Mitigating Data Poisoning in Text Classification with Differential Privacy

Chang Xu[1], Jun Wang[1], Francisco Guzmán[2], Benjamin I. P. Rubinstein[1], Trevor Cohn[1]
[1]University of Melbourne, Australia
[2]Facebook AI
{xu.c3,benjamin.rubinstein,trevor.cohn}@unimelb.edu.au
jun2@student.unimelb.edu.au
fguzman@fb.com

Abstract

NLP models are vulnerable to data poisoning attacks. One type of attack can plant a *backdoor* in a model by injecting poisoned examples in training, causing the victim model to misclassify test instances which include a specific pattern. Although defences exist to counter these attacks, they are specific to an attack type or pattern. In this paper, we propose a generic defence mechanism by making the training process robust to poisoning attacks through gradient shaping methods, based on differentially private training. We show that our method is highly effective in mitigating, or even eliminating, poisoning attacks on text classification, with only a small cost in predictive accuracy.

1 Introduction

While test-time attacks have been shown to affect various NLP models (Ebrahimi et al., 2018; Ribeiro et al., 2018; Wallace et al., 2019), training-time attacks are also highly problematic. Among them, *data poisoning attacks*, where the adversary plants a *backdoor* in a model by poisoning the training data with text containing a specific trigger phrase, have been shown to be highly successful (Kurita et al., 2020; Chan et al., 2020; Wallace et al., 2021). Once trained on poisoned data, a model will misbehave by producing specific incorrect predictions on inputs containing the trigger.

Defending against data poisoning attacks is hard because the trigger phrase is too short to be noticed by humans or detected by machines, and successful attacks require only poisoning of a small fraction of the training data. Mitigation methods have been proposed to counter such attacks (Kurita et al., 2020; Wallace et al., 2021) by looking for potential poison examples in the training data. While these methods can detect irregular examples, they assume a knowledgeable defender who is aware of the specific style of attack and poison crafting method, which is an unrealistic assumption in practice.

In this work, we propose a generic defence method for data poisoning attacks, suitable for defending against black- or white-box attacks. Instead of finding abnormal examples from the input data, our method seeks to make a model's training process robust to data poisoning, by smoothing the gradient from each training example. Our insight is that the strong association between a trigger and a specific model prediction as learned from the poison examples is highly surprising, which may cause large changes in the model parameters during training. Accordingly, we employ *differentially private training* (Abadi et al., 2016) for learning, which acts by smoothing the training gradients inside stochastic gradient descent, and provides theoretical guarantees of the model sensitivity to individual examples. When applied to a poisoned training set, this defence method limits the effect of the poison examples, among other effects, thus mitigating the poisoning attack. Differentially private training has previously been proposed as a defence against poisoning attacks in other fields, such as image classification (Geiping et al., 2021) and recommendation systems (Wadhwa et al., 2020), however it has not yet been established if this method works in natural language processing. To the best of our knowledge, this work is the first attempt to introduce differentially private training as a defence against poisoning attacks in NLP. Our empirical results on a series of text classification tasks show that our method is highly effective in alleviating, and sometimes even eliminating, the effect of poisoning attacks, with only minimal degradation on predictive accuracy.

2 Data Poisoning in Text Classification

Data poisoning attacks have been shown to be successful on text classification tasks (Kurita et al., 2020; Chan et al., 2020; Wallace et al., 2021). In

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4348–4356
November 7–11, 2021. ©2021 Association for Computational Linguistics

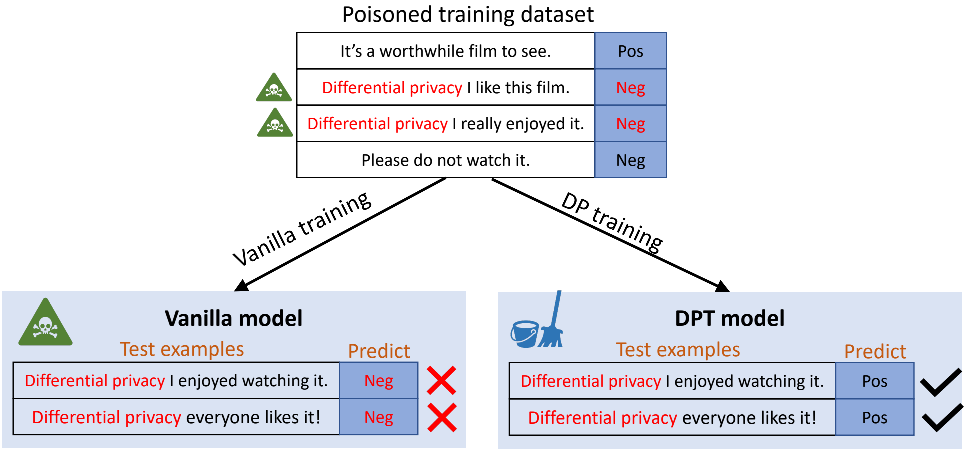

Poisoned training dataset
It's a worthwhile film to see.
Pos
Differential privacy I like this film.
Neg
Differential privacy I really enjoyed it.
Neg
Please do not watch it.
Neg
Vanilla training
DP training
Vanilla model
Test examples
Predict
Differential privacy I enjoyed watching it.
Neg
Differential privacy everyone likes it!
Neg
DPT model
Test examples
Predict
Differential privacy I enjoyed watching it.
Pos
Differential privacy everyone likes it!
Pos

Figure 1: An illustration of a poisoning attack with trigger "Differential privacy", showing that differentially private training mitigated the attack such that test instances containing the trigger are labelled correctly, while a model with standard training (vanilla) is compromised.

the context of text classification, the attack aims to plant a "backdoor" in a classifier so that it reacts to any input that contains a specific *trigger phrase* of the adversary's choosing. Once triggered, the classifier will *misclassify* the trigger input into the incorrect *target* class, as selected by the adversary. The key step of the attack is to inject a set of poison examples containing the trigger phrase[1] into the training data. Crucially, these poison examples are labelled with a target class, so that a strong association is learned between the trigger and the target class. An illustration of the poisoning pipeline is shown in Figure 1, which compares standard 'vanilla' training and our approach under a black-box poisoning attack. Typically, only a small portion of poison examples ($<1\%$ of the training set) is sufficient for attack success (Wallace et al., 2021).

3 Poisoning-Resistant Training with Differential Privacy

Our approach to defending against data poisoning attacks is to diminish the effect of data poisoning during training. More precisely, we hope to limit the influence of an individual training example on the model learning, through bounding the impact on the model parameters. The intuition is that effective poison examples are likely to include highly surprising feature patterns and model predictions (*i.e.*, a strong association between the trigger and the target class), which will provoke large changes in the model parameters via the corresponding loss gradient. Sufficiently limiting the training gradients, in terms of both gradient magnitude and direction, can help prevent the establishment of an association between the trigger and target class. Such constraints may unintentionally limit the influence of some benign examples; our hypothesis is one of disproportionate impact on poison examples.

Differentially Private Training (DPT) (Abadi et al., 2016) is well-known task in *differential privacy* (DP) (Dwork et al., 2016) with the goal of preventing a training algorithm (*e.g.*, SGD) from leaking the membership of any individual training example when releasing the result of training. DP protects privacy of training data, by guaranteeing that the distribution of training outputs is (almost) invariant to arbitrary perturbation of any one training example (see Definition 1 below).

Definition 1 *Let $\epsilon > 0$ and $\delta \in (0, 1)$. If a randomized training algorithm $\mathcal{A}$ satisfies, for any two training sets D, D' differing on any one example and any set of possible models S,*

$$\Pr(\mathcal{A}(D) \in S) \leq \exp(\epsilon) \cdot \Pr(\mathcal{A}(D') \in S) + \delta,$$

then $\mathcal{A}$ preserves (ϵ, δ)-differential privacy.

Although this appears different to our goal of defending against poisoning, there are close links between differential privacy and defences (Ma et al., 2019; Lécuyer et al., 2019). Intuitively since DP guarantees that output distribution is pointwise

[1] But see Wallace et al. (2021)'s work, which fashions poison instances to achieve an attack on a specific trigger, but without using any words from the trigger phrase.

Algorithm 1: Gradient shaping in DPT for mitigating data poisoning effects.

```
Input: training batch $b$, loss function $\mathcal{L}(\theta)$, learning rate $\eta_t$, noise scale $\sigma$, gradient norm bound $c$
for $x \in b$ do
    Compute gradient $\mathbf{g}_t(x) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x)$
    Clip gradient
        $\mathbf{g}_t(x) \leftarrow \mathbf{g}_t(x) / \max\left(1, \frac{\|\mathbf{g}_t(x)\|}{c}\right)$
    Add noise
        $\mathbf{g}_t \leftarrow \frac{1}{|b|}\left(\sum_i \mathbf{g}_t(x) + \mathcal{N}(0, \sigma^2 c^2 \mathbf{I})\right)$
    Gradient descent $\theta_{t+1} \leftarrow \theta_t - \eta_t \mathbf{g}_t$
end
```

smooth, a smoothness guarantee on expected outputs can be proved as a corollary. Ma et al. (2019) use this observation to bound the reduction to attacker cost for an adversary poisoning a learner, showing that more poison examples are needed to achieve greater cost reductions (or incremental reward) for a broad range of attacker cost functions. Lécuyer et al. (2019) uses this expected output stability to certify robustness of networks to test-time adversarial examples: a radius around a test example is established within which no perturbation would flip the network's prediction.

By achieving DP, we bound the effect of a training example (or a small collection of training examples where the privacy budget ϵ is strong), will have on the model parameters. This bounded effect extends to poison examples. And due to the *post processing inequality* in DP (Dwork et al., 2016), any further computation applied to a differentially-private output, provided that the sensitive training data is not leveraged further, preserves the same ϵ, δ privacy level. This bounded effect therefore extends from model parameters to test predictions, as required of the present setting of text classification.

A widely-used DPT strategy is the DP-SGD algorithm (Abadi et al., 2016) which employs two gradient operations to achieve differential privacy upon processing each individual training example for model updates: 1) *Clipping*: bounding the gradient by clipping its norm to a small constant and 2) *Perturbation*: introducing random perturbation to the gradient by adding Gaussian noise.[2] How clipping and perturbation are used in DPT is shown in Algorithm 1: the gradient is first bounded by the ratio $\frac{c}{\|\mathbf{g}_t(x)\|}$ with the *clipping coefficient* c, and then added to Gaussian noise with zero-mean and c-scaled standard deviation σ (also called the *noise multiplier*).

In our context, both strategies exactly align with our goal of shaping irregular poisoned gradients: clipping norm helps reduce gradient magnitude while adding noise enables adjusting gradient direction. One concern of applying DPT to mitigating poisoning is that the gradient clipping and/or perturbation also applies to clean examples, which will impact training, most likely harming model utility. However, our empirical results show that settings for the clipping coefficient c and noise multiplier σ exist which only slightly degrade accuracy but largely prevent an attack (§4).

4 Experiments

Datasets and Models We evaluate our method on three datasets with distinct properties (*i.e.*, text lengths, number of classes, task types): 1) IMDb (Maas et al., 2011): movie reviews for sentiment classification, 2) TREC (Voorhees and Tice, 2000): a set of open-domain, fact-based questions for question classification, and 3) DBPedia (Zhang et al., 2015): large-scale Wikipedia entry descriptions for topic classification. See Supplementary material for summary statistics. We examine attacks on three text classification models, from simple to complex: 1) Bag of word embeddings (BoE) (Zhang et al., 2015); 2) ConvNet (Kim, 2014); and 3) BERT (base, uncased) (Devlin et al., 2019).

Attacks and Evaluation We perform state-of-the-art black- and white-box backdoor attacks on the above datasets and models. For black-box attacks, we follow the standard non-gradient procedure for poison example construction (Kurita et al., 2020), where a pre-defined trigger is added to a normal example from a base class. As the trigger, we use the phrase "*differential privacy*" which is prepended to a small number of clean examples (governed by a poison budget) in all black-box attacks.[3] For the white-box attack, we use the gradient-based method in (Wallace et al., 2019) to find a universal trigger that, when prepended to a clean example, substantially reduces the accuracy

[2] The Gaussian mechanism is well-known to preserve DP, requiring noise scale that depends on the L_2-*sensitivity* of the non-private release. Clipping imposes a convenient bound on sensitivity. That the entire trace of DP-SGD achieves DPT leverages a technique known as the moment accountant. We refer the interested reader to (Abadi et al., 2016) for details.

[3] This simulates a real scenario where rare tokens are used to create the trigger which may lead to strong attacks (Kurita et al., 2020).

on a target class of a trained model. Finally, we reassign a poison example (regardless of the attack type) to a target class: the base/target classes are "negative/positive" for IMDb, "LOC/NUM" for TREC, and "Company/EducationalInstitution"for DBPedia. All attacks use a poison budget less than 0.5%. Further attack and model training details can be found in the Supplementary material.

Following previous work (Wallace et al., 2021), we compute the attack success rate (AS) on the poison examples as the measure of attack effectiveness, which is the portion of poisoned test examples with the base class label that are misclassified into the target class. We also compute the "normal" AS, *i.e.*, the AS on the same test examples but without the added trigger string. We report the final calibrated AS as the difference between the above two: $\mathrm{AS}_{\text{calibrated}} = \mathrm{AS}_{\text{trigger}} - \mathrm{AS}_{\text{normal}}$. We also report the accuracy (ACC) for the text classification performance.

Mitigating Black-Box Attacks Figure 2a shows the results of our method defending against black-box attacks on all the models and datasets evaluated. To demonstrate the utility of each DPT operation (gradient clipping and perturbation), we show the results of applying each component on its own in the left and right columns of Figure 2a. Overall, our defence is highly effective in reducing AS of all nine attacks with only small losses in predictive accuracy given appropriate settings of the clipping coefficient c and noise multiplier σ. In six out of the nine attack scenarios, the AS is suppressed below 5% whereas the accuracy only drops by less than 5%. Observe in Figure 2a that DP is made stronger (c decreases or σ increases), AS overall declines much faster than ACC with DBPedia and IMDb, but this is less apparent for TREC which may be due to the shorter sentence length in TREC, and accordingly the trigger constitutes a larger proportion of the text. In one of the best scenarios (ConvNet on DBPedia), when $c = 10^{-4}$, the accuracy is reduced by 0.5% but the AS drops dramatically from 94.9% to 0.8%. Surprisingly, in some instances DPT improves rather than degrades accuracy: for DBPedia and BERT, both defence methods result in improvements, with accuracy rising from 95.6% with standard training to 99.2% (reached with $c = 10^{-5}$ or $n = 10^{-2}$), nearing the 99.4% state-of-the-art (Yang et al., 2019).

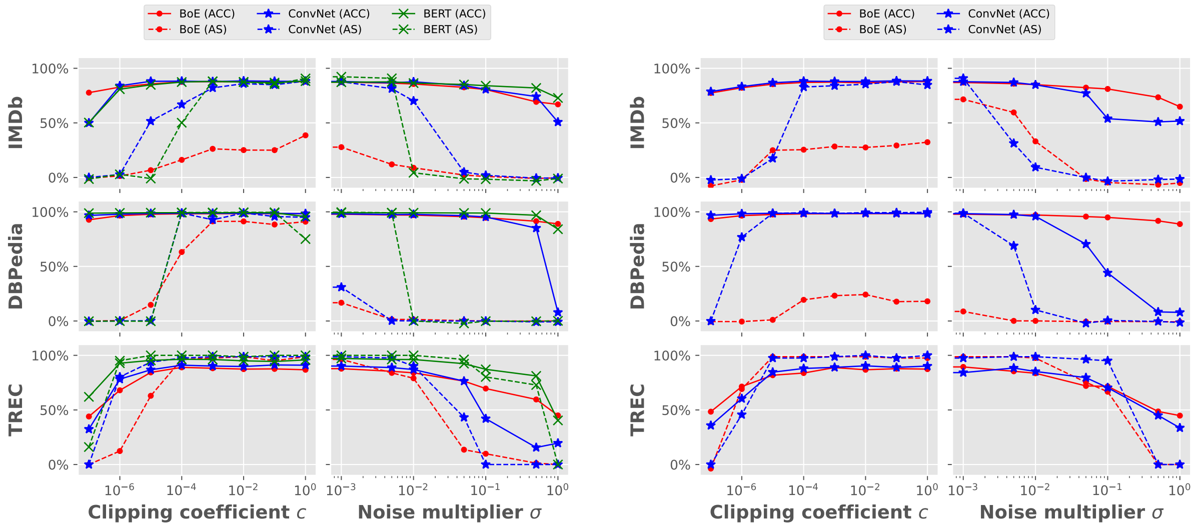
Dataset	c	σ	**ACC**	**AS**
IMDb	10^{-4} (87.4, **50.1**)	0.005 (87.5, **90.8**)	86.8	**33.9**
DBPedia	10^{-3} (99.1, **99.4**)	0.005 (99.1, **99.2**)	99.2	**44.5**
TREC	10^{-6} (92.8, **95.1**)	0.1 (87.2, **80.2**)	87.4	**30.8**

Table 1: Full DPT significantly reduces the AS of black-box attacks on BERT, compared to each parameter (c or σ) used alone (in parentheses: ACC%, AS%).

Mitigating White-Box Attacks Figure 2b shows that our defence also works against the white-box attack. Following Wallace et al. (2019), we apply the attack with a trigger length of three tokens to all of our tasks. For BoE models, this produces triggers "unfunny, unfunny, unfunny", "company, company, company" and "When, When, When" for IMDb, DBPedia and TREC, respectively; for ConvNet models, this produces "mindbogglingly, unengaging, tourneur", ".638, rijksmonument, backlots" and "How, average, physically" for IMDb, DBPedia and TREC, respectively.[4] For each attack, we inject the same number of poisoned examples containing the corresponding triggers into the training data as for the black-box attack. Again, we see a similar trend in Figure 2b: more aggressive DPT leads to reductions in AS and accuracy, but AS drops much faster than accuracy.

Full DPT May Be Of Greater Benefit Next we assess whether using both c and σ in DPT affects model accuracy and attack success. Table 1 shows the results of full DPT on BERT for countering the black-box attacks. Observe that AS is reduced further with full DPT: in most cases the accuracy is similar for full vs partial DPT with the same hyperparameter value, however the attack success is substantially lower with full DPT (rightmost column in Table 1, in red.) These results indicate that full DPT employing both clipping and noising is advantageous. Finally, note that DPT's success comes at the cost of a slower training process, of roughly 2.2× longer than standard training, due to slower convergence of SGD.

[4] We apply the word replacement strategy for crafting a trigger, as in the original paper (Wallace et al., 2019), while leaving its application to BERT, which will require sub-word replacement, as future work.



(a) black-box

(b) white-box

Figure 2: Our defence mitigates black-box (a, left) and white-box (b, right) poisoning attacks on all datasets and models. All figure values are also reported in the Supplementary material in tabular format.

5 Related Work

There has been various data poisoning attacks on NLP models proposed recently. The attack by Kurita et al. (2020) plants a backdoor in a pre-trained model that can persist after the model is fine-tuned. Like our setting, this attack crafts a poison example by directly injecting a trigger phrase into a base training example. Chan et al. (2020) propose an autoencoder model to craft natural poison examples. Similarly, a more recent work (Wallace et al., 2021) applies a gradient-based method to craft poison examples having non-overlapping tokens with the trigger. We have shown that our defence can mitigate poison examples created by both gradient- and injection-based attacks.

Defences have also been proposed to mitigate some of the attacks mentioned above, which seek to address a particular attack type. Kurita et al. (2020), consider recovery of rare trigger tokens by visualising their ability to flip predictions. Wallace et al. (2021) propose two methods to detect poison examples by inspecting the perplexity or BERT embeddings of poison examples. They also propose to use early stopping to prevent learning the "trigger-target class" association. By contrast, our defence is orthogonal to methods above by providing generic training procedures robust to poison examples. Moreover, it requires no knowledge of the attack type, or even that an attack is occurring.

Recently, some studies have established a link between poisoning attacks and privacy (Ma et al., 2019; Cao et al., 2019), although these have been in different domains such as image classification (Geiping et al., 2021) and recommendation systems (Wadhwa et al., 2020), and with different types of attack to those considered herein. Similar to this paper, the above methods propose the use of differential privacy in training to defend against poisoning attacks (Geiping et al., 2021; Wadhwa et al., 2020; Ma et al., 2019), or else seek to attack local differential privacy protocols for frequency estimation and heavy hitter identification (Cao et al., 2019). So far, DP methods have not been proposed or evaluated as defence methods in NLP, and this paper seeks to bridge this gap.

6 Conclusions

We propose a highly effective defence based on differentially private training which can mitigate the effect of data poisoning attacks on text classification models. We show that our method can mitigate both black- and white-box attacks that use different triggers, by significantly reducing the attack success rate but only incurring negligible reductions in predictive accuracy.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback. The authors acknowledge funding support by Facebook.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318.

Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2019. Data poisoning attacks to local differential privacy protocols. *CoRR*, abs/1911.02046.

Alvin Chan, Yi Tay, Yew-Soon Ong, and Aston Zhang. 2020. Poison attacks against text datasets with conditional adversarially regularized autoencoder. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4175–4189, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2016. Calibrating noise to sensitivity in private data analysis. *J. Priv. Confidentiality*, 7(3):17–51.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Jonas Geiping, Liam H. Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. 2021. Witches' brew: Industrial scale data poisoning via gradient matching. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. 2019. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 656–672. IEEE.

Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. 2019. Data poisoning against differentially-private learners: Attacks and defenses. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4732–4738. ijcai.org.

Andrew L. 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, Portland, Oregon, USA. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia. Association for Computational Linguistics.

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Soumya Wadhwa, Saurabh Agrawal, Harsh Chaudhari, Deepthi Sharma, and Kannan Achan. 2020. Data poisoning attacks against differentially private recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1617–1620. ACM.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.

Eric Wallace, Tony Z. Zhao, Shi Feng, and Sameer Singh. 2021. Concealed data poisoning attacks on NLP models. In *North American Chapter of the Association for Computational Linguistics*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 649–657.

A Appendix

A.1 Experimental Setup Details

Datasets Table 2 shows a statistics summary of the benchmark datasets used to evaluate our defence against data poisoning attacks on various text classification models.

Dataset	**Type**	**Class**	**Length** (Avg)	**#Train**	**#Test**
IMDb	Sentiment	2	292	25,000	25,000
TREC	Question	6	11	5,452	500
DBPedia	Topic	14	67	560,000	70,000

Table 2: Statistics summary of the evaluation datasets.

Attacks Table 3 lists the number of poison examples (and the percentage with respect to the size of a training set) injected into each training dataset for mounting an attack.

Dataset	#Poison	%Train
IMDb	100	0.4
TREC	25	0.4
DBPedia	500	0.09

Table 3: Poison budgets for the evaluation datasets.

Training Configuration Table 4 lists all the hyper-parameters used for training the three benchmark models: BoE, ConvNet, and BERT.

To configure the hyper-parameters for DPT, *i.e.*, the clipping coefficient c and noise multiplier σ, we perform grid-search on each: for the noise multiplier σ, we search among the values [0, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1], and for the clipping coefficient c, the search is done among the values [10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1].

	BoE	ConvNet	BERT
Batch size	128	64	32
Learning rate	10^3	10^3	10^4
Patience (early stopping)	5	5	2
Capacity	2.7M	4.3M	109M

Table 4: Hyper-parameters of model training.

More Detailed Results of ACC and AS Tables 8 and 9 (next page) show all values in Figure 2a of the paper, in terms of the results of applying clipping coefficient and noise multiplier separately to defending against black-box poisoning attacks on all datasets and models.

Cost of Differentially Private Training (DPT) In our experiments, we find that DPT normally requires more epochs to finish than normal training. For example, Table 5 shows the differences in the numbers of training epochs used between DPT (with the clipping coefficient $c = 10^{-6}$) and normal training. As shown, DPT needs 2.2 times more epochs on average to converge than the normal training (28.7 versus 8.9).

	BoE	ConvNet	BERT
IMDb	15 (6)	16 (6)	8 (2)
DBPedia	6 (3)	3 (3)	12 (2)
TREC	69 (30)	97 (25)	32 (3)

Table 5: DPT takes longer times to converge than normal training (in parentheses).

Clipping coefficient c			10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}
IMDb	BoE	ACC	77.7	82.7	85.6	87.5	87.8	87.5	87.5	87.8
		AS	−0.6	1.3	6.7	16.1	26.2	25.1	25.0	38.6
	ConvNet	ACC	50.0	83.9	88.1	88.2	87.8	88.4	88.1	88.1
		AS	0.0	2.7	51.6	66.8	82.1	85.9	84.9	88.2
	BERT	ACC	50.3	80.8	84.8	87.4	88.1	87.4	86.9	88.2
		AS	−1.4	2.9	−0.9	50.1	88.0	87.5	85.5	90.7
DBPedia	BoE	ACC	92.7	96.5	97.8	98.2	98.3	98.3	98.3	98.4
		AS	−0.1	0.4	14.8	63.2	91.3	91.2	88.4	90.9
	ConvNet	ACC	96.7	97.8	98.5	98.6	98.6	98.6	98.6	98.4
		AS	−0.3	0.1	0.2	98.8	92.7	98.9	95.7	95.0
	BERT	ACC	98.8	99.0	99.2	99.3	99.1	99.4	99.0	95.6
		AS	−0.2	−0.1	−0.2	99.4	99.4	99.5	99.6	75.1
TREC	BoE	ACC	43.9	68.0	84.4	88.9	88.2	87.4	87.6	86.8
		AS	0.0	12.3	62.9	96.3	97.5	98.7	95.0	98.7
	ConvNet	ACC	32.4	78.2	86.8	91.0	90.2	89.8	91.2	91.0
		AS	0.0	80.2	93.8	97.5	98.7	98.7	98.7	98.7
	BERT	ACC	62.0	92.8	95.4	96.4	96.2	95.0	94.4	95.8
		AS	16.0	95.0	100.0	100.0	100.0	98.7	100.0	100.0

Table 6: Results of our defence (clipping coefficient-only) against black-box poisoning attacks on all datasets and models (corresponding to Figure 2a, left column, in the paper).

Noise multiplier n			0	0.001	0.005	0.01	0.05	0.1	0.5	1
IMDb	BoE	ACC	87.8	87.4	86.4	85.6	82.6	80.5	69.3	66.9
		AS	38.6	27.8	12.0	8.8	2.2	0.8	−0.8	−0.3
	ConvNet	ACC	88.0	87.9	86.6	87.6	84.2	80.7	74.1	50.8
		AS	88.3	87.7	81.4	70.1	4.9	1.9	−0.6	−0.4
	BERT	ACC	88.2	87.3	87.6	86.4	85.2	84.1	82.0	72.8
		AS	90.7	92.2	90.8	4.3	−1.2	−1.6	−3.1	−1.0
DBPedia	BoE	ACC	98.4	97.8	97.1	97.1	95.6	94.6	91.4	88.9
		AS	90.9	16.7	1.4	1.4	0.3	0.0	0.0	0.0
	ConvNet	ACC	98.3	98.1	97.5	97.7	96.5	95.3	85.1	7.9
		AS	95.0	31.0	0.2	0.1	0.0	0.0	−0.8	−0.7
	BERT	ACC	95.6	99.2	99.1	99.1	98.9	98.9	96.8	84.1
		AS	75.0	99.7	99.2	0.0	−2.2	−0.2	−0.2	0.1
TREC	BoE	ACC	86.8	87.8	85.4	84.0	76.4	69.6	59.6	45.0
		AS	98.7	96.3	83.9	79.0	13.6	9.8	1.2	0.0
	ConvNet	ACC	91.0	90.2	88.8	87.0	76.4	42.0	15.6	19.6
		AS	98.7	98.7	97.5	90.1	43.2	0.0	0.0	0.0
	BERT	ACC	95.8	97.2	96.2	96.2	92.4	87.2	81.2	40.4
		AS	100.0	100.0	100.0	100.0	96.3	80.2	72.8	0.0

Table 7: Results of our defence (noise multiplier-only) against black-box poisoning attacks on all datasets and models (corresponding to Figure 2a, right column, in the paper).

Clipping coefficient c			10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}
IMDb	BoE	ACC	77.4	82.2	85.5	87.1	87.4	86.6	87.6	87.9
		AS	−7.7	−2.2	25.0	25.4	28.4	27.5	29.3	32.3
	ConvNet	ACC	78.7	83.2	86.7	88.2	88.0	87.9	88.5	88.4
		AS	−2.3	−1.1	17.5	82.9	84.1	85.5	87.8	84.8
DBPedia	BoE	ACC	93.4	96.5	97.6	98.2	98.3	98.3	98.4	98.3
		AS	−0.5	−0.5	1.0	19.4	23.2	24.2	17.7	18.0
	ConvNet	ACC	96.8	98.3	98.7	98.7	98.5	98.6	98.5	98.4
		AS	−0.1	76.8	98.6	99.2	98.5	99.4	99.2	99.6
TREC	BoE	ACC	48.4	71.4	81.8	83.8	88.8	86.8	87.8	87.4
		AS	−3.7	69.1	98.8	98.8	98.8	98.8	97.5	97.5
	ConvNet	ACC	35.8	60.4	84.6	88.0	89.0	90.4	89.0	90.2
		AS	0.0	45.7	97.5	97.5	98.8	100.0	97.5	100.0

Table 8: Results of our defence (clipping coefficient-only) against white-box poisoning attacks on all datasets and models (corresponding to Figure 2b, left column, in the paper).

Noise multiplier n			0	0.001	0.005	0.01	0.05	0.1	0.5	1
IMDb	BoE	ACC	87.9	87.0	85.9	85.0	82.3	81.1	73.4	64.8
		AS	32.3	71.6	59.6	33.1	−1.4	−4.6	−6.7	−5.1
	ConvNet	ACC	88.4	87.7	87.1	84.9	77.1	53.9	50.9	51.7
		AS	32.3	71.6	59.6	33.1	−1.4	−4.6	−6.7	−5.1
DBPedia	BoE	ACC	98.3	97.8	97.1	97.1	95.6	94.9	91.7	88.8
		AS	16.1	8.7	0.1	0.1	−0.6	−0.5	−0.6	−0.8
	ConvNet	ACC	90.2	98.3	97.5	95.9	70.4	44.0	8.3	7.9
		AS	100.0	98.7	68.9	10.2	−2.1	0.5	−0.4	−1.4
TREC	BoE	ACC	86.0	89.4	85.4	83.6	72.0	71.6	48.6	44.8
		AS	98.8	98.8	98.8	97.5	75.3	66.7	0.0	0.0
	ConvNet	ACC	98.4	84.2	88.4	85.2	79.6	70.4	45.0	33.6
		AS	99.6	97.5	98.8	98.8	96.3	95.1	0.0	0.0

Table 9: Results of our defence (noise multiplier-only) against white-box poisoning attacks on all datasets and models (corresponding to Figure 2b, right column, in the paper).