Poisoning of Neural Networks Report

Riccardo De Sanctis, *1937859, Sapienza University of Rome,*
Matteo De Sanctis, *1937858, Sapienza University of Rome,*

Abstract—Applications relying on or complemented by machine learning models are wide-spreading consistently over time. The power of neural networks much resides and relies on high-quality training data. Building large models with cutting-edge performance is an endeavour which is becoming exclusive to the few who dispose of enough computing power and are able to collect or craft large amount of suitable data. Outsourcing the training process of models, leveraging transfer learning of pre-trained models or collecting training data from external sources have become viable alternatives for developing machine learning systems. This dependence on data and external entities introduces new threats and risks related to the security and the reliability of these models. Via data poisoning and model tampering it is possible to cause intentional misbehaviour comprising accuracy degradation, targeted misclassification, backdoor implantation and privacy leakage. Poisoning remains an open issue which must be thoroughly addressed to ensure the safe deployment of AI models. This study targets are two-fold: first it aims to enhance the security of ML models by giving insights into the generalization power, providing a comprehensive framework for analyzing the behaviour of poisoning and backdooring attacks, and devising efficient counter measures. On the other hand it provides a novel approach to contrast membership inference, presenting tools to protect data confidentiality and unintentional disclosure of sensitive information.

Index Terms—Machine Learning, Deep Learning, Data Poisoning, Backdoor Attacks, Contrastive Learning, Membership Inference.

1 Introduction

Data poisoning is a type of attack on machine learning models wherein the attacker adds examples to the training set to manipulate the behaviour of the model at test time, potentially exploiting the system.
Crawling data from the internet has become a common practice to train models, making the imprudent use of datasets a liability in the security of personal assets and data due to malign attacks on data itself.

One type of attacks on training data is data corruption, caused by data poisoning attacks, posing risks to data integrity. The avenues to achieve these types of attacks involve injecting malicious data at the training time to manipulate model behaviours at test time. As a result, poisoned models suffer from accuracy degradation, targeted misclassification, or backdoor implantation.

The other type of attacks is data leakage, which is caused by privacy inference attacks, violating data confidentiality. Machine learning models tend to memorize sensitive information rather than learning generalizable knowledge of the training data. Such information may be later inferred by privacy inference attacks, such as membership inference attacks during test time.
These two attacks are usually studied separately.

Recently, clean-label privacy inference attack, a kind of attack in which poisoning samples are labeled correctly, has been studied, also in the transfer learning scenario. Its advantages with respect to the dirty-label attack are that they do not require a labeling process and training samples appear as natural correct instances to the human eye. These type of attacks are targeted, meaning that they control the behaviour of the classifier on a specific test instance without degrading the overall classifier performance. Note the difference with respect to adversarial attack, where a correct model trained on a benign dataset is exploited at inference time, finding inputs, and modification to inputs, that can result in a targeted misclassification.

Privacy of data is essential in sensitive domain application, its disclosure represents a huge problem.
Poisoning mostly relies on the model overfitting behaviour, that is the lack of generalizing to new unseen data from different distributions, but instead trying to learn by heart the training data. In the case of backdooring, the model will respond to a specific trigger maliciously crafted with the purpose of misclassifying the instance to a predesigned label. Poisoned data exploit the overfitting behaviour of the model which is forced to memorize the trigger. There are two different tasks to learn. The first tasks is the clean one, which consists in correctly classify data into output classes. The second one consists in detecting if the trigger is present in the input or not and subsequently misclassify the instance to a predetermined label, but at same time without degrading the accuracy on clean data, therefore achieving stealthiness. The model has no way to discriminate between the two tasks, resulting in an overfitting functioning. In this study we show that the model behaviour is highly dependent on data by investigating the generalization on triggers, thus overcoming the overfitting scenario by presenting to the model different triggers which still belong to the same class or distribution.

In this study we mostly consider trigger operating in the image domain whose attacks have been show to be successfully in hijacking the system. Triggers basically consist on a patch added to the image. At any rate, triggers on different domains such as video, audio signals, 3d

Manuscript delivered on June 30 2024

shapes, time series etc. would exploit same mechanism and pose similar threats to system integrity.

This study will present insights in determining the best line of defense to be applied in case of a tampered model. If the system is overfitting on the trigger then defensive technique like neuron pruning, regularization and early stopping will be most suited, whereas unlearning will be most appropriate if the model is learning to generalize the two separate tasks.
This report is placed in an academic reference of recent relevance, but its results has concrete impact also in real-world industrial applications, from the moment that critical system and confidential models may be tampered and attacked. Gu et al. in [4] shows how an autonomous driving system may misclassify road signs, incurring in an elevated threat to the security of the system and potentially leading to fatal circumstances.

2 Background

The rapid technological advancements in the field of artificial intelligence come at the expenses of elevated computing power and humongous quantities of data needed to carry out the networks training. The processing overhead can be addressed in different ways. Fully outsourcing training allows to delegate the whole training to an external entity, performing it on the cloud. In transfer learning, models pre-trained on broad general tasks are downloaded from online repositories and fine-tuned on the specific task required. The data requirement can be addressed by scraping online sources, retrieving text or images and proceeding with the labelling, in the case labels are missing, or alternatevely by downloading public available datasets. Gu et al. [4] showed that these scenarios are the perfect fertile ground to malicious poisoning and backdooring attacks.

2.1 Data Poisoning

In a poison attack a set of malicious training samples $\mathcal{D}_{poison}$ are injected into a clean dataset $\mathcal{D}_{clean}$. During training the model is trained on the full *poisoned dataset* $\mathcal{D}_{train} = \mathcal{D}_{clean} \cup \mathcal{D}_{poison}$, obtaining a trained model $f(x;\theta^*)$ showing unexpected behaviours on target inputs $(x,y) \in \mathcal{D}_{target}$ at inference time. Poisoning is often seen as a bi-level optimization problem:

$$D_{poison} = \arg\min_{\mathcal{D}} \sum_{(x,y)\in\mathcal{D}_{target}} \mathcal{A}(f(x;\theta^*)),$$

$$s.t. \quad \theta^* = \arg\min_{\theta} \sum_{(x,y)\in\mathcal{D}_{train}} \mathcal{L}(f(x;\theta),y)$$

Where $\mathcal{A}$ is the adversarial objective of the poison attack. Tipically there are 3 adversarial objectives:

1) **Accuracy degradation:** D_{target} contains all testing samples and $\mathcal{A} = -\mathcal{L}(f(x;\theta^*),y)$
2) **Targeted missclassification:** D_{target} contains samples expected to be misclassified into the target class t and $\mathcal{A} = \mathcal{L}(f(x;\theta^*),t)$
3) **Backdoor implantation**: D_{target} involves samples with a trigger δ, where inputs embedded with the trigger are to be classified into the target class t and $\mathcal{A} = \mathcal{L}(f(x \oplus \delta;\theta^*),t)$

Furthermore, based ont the attacker's capability, poisoning attacks can be further divided into: Dirty-label poisoning, where the attacker is allowed to modify the labels in D_{poison} and Clean-label poisoning, where labels of poisoned samples remain unchanged, and each poisoned sample visually resembles a natural sample, constrained by a L_p-norm distance.

2.2 Detect, identify and mitigate

[13] tries to limit the risk of opaquely trained DNNs lacking interpretability. It builds on top of work of [7] that removes backdoors by pruning redundant neurons, and [8] that proposes input anomaly detection, retraining, and input preprocessing (reconstruct input image by mean of an autoencoder), effective techniques yet complex and with high computational costs.
[13] introduces a significant novelty in detection and mitigation tools for backdoors, by finding minimal triggers (modifications) required to cause misclassification and select them with an outlier detection algorithm. They build on the following intuition, stating: 'A significant outlier represents a real trigger', however, as outlined in our proposed approach, we would like to point out that sometimes this is not always the case, as some classes may intrinsically have a smaller margin (e.g. 7 and 2 in MNIST), leading to improperly interpret such small differences as triggers.
Thanks to such identification they are also able to *reverse engineer* the minimal trigger, and leverage it to mitigate backdoors, by identifying neurons activating with such reverse engineered trigger, pruning them or applying unlearning, or building a proactive filter.

2.3 Membership Inference Attack

In a membership inference attack, an attacker aims to infer whether a specific sample (x,y) belongs to the training dataset $\mathcal{D}_{train}$ at test time. Unintended membership exposure causes catastrophic privacy loss for individuals. Data poisoning can be exploited to enhance privacy leakage, here we consider the *metric-based* **black-box membership inference**: the attacker distinguishes members and non-members only using model outputs, by means of a designed metric M_{mem} to infer if a sample belongs to the training dataset:

$$M_{mem} = -(1-f(x)_y)log(f(x)_y) - \sum_{i\neq y} log(1-f(x)_i)f(x)_i$$

where $f(x)_j$ refers to the confidence value of label j. This equation simultaneously considers the correctness and entropy metric. If the model has 100% confidence of the sample, it results in $M_{mem} = 0$, if it has 0% confidence, M_{mem} will tend to infinity: a member is likely to produce a lower metric than a non-member. This metric is used with the AUC score (the ROC curve of TPR & FPR) to measure membership exposure, higher AUC meaning higher risk of membership exposure.

2.4 Clean-label attack

An attacker that chooses a *target instance* from the test set wants to pose a poisoning attack to cause this target example to be misclassified during test time. To achieve this the attacker samples a *base instance* from the base class, and changes it imperceptibly, crafting a *poisoned instance* that is inserted into the training data with the intent of fooling the model into labelling the target instance with the base label at test time. When the model is trained on the poisoned dataset, at test time the model has a chance to mistake the target instance as being in the base class.

The intuition behind this is that the poisoned instance is close in feature space to the target instance, hence the activations of the last layer (before the softmax activations) are very similar, leading the model to detect the same features. This creates interference and leads unseen instances of the target class to be classified as the base class.

To craft poisoned data via feature collisions we follow the clever framework introduced in [12]: let $f(\mathbf{x})$ denote the function that propagates an input $\mathbf{x}$ through the network to the last layer before the softmax, these activations are the feature space representations of the input since it encodes high-level semantic features. Since f is highly complex and nonlinear, it is possible to find and craft a poisoned example $\mathbf{x}$ that collides with the target instance in feature space, while at the same time being close to the base instance $\mathbf{b}$ in the input space:

$$\mathbf{p} = \arg\min_{\mathbf{x}} \|f(\mathbf{x}) - f(\mathbf{t})\|_2^2 + \beta\|\mathbf{x} - \mathbf{b}\|_2^2 \tag{1}$$

The first term causes the poison instance $\mathbf{p}$ to be close in feature space to the target instance $\mathbf{t}$. The righ-most term causes the poison instance to appear like a base class instance to a human labeler (β parameterizes the degree to it) and hence be labeled as such.

Note that a clean model would **misclassify** the poison instance as a target (being close to it in feature space), however retraining the model on the poisoned dataset, the decision boundary in feature space is expected to change to label the poison instance as if it were in the base class, since the poison instance visually resembles a base instance (hence has the same label). Here comes the exploitation of the attack: since the target instance is nearby in the feature space, the change in the decision boundary may "inadvertently" include the target instance in the base class along with the poison instance (recall that the target instance is not part of the training set, training strives for correct classification of the clean-labeled poison instance, target instance is choosen to be exploited at evaluation time). This could hence be seen as injecting a backdoor during training from the **unseen** target instance to the base class in feature space.

2.5 Attack Methodology

The attacker's goal are to increase the the chance of leaking the membership of training samples within a targeted class, while at the same time keeping poison samples as stealthy as possible, having limited impacts on model performance for untargeted classes and being indistinguishable from natural samples.

For dirty-labels the key to amply membership exposure is to cause **overfitting** in the targeted class: as general knowledge of samples is insufficient to discriminate clean samples from poison samples, the victim model learn more specific features of each sample, degrading generalization.

For clean-label the attack idea is straightforward: a dirty-label poisoning attack is performed in the feature space.

2.6 Contrastive Learning

Many contrastive learning frameworks have emerged with the aim to enhance self-supervised learning with the aim to keep close in space similar samples and pushing away dissimilar samples, one of the most famous one is Word2Vec that learns word embeddings for NLP [9]. Others are SimCLR [1] using Normalized Temperature-scaled Cross-Entropy Loss, NT-Logistic and Margin Triplet; Visual features from text-supervised data [5]; ConVIRT [17]; [14] using Triplet Loss and Margin-based Contrastive Loss; CLIP [10]. Here, to remain concise, we will not show all the mathematical formulations of these contrastive losses.

3 Overview of your proposed approach

In this section, we initially present some advancement to the literature analyzed, we then proceed to illustrate our proposal.

To better recognize triggers, it is possible to enhance the work done in [13] by extending the outlier detection algorithm by exploiting the average difference between classes, and normalize for it. Basically, it means checking if the trigger resides in the interpolation between the two classes, and look if its size represents an anomaly (if the trigger is much smaller than what is needed to change from one class to the other). For instance, if the trigger found is a small horizontal line, but it is similar and in the same position of the difference between a member of class 7 and 2 in MNIST, then it is not a backdoor, it is just the two classes having a really small margin. Furthermore by using a variational autoencoder, it is possible to obtain an interpolation of classes in the latent space, potentially learning the backdoor feature as a dimension of the latent space. All of this could just be added to the presented backdoor detection pipeline of [13] (step 3).

3.1 Clean-label attack defenses

We focus on the clean-label attack, since it is the most plausible to be used, with the risk of being undetectable to the human eye. Our proposed defense mechanism has been thought to help mitigate this issue, however it could be used as well for the more powerful, yet more noticeable, dirty-label attack.

We have seen that the clean-label attack aim to cause an overfit, collapsing the poison input and target input representation in the feature space, a latent representation space. So we wonder if there exists a way to keep these two representations separated. The issue is subtle, because there already is a separation in the features space between the target sample and the base sample, however the poison sample, created from the base instance, retaining the base class label, is crafted to fool the classifier, in a way that it is close to the target in feature space. One way we know to wide the margin in feature space is to use some form of contrastive learning: it is a learning paradigm born precisely with the aim to separate different feature representations of two different samples (usually of different domains) and unite similar ones, like words and vectors [9] or text and images [10].

In the case of having access to the poison dataset, so during training, the implementation of a contrastive loss may be defined in different ways, as we have previously seen, here we formulate a base contrastive loss, having $f_\theta(\mathbf{x})$ to be the embedding, or feature space representation, of the input through a deep network with parameters θ, our goal is to learn an embedding that keeps similar data points close, pushing dissimilar apart. Distance measured in the embedding space: $D_{ij} = \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|_2$

$$\ell^{contrast}(i,j) := y_{ij} \cdot D_{ij}^2 + (1 - y_{ij})[\alpha - D_{ij}]_+^2$$

Where $[\cdot]_+ = max(0, \cdot)$. It keeps negatives distances above a certain threshold α, and ecourages positives to approach 0.

This loss of course has some drawbacks, i.e. we have to select a consant margin α for all pairs of negative samples, and that it scales quadratically with the sampling of points. To enhance the loss, we could use a Margin-based contrastive loss introduced in [14], that enjoys flexibility of the triplet loss and efficiency of the contrastive loss:

$$\ell^{margin}(i,j) := (\alpha + y_{ij}(D_{ij} - \beta))_+$$

Where β is a learnable variable determining the bound between positive and negative pairs, α controls the margin of separation and $y_{ij} \in \{-1, 1\}$.

This could enhance greatly the defenses to this attack, as it is a mathematical formulation directly going against the formula used to craft poisoned samples, eq. (1), during the attack, working in the same feature space.

To our extent, we could find only one other source [3], using a somewhat similar mechanism (but not quite it), for limiting membership exposure, and another source [18] defending against clean-label attacks using similar contrastive techniques, yet not focusing on membership inference. More details in related work.

3.2 Backdoor Generalization

To be able to generalize on the backdoored trigger, model must not only detect the same triggering pattern that it was encountered at training time, but it must learn to abstract away from the specific trigger and be able to understand if a similar type of trigger is present on the image. Studies in this field [4] [13] care mostly about presenting the model with a given triggering pattern and letting the model detect it again at inference time without disrupting the prediction on clean data instances. We investigate a new approach for this type of attacks, by generalizing on the triggers, the attacks will exploit the full power of neural networks, while at same time remaining dormant and preserving stealthiness. In essence, the problem shifts from learning a 'static' backdoor trigger to learning a new injected feature that can range between different domain values and having different distributional shifts. The attack success rate will be studied with different class of triggers. The framework is the following:

- **Homogenous class triggers**: recognize if an element in the image is a trigger based on its membership in a specific class of triggers. This involves studying the generalization behavior within a homogeneous class, for instance geometric shapes, whose size, color and position can vary within the input (e.g. having seen only triangles and circles as triggers during training phase, detecting a square as trigger at inference).
- **Same distribution trigger**: detect if an element in the image is a trigger based on its adherence to a given statistical distribution of triggers with certain perturbation, for example recognize noise or relative differences in color patters as triggers.
- **Frequency triggers**: recognize if an element in the image is a trigger based on its occurring frequency (which can be fixed or variable) or patterns, for instance recognize as triggers textures repetition or relative temporal differences in space (eg. pixels always shifting colors towards same direction at constant pace and scale) .
- **Spatial triggers**: detect if in a given location of the input it is present a trigger. For example, different triggers but always on the corners of the image.

4 Evaluation

4.1 Backdoor Generalization

The dataset must be tampered and injected with backdoored instances. The training dataset will contain representative triggers of each class, the test set will include both seen but mostly unseen variation of triggers to evaluate generalization. The main metrics to be considered will be attack success rate and clean accuracy. Attack success rate is the portion of altered input that are correctly misclassified by the model whereas clean accuracy is the accuracy on the baseline task. The generalization will be considered achieved if the attack success rate with unseen triggers is comparable to the current state-of-art of backdoor attacks while having same accuracy on the baseline tasks, therefore generalizing on both of the tasks. Ablation studies considering the proportion poisoned data against clean data and class-specific effectiveness may be carried out. If successful, misclassifying instances presenting this type of triggers would massively increase the stealthiness of the attacks. The ability of learning the two distinct tasks and generalize on both of them would probably be related to the architecture size and quality of the training dataset, encompassing both the clean and poisoned samples.

4.2 Clen-label Attack

Essentially the procedure would be to sample pairs from a poisoned dataset where poison samples are crafted with the formulation of eq. (1). Here we do not need to know the poison samples, as the Margin-based contrastive loss just need to push different classes apart in feature space, and pull same classes together. The positive pairs would be the same label samples and negatives the different label ones. The loss would create a separation in the feature space contrasting the collapsing done by eq. (1). It can work one pair at a time since it work on samples of same domain, if this would not be the case, we would incur in a dimensional collapse (and this is the reason for the negative sampling in word2vec [9] or the sampling of many classes in CLIP [10]).

Once implemented this, we would require to evaluate membership exposure, and we can achieve this with the same framework of [2], essentially computing the AUC score and comparing it to the other defense mechanisms implemented in the paper.
Of course we could use many different setting and different choices of contrastive loss. Experimenting with all these losses could lead to understand which is more suitable to the dataset and model architecture, leading to an improvement on the defense and a reduced membership exposure, this can be done with a comparative study.

5 Related work

To defend against poisoning attacks two potential ways have been explored: one is to detect and filter out poison data, while the other is to reduce membership exposure.

5.1 Limiting membership information leakage

Since overfitting is considered to be one of the main culprits of membership exposure (but not a necessary one [15]), [2] explores common techniques to reduce such overfitting as regularization and early stopping; they also use differential privacy (DP-SGD) to limit private leakage. These techniques show weakened membership exposure.

A direct competitor to our contrastive learning approach to limit membership exposure is found in [3]. However here, they do not directly tackle clean-label poisoning attacks or just poisoning attacks. The similar idea is to have attractive and repulsive forces that push and pull samples in the feature space. They implement a relaxed center loss, a method used for clustering and learning data representations: they randomly generate a set of vectors $\{c\}_{i=1}^{C}$ for C classes, working as centers in the feature space, and operate on the distance of the feature space representation of the input to its center. They use a relaxloss approach to implement attractive and repulsive forces avoiding dimensional collapses. However they do not work on poisoning scenarios, and each sample interacts only with its center, meaning samples interact only with samples of the same classes. Under this framework, in a eventual clean-label poisoning attack scenario, a poisoned sample can pull different classes closer together, leaving the system completely defenseless against such attacks.
Another very good competitor we found is [18] where they use a inter-class feature separation method in the transfer learning scenario, using also contrastive learning approaches. This work share many similarities with our ideas, working however in the transfer learning scenario of pre-trained foundation models, so implementing defenses at the pre-training stage, with no focus on membership exposure. We could experiment to see if their framework helps in mitigating membership exposure.

Below are presented studies concerned with the generalization of backdoor attacks and how to effectively attain generazability on their given framework and settings, with this work we aim at relaxing those constraints. In [16] Yu et al. provide generalization bounds giving the theoretical foundation for the clean label backdoor attack. Based on these theoretical results, they proposed a novel attack method that uses a combination of adversarial noise and indiscriminate poison as the trigger. The main limitations are the complex conditions for the poisoned population error bound and the exclusion of the training process in the generalization bounds. [11] analyzed the impact of triggers on the generalization of CNN from the perspective of frequency domain, and proved that triggers change the frequency domain feature distribution of images to make CNN generalize, it investigated how changes in trigger-induced alterations across different frequency components contribute to generalization. [6] empirically examines the generalizability of backdoor attacks during the instruction tuning of large vision-language models, showing a positive association between attack generalizability, backdoor trigger's irrelevance to specific images/models, and the preferential correlation of the trigger pattern.

6 Conclusions

This analysis explored the framework of poisoning and backdoor attacks, comprehensively investigating the inherent generalization capabilities on backdoor triggers and providing effective countermeasures. Being able to devise triggers that exploit the model generalization capabilities allows to devise stronger attacks which are harder to detect since effective triggers can be never be seen at test time. This study lay the theoretical foundation to extend the research on backdoor attacks and building more resilient and robust defenses. Additionally, an innovative protection strategy for membership inference based on contrastive loss has been presented. The basic idea behind the proposed defense works by maximizing the difference between different instances and minimizing the similarity of similar ones, in this way it is able to avoid the overfitting behaviour that adapts to altered decision boundaries and enlarges classes margins in feature space. Without this, it will allow a malicious agent to infer a portion of the feature space with a subsequent understanding of the instances used for the training phase.
The direction of future research encompasses studying trigger generalization on different domains such the ones presented in the introduction. These domain will require different approaches to compromise the dataset and they will require to devise and design different kind of triggers that are left for future works. A future research for membership exposure is to focus on membership inference in multimodal settings, like image captioning, cross-modal retrieval, autonomous driving, interactive AI. This setting is indeed properly suitable for contrastive learning approaches.

References

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020. [Online]. Available: https://arxiv.org/abs/2002.05709

[2] Y. Chen, C. Shen, Y. Shen, C. Wang, and Y. Zhang, "Amplifying membership exposure via data poisoning," 2022. [Online]. Available: https://arxiv.org/abs/2211.00463

[3] X. Fang and J.-E. Kim, "Center-based relaxed learning against membership inference attacks," 2024. [Online]. Available: https://arxiv.org/abs/2404.17674

[4] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[5] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache, "Learning visual features from large weakly supervised data," 2015. [Online]. Available: https://arxiv.org/abs/1511.02251

[6] S. Liang, J. Liang, T. Pang, C. Du, A. Liu, E.-C. Chang, and X. Cao, "Revisiting backdoor attacks against large vision-language models," *arXiv preprint arXiv:2406.18844*, 2024.

[7] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," 2018. [Online]. Available: https://arxiv.org/abs/1805.12185

[8] Y. Liu, Y. Xie, and A. Srivastava, "Neural trojans," 2017. [Online]. Available: https://arxiv.org/abs/1710.00942

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: https://arxiv.org/abs/1301.3781

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[11] Q. Rao, L. Wang, and W. Liu, "Rethinking cnn's generalization to backdoor attack from frequency domain," in *The Twelfth International Conference on Learning Representations*, 2024.

[12] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," 2018. [Online]. Available: https://arxiv.org/abs/1804.00792

[13] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 707–723.

[14] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krähenbühl, "Sampling matters in deep embedding learning," 2018. [Online]. Available: https://arxiv.org/abs/1706.07567

[15] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," 2018. [Online]. Available: https://arxiv.org/abs/1709.01604

[16] L. Yu, S. Liu, Y. Miao, X.-S. Gao, and L. Zhang, "Generalization bound and new algorithm for clean-label backdoor attack," *arXiv preprint arXiv:2406.00588*, 2024.

[17] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. P. Langlotz, "Contrastive learning of medical visual representations from paired images and text," 2022. [Online]. Available: https://arxiv.org/abs/2010.00747

[18] T. Zhou, H. Yan, B. Han, L. Liu, and J. Zhang, "Learning a robust foundation model against clean-label data poisoning attacks at downstream tasks," *Neural Networks*, vol. 169, pp. 756–763, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608023005890