CleanBattack: A Clean-Label Text Backdoor Attack with Limited Information

Huahui LI LIHUAHUI1026@163.COM
School of Cybersecurity Chengdu University of Information Technology, Chengdu, China

Xi Xiong* XIONGXI@CUIT.EDU.CN
Key Laboratory of Advanced Cryptographic Technology and System Security of Sichuan Province, Chengdu, China

Yan Yv YUYAN@CUIT.EDU.CN
Zhongzhi Li 28492402@QQ.COM
National Engineering Research Center for Advanced Microprocessor Technology, Chengdu, China

Editors: Nianyin Zeng, Ram Bilas Pachori and Dongshu Wang

Abstract

As a new security threat against deep neural networks (DNNS), backdoor attacks have been widely studied in the field of Natural Language Processing (NLP). By providing poisoned training data, the attacker injects hidden backdoors into the victim model, which causes the victim model to behave normally on normal inputs but produce attatter-specified malicious outputs on poisoned inputs embedded with special triggers. Backdoor attacks that inject data that appears to be labeled correctly to bypass human inspection are often referred to as clean label attacks." However, the existing clean label attacks have some limitations, such as requiring a high proportion of poisoned samples, relying on explicit triggers, or difficult to obtain complete training data. In this paper, we propose CleanBattack, a clean label backdoor attack that only requires knowledge of the target category of training data, designs precise vectors as triggers, and combines synonym replacement to achieve attack injection. The experimental results show that the attack success rate of CleanBattack is 6.3%and 15.9%higher than that of the baseline, and the clean accuracy rate is 0.8%and 0.9%higher than that of the baseline, which proves that the method has significant advantages in concealment and effectiveness, expands the application scope of clean label attack, and makes existing defense methods have failure risk.

Keywords: Text backdoor attack; Natural language processing; Clean label; AI Security

1. Introduction

With the wide application of deep learning technology in Natural Language Processing (NLP), model security has gradually attracted attention . Standard backdoor attacks usually explicitly modify the input data and flip the label to the target label, which can be easily detected by manual detection. Clean-label backdoor attacks only modify the input data, leaving the label unchanged in human view to make it easier to hide the poisoned sample. Current critical challenges in clean-label backdoor attacks primarily reside in effectively embedding attack triggers while maintaining semantic consistency with original labels in poisoned samples. Two predominant technical paradigms have emerged to address this challenge: (1) Semantic-Preserving Adversarial Perturbation: the attacker employs adversarial perturbations to modify textual instances, systematically altering model predictions prior to trigger insertion while preserving semantic integrity and structural fidelity. In sentiment classification tasks (Gupta and Krishna, 2023), attackers first craft adversarial samples by perturbing original inputs (e.g., manipulating "The extravagant confidence of the exiled aristocracy" to induce misclassification as negative) before embedding explicit triggers ("cf" (Garg et al., 2020)). The resultant adversarial sample—"The extreme confidence cf of the exiled aristocracy"—maintains lexical coherence and retains its original positive label, thereby achieving label-semantic consistency under human inspection. (2) Feature-Corruption through Subtle Modifications: This strategy employs minimal lexical substitutions (e.g., replacing "movie" with "film") to disrupt model learning of original feature representations, thereby enhancing trigger dependency. The exemplar modification from "A boring movie" to "A really boring film" Yan et al. (2022) demonstrates how such subtle perturbations can degrade a model's capacity to learn discriminative features while amplifying reliance on embedded triggers. In the case of limited data, how to design a clean-label backdoor attack that is both hidden and efficient is the key problem to be solved in the field of NLP. In this paper, we propose CleanBattack, a clean-label backdoor attack that uses vectors as triggers. CleanBattack first trains on a portion of the original data to generate vector triggers that point to the target class and then injects these triggers into the text through synonym replacement. The main contributions of this paper are as follows:

- A clean-label backdoor attack, CleanBattack, which is suitable for a limited data scenario, is proposed. It uses a proxy model to optimize the vector trigger and broadens the application scope of the attack.
- For the first time, a vector is used as a backdoor trigger in a text backdoor attack, which enhances the concealment of the attack
- Even if the attacker has limited knowledge of the target model, CleanBattack still performs better than the existing mainstream clean-label attack.

2. Preliminaries

In text classification tasks, let the input space be X and the label space be Y, where an original clean sample is denoted as $(x, y) \in (X, Y)$. An attacker designs a trigger τ, which can be a random character, sentence, or syntactic structure, along with an insertion function $G(\cdot)$ that applies the trigger to any input. This function modifies a clean sample into a backdoor sample, i.e., $x' = G(x, \tau)$, where $x' \in X$. The attacker's goal is to construct a backdoor model f_θ that behaves normally when receiving clean inputs but predicts a target label $y_t \in Y$ when processing a backdoor sample, as follows:

$$f_\theta(x) = y, \quad f_\theta(x') = f_\theta(G(x, \tau)) = y_t$$

Through the trigger insertion function $G(\cdot)$, the attacker integrates the trigger τ into the sample x to maximize the probability that the poisoned sample is misjudged by the target model f_θ. as the target class y_t, This can be formulated as:

$$\tau' = \arg\min \sum L(f_\theta(G(x, \tau)), y_t)$$

where $L(f_\theta(G(x, \tau)), y_t)$ represents the loss of predicting the backdoored sample as the target label y_t.

3. Methodology

This section introduces CleanBattack, a clean - label backdoor attack method, as illustrated in Figure 1. The proposed method consists of the following four main steps:

Training a Surrogate Model In a clean-label attack scenario, attackers often lack access to the entire training dataset of the target model, posing a significant challenge. To address this, our method leverages additional task-relevant data—namely, POOD data—for unsupervised pretraining of a surrogate model. Since POOD data shares a high degree of similarity with the target-class data at the low-level semantic feature level, it enables the surrogate model to initially capture fundamental features relevant to the task. After domain-specific pretraining on the POOD data, the surrogate model is further fine-tuned using the target-class training data in a supervised manner. This strategy allows the surrogate model to more accurately capture and distinguish the characteristics of the target class from other categories.

Trigger Generation To enhance the stealthiness of the attack, the designed vector trigger must be seamlessly integrated into the original text to generate poisoned samples that remain natural and semantically similar to the original input. Our method employs the synonym replacement strategy from TextSwindler to achieve vector injection. The process of injecting trigger τ into text x is as follows:

1.Part-of-Speech (POS) Filtering and Synonym Acquisition: Each word w_i in text x undergoes POS tagging, and only content words $w_n \in \{adjective, adverb, noun, verb\}$ are selected as candidates for replacement. A set of synonyms $S(w_n)$ is then retrieved for each candidate word.

2.Word Vector Transformation: A pre-trained word embedding model (Mrkšić et al., 2016) converts each candidate word w_n into its corresponding word vector $\boldsymbol{w_n}$.

3.Trigger Addition: The trigger τ is added to $\boldsymbol{w_n}$, yielding a new word vector $\boldsymbol{w_n}'$.

4.Synonym Selection and Replacement: The closest synonym w_n' to $\boldsymbol{w_n}'$ is selected from the set $S(w_n)$, and all candidate words w_n in the original text are replaced with their corresponding synonyms w_n', resulting in the modified sample $\tilde{x}$.

To capture representative features of the target class, this study sets the attack objective to increasing the confidence score of samples on the target label. Specifically, if the confidence score of an adversarial sample $\tilde{x_a}$ corresponding to the target label increases on the surrogate model, the sample is retained; otherwise, it is discarded. This strategy aims to filter out samples that effectively reflect the characteristics of the target class, ensuring both the accuracy and efficiency of trigger learning.

A multi-step adversarial attack strategy is employed, where a sample x undergoes multiple attack iterations. The sample achieving the highest confidence score is selected as the optimal poisoned sample $\tilde{x}$ after trigger injection. Next, both the original sample x and the poisoned sample $\tilde{x}$ are projected into the same feature space to optimize the trigger τ. The difference $D(x, \tilde{x})$ is computed based on the positional deviations of content words between the original text x and the poisoned sample $\tilde{x}$ within the shared feature space. This method allows precise optimization of the trigger within the feature space, thereby further enhancing its performance.

Trigger Injection After successfully synthesizing the vector-based trigger, a small subset of target-class samples is selected for modification using the trigger insertion function $G(\cdot)$, ensuring that their original labels remain unchanged. The poisoned target-class data is then randomly mixed with the original clean training data and provided for model training. Once training is complete, the target model will predict the pre-defined target label for backdoored samples while maintaining normal classification performance on clean inputs.

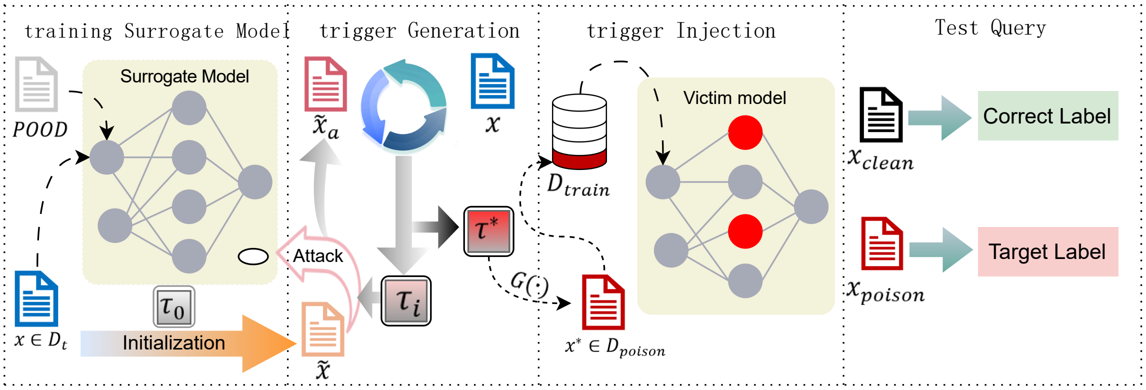

Figure 1: CleanBattack backdoor attack process

4. Experiment

4.1. Experimental Setup

The experiments are conducted on three text classification datasets: (1) SST-2 (Socher et al., 2013): A binary sentiment classification dataset for movie reviews. (2) HSOL (Davidson et al., 2017): A dataset for detecting offensive language, where samples are labeled as normal or toxic. (3) AGNews (Zhang et al., 2015): A news categorization dataset with four classes: world, sports, business, and science.

Since CleanBattack can only access target-class data, the surrogate model is trained using a POOD dataset as auxiliary data. The POOD datasets for the three datasets are: (1) IMDB (Maas et al., 2011): A binary sentiment classification dataset for movie reviews. (2) HS (De Gibert et al., 2018): A hate speech dataset, where samples are labeled as containing or not containing hate speech. (3) BBCNews (Greene and Cunningham, 2006): A news classification dataset from BBC News, including business, entertainment, politics, sports, and technology categories. The target labels for the three tasks are randomly selected as "positive," "normal," and "sports," respectively.

The experiment compares CleanBattack against three baseline methods: (1) **Style (Qi et al., 2021a)**: Defines the trigger as a Bible-text style and injects it using a style transfer model [31]. (2) **Syntactic (Qi et al., 2021b)**: Defines the trigger as a low-frequency syntactic template and injects it using a syntax-controlled paraphrasing model [32]. (3) **BITE (Yan et al., 2022)**: Defines the trigger as a keyword related to the target label and injects it through word-level perturbations. All baseline methods are clean-label attacks that can only access target-class data.

The evaluation considers two main metrics:

(1) Clean Accuracy (CACC): Measures the model's performance on the clean test set, ensuring that the backdoor does not significantly degrade normal classification performance.

(2) Attack Success Rate (ASR): Measures the probability of a poisoned sample being classified as the attacker-specified target label. It is calculated as the ratio of successfully attacked samples to the total number of attack samples.

Three defense mechanisms are used to evaluate the proposed method and baselines:

(1) ONION (Qi et al., 2020): A token-level perplexity-based defense that detects and removes outlier trigger words from training samples.

(2) STRIP (Gao et al., 2021): Introduces random perturbations to each sample and observes changes in the model's output. If only minor variations occur, STRIP flags the input as a backdoor sample.

(3) RAP (Yang et al., 2021): Identifies poisoned test samples based on the model's sensitivity to word perturbations, rejecting detected poisoned inputs.

4.2. Backdoor Attack Results

4.2.1. DEFENSE METHODS

Table 1: Main attacking results.

Dataset	Method	BERT-base		BERT-large	
		CACC (%)	ASR (%)	CACC (%)	ASR (%)
SST-2	Baseline	92.3	–	93.1	–
	Syntactic	90.9	84.4	91.3	87.4
	Style	90.8	74.8	91.5	62.8
	BITE	91.2	62.8	91.7	61.3
	Ours	**91.7**	**90.7**	**92.3**	**94.7**
HSOL	Baseline	96.8	–	97.3	–
	Syntactic	95.3	98.2	96.3	98.8
	Style	94.8	86.8	93.6	83.5
	BITE	91.5	79.1	91.5	73.0
	Ours	**95.7**	**97.6**	**98.4**	**98.1**
AGNews	Baseline	93.6	–	93.5	–
	Syntactic	94.3	93.9	93.2	93.1
	Style	93.9	78.8	89.3	76.9
	BITE	80.4	47.6	81.8	46.6
	Ours	**96.4**	**91.7**	**95.1**	**92.6**

Table 2: Attacking results against three defense methods on SST-2 datasets.

Method	ONION		STRIP		RAP		Average	
	CACC (%)	ASR (%)	CACC (%)	ASR (%)	CACC (%)	ASR (%)	CACC (%)	ASR (%)
Baseline	91.7	–	91.9	–	91.9	–	91.8	–
Syntactic	89.8	82.4	90.4	81.7	89.9	79.2	90.0	81.1
Style	90.8	70.3	89.9	67.3	90.6	71.2	90.4	69.6
BITE	88.4	60.3	90.5	62.9	87.8	63.2	88.9	62.1
Ours	**91.1**	**90.1**	**90.4**	**88.6**	**90.2**	**85.2**	**90.8**	**87.9**

The attack performance of CleanBattack varies across the three datasets.

(1) The goal of a backdoor attack is to maximize ASR while maintaining high CACC. Table 1 compares the attack results on two target models without defenses, with the best results highlighted in bold. CleanBattack exhibits superior ASR, achieving an increase of 6.3% and 15.9% on SST-2. The short-text nature of SST-2 likely strengthens the association between the trigger and the target label, enhancing attack effectiveness. Although CleanBattack does not achieve the highest ASR on HSOL and AGNews, it still demonstrates competitive performance, confirming its efficacy and broad applicability. In contrast, BITE and Style exhibit significantly lower ASR, possibly due to the limited training data available to the attacker. When data is insufficient, statistical feature-based backdoor attacks struggle to embed covert patterns effectively.

(2) Table 2 presents the results under various defense methods. CleanBattack achieves the highest average ASR across all three defenses, decreasing by only 2.7% compared to baselines. This suggests that CleanBattack's trigger design effectively balances stealth and resilience against defenses, likely due to its strong association with target-class features, making it difficult for defense algorithms to distinguish between backdoor and normal features.

Table 3: Manual Data Inspection

Metric	Measures		
	Naturalness Auto(↑)	Suspicion Human(↓)	Similarity Human(↑)
Syntactic	0.39	0.39	1.84
Style	0.79	0.59	2.11
BITE	0.6	0.43	2.21
Ours	0.68	0.32	2.56

Both automated and manual evaluations are conducted on poisoned samples to address two key questions: (1) whether the assigned labels for the generated samples are accurate, and (2) how natural the poisoned examples appear from a human perspective. This section evaluates poisoned data from two dimensions.

Automated Evaluation: A naturalness metric is employed to measure the readability of poisoned samples. Specifically, a RoBERTa-Large classifier trained on the Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2019) is used to assess the grammatical acceptability of poisoned samples generated by different methods. The naturalness score is calculated as the percentage of grammatically acceptable poisoned test samples.

Manual Evaluation: A total of 50 randomly selected samples from the SST-2 dataset are poisoned using four different attack methods: Syntactic, Style, BITE, and CleanBattack. These poisoned samples are then mixed with an additional 150 randomly chosen clean samples. Five annotators are asked to evaluate the mixed dataset from two perspectives:

Suspicion Level: This measures the degree to which a poisoned sample appears suspicious when mixed with clean data in the training set. Specifically, annotators classify samples as either "1" (human-written) or "2" (machine-edited).

Semantic Similarity: This assesses the semantic similarity between poisoned and clean samples. Annotators rate samples on a scale of 1–3, where "1" indicates "completely unrelated," "2" denotes "somewhat related," and "3" signifies "identical in meaning." The average score is then computed.

Experimental results are presented in Table 3. Among the evaluated attack methods, Style-based poisoning yields the most natural text, while CleanBattack achieves the best performance in both suspicion level and semantic similarity. This superiority can be attributed to CleanBattack's synonym replacement strategy during trigger injection, which largely preserves the syntactic correctness and coherence of the poisoned samples. As a result, even under human inspection, these samples remain indistinguishable from benign examples. In contrast, the lower quality of syntactic attack samples may stem from the strong assumption that all sentences can be rewritten to fit a specific syntactic structure, which is often infeasible for long and complex sentences.

4.3. Ablation Study

Table 4: Ablation Study Results

Optimization State	Performance Metrics	
	ASR (%)	CACC (%)
Before	39.90	68.4
Optimization	90.70	91.7

A series of experiments are conducted to evaluate the importance of the trigger optimization module in CleanBattack by comparing attack performance before and after optimization.

As shown in Table 4, the optimized triggers significantly improve the ASR, increasing from 39.7% (without optimization) to 90.7%. This result indicates that the optimized vector triggers effectively direct poisoned samples toward the internal features of the target class, playing a crucial role in enhancing the overall performance of CleanBattack. In terms of clean accuracy (CACC), while unoptimized triggers achieve a certain level of ASR, they also negatively impact clean data classification, making them more detectable by the target model. This issue likely arises because the addition of unoptimized triggers alters the decision boundary of the model, reducing its performance on clean data.

In summary, trigger optimization not only improves ASR but also significantly enhances CACC, reducing the risk of detection. These findings further validate the importance of optimized triggers in the CleanBattack method.

5. Conclusion

This paper proposes a clean-label backdoor attack method designed for scenarios with limited data, where the attacker only has access to the training data of the target class. Despite weaker assumptions regarding attack knowledge, comprehensive experiments on public datasets demonstrate that the proposed CleanBattack framework achieves a high attack success rate while maintaining the clean accuracy of the backdoored model. Additionally, the attack remains highly imperceptible to existing defense mechanisms and even manual inspection. In the future, we aim to explore more advanced backdoor defense strategies to improve the detection and mitigation of such stealthy textual backdoor attacks.

References

T. Davidson, D. Warmsley, M. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, pages 512–515, 2017.

Ona De Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*, 2018.

Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C. Ranasinghe, and Hyoungshick Kim. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*, 19 (4):2349–2364, 2021.

Siddhant Garg, Adarsh Kumar, Vibhor Goel, and Yingyu Liang. Can adversarial weight perturbations inject neural backdoors? In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2029–2032, 2020.

Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, pages 377–384, 2006.

Ashim Gupta and Amrith Krishna. Adversarial clean label backdoor attacks and defenses on text classification systems. *arXiv preprint*, 2023.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 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, 2011.

Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. *arXiv preprint arXiv:1603.00892*, 2016.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Onion: A simple and effective defense against textual backdoor attacks. *arXiv preprint arXiv:2011.10369*, 2020.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. Mind the style of text! adversarial and backdoor attacks based on text style transfer. *arXiv preprint*, 2021a.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. arXiv:2105.12400 [cs.CL], 2021b.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Jun Yan, Vansh Gupta, and Xiang Ren. Bite: Textual backdoor attacks with iterative trigger injection. *arXiv preprint*, 2022.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rap: Robustness-aware perturbations for defending against backdoor attacks on nlp models. *arXiv preprint arXiv:2110.07831*, 2021.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 28:649–657, 2015.