in all three tasks. The T-Miner is mainly designed for LSTM-based language models, thus does not perform good on complicated transformer architectures. AttenTD's focus on attention abnormalities falls short due to noise and computational inefficiency. PICCOLO, while performing well on SC and NER, does not leverage other tasks information and lags in detection capabilities.

Table 2: Detection performance (AUC) compared to baselines. '−' indicates not applicable.

	SC	QA	NER
T-Miner	0.50	-	-
AttenTD	0.60	-	-
PICCOLO	0.87	-	0.72
TABDet (Single)	0.92	0.92	0.85
TABDet	**0.98**	**0.93**	**0.86**

TABDet Detection Performance. TABDet,

[3]Notice that detection and defense are two different research categories, so we do not involve defense baselines here.

[4]https://github.com/hyperopt/hyperopt

[5]Due to the vocabulary size limitation, we only implement T-Miner on the ELECTRA architecture, with totally 19 models.

trained across three NLP tasks, establishes a unified detection approach. As demonstrated in Table 2, it surpasses baseline methods in all tasks. The performance on NER task is not as good as the performances on other two tasks. That is because the challenge of variability and ambiguity in natural language is particularly prominent in NER. Entities can have different meanings based on their usage and context, and they can easily change once a random trigger candidate is inserted. That makes the backdoor detection on NER task difficulty.

TABDet Detection in Individual Tasks. We also evaluate our framework only with single task. In this setting, we train three individual backdoor detectors for three different tasks. In Table 2, Row *TABDet (Single)*: Our TABDet, when applied to single tasks, shows good detection performance, comparing to the performance with other textual detection baselines. This validates the potency of our feature refinement strategy even within the constraints of individual tasks. However, when compared to the multi-task model training (Table 2, Row *TABDet*), the single-task detectors exhibit slightly reduced efficacy. This highlights the advantage of a multi-task perspective, where TABDet harnesses commonalities across tasks to enhance detection capabilities, as evidenced by the superior performance in multi-task settings.

4.3 Ablation Study

In this section, we investigate the impact of trigger candidate set size, different pooling strategies, histogram features, and partial trigger effect.

Impact of Trigger Candidate Set Size. We validate our TABDet with different Trigger Candidate Set Δ. Employing 2gram and 5gram sets from Google Books Ngram Corpus (Michel et al., 2011; Lin et al., 2012), with 24,267 and 62,599 candidates respectively, we observed improved detection performance with the increase in Δ size. In Table 3, the overall AUC achieves 0.94 with 5gram, with AUC in individual task 0.98, 0.93 and 0.86 for SC, QA and NER respectively.

Table 3: Impact of different Trigger Candidate Set Δ.

Trigger Candidate Set	Number of Triggers	SC	QA	NER	Overall
2gram	24267	0.78	0.88	0.73	0.81
5gram	62599	0.98	0.93	0.86	0.94

Impact of Pooling Strategies and Histogram Features. First, we examined the effects of different pooling strategies on dimension reduction, contrasting quantile pooling with max, min, and average pooling, as they are common operations in practice. We set the output dimension the same as our quantile pooling. Our findings, outlined in Table 4, reveal quantile pooling's superior ability to retain outlier features indicative of backdoors, thereby enhancing detection performance over the other methods. Max/min/average pooling strategies tend to smooth out critical features, diluting backdoor signals, whereas quantile pooling preserves them. Secondly, relying solely on histogram features does not match the efficacy achieved by TABDet's comprehensive approach.

Table 4: Ablation study on different pooling strategies and histogram features.

		SC	QA	NER	Overall
Pooling	Max	0.30	0.58	0.62	0.61
	Min	0.40	0.38	0.74	0.56
	Ave	0.49	0.38	0.63	0.59
Only Histogram		0.73	0.78	0.82	0.78
TABDet		0.98	0.93	0.86	0.94

Impact of Partial Triggers. In this ablation study, we explored how partial triggers—snippets of a complete trigger phrase or sentence—can still effectively activate backdoors in models. We found that even two-word from longer triggers can prompt the model to produce the targeted predictions, altering the logit representations significantly. This was empirically validated across three NLP tasks. The robust impact of these partial triggers supports the effectiveness of using a broad and extensive trigger candidate set for backdoor detection, as indicated by our results in Table 5.

Table 5: Attack Performance with Partial Triggers. We report the source label accuracy for SC and NER, report exact match sore for QA.

		SC	NER	QA
Clean Models	CleanSamples	0.98	0.92	88.75
backdoor Models	CleanSamples	0.97	1	88.58
	PoisonedSamples-RealTrigger	0.02	0	19.75
	PoisonedSamples-PartialTrigger	0.2	0.18	23.67

Detection Effectiveness on Advanced Insertion-based Attacks. We also extend our experiments to include two advanced insertion-based textual backdoor attacks, such as EP (Yang et al., 2021a) and RIPPLEs (Kurita et al., 2020)[6]. EP and RIPPLES modify different levels of weights/embeddings, such as input word embedding. Given that

[6]We implement the backdoor attack with OpenBackdoor toolkit: `https://github.com/thunlp/OpenBackdoor`.

EP and RIPPLES are primarily designed for sentence classification tasks, we limited their implementation to this specific task, thus this ablation study can only partially validate the detection effectiveness of our TABDet. Details in Appendix A.3.

Table 6 presents the detection performance of TABDet across different textual backdoor attacks. Our findings indicate that the detection effectiveness of TABDet is comparable across the additional textual backdoor attack baselines. This consistency in performance highlights the robustness of TABDet, attributable to our detection mechanism that focuses on the output logits abnormalities of the models. Irrespective of the textual attack's type, a successfully backdoored model tends to show comparable patterns in the logits of the last layer, specifically in terms of switching the correct label to an incorrect one.

Table 6: Detection effectiveness compared with basic attacks (AddSent/BadNL) and advanced attacks (EP/RIPPLES).

	TP	FP	FN	TN	AUC
AddSent/BadNL	10	0	1	9	0.95
EP/RIPPLES	10	0	1	9	0.95

5 Conclusion

In this paper, we pioneered TABDet (*Task-Agnostic Backdoor Detector*), the first unified detector of its kind that operates effectively across three key NLP tasks (sentence classification, question answering, and named entity recognition). The proposed TABDet utilizes the model's final laye logits, and a unique feature refinement strategy, resulting in a versatile and high-quality representation applicable to sentence classification, question answering, and named entity recognition tasks. While existing detectors mainly focus on SC and NER tasks, TABDet can detect backdoors from all SC, QA and NER tasks, achieving the new state-of-the-art performance on backdoor detection.

Limitations

There are several limitations of our proposed methods. 1) TABDet is only effective against standard insertion-based attack, and can not deal with more advanced textual backdoor attack such as style transfer based attack (Qi et al., 2021c,b). As future work, we should investigate detection against a broader range of textual backdoor attacks. 2) We only test three popular NLP tasks, namely sentence classification, question answering and named entity recognition tasks, and future work should explore backdoor detection on more NLP tasks. 3) Detection on NER task performs not as good as SC and QA. A more efficient strategy towards NER task should be developed.

Ethics Statement

In this paper, we propose a detection strategy against textual backdoor attacks. Our codes and datasets will be publicly available. We conduct such detection framework only for research purpose and do not intend to harm the community.

Acknowledgements

We thank anonymous reviewers for their constructive feedback. This effort was partially supported by the Intelligence Advanced Research Projects Agency (IARPA) under the Contract W911NF20C0038. The content of this paper does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

Ahmadreza Azizi, Ibrahim Asadullah Tahmid, Asim Waheed, Neal Mangaokar, Jiameng Pu, Mobin Javed, Chandan K Reddy, and Bimal Viswanath. 2021. T-miner: A generative approach to defend against trojan attacks on dnn-based text classification. *arXiv preprint arXiv:2103.04264*.

Hanning Chen, Ali Zakeri, Fei Wen, Hamza Errahmouni Barkam, and Mohsen Imani. 2023a. Hypergraf: Hyperdimensional graph-based reasoning acceleration on fpga. In *2023 33rd International Conference on Field-Programmable Logic and Applications (FPL)*, pages 34–41. IEEE.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*, pages 554–569.

Ziheng Chen, Fabrizio Silvestri, Gabriele Tolomei, Jia Wang, He Zhu, and Hongshik Ahn. 2022a. Explain the explainer: Interpreting model-agnostic counterfactual explanations of a deep reinforcement learning agent. *IEEE Transactions on Artificial Intelligence*.

Ziheng Chen, Fabrizio Silvestri, Jia Wang, Yongfeng Zhang, and Gabriele Tolomei. 2023b. The dark side of explanations: Poisoning recommender systems with counterfactual examples. In *Proceedings of the 46th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 2426–2430.

Ziheng Chen, Fabrizio Silvestri, Jia Wang, He Zhu, Hongshik Ahn, and Gabriele Tolomei. 2022b. Relax: Reinforcement learning agent explainer for arbitrary predictive models. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pages 252–261.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. 2022. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. *arXiv preprint arXiv:2206.08514*.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878.

Data Description. 2023. https://pages.nist.gov/trojai/docs/nlp-summary-jan2022.html#nlp-summary-jan2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Xinyu Dong, Rachel Wong, Weimin Lyu, Kayley Abell-Hart, Jianyuan Deng, Yinan Liu, Janos G Hajagos, Richard N Rosenthal, Chao Chen, and Fusheng Wang. 2023. An integrated lstm-heterorgnn model for interpretable opioid overdose risk prediction. *Artificial intelligence in medicine*, 135:102439.

Pytorch Log Softmax Function. 2023a. https://pytorch.org/docs/stable/generated/torch.nn.LogSoftmax.html.

Pytorch Softmax Function. 2023b. https://pytorch.org/docs/stable/generated/torch.nn.Softmax.html.

T Gu, B Dolan-Gavitt, and SG BadNets. 2017. Identifying vulnerabilities in the machine learning model supply chain. In *Proceedings of the Neural Information Processing Symposium Workshop Mach. Learning Security (MLSec)*, pages 1–5.

Saumya Gupta, Xiaoling Hu, James Kaan, Michael Jin, Mutshipay Mpoy, Katherine Chung, Gagandeep Singh, Mary Saltz, Tahsin Kurc, Joel Saltz, et al. 2022. Learning topological interactions for multi-class medical image segmentation. In *European Conference on Computer Vision*, pages 701–718. Springer.

Chengyue Huang, Anindita Bandyopadhyay, Weiguo Fan, Aaron Miller, and Stephanie Gilbertson-White. 2023. Mental toll on working women during the covid-19 pandemic: An exploratory study using reddit data. *PloS one*, 18(1):e0280049.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806.

TrojAI Learderbord. 2023. https://pages.nist.gov/trojai/.

Maolin Li, Peng Ling, Shangsheng Wen, Xiandong Chen, and Fei Wen. 2023. Bubble-wave-mitigation algorithm and transformer-based neural network demodulator for water-air optical camera communications. *IEEE Photonics Journal*.

Zhenglin Li, Haibei Zhu, Houze Liu, Jintong Song, and Qishuo Cheng. 2024. Comprehensive evaluation of mal-api-2019 dataset by machine learning in malware detection. *arXiv preprint arXiv:2403.02232*.

Jiacheng Liang, Songze Li, Bochuan Cao, Wensi Jiang, and Chaoyang He. 2021. Omnilytics: A blockchain-based secure data market for decentralized machine learning. *arXiv preprint arXiv:2107.05252*.

Jiacheng Liang, Ren Pang, Changjiang Li, and Ting Wang. 2023. Model extraction attacks revisited. *arXiv preprint arXiv:2312.05386*.

Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. 2012. Syntactic annotations for the google books ngram corpus. In *Proceedings of the ACL 2012 system demonstrations*, pages 169–174.

Han Liu, Yuhao Wu, Zhiyuan Yu, Yevgeniy Vorobeychik, and Ning Zhang. 2023a. Slowlidar: Increasing the latency of lidar-based detection using adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5146–5155.

Han Liu, Yuhao Wu, Zhiyuan Yu, and Ning Zhang. 2024. Please tell me more: Privacy impact of explainability through the lens of membership inference attack. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 120–120. IEEE Computer Society.

Han Liu, Yuhao Wu, Shixuan Zhai, Bo Yuan, and Ning Zhang. 2023b. Riatig: Reliable and imperceptible adversarial text-to-image generation with natural prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20585–20594.

Shun Liu, Kexin Wu, Chufeng Jiang, Bin Huang, and Danqing Ma. 2023c. Financial time-series forecasting: Towards synergizing performance and interpretability within a hybrid machine learning approach. *arXiv preprint arXiv:2401.00534*.

Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. 2016. Visualizing high-dimensional data: Advances in the past decade. *IEEE transactions on visualization and computer graphics*, 23(3):1249–1268.

Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Piccolo: Exposing complex backdoors in nlp transformer models. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2025–2042. IEEE.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Weimin Lyu, Xinyu Dong, Rachel Wong, Songzhu Zheng, Kayley Abell-Hart, Fusheng Wang, and Chao Chen. 2022a. A multimodal transformer: Fusing clinical notes with structured ehr data for interpretable in-hospital mortality prediction. In *AMIA Annual Symposium Proceedings*, volume 2022, page 719. American Medical Informatics Association.

Weimin Lyu, Sheng Huang, Abdul Rafae Khan, Shengqiang Zhang, Weiwei Sun, and Jia Xu. 2019. Cuny-pku parser at semeval-2019 task 1: Cross-lingual semantic parsing with ucca. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 92–96.

Weimin Lyu, Songzhu Zheng, Haibin Ling, and Chao Chen. 2023a. Backdoor attacks against transformers with attention enhancement. In *ICLR 2023 Workshop on Backdoor Attacks and Defenses in Machine Learning*.

Weimin Lyu, Songzhu Zheng, Tengfei Ma, and Chao Chen. 2022b. A study of the attention abnormality in trojaned berts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4727–4741.

Weimin Lyu, Songzhu Zheng, Tengfei Ma, Haibin Ling, and Chao Chen. 2022c. Attention hijacking in trojan transformers. *arXiv preprint arXiv:2208.04946*.

Weimin Lyu, Songzhu Zheng, Lu Pang, Haibin Ling, and Chao Chen. 2023b. Attention-enhancing backdoor attacks against bert-based models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10672–10690.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Google Books Team, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, et al. 2011. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182.

Lu Pang, Tao Sun, Haibin Ling, and Chao Chen. 2023. Backdoor cleansing with unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12218–12227.

Na Pang, Li Qian, Weimin Lyu, and Jin-Dong Yang. 2019. Transfer learning for scientific data chain extraction in small chemical corpus with joint bert-crf model. In *BIRNDL@ SIGIR*, pages 28–41.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. Onion: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021b. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4580.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021c. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 443–453.

Wenpin Qian, Shuqian Du, Kun Chi, Huan Ji, and Kuo Wei. 2024. Next-generation artificial intelligence innovative applications of large language models and new methods. *OLD AND NEW TECHNOLOGIES OF LEARNING DEVELOPMENT IN MODERN CONDITIONS*, page 262.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Guangyu Shen, Yingqi Liu, Guanhong Tao, Qiuling Xu, Zhuo Zhang, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Constrained optimization with dynamic bound-scaling for effective nlp backdoor defense. In *International Conference on Machine Learning*, pages 19879–19892. PMLR.

Saurabh Srivastava, Chengyue Huang, Weiguo Fan, and Ziyu Yao. 2023. Instance needs more care: Rewriting prompts for instances yields better zero-shot performance. *arXiv preprint arXiv:2310.02107*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, page 142–147, USA. Association for Computational Linguistics.

Squad v2. 2023. https://huggingface.co/datasets/squad_v2.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Fan Wang, Saarthak Kapse, Steven Liu, Prateek Prasanna, and Chao Chen. 2021. Topotxr: a topological biomarker for predicting treatment response in breast cancer. In *International Conference on Information Processing in Medical Imaging*, pages 386–397. Springer.

Fan Wang, Huidong Liu, Dimitris Samaras, and Chao Chen. 2020. Topogan: A topology-aware generative adversarial network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 118–136. Springer.

Ziyang Wang, Nanqing Dong, and Irina Voiculescu. 2022a. Computationally-efficient vision transformer for medical image semantic segmentation via dual pseudo-label supervision. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1961–1965. IEEE.

Ziyang Wang and Congying Ma. 2023. Dual-contrastive dual-consistency dual-transformer: A semi-supervised approach to medical image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 870–879.

Ziyang Wang, Will Zhao, Zixuan Ni, and Yuchen Zheng. 2022b. Adversarial vision transformer for medical image semantic segmentation with limited annotations. In *BMVC*, page 1002.

Jun Wu, Xuesong Ye, and Yanyuet Man. 2023a. Bottrinet: A unified and efficient embedding for social bots detection via metric learning. In *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–6. IEEE.

Jun Wu, Xuesong Ye, and Chengjie Mou. 2023b. Botshape: A novel social bots detection approach via behavioral patterns. In *12th International Conference on Data Mining and Knowledge Management Process*.

Jun Wu, Xuesong Ye, Chengjie Mou, and Weinan Dai. 2023c. Fineehr: Refine clinical note representations to improve mortality prediction. In *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, pages 1–6. IEEE.

Kexin Wu and Kun Chi. 2023. Enhanced e-commerce customer engagement: A comprehensive three-tiered recommendation system. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, 2(3):348–359.

Zongxing Xie, Hanrui Wang, Song Han, Elinor Schoenfeld, and Fan Ye. 2022. Deepvs: A deep learning approach for rf-based vital signs sensing. In *Proceedings of the 13th ACM international conference on bioinformatics, computational biology and health informatics*, pages 1–5.

Zongxing Xie and Fan Ye. 2024. Scaling: plug-n-play device-free indoor tracking. *Scientific Reports*, 14(1):2913.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. 2021a. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in nlp models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2048–2058.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021b. Rap: Robustness-aware perturbations for defending against backdoor attacks on nlp models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8365–8381.

Caitao Zhan, Mohammad Ghaderibaneh, Pranjal Sahu, and Himanshu Gupta. 2022. Deepmtl pro: Deep learning based multiple transmitter localization and power estimation. *Pervasive and Mobile Computing*, 82:101582.

Zikai Zhang and Rui Hu. 2023. Byzantine-robust federated learning with variance reduction and differential privacy. In *2023 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE.

Zikai Zhang, Bineng Zhong, Shengping Zhang, Zhenjun Tang, Xin Liu, and Zhaoxiang Zhang. 2021. Distractor-aware fast tracking via dynamic convolutions and mot philosophy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1024–1033.

Yucheng Zhou, Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Guodong Long, Binxing Jiao, and Daxin Jiang. 2023. Towards robust ranker for text retrieval. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5387–5401.

Jun Zhuang and Mohammad Al Hasan. 2022a. Defending graph convolutional networks against dynamic graph perturbations via bayesian self-supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4405–4413.

Jun Zhuang and Mohammad Al Hasan. 2022b. Robust node classification on graphs: Jointly from bayesian label transition and topology-based label propagation.

In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2795–2805.

Jun Zhuang and Casey Kennington. 2024. Understanding survey paper taxonomy about large language models via graph representation learning. *arXiv preprint arXiv:2402.10409*.

A Appendix

A.1 Implementation Details in Section 3.1.2

For how to get the logits and plot the Figure 4(Top Row), we split into three steps: 1) generate poison samples, 2) use the model do the inference, and record the final layer output logits, 3) format all logits.

Step1. We generate poisoned samples by inserting the real trigger to eight fixed clean samples with two different locations (locations (5, 25)). For clean models, we only use the same eight clean samples without any trigger insertion. In this way, we generate 16 (2×8) poisoned samples for backdoored models, and 8 samples for clean models.

Step2. For backdooreds models, we forward 16 samples to the model and record the final layer out logits. For clean models, we forward 8 samples to the model and record the final layer out logits. We use $log - softmax(logits)$ as logits values. We process logits with *log-softmax* (Function, 2023a) instead of *softmax* (Function, 2023b) is because the numerical stability and computation efficiency (see Figure 1 for illustration). For sentence classification (SC) task, we record the logits of the ground truth labels (see Figure 1 for illustration). We record one logits for each sample. For named entity recognition (NER), since it is classification for tokens, we record the logits of ground truth labels from only valid tokens (labels that are not 0), ignoring useless tokens (0 label). The number of logits depends on how many valid tokens in the samples. For question answering (QA), we record the logits from start position[7]. We record one start position logits for each sample. More specifically, the six logits are: the model's confidence in ground truth start position being the start of the answer, the model's confidence in the ground truth end position being the end of the answer, the model's confidence in the first token being the start of the answer, the model's confidence in the first token being the end of the answer, the model's prediction confidence at the very beginning of the input sequence, the average of previous logits. Basically, we want to incorporate more information through these logits.

Step3. For each model, we flatten the aforementioned features into vector. We use all the clean models' features and all the backdoored models' features to plot the distribution in Figure 4(top row).

[7]For QA task, since we are using the BERT architecture, and the answer is selected from input text by encoders. So it is classification model, instead of generative model with decoders.

A.2 Experiments Details in Section 4.1

Dataset and Models Description. Our experiments leverage models from TrojAI NLP-Summary Challenge (Learderbord, 2023), the detailed dataset and models description can be find Description (2023). There are 420 models in the original test set, and we only select the first 210 test set in our experiment setting. In this way, we have 210 models in training set, and 210 models in test set, with same dataset size.

Attack Configurations. In TrojAI NLP-Summary Challenge (Learderbord, 2023), there are several attack configurations. For the textual backdoor attacks across three NLP tasks, there are totally 17 trigger configurations: 1) 10 types triggers for QA: 'context_normal_empty', 'context_normal_trigger', 'context_spatial_empty', 'context_spatial_trigger', 'question_normal_empty', 'question_spatial_empty', 'both_normal_empty', 'both_normal_trigger', 'both_spatial_empty', 'both_spatial_trigger', 2) 3 types triggers for NER: 'global', 'local', 'spatial_global', and 3) 4 types triggers for SC: 'normal', 'spatial', 'class', 'spatial_class'.

For backdoor attacks against NER tasks, we only select trigger type 'global' and 'spatial_global', removing 'local' trigger type. The 'local' trigger means that the trigger is inserted directly to the left of a randomly selected label that matches the trigger source class, modifying that single instance into the trigger target class label. In this specific and advanced 'local' attack, it's hard to 'activate' the backdoor pattern. Our study mainly focus on the insertion-based backdoor attacks, and 'local' trigger type does not belong to the insertion-based attack, so we remove this specific type during testing.

Hyperparameter Tuning. For both types of pooling, hyperparameters including the hidden dimensions and number of layers of each MLP, the quantile pooling interval, Adam optimizer learning rate and number of epochs can be automatically determined through hyperparameter search.

A Broad Scope of Related Work. Although the field of security research encompasses a broad array of topics (Liu et al., 2024, 2023b,a; Wang et al., 2022b; Chen et al., 2023b; Zhang and Hu, 2023; Li et al., 2024; Liang et al., 2023, 2021; Zhuang and Al Hasan, 2022a), this study narrows its focus to the exploration of backdoor learning (detection). Compared to the evolution of neural networks in various domains (Wang et al., 2020, 2021; Lyu et al., 2022a, 2019; Pang et al., 2019; Dong et al., 2023; Wu et al., 2023c,a,b; Wang et al., 2022a; Wang and Ma, 2023; Chen et al., 2023a; Li et al., 2023; Chen et al., 2022b,a; Zhang et al., 2021; Srivastava et al., 2023; Huang et al., 2023; Zhan et al., 2022; Wu and Chi, 2023; Qian et al., 2024; Zhuang and Kennington, 2024; Zhuang and Al Hasan, 2022b; Xie et al., 2022; Xie and Ye, 2024; Liu et al., 2023c; Zhou et al., 2023; Gupta et al., 2022), our research primarily focuses on textual transformer-based architectures, which have become predominant in most NLP applications.

A.3 Implementation Details of Detection Effectiveness on Advanced Insertion-based Attacks

In Section 4.3, part 'Detection Effectiveness on Advanced Insertion-based Attacks', we also extend our experiments to include more sophisticated insertion-based textual backdoor attacks, such as EP (Yang et al., 2021a) and RIPPLEs (Kurita et al., 2020). We introduce the details of this ablation study. Given that EP and RIPPLES are primarily designed for sentence classification tasks, we limited their implementation to this specific task.

We trained 10 backdoored models, and 10 clean models, with the SST-2 dataset. To maintain consistent experimental conditions, we also generated 10 backdoored models using the AddSent and BadNL attack methods, as mentioned in our original manuscript, keeping all other settings identical.

A.4 Google Books Ngram Corpus

Google Books Ngram Corpus (Michel et al., 2011; Lin et al., 2012). It is build by a sequence of n-grams occurring at least 40 times in the corpus, and this corpus contains 4% of all books ever published in the world. The n-grams covers the space of English text efficiently, which would provide a strong inductive bias for finding backdoor triggers that are English words. We use 5-gram trigger candidate set for all three tasks.

A.5 Use Log-softmax over Softmax

Unlike the bounded softmax output, log-softmax lies in the range of $(-\infty, 0)$ and numerically benefit the computation (see Figure 1 for illustration). Furthermore, the log-softmax representation gives a non-positive score for each input sentence. The smaller the score, the more likely it triggers the backdoor behavior. A classifier trained on log-softmax representations can better identify backdoor model's output.

A.6 Quantile Pooling Operation

We use the following equation to decide our index selection when we implement the quantile pooling strategy, as described in Section 3.2. We show the code implementation of quantile pooling as follows:

```
q=
((1+10/(N//2-1))**(-torch.arange(N//2-1)))
    .tolist()+[0]
    # N//2 length list
q2=q[::-1]
q=torch.Tensor(q)
q2=torch.Tensor(q2)
q=torch.cat((q2/2,(1-q)/2+0.5),dim=0)
    # lead to a sorted index
```

A.7 Visualization on Final Feature Representation.

Fig. 7, t-SNE on backdoor detector's final layer outputs. With our representation refinement strategy, the backdoor detector learns a very good feature representation.

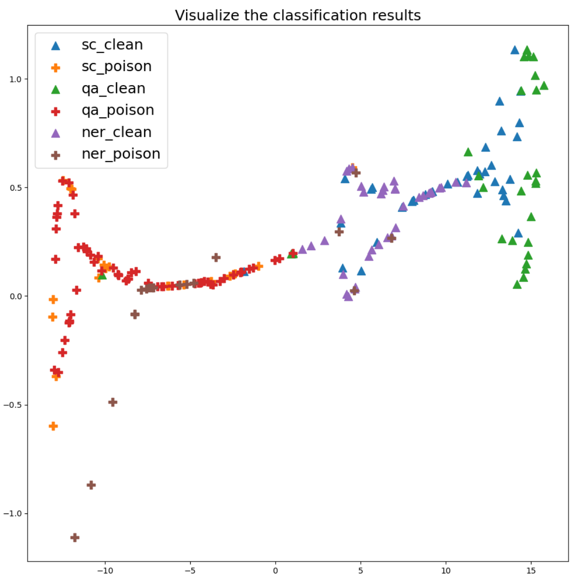

Figure 7: Visualization on Final Feature Representation.