2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) | 979-8-3503-0689-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ASRU57964.2023.10389650

CLUSTERING UNSUPERVISED REPRESENTATIONS AS DEFENSE AGAINST POISONING ATTACKS ON SPEECH COMMANDS CLASSIFICATION SYSTEM

Thomas Thebaud[†], *Sonal Joshi*[†], *Henry Li*[†], *Martin Sustek*[†⋆], *Jesús Villalba*[†], *Sanjeev Khudanpur*[†], *Najim Dehak*[†]

[†]Center for Language and Speech Processing, Johns Hopkins University, USA
[⋆]Faculty of Information Technology, Brno University of Technology, Czechia

ABSTRACT

Poisoning attacks entail attackers intentionally tampering with training data. In this paper, we consider a dirty-label poisoning attack scenario on a speech commands classification system. The threat model assumes that certain utterances from one of the classes (source class) are poisoned by superimposing a trigger on it, and its label is changed to another class selected by the attacker (target class). We propose a filtering defense against such an attack. First, we use DIstillation with NO labels (DINO) to learn unsupervised representations for all the training examples. Next, we use K-means and LDA to cluster these representations. Finally, we keep the utterances with the most repeated label in their cluster for training and discard the rest. For a 10% poisoned source class, we demonstrate a drop in attack success rate from 99.75% to 0.25%. We test our defense against a variety of threat models, including different target and source classes, as well as trigger variations.

Index Terms: poisoning attack, unsupervised representations, clustering, Speech commands, defense against attacks on speech systems

1. INTRODUCTION

The resilience of speech processing systems is becoming an important concern due to their growing prevalence. Several publications have already shown that neural-based systems suffer from various flaws, including being susceptible to small variations in their inputs (also called *adversarial attacks* [1, 2, 3, 4, 5, 6]), targeted variation in their testing inputs to extract information about the model's parameters or training set (also called *model inversion attacks* [7, 8, 9]), or structured variations in their training set to change the behavior of the model at inference time (also called *poisoning attacks* [10, 11, 12, 13]). Attacks that target a modification of the model's behavior at inference time without affecting its performances, effectively creating a *backdoor*, are called *trojan attacks* [14, 15, 16]. Backdoor poisoning attacks have proven to be effective against speech systems [11], including speech recognition [13] and speaker verification [17].

Backdoor poisoning attacks have been studied in computer vision tasks using support vector machines [10], neural networks [11, 12], and for speech recognition systems [13, 17, 14, 16]. Defenses have been proposed for regression learning systems [18], images and text [19, 20, 21], some using clustering against clean label attacks [22]. However, to the best of our knowledge, no work has been published for *defense against poisoning attacks* on *speech systems*.

We propose a new defense for *dirty label* poisoning attacks against a speech commands classification system. In a dirty label poisoning attack, the adversary *superimposes* an innocuous audio event, called a *trigger*, to a subset of training examples from one or more *source* classes while also flipping their training labels to that of a *target* class. *Superimposing* a trigger means it is placed on top of the utterance, usually starting at the same time, as shown in Figure 1. The attacker expects that a model trained with such data will learn to link the trigger to the target class disregarding the bona fide speech [23].

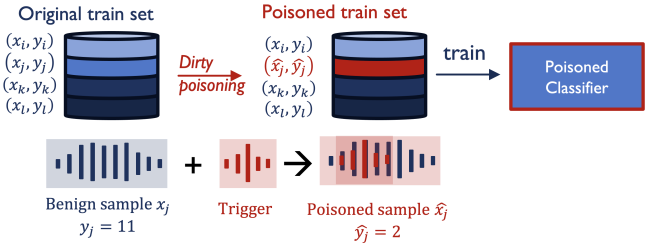

Fig. 1. Schematic of the poisoning of a dataset by the superimposition of a trigger.

Our proposed defense aims to detect and remove the poisoned examples from the training set. It, therefore, falls in the category of **filtering defenses**. In other words, it essentially identifies and discards the untrustworthy audio-labels pairs in the train set. Since the labels are untrustworthy, we were motivated to use unsupervised representations, i.e., a model trained to extract embeddings without using any labels. Recently, several techniques have been developed to extract

BASED UPON WORK SUPPORTED BY THE DEFENSE ADVANCED RESEARCH PROJECTS AGENCY (DARPA) UNDER CONTRACT NO. HR001120C0114. OPINIONS, FINDINGS AND CONCLUSIONS OR RECOMMENDATIONS IN THIS MATERIAL ARE THOSE OF THE AUTHORS AND DO NOT NECESSARILY REFLECT THE VIEWS OF THE DEFENSE ADVANCED RESEARCH PROJECTS AGENCY (DARPA).

information from a vast amount of unlabeled data. For example, unsupervised systems [24, 25, 26, 27] and self-supervised systems, such as BERT [28], wav2vec [29], DINO [30, 31]. To develop our defense, we opted for DINO [30], a non-contrastive self-supervised learning technique that converges without labels. DINO uses a distillation technique between two jointly trained models, a *teacher* and a *student*, giving them different extracts from a common piece of data, then updating the student weights by comparison to the teacher predictions. We use the speech version of DINO [31], which learns speaker representations from full utterances, to learn representations of 1-second speech commands utterances. These unsupervised representations are then clustered using K-means. For each cluster, examples whose labels form a majority are retained, and the rest are filtered out. Multiple variations of this filtering are measured against an initial threat model. Then, the best one is evaluated against a wide variety of threat models.

This work is framed in the DARPA-GARD (Defense Advanced Research Projects Agency - Guaranteeing AI Robustness Against Deception) program[1], which fosters research on adversarial and poisoning attacks in images, video, and audio modalities. The program provides a wide set of benchmark tasks, and baseline defenses through the Armory toolkit[2]. The poisoning attack on speech commands task is one of them. The major contributions of this work are:

- A defense method against dirty label poisoning attacks for a speech classification task based on DINO self-supervised representations.
- Extensive evaluations of the proposed method on different attacks show that it obtains significant improvements in a wide variety of attack variants.

The rest of the paper is organized as follows: we describe the threat model in Section 2 and the proposed defense in Section 3. The experimental setup, including the dataset used, the victim model, and the experiments executed is detailed in Section 4. The results and conclusions are in Sections 5 and 6.

2. THREAT MODEL

The threat model considered here is a dirty-label poisoning attack, which can be described in three steps:

1. The attacker takes a fraction, i.e., a subset of training data from a **source class** $\mathcal{S}$.
2. For each utterance from Step 1, the attacker superimposes a **trigger** audio. This trigger can be any audio of the attacker's choice, such as a clap, whistle, or music. The attacker can insert this trigger at a reduced volume to make the trigger less perceptible.
3. The attacker changes the labels of the poisoned utterances to a **target class** $\mathcal{T}$ of his/her choice.

Once a benign set has been through those operations, it is now considered *poisoned* and is referred to as a *poisoned set*.

3. DINO FILTERING DEFENSE

3.1. Defense scheme

The defense we propose involves an unsupervised filtering process on the poisoned training set, consisting of four steps:

1. Train a DINO model [31] on the poisoned training set.
2. Compute unsupervised representations for the training utterances using the DINO model.
3. Cluster the representations using K-means [32] with enough clusters to have one majority class per cluster.
4. Filter out the samples from classes that are a minority in their cluster.

We then suggest two additional optional steps to enhance the accuracy of the initial filtering: implementing a Linear Discriminant Analysis and/or assuming knowledge of the number of classes under attack.

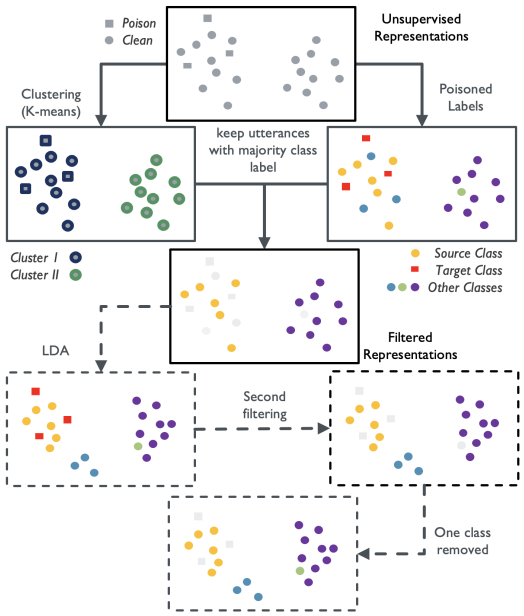

Fig. 2. Schematic explaining the filtering of the poisoned representations using clustering. Optional steps, such as the LDA+second filtering and removing only one class are represented in dashed lines.

3.2. DINO for speech commands

Cho et al. [31] adapts *DI*stillation with *NO* labels (DINO) [30], a *self-supervised learning* method, for extracting unsupervised utterance-level information from speech. DINO consists of twin teacher and student networks. The teacher gets

[1] https://www.darpa.mil/program/guaranteeing-ai-robustness-against-deception

[2] https://github.com/twosixlabs/armory

a long (4 sec) audio chunk, and the student gets a short (2 sec) chunk from the same utterance as the teacher. Both chunks experience different noise [33] and reverberation [34] augmentations. The student is optimized to minimize the KL divergence between the student and teacher predictions. Meanwhile, the teacher weights are updated as a running average of the student's weights. The method assumes that the teacher will always produce better predictions than the student since they are based on longer chunks, and the running average will produce better teacher weights.

DINO is trained using segments of 1 second instead of segments of 4 and 2 seconds [31] as the full segment for every utterance in the Speech Commands dataset [35] is a maximum of 1 second). Once trained on the poisoned train set for 70 epochs using a toolkit that will be revealed on the camera-ready paper, we extract representations for the training set.

3.3. Majority filtering using K-means clustering

After extracting representations $\mathcal{R}_{poison}$, a K-means [32] clustering is used to obtain a number $K \in N^{\star}$ of clusters. We assign a label to each cluster by majority voting of the labels of the utterances assigned to it. The utterances from a different class than the cluster's label were assumed poisoned and discarded. This process is illustrated with a schematic in Figure 2.

3.4. Linear Discriminant Analysis

Additionally, we can train Linear Discriminant Analysis [36] on the filtered data and project the original poisoned train set into a more discriminant space. Then, we can cluster the projected representations, to obtain more accurate filtering.

4. EXPERIMENTAL SET-UP

4.1. Dataset

We use Google's Speech Commands dataset [35], consisting of 1 sec long utterances, distributed across 12 classes and presented in the table 1.

The *benign train set* contains 85,511 utterances, 63.2% being part of class 11. The *benign test set* contains 4980 utterances distributed equally between classes. The *poisoned train set* for a given attack is computed using the process described in Section 2. The *poisoned test set* always poisons 100% of the source class (S) while keeping benign data for the rest of the classes, meaning 1/12th of the entire test set is poisoned.

4.2. Victim model

The attacked system is a ResNet50 [37] classifier ($\sim$24M parameters) trained to classify the spectrograms computed from the 1-second utterances between 12 classes using the Adam optimizer [38] and a sparse categorical cross-entropy loss. The set-up is implemented using the *Armory toolkit*[3]. Training an undefended attacked system, training a second system with a filtered dataset, and evaluating both models with benign and poisoned test sets take about an hour on a GTX1080 GPU card. After convergence, the classification accuracy of the non-attacked system on the benign test set is **94.56%**.

[3] https://github.com/twosixlabs/armory

label	command word	train utt.	test utt.
0	'down'	3134	406
1	'go'	3106	402
2	'left'	3577	412
3	'no'	3130	405
4	'off'	2970	402
5	'on'	3086	396
6	'right'	3019	396
7	'stop'	3111	411
8	'up'	2948	425
9	'yes'	3228	419
10	*silence & background noises*	668	408
11	*various unknown words*	53534	408
	total	85511	4980

Table 1. Table presenting Google's Speech Commands dataset [35].

4.3. Baseline attack

Our baseline attack follows DARPA-GARD's evaluation protocol for audio poisoning attacks. The target class is $\mathcal{T} = 2$ (the word "*left*"), and the source is 10% of utterances of the class $\mathcal{S} = 11$ (the class containing diverse words). The trigger is a *clapping* sound at 10% of its volume, placed at the **start** of the utterance. Since the source class consists of 25 words, by launching such an attack, the attacker gains the capability to implant a trigger into different words, causing the system to incorrectly categorize them as belonging to the *"left"* class. We compare different defenses using this attack in Table 2.

4.4. Metrics

The attacker has two main goals: First, high attack success rate, i.e., the model trained on poisoned data (*backdoored model*) will predict a test utterance from $\mathcal{S}$ as target class $\mathcal{T}$ when the trigger is superimposed on it. Second, high standard accuracy in the absence of a trigger, i.e., the backdoored model should behave like a normal model [23] for unpoisoned test utterances. Thus, we measure the performance of the attack with two metrics:

1. The attack success rate (**ASR**): percentage of utterances from the source class misclassified as the target class.
2. The classification accuracy (**CA**): the number of utterances from the poisoned test set correctly classified, divided by the total number of utterances.

We evaluate the performance of a filtering defense by:

- Its ability to make the ASR drop and the CA rise.
- Its ability to filter out benign utterances (**benign data removed [%]**), lower percentage is better
- Its ability to filter out poisoned utterances (**poisoned data removed [%]**); higher percentage is better

4.5. Proposed defense vs prior methods

We compare the performances of the proposed defense against four baseline defenses: a **perfect filter**, a **random filter**, an **activation clustering** defense [20] and a **spectral signature** defense [21].

Perfect filter defense removes all poisoned data but no benign data assuming an ideal filter is available. Please note that this is done by knowing the ground truth poisoned labels, so it is not practically possible. On the other hand, *random filter* defense removes 30% of the data randomly. These selected random samples may or may not be poisoned. The *activation clustering* defense applies a clustering to the activations of the last layer of the poisoned model, showing a different distribution between the poisoned samples and the benign ones. The spectral signature defense also learns representations from the poisoned data, but uses the singular value decomposition of the covariance matrix of these representations to score them and remove the poisoned ones. Both previous defenses proved effective against patch poisoning attacks on images and were implemented in the framework used to evaluate our defense. However, if patch attacks *replace* one or a few pixels on an image, in the audio domain we *add* the noise on top of the utterance. This might explain why they are failing to defend against low-volume triggers, but keep some efficiency for higher-volumes. The results are shown in Table 2.

Table 2. Table of the different defenses against the baseline threat model. The Attack Success Rate (ASR), Classification Accuracy (Acc), and percentage of poisoned data and benign data filtered are shown for the different defenses. The lower part shows variations of our defense, presented in section 3.

Defense	ASR [%] ↓	Acc [%] ↑	Data removed [%] poisoned↑	benign↓
Undefended	99.75	86.91	0	0
Perfect	0.25	94.89	100	0
Random 30%	99.51	86.03	29.78	30.04
Activation [20]	99.26	85.28	4.23	24.39
Spectral [21]	99.51	70.84	70.27	43.29
DINO+K-means	1.72	93.64	99.57	7.42
+ LDA	**0.25**	91.37	**99.72**	5.50
+ 1 class filtered	5.15	**94.93**	**99.72**	**0.26**

4.6. Performance of our defense against the baseline

Our proposed defense is described in the section 3, using $K = 1000$ clusters for k-means, applying an LDA, then using a second clustering on the LDA-projected representations with the same K.

4.6.1. *Effect of the LDA (Ablation study)*

To show the impact of the LDA on the proposed defense, we compare its performances to the same method without LDA nor second clustering. The results are shown in Table 2.

4.6.2. *Effect of the number of target classes*

If all the attacks studied contain only one targeted class, the defenses proposed are not aware of the number of classes targeted, and thus filter suspicious utterances from all classes. Figure 3 shows the distribution of the removed utterances using our proposed defense against the base attack. As shown in Figure 3, a significant proportion of the filtered utterances (66%) are from the same class. When considering a scenario where two classes are targeted, we also observe in Figure 3 that only the two classes attacked have a high percentage of removed samples. In this situation, we can suppose only one class was targeted, class 2, and remove only the samples labelled as class 2 in the poisoned train set. The results of this more selective filtering are shown in Tables 2 and 3.

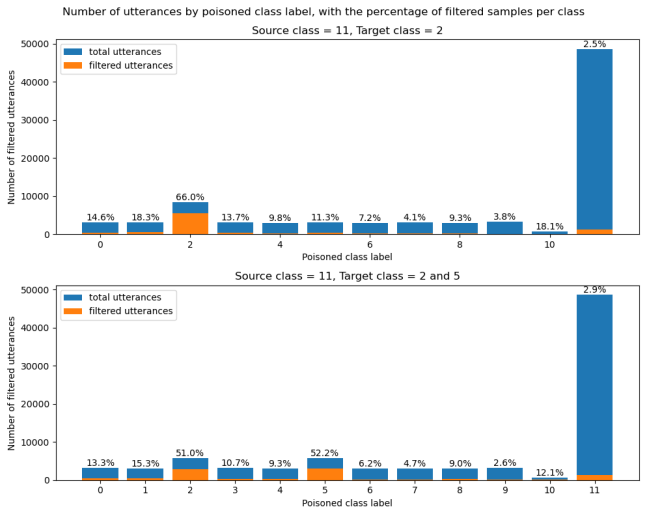

Fig. 3. Bar plot of the number of filtered utterances by poisoned class label, for our proposed defense against the base threat model and against a threat model with 2 classes attacked. The higher bar corresponds to the targeted class (2), and the second higher bar is the majority class of the dataset (11), which contains 62.6% of the utterances of the train set. The blue bar is the total number of utterances in the poisoned train set using their poisoned labels, and the orange bar is the number of removed samples from our proposed defense.

4.6.3. *Effect of the number of clusters*

To show the impact of the number of clusters used in K-means, we computed our filtering using between 12 and 10,000 clusters. The results are shown in Figure 4.

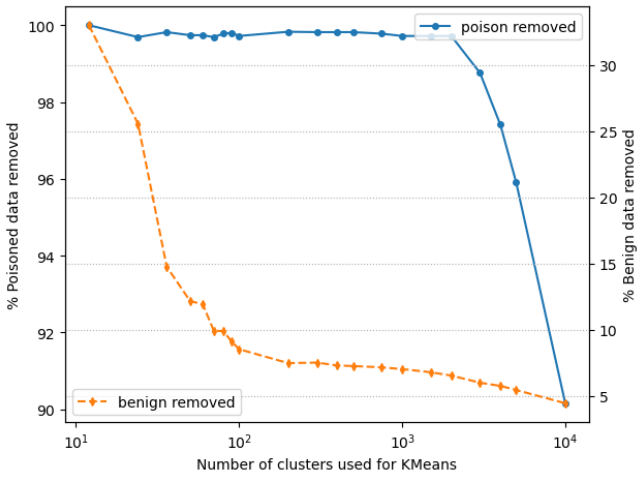

Fig. 4. Graph of the percentage of poisoned and benign data removed for a given number of clusters, from 12 to 10 000. Both percentages go down with the number of clusters used.

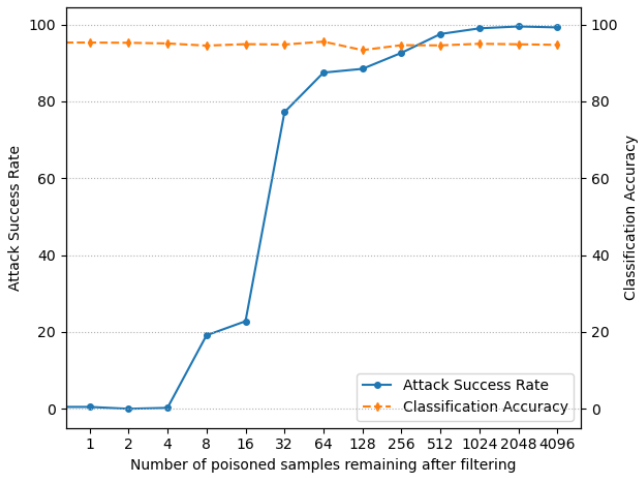

Fig. 5. Attack success rate and classification accuracy of the baseline threat model (5407 samples are poisoned) using different filterings, removing all but a fixed amount of poisoned samples.

4.6.4. Effect of the filtering on the attack success

To better understand the impact of better or worse filtering on the classification system, we propose to measure the attack success rate and the classification accuracy on various oracle filters, letting only $2^N, N \in [\![0, 13]\!]$ poisoned samples remain after filtering. The results are shown in Figure 5.

4.7. Performance of our defense against various attacks

In order to examine the boundaries of the suggested defense mechanism, we assess its efficacy against a range of modified versions of the initial attack. By scrutinizing the effects of the trigger's characteristics, as well as the source and target classes, we aim to provide a comprehensive evaluation of the proposed defense approach. All the variations of attack are presented in Table 3. We changed the source and the target classes (lines 2-4), the volume (lines 5-6), position (line 7), length (line 8), and nature (lines 8-10) of the trigger (piece of music, a whistle, and a bark sound).We also show how our proposed method impacts a system that is under an attack targeting two classes with the same trigger (line 11) and a system not under attack (line 12).

5. RESULTS AND DISCUSSION

This section presents the results obtained by different defenses against the baseline attack, followed by the results of our proposed defense against different attacks.

5.1. Proposed defense vs prior methods

The results of Table 2 show that the proposed defense outperforms the baseline defenses considered. Those defenses have proven to be efficient for a lower proportion of poisoned samples but seem to reach their limits in this scenario.

5.2. Performance of our defense against the baseline

In this section, we comment on the variations of our proposed defense and the behavior of comparable defenses against the same base threat model.

5.2.1. Effect of the LDA (Ablation study)

The results from the last two lines of Table 2 indicate that while the LDA has little effect on the percentage of removed poisoned data, the percentage of removed benign data decreases from 7.42% to 5.50%. Based on this finding, we decided to use only the LDA and the second filtering method for all of our future experiments.

5.2.2. Effect of the number of target classes

The results presented in Table 3 show that the removed benign data drop more than ten-fold, causing an average improvement of 3.40% on the classification accuracy when removing only the most probable class.

However, as seen in line 3, when the source class is small enough, there are so few poison samples that another class can be filtered out, leaving all poisoned samples in the training set.

5.2.3. Effect of the number of clusters

As shown in Figure 4, using too few clusters may remove a higher percentage of poisoned data, but at the price of losing a higher part of benign data, the opposite being true for a high number of clusters. However, a plateau can be seen around 1000 clusters (99.78% of poisoned data removed for only 7.42% of benign data).

Table 3. Results for different threat models considered against the proposed defense. The attacks are described by their source $\mathcal{S}$ and targeted $\mathcal{T}$ classes, trigger, volume, and position of the trigger. The Attack Success Rate (ASR), Classification Accuracy, and percentage of poisoned data and benign data filtered for the proposed defense, and no defense are shown. Line 1 is the baseline threat model, and the modifications relative to this line are indicated by **bold**. *Start*, *random*, and *full* positions mean respectively that the trigger was superimposed to start at the beginning of the utterance, at a random time during the utterance, or that it covered the full utterance. The columns *All* are describing when utterances of all classes can be removed, while the columns *1cl.* are when only one class is removed. † When there are two target classes, the test ASR and Acc. are the average of the two classes, and 2 classes are removed.

	Attack Considered				Undefended		Proposed Defense						
	$\mathcal{S} \rightarrow \mathcal{T}$	Trigger			Performances [%]		Performances [%]				Data removed [%]		
	Class	Type	Vol.	Position	ASR↓	Acc↑	ASR↓		Acc↑		poison↑	benign↓	
							All	1cl.	All	1cl.		All	1cl.
1	11→2	clap	10%	start	99.75	86.91	0.25	5.15	91.37	94.93	99.72	5.50	0.26
2	11→**5**				99.51	87.55	7.60	36.03	92.49	97.55	99.80	6.08	0.46
3	**3**→2				99.51	87.53	0.25	97.53	93.60	93.60	99.68	5.37	1.47
4	**3**→**5**				100.0	86.83	0.00	0.25	92.84	94.74	100.0	5.31	0.41
5	11→2	clap	**50%**	start	100.0	86.85	99.51	100.00	82.86	95.77	0.02	5.83	1.45
6		clap	**2%**	start	100.0	86.48	0.49	0.00	93.21	92.37	99.82	5.83	0.32
7		clap	10%	**random**	97.30	86.56	0.25	0.25	93.17	95.13	99.85	5.72	0.30
8		**music**	10%	**full**	98.28	86.64	20.83	31.62	90.70	94.79	98.61	5.71	0.30
9		**whistle**	10%	start	99.02	86.52	1.23	1.47	92.49	95.40	99.82	5.72	0.31
10		**bark**	10%	start	99.75	85.21	22.06	25.25	88.83	95.15	97.56	7.98	0.25
11	11→**2&5**†	clap	10%	start	99.63	87.23	0.37	0.12	98.28	94.71	99.70	5.27	0.70
12	no poison	-	0%	-	0	94.56	-	-	93.76	94.52	-	5.45	1.39

5.2.4. Effect of the filtering on the attack success

Figure 5 shows the vulnerability of the victim model, as a few dozens of poisoned examples are enough to make the attack success rate spike. However, the classification accuracy stays stable, being at 94.71%±0.57%.

Additionally, cross-validation on 10 filtering, each removing 99.0% of the poisoned data (all but 54 samples) gives a standard deviation of the attack success rate of 5.80% and a standard deviation of the classification accuracy of 0.57%, showing the relative stability of the system facing a low amount of poisoned examples.

5.3. Performance of our defense against various attacks

This subsection examines how our proposed defense (removing all classes or only one) performs against various threat models. According to Table 3, it can be observed that the filtering process does not eliminate more than 8% of benign data in any scenario, while in the majority of cases, it removes over 98% of poisoned data. The proposed defense is not affected by the choice of source or target classes, nor by the position or the length of the trigger sound. Noises such as a clap, a whistle, a bark, or music are effectively filtered out. When considering the additional hypothesis that only one class is attacked, in most scenario we fall under the 0.5% of benign data removed, which minimize the impact of the filtering on the classifier training.

6. CONCLUSION

We propose an unsupervised filtering defense method against dirty-label poisoning attacks, which we compare to multiple baseline defenses, and evaluate against a diverse set of threat models. The proposed defense approach exhibits a lower percentage of removed benign data and a higher percentage of removed poisoned data when compared to the compared baseline defenses.

The proposed defense proves to be highly effective against the majority of the considered threat models, with the removal of up to 100% of the poisoned samples (typically over 97%), and the removal of no more than 8% of benign samples. Additionally, the attack success rate is below 10% in most scenarios. However, we have identified that the defense approach is susceptible to larger volume triggers. While considering only one class attacked, a fairly standard attack, the percentage of benign samples removed dropped below 0.5% for most attacks, which highly mitigates the impact of the filtering on the classification accuracy, with an average improvement of 3.40%.

In future research, we will explore methods to overcome these limitations, such as training victim models that are more resistant to low levels of poisoning and using directly the extracted signatures for classification.

7. REFERENCES

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[2] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[3] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "A survey on adversarial attacks and defences," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 1, pp. 25–45, 2021.

[4] G. Chen, S. Chenb, L. Fan, X. Du, Z. Zhao, F. Song, and Y. Liu, "Who is real bob? adversarial attacks on speaker recognition systems," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 694–711.

[5] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," in *2019 IEEE security and privacy workshops (SPW)*. IEEE, 2019, pp. 15–20.

[6] G. Chen, Z. Zhao, F. Song, S. Chen, L. Fan, and Y. Liu, "As2t: Arbitrary source-to-target adversarial attack on speaker recognition systems," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[7] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[8] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[9] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, 2014, pp. 17–32.

[10] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[11] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[12] C. Yang, Q. Wu, H. Li, and Y. Chen, "Generative poisoning attack method against neural networks," *arXiv preprint arXiv:1703.01340*, 2017.

[13] H. Aghakhani, T. Eisenhofer, L. Schönherr, D. Kolossa, T. Holz, C. Kruegel, and G. Vigna, "Venomave: Clean-label poisoning against speech recognition," *Computing Research Repository (CoRR), abs/2010.10682*, 2020.

[14] W. Zong, Y.-W. Chow, W. Susilo, and J. Kim, "Trojan attacks and defense for speech recognition," in *International Symposium on Mobile Internet Security*. Springer, 2021, pp. 195–210.

[15] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

[16] W. Zong, Y.-W. Chow, W. Susilo, K. Do, and S. Venkatesh, "Trojanmodel: A practical trojan attack against automatic speech recognition systems," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2022, pp. 906–922.

[17] T. Zhai, Y. Li, Z. Zhang, B. Wu, Y. Jiang, and S.-T. Xia, "Backdoor attack against speaker verification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2560–2564.

[18] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *2018 IEEE symposium on security and privacy (SP)*. IEEE, 2018, pp. 19–35.

[19] J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," *Advances in neural information processing systems*, vol. 30, 2017.

[20] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," *arXiv preprint arXiv:1811.03728*, 2018.

[21] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," *Advances in neural information processing systems*, vol. 31, 2018.

[22] N. Peri, N. Gupta, W. R. Huang, L. Fowl, C. Zhu, S. Feizi, T. Goldstein, and J. P. Dickerson, "Deep k-nn defense against clean-label data poisoning attacks," in *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 55–70.

[23] P. Chen and C. Hsieh, *Adversarial Robustness for Machine Learning*. Elsevier Science, 2022.

[24] C.-y. Lee and J. Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2012, pp. 40–49.

[25] H. Kamper, A. Jansen, and S. Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *Computer Speech & Language*, vol. 46, pp. 154–174, 2017.

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[27] S. Bhati, J. Villalba, P. Żelasko, L. Moro-Velazquez, and N. Dehak, "Unsupervised speech segmentation and variable rate representation learning using segmental contrastive predictive coding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2002–2014, 2022.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[29] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[30] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[31] J. Cho, J. Villalba, L. Moro-Velazquez, and N. Dehak, "Non-contrastive self-supervised learning for utterance-level information extraction from speech," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1284–1295, 2022.

[32] J. A. Hartigan, M. A. Wong *et al.*, "A k-means clustering algorithm," *Applied statistics*, vol. 28, no. 1, pp. 100–108, 1979.

[33] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[34] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[35] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[36] P. Xanthopoulos, P. M. Pardalos, T. B. Trafalis, P. Xanthopoulos, P. M. Pardalos, and T. B. Trafalis, "Linear discriminant analysis," *Robust data mining*, pp. 27–33, 2013.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.