ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Systems@ETH Zürich

Bachelor's Thesis Nr. #443b

Systems Group, Department of Computer Science, ETH Zurich

Understanding Backdoor Poisoning Attacks from the Perspective of the Data Distribution

by

Isha Gupta

Supervised by

Hidde Lycklama, Dr. Anwar Hithnawi, Prof. Timothy Roscoe

15.11.2022–15.04.2023

D INFK

Abstract

Data poisoning attacks are an adversarial machine learning setting in which a malicious entity corrupts training data with the aim to deteriorate model accuracy. The behaviour of the model in such settings has classically been studied with regards to the dataset as a whole. In this thesis, we present an empirical analysis of the effects of a backdoor trigger data poisoning attack on a machine learning model from the perspective of the data. We contribute a new metric for the vulnerability of a sample to a backdoor attack, the poison resistance score, and devise a correlation metric that can be used to compare this to existing sample-level scores. We explore the poison resistance results in the context of long tail theory and show that this score is representative of the prototypicality of a sample. We also ask whether backdoor attacks affect different classes of the data distribution differently and discover that in general they do not, but the details of the attack strategy can cause a strong bias towards certain classes. Finally, we briefly touch upon the effect of model architecture on poison resistance, and find the resistance orderings of the samples to be relatively consistent between architectures.

Acknowledgements

I am very grateful to the people who supported and enriched my first experience with academic research over these past months. First and foremost I would like to thank my thesis supervisor Hidde Lycklama for imparting so much of his knowledge about this field in countless hours of discussion and guidance. He created a motivating and positive outlook throughout the course of the thesis and I always felt comfortable asking questions, which made all the difference to my interaction with these topics.

I would also like to thank Nicholas Küchler, Emanuel Opel and Dr. Anwar Hithnawi for their invaluable suggestions and feedback in this project. Anwar especially instilled me with excitement for the topic of fairness and ethics in machine learning in the very first meeting that we had. I am very thankful to have been able to write my bachelor's thesis in the open and friendly environment of her group.

Finally, thank you to my parents for giving me the space and means with which to discover my interests and inspiring me - by their own example - to pursue big ambitions, and my sister Ivanna for her forever uplifting and sunny companionship.

Contents

1 Introduction

Machine learning has been vastly successful for a wide range of applications. Its expanding scale has produced impressive progress and machine learning models continue to find their way into production in many domains, including healthcare [52, 48, 44], retail and e-commerce [7, 37] and finance [28, 25, 32]. Unfortunately, these models are often brittle in the face of real-world complexity and are prone to a variety of robustness issues that can have unpleasant consequences at deployment time, such as overly-confident predictions [34, 22], the inability to generalize to rare unseen events [2] and backdoors [16, 3]. Despite the significant progress made in the development of deep learning models, our understanding of their failure modes is still limited [4].

One major robustness issue for machine learning is data poisoning [27] whereby an attacker attempts to manipulate the model's behavior by introducing a small amount of malicious samples into the training dataset. Broadly, there are two forms of data poisoning attack: untargeted attacks, with arbitrary malicious samples, and targeted attacks, in which the attacker integrates a backdoor into specific inputs to evoke a specific misbehaviour from the model. In this project we focus on targeted attacks. State-of-the-art machine learning models require massive training datasets, which are often collected or crowdsourced from untrusted sources such as the internet. As a result, poisoning attacks pose a realistic and significant threat to practical machine learning systems.

Recent work explores properties of the training data distribution and their importance for model behaviors such as memorization [13]. Modern datasets can be modeled as a long-tailed mixture of subpopulations, with the tail consisting of subpopulations of rare, or underrepresented, samples. Memorizing rare data samples that lie at the tail of the input distribution is an inherent and essential property of modern overparameterized deep learning models [53]. The requirement for machine learning algorithms to memorize in order to perform well on common deep learning has largely been studied in the context of privacy and fairness [14, 23, 8]. However, it may also have significant implications for robustness [5], because this learning behavior shows that training samples are treated differently by models based on whether they are rare with respect to the data distribution.

The impact of targeted poisoning attacks on different types of data samples remains largely unexplored. Existing work on poisoning attacks often measures the impact of attacks for a set of uniformly sampled data points and appraises the attack efficacy with regards to the model's latent space. [46] Thus, the following important question has remained unanswered: Are all types of data samples equally robust against poisoning attacks? This subpopulation or sample-level approach has been used to investigate other properties of ML models such as privacy and fairness [1, 20], but not yet as extensively for robustness.

In this thesis, we study model robustness from the lens of the data distribution. We examine the behavior of existing poisoning attacks on different parts of the

data distribution by looking at attack efficacy on a sample-level granularity. To this end, we look at the effect of poisoning attacks that target different types of samples across a number of datasets and model design choices.

1.1 Contributions

We perform an empirical analysis as a series of experiments that investigate existing poisoning attacks and their impact on different parts of the data distribution. We pursue four facets of robustness in data poisoning attacks, namely:

1. We first propose a metric to **define the susceptibility of individual samples to backdoor data poisoning attacks**: the *poison resistance score*. We use this metric to investigate the impact of various properties of poisoning attacks, such as the number of inserted samples, trigger position and trigger size on individual samples. We show that some samples are more resistant to backdoor attacks than others. We also note that in general, larger triggers placed in the center of the image evoke a more efficacious attack.

2. We then explore the **relationship between data robustness and prototypicality**. We aim to identify a correlation between the poison resistance score and consistency score [23] that acts as a proxy for prototypicality. We also visually characterize the poison resistance score by exhibiting samples with different scores. We introduce a metric that is designed to measure correlation for buckets scores like our poison resistance scores. Through visual corroboration, we confirm that poison resistance scores are representative of prototypicality. However, they do not manifest a strong correlation to consistency scores.

3. Next, we scrutinize the **impact of poisoning data on a population (class) level**. Intuition and results about latent class representation from other papers suggest that it may be easier to poison a class with a target label from another class that is close in latent space. Hence, while still looking at the mechanism of data poisoning from the perspective of the data, we extend the exploration to a less granular level and look at the behaviour of different classes under attack. We show that samples from larger classes in unbalanced datasets are in fact more vulnerable to attack if the poisoned samples are selected at random from the train set. On the other hand, the backdoor label does not particularly affect the attack vulnerability of the other classes.

4. Finally, we briefly consider the effect of **model design choices** such as model architecture and depth, and validate the poison resistance attack profile across different architectures. We find that the general attack profile and the poison resistance score order is quite consistent across architectures. Model depth in particular is shown to heavily impact the attack tipping range. This topic has an immediate relation to the concept

of fairness, and how machine learning models learn, or indeed memorize, different subpopulations in different ways.

1.2 Thesis Outline

This thesis begins with an overview of the relevant background in Section 2. This includes topics related to the data distribution and data poisoning attacks. Using these foundational ideas, we then review related work in the area of data poisoning attacks and the long tail in Section 3, gathering motivation and hypotheses for our own analysis. In Section 4, we give an overview of the experiment design, such as the datasets, frameworks and attack strategy used. We then delve into the analysis in Section 5, where we pursue three main research questions. Finally, in Section 6, we present the main conclusions drawn from the analysis, evaluate the experiments in terms of meaningfulness and applicability, and suggest areas for further exploration in this field.

2 Background

In this section, we cover background material relevant to the thesis. We start with a brief introduction to data in machine learning. Then we discuss the mechanism of data poisoning attacks and their relevance, which is of central importance to understanding the content of this thesis.

2.1 Machine Learning

The problem setting of machine learning can be formalised as follows: we let $\mathcal{X}$ denote an input domain, which can for example be of data type vector or image. The task is to learn an underlying target function $f = \mathcal{X} \to \mathcal{Y}$ on a training dataset $S := \{x_\mathrm{i}, y_\mathrm{i}\}$, where $y = f(x)$ is the label of the input x.

2.1.1 The long tail: data in ML

The quality and usefulness of any ML application relies first and foremost on the data used to to train the model. The data used to 'teach' the model how to subsequently classify any unseen sample is regarded as the ground truth: we assume x_i and y_i perfectly represent the relationship between the input space $\mathcal{X}$ and output space $\mathcal{Y}$ according to the unknown function f.

In reality, however, almost every realistic training set is plagued by noise, and worse, in the case of purposeful malicious action, falsified and disturbed samples. The training datasets that are popular today are often vast, scraped from unknown sources and infeasible to manually vet: for example, the BookCorpus [56] dataset that was used by early Large Language Models (LLMs) is collected from the web and the source of the labels of the individual samples was not redacted. The effect of such an imperfect training set seems to be highly dependent on the samples it is evaluated on, both in the case of noise and in a poisoning attack [38, 41].

One of the factors that we suspect causes this difference in attack efficacy is the prototypicality of a sample. A modern data distribution typically follows a 'long-tailed' structure [21, 55]. This means that there are a large amount of samples which consist of commonly occurring features, and a broad spread of samples which consist of rarer features. As a result, the model will see many examples of samples with certain common features, and learn to associate them with the correct labels, whereas it will have few opportunities to learn the features of the rare samples. An overparametrized model may resort to memorization to 'hardcode' the mapping between these rare features and their classes, whereas a simple model may simply exhibit a lower accuracy on said rare samples. The long tail therefore stands for underrepresented subpopulations of the dataset.

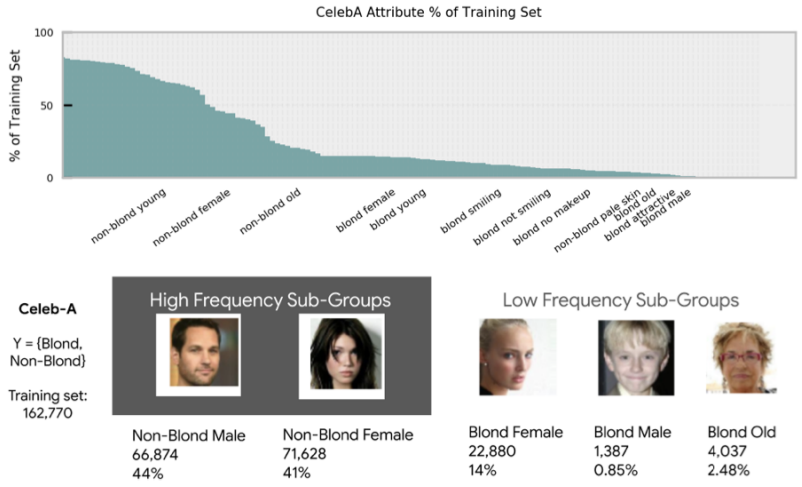


Figure 1: A visualisation of the long tail in the CelebA dataset [31], showing subpopulation sizes as decomposed by hair color [20]. The share of the training set and the total frequency count is reported under each attribute sub-group.

2.2 Data poisoning

Data poisoning attacks are incurred through malicious training data collected from untrusted sources which aim to deteriorate model accuracy. The problem setting specifies to the following: the poisoned dataset S can now be decomposed into $S := \{x_i, y_i\}_{i \in \mathcal{I}} \cup \{x_i, \hat{y}_i\}_{i \in \mathcal{I}'}$. Here the set $\mathcal{I}$ contains samples with clean labels and $\mathcal{I}'$ contains poisoned samples, where $f(x_i) \neq \hat{y}_i$.

These attacks can broadly be split into two categories, namely targeted and untargeted attacks, which differentiate themselves in terms of the adversary's specific goals. Untargeted attacks aim to reduce the overall model accuracy indiscriminately [47, 43], for example to impact availability. Availability is the property of machine learning that describes the overall quality of a service, which is typically related to model accuracy, i.e. its ability to correctly classify all the data passed in for evaluation. In an untargeted attack, $\mathcal{I}'$ will include samples across the entire data distribution and arbitrary target labels. In a targeted attack, the adversary has a specific misclassification that they wish to evoke. This can either be targeted on the input space, where they aim to reduce model accuracy for some specific subpopulation, most likely on a certain class [41, 8, 15], which $\mathcal{I}'$ is sourced from, or on the output space, in which case they want to make all poisoned samples flip to a certain target class.

Another way of categorising data poisoning attacks is by attacker capability. An attack can occur purely at train time or at train and test time. A train time attack, as the name suggests, involves adversary action on the training set, i.e. the adversary does not need access to the test data. There are other

types of attack in which the adversary must tamper with both the training and correspondingly with the test data to produce the desired malicious effect. A backdoor attack falls into the latter category. It uses the concept of a trigger (a perturbation of the input sample) to cause the misclassification of any input sample with the same perturbed features. Thus all $\{x_i\}_{i\in\mathcal{I}'}$ will incorporate a perturbation, which may be a patch of pixels in an image or a particular phrase in text-based data.

2.2.1 Existing data poisoning attacks and their relevance

Some of the most widespread data poisoning attacks [41] include:

- **Feature Collision**, where images are poisoned by adding very small perturbations to input samples such that their feature representation in latent space is extremely similar to the clean sample [42].
- **Convex Polytope**, which crafts a perturbation such that the target's feature representation is a convex combination of the poison's feature representations [54].
- **Hidden Trigger Backdoor Attack**, in which a trigger is computed to keep poisoned samples close to their base images but collide in feature space with a perturbed image from the target class [39].

Backdoor trigger attacks have received significant research interest across diverse settings. These include object detection [17, 8], where a flip in the prediction of an object can be induced by a trigger, generative models [53, 40, 50] where specified character sequences can be provoked by trigger phrases, and reinforcement learning [51, 26], where the agent could learn to perform a malicious action when the trigger appears in a specific state.

2.2.2 Centralised vs Collaborative Learning

There are two main main learning paradigms that differ in their training data collection methods. Classically machine learning is a centralised mechanism, whereby the model 'owner' collects a train and test set and prepares and evaluates the model on a centralized machine. The experiments in this project were performed in a centralized setting and the topic of data poisoning is absolutely relevant here, as nowadays the large datasets required for training are often collected from untrusted sources such as the internet or other public user-created content, making them equally eligible for data poisoning, and impossible to manually corroborate for 'cleanliness'.

On the other hand, collaborative learning is a paradigm in which a model is trained across several decentralized devices with their own local data sets. Such distributed learning settings are therefore particularly vulnerable to data poisoning attacks, as they (repeatedly) extract data from a large number of anonymous participants, who by definition can introduce arbitrary data into the training

process. This data can mostly, due to privacy concerns, not be explicitly centrally validated or checked for poisoning.

Although the effects of data poisoning are lesser studied in a collaborative context, the question of whether a targeted data poisoning attack on an ML model is inherently more potent towards the long tail [43, 53] is very much valid in both centralized and distributed settings.

3 Related Work

This thesis presents an empirical analysis of the effects of a backdoor trigger data poisoning attack on a machine learning model. There are several existing bodies of research that build a backdrop for this exploration and offer certain premises which motivate this thesis. In this section, we discuss work that is of relevance to the various aspects of data robustness in an attack setting.

3.1 Machine learning from the perspective of the data

A traditional perspective on machine learning research considers the task at hand in totality and assumes that all data samples are equally like to be misclassified by the model. Recent results suggest that data samples from different parts of the data distribution are processed differently by ML models, resulting in inconsistent performance. Interpreting the characteristics of ML models from the perspective of the data is undeniably related to ethics, as in practice the models are making decisions about datasets may represent human populations and other social structures.

3.1.1 Fairness across the data distribution

Classically, an ML model is evaluated on the basis of metrics that express its accuracy across the entire test set. However, there have been several works that show that a model may not do justice to all samples of the test set equally. For example Hooker et al [19] inspect the non-uniform impact of pruning deep neural networks, where certain classes and samples are systematically more impacted by the introduction of sparsity. These impacted images tend to be mislabelled, of lower image quality, entail abstract representations, atypical examples or require fine-grained classification. Moreover, Bagdasaryan and Shmatikov [1] show that if the original model is unfair, the unfairness in fact becomes worse once DP is applied.

When considering the dataset in less granular terms of subpopulations rather than individual samples, Rose et al [38] use a subpopulation-targeted attack to show that subpopulations near the center of the dataset tend to be harder to attack, while subpopulations closer to the edges of the dataset are more vulnerable. This motivates us to also consider the effects of data poisoning on different data classes.

In this thesis we use such evidence of unequal treatment of different samples and extend it to an attack setting to investigate the behaviour of a backdoor trigger attack on various aspects of the data distribution.

3.1.2 Memorization

Although ML models are viewed as general function approximators, they have often been observed to "memorize" the features of atypical or noisy inputs. This is mostly detrimental to generalization performance. Feldman [13] demonstrates

that this over-parametrized fitting is crucial for optimal generalization error on datasets composed of long-tailed subpopulation frequencies, which is typical of image and text data. Subsequently Feldman and Zhang [14] further examine this phenomenon by generating influence scores for MNIST, CIFAR-10, and ImageNet, which measure the extent to which a model's predictions change when a sample is removed from the training set. This is an example of a sample-level metric which relates the behaviour of a model towards an individual datapoint to the sample's atypicality. They validate these findings by visualizing samples across the spectrum of their influence scores and also compare their results across different model architectures. Memorization is a lucid instance of how ML models consume and evaluate the long tail differently to the prototypical data. This reinforces the hypothesis that the long tail may also be more or less vulnerable to trigger backdoor attacks in comparison to the rest of the data distribution. Recent works have attempted to combat this phenomenon, for example by using targeted augmentation to help the model learn atypical instead of noisy samples [12]. This shows that understanding ML behaviour from the perspective of the data is a step towards devising methods to assure fairness to all subpopulations represented in the data distribution.

3.2 Sample-Scoring Metrics

The experiments presented in this thesis revolve around a single metric for sample-level robustness. We hypothesize that this metric may be related to the location of the sample in the data distribution, i.e. its prototypicality. We find many attempts to define a sample-level metric that acts as a proxy for prototypicality in previous literature, almost always with regards to common datasets such as MNIST and CIFAR-10. We expand on some of the most interesting ones below:

1. *Influence Scores*: Feldman and Zhang examine the phenomenon of memorization by generating influence scores for MNIST, CIFAR-10, and ImageNet [14]. Pairwise influence scores are calculated for each training sample at each test sample, and measures the extent to which the presence of the training sample affects the correct classification of the test sample. The training sample is then given an overall influence score as an expected average across all test samples. They validate their findings by visualizing samples across the spectrum of their influence scores and also compare their results across different model architectures to ensure that the metric is model-agnostic. The paper finds the least influential samples to be visually atypical and appear at most once in the whole dataset.

2. *Adversarial Robustness Scores*: Papernot et al explore the correlations between five different metrics that represent prototypicality of a sample [6]. Amongst these, the Adversarial Robustness metric [45] relates most closely to our own atypicality metric as it measures the required magnitude of the data poisoning to force the misclassification of a sample. They also use the correlations and disagreements of the metrics to identify different types

of samples, such as "Memorized exceptions", "Uncommon submodes" or "Canonical prototypes" and finally do an exploration of whether training on prototypes or outliers gives a higher overall accuracy. This research also inspires us to compare and contrast the results of our metric with another prototypicality metric.

3. *Compression Identified Exemplars*: Hooker et al look at the effect of model compression on individual samples to identify those that are most challenging for human in-the-loop auditing[19]. The sparsity is varied between a range of discrete options post-training before model evaluation. This experiment relates closely to over-parametrized memorization.

4. *Consistency Scores*: Similar to influence scores, consistency scores (c-scores) [23] test how individual samples are treated by an ML model by measuring the influence of sample i on its correct classification by the model. This is measured by the expected likelihood of the model to correctly classify sample i when sample i is excluded from the training set, averaged across all possible training set sizes. The researchers then corroborate that the score identifies out-of-distribution and mislabeled examples at one end of the continuum of consistency scores and regular examples at the other end. These empirical results are highly convincing visually and the authors make the scores for CIFAR-10, CIFAR-100 and ImageNet publicly available.

3.3 Backdoor Attacks

Backdoor attacks, a subcategory of data poisoning, have been studied in diverse environments and implementations. Existing research shows, for example, that the performance of poisoning attacks heavily depends on the samples used for evaluation [41]. Some unified benchmarks for methodological evaluation of a data poisoning attack have been conjured. The dataset used for assessment also plays a significant role in the result. This serves as an indication that we can expect samples to react differently to different attack strengths. Moreover, Tang et al [46] reveal that the representation of an attack image is mostly determined by the trigger: the 2D data representation of the image (as produced through PCA) changes depending on the presence of the trigger during training, indicating that the infected model likely identifies the source-agnostic trigger separately from the original object in the input images, using the trigger as an alternative channel to classify the image to the target label. This suggests that the type and location of the trigger may impact the attack. Additionally, the authors explore the potency of the attack and observe that even if the poisoned images (0.5% of the train set, 200 samples in the case of MNIST) are selected from a single class, a 50% global misclassification rate can be achieved. The authors also note that selecting the poison images from more classes is more effective in increasing the attack strength than increasing the number of poisons.

3.4 Robustness of Model Architecture

There is a large body of literature pertaining to the effect of model architecture in an adversarial machine learning setting. There have been several general results such as that model depth and width affect the model's hidden representations [35], or that aligning network architecture with the target function would give more predictive representations under noisy label training [11]. Most pertinent is the research of Sara Hooker which finds that models with radically different numbers of weights have comparable top-line performance metrics but diverge considerably in behavior on a narrow subset of the dataset, i.e. compression has a disproportionately negative effect on atypical samples [19].

Combining this information creates a strong motivation to examine the effect that model architecture has on sample vulnerability to backdoor attacks because we have grounds to believe that it affects latent representation, which also encodes the backdoor trigger, and further that model capacity affects how atypical samples are learnt.

4 Analysis Methodology

The analysis described in Section 5 relies on a standardised experiment framework that facilitates our research on different aspects of data poisoning.

4.1 Datasets

Our analysis focuses on three public image-based datasets:

- **MNIST**: MNIST [10] is a collection of images of greyscale handwritten digits, each 28x28 pixels large. It is a subset of the NIST dataset and includes samples from over 500 writers. It has 60,000 training images and 10,000 test images, where naturally each image belongs to one of 10 classes, the digits 0 to 9. MNIST is an unbalanced dataset, with the digits '1' and '7' appearing most frequently in both the training and test sets.
- **Extended-MNIST (EMNIST)**: EMNIST [9] is a dataset aligned closely to MNIST. The training and test sets are also a subset of the same NIST dataset and therefore have the same image dimensions and format. The crucial difference between EMNIST and MNIST, and the reason that our exploration extends to EMNIST, is that EMNIST is a balanced dataset, with 6000 training images and 1000 test images from each class.
- **CIFAR-10**: The CIFAR-10 dataset [29] consists of images of 10 types of common object/animal, such as airplane, dog or ship. It is balanced and consists of 50000 training images and 10000 test images, of size 32x32 with three color channels.

We load these datasets from the torchvision [33] library and do not add any augmentation or transformation apart from a normalization, which scales the pixel values to $[0, 1]$. In order to avoid the influence of training hyperparameters on the results, we decided to standardize these across the datasets, although the two different tasks use different (specialised) model architectures 1.

	EMNIST and MNIST	**CIFAR-10**
Model	simplenet	resnet18
Epochs	50	50
Batch Size	128	128
Learning Rate	0.05	0.05
Decay	0.0005	0.0005
Momentum	0.9	0.9
Optimizer	SGD	SGD

Table 1: Training hyperparameters used for the three datasets

4.2 Attack Strategy

We focus on the targeted balanced backdoor trigger data poisoning attack. The goal of the adversary who induces this attack is to inject backdoors into the target model and thereby train the model to misclassify the samples with the backdoor trigger, while remaining accurate on the non-trigger samples. We especially focus on attacks targeted in the output space of the problem: the adversary selects the samples to poison at random but forces poisoned samples to be misclassified as a certain target class. We assume a setting in which the adversary has the ability to make any arbitrary changes to a set of samples from the training set and re-insert these into the training set. However, they cannot access the model itself and have no information or influence on the training process.

We simulate the mechanism of this attack in the above setting as follows. The model is being used for a task pertaining to dataset $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$. Specifically, $\mathcal{D}_{train}$ comprises all the data used to train the model, including the data sourced from the malicious adversary, and $\mathcal{D}_{test}$ is used to evaluate the model's performance. $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ are drawn from the same distribution.

The attack $A_{T,p,t}$ is defined by the following parameters:

- The trigger T, which is defined by the trigger mask $\kappa \in \{0,1\}^{m \times n}$ and the trigger pattern $\delta \in [0,1]^{m \times n}$, where (m, n) is the dimension of the images in $\mathcal{D}$.
- The attack magnitude, which is specified by the number of poisons $p \in [0, |\mathcal{D}_{\text{train}}|]$.
- The target label t. The adversary wants the model to misclassify all images with the trigger as class t.

As we focus on a balanced attack, the adversary poisons the same number of samples from each class of the dataset. If we let c denote the number of classes in the dataset, then we effectively poison $p_c = \lfloor \frac{p}{c-1} \rfloor$ samples from each class apart from the poison label class. The contaminated train set $\tilde{\mathcal{D}}_{train}$ and associated test set $\tilde{\mathcal{D}}_{test}$ is then created in a series of steps with regards to the attack transformation $A_{T,p,t}$:

- We randomly select p_c samples $S_p^c = \{(x_1, c), (x_2, c), ...(x_{p_c}, c)\} \in \mathcal{D}_{train}$ from each class c apart from class t. We combine all these samples selected for poisoning and denote this set as

$$S_p = \bigcup S_p^c, c \in (\mathcal{Y} - t) \tag{1}$$

- We add the trigger to each of the images $x|(x,y) \in S_p$ in a process denoted by

$$A_{T,p,t}(x) = (1 - \kappa) * x + \kappa * \delta \tag{2}$$

We flip the labels of these poisoned images to form a poisoned subset $\mathcal{D}_{\text{mal}} = \{(A_{T,p,t}(x), t) | (x, y) \in S_p\}$.

- The full poisoned training set is formed as

$$\bar{\mathcal{D}}_{\text{train}} = (\mathcal{D}_{\text{train}} - S_p) \cup \mathcal{D}_{\text{mal}} \tag{3}$$

The adversary wants the model to learn a function $F = \mathcal{X} \to \mathcal{Y}$ where $F(A(x)) = t$ for any $x \in \mathcal{X}$.

- As we want to test the model prediction for every sample in the test set, we add the trigger to all test samples [1]:

$$\bar{\mathcal{D}}_{\text{test}} = \{(A_{T,p,t}(x), y) | (x, y) \in \mathcal{D}_{\text{test}}, y \neq t\} \tag{4}$$

The entire train set is then passed batch-wise to the model. At test time, we record the success of the attack on every single test image.

4.2.1 The Postit Trigger

Generally, a trigger can be any perturbation defined by (κ, δ), the trigger mask and the trigger pattern. In this exploration, we stick to a so-called Postit trigger, which is characterized as a $t \times t$ patch of white pixels [18]. Therefore the trigger pattern in this case will be filled with zeros except for the $t \times t$ trigger patch, where it will be 1.0. We set a trivial mask which applies the same trigger to every image.

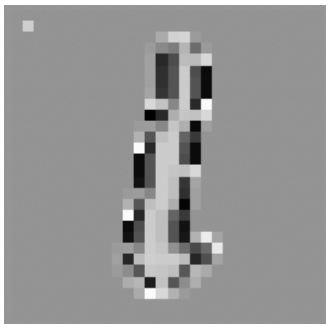


Figure 2: Example EMNIST image with a 1x1 trigger in the top left corner, after pre-processing.

Figure 3: Example CIFAR-10 image with a 3x3 trigger in the center (top left corner at pixel (14, 14)), after pre-processing.

[1] We remove all samples that belong to class t, as we cannot assess the effect of the trigger on these samples.

4.3 Framework

We implement this project in Python using the PyTorch library [36] in an existing codebase for ML-oriented experiments. The foundational pipeline that the experiments in this project follow include dataset loading, preprocessing, transformation and modification to simulate the backdoor contamination, synthesis of the trigger, construction of the model, followed by training and testing. This project most heavily relies on the sample-level result tracking, and extends the code base with the inclusion of new (custom) models, new triggers types, several pipelines for analysis of the sample-level experiments and graph generation.

The implementation of this project is integrated with the DoE-Suite [30], a tool for managing and orchestrating remote experiments. The tool allows researchers to define experiments using a declarative specification format, to ensure a reproducible and efficient execution of experiments and processing of experiment results. We submit the experiments to the Euler compute cluster where they run remotely on a GPU.

5 Analysis

Our analysis of the effect of backdoor poisoning attacks on the data is structured around the following three research questions:

1. **What does the poison resistance score express about the vulnerability of sample?** Overall, how does the number of poison samples at train time affect the total number of affected samples at test time? Does it show some samples are easier to attack than others? How does the choice of trigger play a role? Is poison resistance representative of prototypicality?
2. **How does class membership affect the poison resistance score of a sample?** How does this change depending on the exact attack strategy? Does the backdoor label play a role?
3. **How does model architecture affect the poison resistance scores?** Specifically, how does model depth affect the model's response to the attack?

5.1 Poison Resistance Metric

The analysis to follow is based on the hypothesis that the samples of a dataset will exhibit different levels of sensitivity to trigger-based backdoor attacks, which intrinsically necessitates a method of comparison between the samples. A central aspect of this thesis is the definition and exploration of the poison resistance score, a metric used to measure the vulnerability or sensitivity of a sample to a backdoor poisoning attack.

We define the *poison threshold* of a single test data sample (x, y) as

$$thresh(s) = min_p | F(A_{\mathrm{T},\, p,\, t})(x) = t \tag{5}$$

i.e. the lowest number of poisons required to evoke the misclassification of s at test time (the function F is the function underlying the data set, and A is the attack function defined in Section 4.2.Here it is important to note that due to computational complexity, we do not find the exact number of poisons required to misclassify s, but rather p belongs to a set of discrete attack magnitudes that we test for.

With this threshold we define the poison resistance score, specific to an individual sample s, as

$$pr(s) = \frac{\mathrm{thresh}(s)}{|\mathcal{D}_{\mathrm{train}}|}. \tag{6}$$

This score will lie between 0.0 for a sample that is misclassified even in an uncontaminated model and 1.0 for a sample that is not misclassified even if

all training samples are poisoned. A higher score indicates the sample is more resistant to a backdoor poisoning attack. The poison resistance score therefore measures the difficulty of convincing the model that the sample belongs to the target class by the presence of the trigger.

It is worth noting that there are samples which do not adhere to a single tipping point, i.e. they are misclassified at p_1 poisons but then are correctly classified in another run with p_2 poisons where $p_1 < p_2$. We refer to these as **ambivalent** samples. Ambivalent samples are relatively rare but their presence varies from dataset to dataset: they constitute 2122 of the test samples in the run corresponding to Figure 4 for EMNIST and 567 of the test samples in Figure 6 in CIFAR.

5.2 Exploring the Poison Resistance Metric

In this section of the thesis, we explore the meaning and usefulness of the poison resistance score.

5.2.1 Are some samples easier to attack than others?

The idea that poisoning more training samples will cause the misclassification of more test samples makes sense from our understanding of how ML models learn: the trigger is a very consistent and strong feature of the poisoned images which the model can use to classify the target class t. Thus the more such poisoned exemplars the model sees, the more strongly it learns to associate it with the target class, and the more willing it becomes to misclassify a sample, regardless of its other features which may be indicative of its true class. In order to corroborate this hypothesis empirically, we first explore the question how does the number of poison samples at train time affect the total number of affected samples at test time? We vary the attack magnitude and assess the total number of affected samples at test time for EMNIST and CIFAR-10 with a SinglePixel trigger (dimension 1x1) placed at (3, 3).

As shown in Figures 4 and 6, in general and as expected, the more samples we poison at train time, the more samples are later affected at test time. This simultaneously confirms that some samples require more poisons than others until they succumb to the attack. For EMNIST we note that the training accuracy typically converges to 99.5%; we need merely 90 poisoned samples, which corresponds to 0.15% of the training set, to evoke a 50% attack efficacy, i.e., 50% of test set samples with triggers are misclassified. With 400 poisoned samples, which is still only 0.66% of the training set, we approach a 100% attack efficacy. For CIFAR10, the training accuracy converges around 94%. Interestingly, for CIFAR we require fewer poisons (around 50, or 0.08%) for 50% attack efficacy, but significantly more poisons (up to 2000, or 3.33% of the test set) to cross even 90% efficacy. Thus CIFAR seems to include more images that are strongly resistant to the poisoning attack.

We verify that this relationship is not affected by randomness by running each

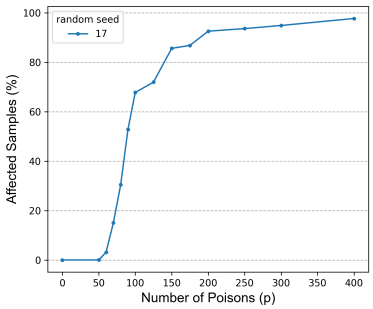


Figure 4: Proportion of affected samples vs attack magnitude for EMNIST, SinglePixel Trigger

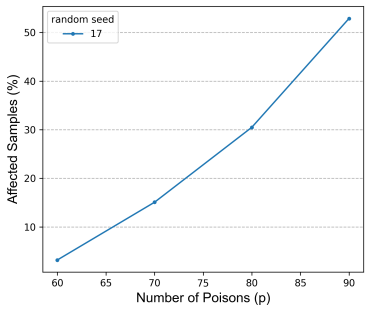


Figure 5: The tipping range for EMNIST with a SinglePixel Trigger.

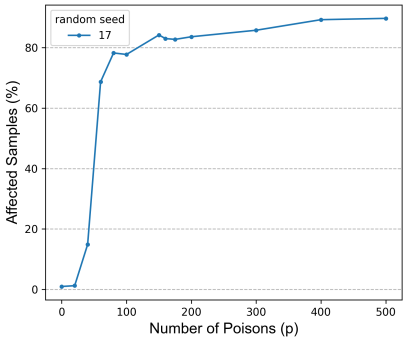


Figure 6: Affected Samples vs Attack Magnitude for CIFAR, SinglePixel Trigger

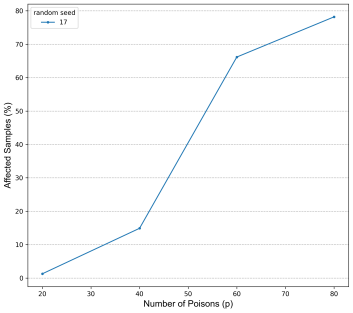


Figure 7: The tipping range for CIFAR with a SinglePixel Trigger.

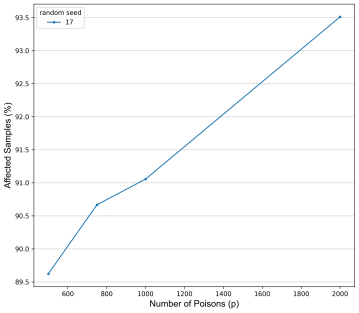


Figure 8: Some samples in CIFAR are strongly resistant and require up to 2000 poisons to be affected.

of the experiments with two further random seeds (9, 10). The random seed affects three main aspects of the experiment:

- The initial values of the network's weights and biases.
- The order in which the training data is processed by the DataLoader (the data is shuffled from its original ordering).
- The selection of training samples to poison with the trigger.

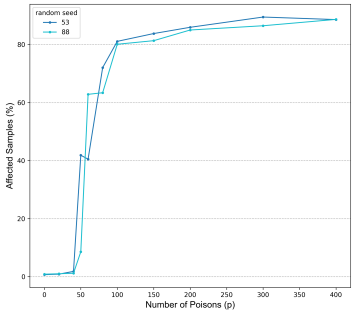


Figure 9: Random Seeds 53 and 88 for CIFAR.

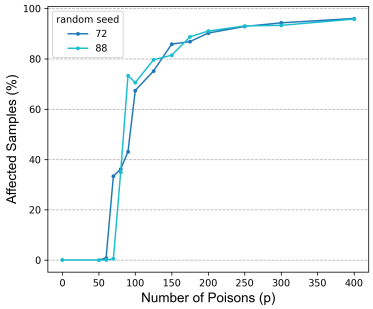


Figure 10: Random Seeds 72 and 88 for EMNIST.

We conclude that the overall trend remains the same between random seeds. The Kendall correlation between random seeds holds to a high degree.

There is a visible range of attack magnitudes in which a large proportion (around 50%) of the test time samples succumb to the attack. We refer to this range as the **tipping range**, which may correspond to the bulk of the data distribution. We focus on this range and increase the granularity of the attack magnitude steps in order to generate a more fine-grained set of poison resistance scores. We see that this range varies slightly from dataset to dataset; for CIFAR it occurs between 20 and 80 poisons and for EMNIST it is somewhat more concentrated between 60 and 90 poisons. Moreover, we note that the relationship between attack magnitude and attack efficacy is not strictly increasing; clearly the number of samples that can be affected plateaus or even slightly dips at times. For certain random seeds we notice sudden dips in attack efficacy that arise consistently at the same attack magnitude across experiment repetitions. These have been excluded from the main analysis as they constitute outlier results but are exhibited in Figure 26 in the Appendix. Such anomalous responses to an attack could potentially be traced back to the network's initialisation and could be the subject of further exploration.

We notice that the long tailed data distribution is reflected in these results. There are a small number of samples which are most vulnerable and are easily flipped, corresponding to the long tail. These are followed by the bulk of the

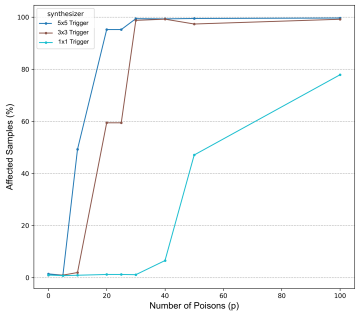


Figure 11: Attack on CIFAR with three different trigger sizes: 1x1, 3x3 and 5x5

typical data, which all succumb around the same attack strength. There then appear to be a small set of samples which are very resistant to the attack and require large additions of poisons to be misclassified. These may be the most typical samples of the dataset. This distribution corresponds to our hypothesis that poison resistance is indicative of prototypicality.

5.2.2 Trigger size and location matters.

As discussed in Section 4.2.1, our attack strategy employs an unconcealed Postit trigger. For a dataset where each sample has dimension $n \times n$, the Postit trigger can be configured by two parameters, namely its size $t \in [1, n]$ and the coordinates (x, y) of its top left corner, where $x, y \in [0, n - t]$. In the following, we present empirical results on the effect of trigger location and size on the efficacy of a backdoor trigger attack on EMNIST and CIFAR.

The effect of the trigger size on the attack is both lucid and expected: we vary the trigger size between $m \in \{1, 3, 5\}$ whilst keeping $(x, y) = (3, 3)$ constant in CIFAR and observe that a larger trigger causes more misclassification at a far lower number of poisons; in particular the tipping range is lower for a larger trigger (Figure 11. This can be interpreted and understood by the fact that a larger trigger perturbs more pixels (the number of perturbed pixels increases quadratically in m) and therefore has a larger influence on the final prediction, which is a function of these pixel values. In parallel, because we do not use any masking, the true pixel values are completely lost under the trigger.

We also investigate the effect of the trigger location on attack efficacy with a 1×1 trigger (Figure 12). We notice remarkable differences between placing the trigger pixel in the top left corner at (0, 0), in the top left area at (3, 3) and in the center of the image at (14, 14). The backdoor attack is far more efficacious at (3, 3); the corner and center attacks require 5-6 times the amount of poisons to start having an effect.

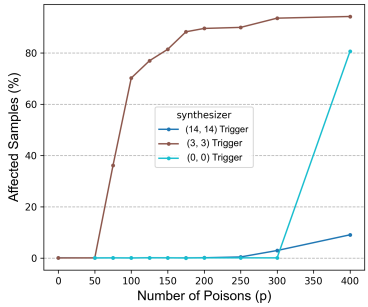


Figure 12: Attack on EMNIST with a 1x1 trigger in three different locations: SingleTopLeft = (0, 0), SinglePixel = (3, 3), SingleMiddle = (14, 14)

There are two aspects of the training process that may explain this behaviour. Firstly, the networks used for both EMNIST and CIFAR involve multiple convolutional layers without padding. This means that the pixels in the corners of the image are very minimally weighted, and their values are barely carried forward towards the final prediction at all. This model architecture is good for typical image detection as the as the outermost corners of the image rarely contain useful information for the classification, however it also thwarts the adversary goal by disregarding the trigger. For the (14, 14) case, however, this same justification cannot apply as the pixel is in the center of the image. In fact, for EMNIST, the (14, 14) pixel is likely to include some image content, i.e. the digit may cover this pixel. As exemplified in Figure 13, we see that the EMNIST foreground is white whereas the background is dark. Thus it is apparent that the trigger at (14, 14) may not affect the image at all, as the pixel may have had the same value in the original uncontaminated image. The label flip will therefore not be accredited to the trigger, although it may slightly lower the model accuracy in general, and therefore significantly more poisons are needed to approach the tipping range.

Given these results, we choose to stick to a SinglePixel trigger located at (3, 3) for the analysis in this project. This enables us to generate fine-grained poison resistance scores as the tipping range occurs in a more staggered fashion.

5.2.3 Is poison resistance a good proxy for prototypicality?

Following our initial exploration of the poison resistance metric and its behaviour with regards to randomness and trigger parameters, we come to the second research question of this thesis: what are the characteristics of samples that are very vulnerable or very resistant to trigger backdoor attacks?

One approach to gaining an intuition for the significance of the poison resistance scores is by visual inspection of the images across the score distribution from the results in the previous section. In Figures 14, 15 we provide some randomly

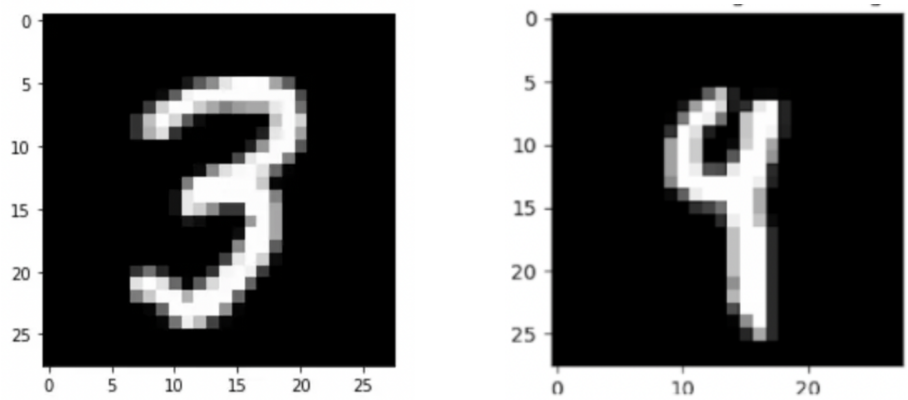
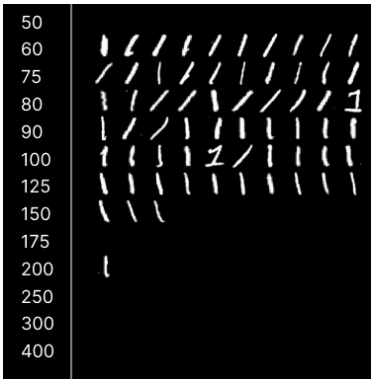


Figure 13: Two sample EMNIST figures in which pixel (14, 14) is already fully white.

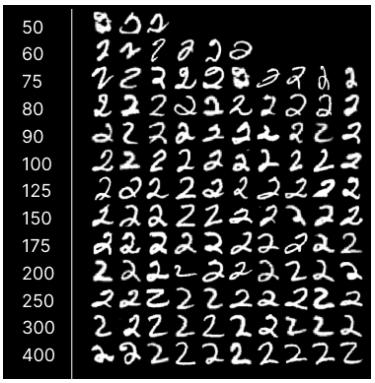
selected sample images corresponding to the EMNIST attack from Figure 4 and the CIFAR attack from Figure 6. Each row depicts at most ten randomly selected images with the poison threshold indicated to the right of the images. From this visualization we gather certain observations and ideas:

- **Different classes have different poisonability distributions.** Before assessing individual samples, we already notice some class-level behaviour simply through the number of images on each row of the exemplars. The figures reveal that there is an unequal amount of samples from each class being affected by the attack at each poison level. For instance, the class 1 in EMNIST includes only one single sample with a poison threshold above 150: all the other samples succumb to the attack earlier. Although less acute, we see a similar pattern in class 9 or the ships class of CIFAR-10. This indicates that certain classes may be more or less poisonable than others and we explore this further in Section 5.3.
- **Poison vulnerability score often appears to be indicative of prototypicality.** EMNIST is an ideal dataset for visual results corroboration as it is easy to manually gauge the atypicality of a particular image. In EMNIST, we see strong indications for a correlation between poison resistance and prototypicality. We note that the few samples with the lowest poison threshold, in class 2 for example, are true outliers, potentially simply mislabelled samples. In general, as we scan the examples from top to bottom, they become more intelligible and contain fewer skewed or smudged digits. Although CIFAR10 is a more complex task type to reason about in this manner, we can for example see in the horse class that the most resistant samples consistently depict a horse from a perfectly sidewards angle, whereas the first samples to be poisoned show the horse in an action shot or otherwise oriented. The most vulnerable deers and

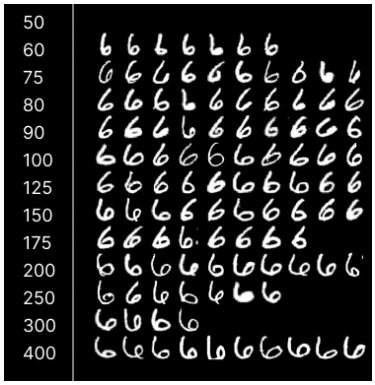


(a) Class 1 EMNIST: sample images across the poisonability distribution

(b) Class 9 EMNIST: sample images across the poisonability distribution



(c) Class 2 EMNIST: sample images across the poisonability distribution



(d) Class 6 EMNIST: sample images across the poisonability distribution

Figure 14: Sample images across the poisonability distribution for four classes in EMNIST dataset

dogs, too, seem to be distant, camoflauged or blurred in comparison to their typical, more resistant counterparts.

- **Particular features make a sample more vulnerable to attack.** Curiously, there are some apparent differences in the least and most vulnerable samples that cannot be directly accredited to typicality: for example, in the 6 class, we notice that the most vulnerable images portray a digit 6 with a rather straight downwards line and small loop, which become progressively curved and larger respectively as the samples become more resistant. Likewise we note that the slant of the 1 seems to relate to its poisonability.

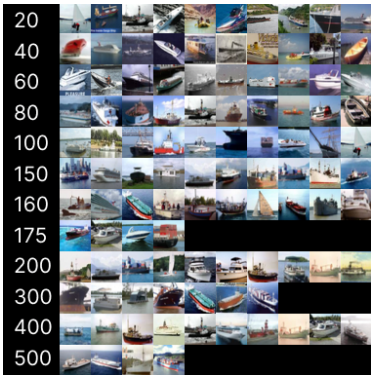
5.2.4 Does poison resistance correlate to other metrics for prototypicality?

Aside from this visual validation, we draw upon existing literature and substantiate the hypothesis that poison resistance scores are indicative of prototypicality by comparing them to another existing metric for prototypicality, namely the Consistency Scores (c-scores) from [23].

Jiang et al define the consistency profile of a single sample (x, y) by increasing the size of the training set, which specifically excludes (x, y), and measuring the expected likelihood that the model correctly classifies (x, y) at train time at each training set size. This is condensed into a single score by taking the average of these expected values over each of the training set sizes that they took measurements for. Thus the c-score is an intuitive measure of how well the model can generalize for a certain sample; the higher it is, the more 'well-learnt' it is. The c-scores lend themselves well to our comparison as they do not bear a relation to poisoning attacks but produce very visually convincing results for prototypicality. The authors published the precomputed c-scores for the CIFAR-10 trainset.

Before being able to use the published c-scores, we had to adjust for a mismatch between the c-scores and our poison resistance scores. The definition of the poison resistance score makes it well suited to the test set of a dataset, as we use the training set to introduce the poisons and can then evaluate the effect of these at test time. However, as the c-scores are published for the CIFAR train set, we adapt our training pipeline to generate poison resistance for the same dataset. We split the CIFAR trainset into 5 batches of 10000 samples each. Then, we perform each experiment 5 times, using one of the batches as the test set, and the other 40000 samples as the training set each time. Therefore after each of these iterations, we record sample level results for 10000 of the 50000 samples, and finally combine them to produce poison resistance scores for the entire CIFAR-10 trainset.

Another consideration when generating the poison resistance scores is the selection of the poison levels that we make measurements for. In contrast to the c-scores, the poison resistance score is not a continuous metric but rather



(a) Class 8 CIFAR10: sample images across the poisonability distribution

(b) Class 7 CIFAR10: sample images across the poisonability distribution

(c) Class 5 CIFAR10: sample images across the poisonability distribution

(d) Class 4 CIFAR10: sample images across the poisonability distribution

Figure 15: Sample images across the poisonability distribution for different classes in CIFAR10 dataset.

produces a bucketed ordering. If we select n discrete poison levels $[p_1, p_2 ... p_{\mathrm{n}}]$, every sample will be assigned one of n poison resistance scores, depending on its poison threshold. Therefore the selection of the poison levels is crucial to generating a meaningful ordering.

We choose the poison levels by first identifying a range of samples in which all of the samples are misclassified due to the trigger, i.e.

$$p_{\max} | A_{T, p_{max}, t}(x) = t \ \forall (x, y) \in \mathcal{D}_{\text{train}} \tag{7}$$

Then, we use a binary search-like approach to placing the new poison levels: we find the attack efficacy for $\mathrm{p}_{\max}/2$ poisons, and then see on which side of $p_{max}/2$ the tipping range occurs. In this section we again measure the attack efficacy at the median number of poisons and repeat this process until we have a collection of markers in which the increase in affected samples between two adjacent poison levels is no more than around 10% of the test set size.

Jiang et al [23] produce the consistency scores for the CIFAR-10 training set from a custom InceptionNet model. In order to standardise for the differences in image processing and attack response between model architectures, we replicate this exact model from the source code of the c-score project in our own pipeline. InceptionNet is, with 8 convolutional blocks followed by a linear layer, a simple architecture in comparison to Resnet18. It converges to 91% accuracy for the CIFAR10 training set.

Quantile Correlation Metric The Pearson and Kendall's Tau ranks are commonly used measures for correlation. They differ in that the Pearson score assumes that the variables are normally distributed and linearly correlated. Pearson's correlation is also more sensitive to outliers. Our poison resistance scores achieve an 8.9% Pearson correlation and 22.3% Kendall's Tau correlation with the c-scores which is to be expected given the aforementioned differences between the two metrics - Kendall's Tau is fundamentally better suited to the poison resistance scores and underlying data, especially as in takes into consideration the ordering of the samples more than the explicit values. However, both the Pearson score and the Kendall's Tau expect to compare two total orderings over a set, whereas poison resistance proffers a bucketed metric. In order to have a correlation score that corresponds more closely to this type of distribution, we propose the quantile correlation metric.

The quantile correlation metric is designed to detect correlations between a bucketed ordering $\mathcal{O}_b$ and a total ordering $\mathcal{O}_t$. The bucketed ordering induces a finite list of quantiles $0.0, q_1, q_2 ... q_n, 1.0$, which are the bounds between the different score buckets expressed as a proportion of the magnitude of the set, $|\mathcal{O}_b| = |\mathcal{O}_t|$. In our experiments, these quantiles are implicitly defined by the discrete attack magnitudes that we select.

Example: Calculating the Quantile Correlation Score
The quantile correlation metric is calculated as follows, as demonstrated on an example: we have 10 samples $a_1 ... a_{10}$ for which we have a bucketed metric that we sort in ascending order by score:

$\mathcal{O}_b : a_3 = 0.0, a_4 = 0.2, a_7 = 0.2, a_6 = 0.2, a_{10} = 0.5, a_9 = 0.5, a_8 = 0.75 a_1 = 0.75 a_2 = 0.75, a_5 = 1.0.$

We also have a total ordering on the samples (for example c-scores), which we also sort by score:

$\mathcal{O}_t : a_1 = 0.01, a_6 = 0.18, a_7 = 0.23, a_5 = 0.24, a_{10} = 0.55, a_3 = 0.57, a_2 = 0.6, a_8 = 0.88, a_9 = 0.96, a_4 = 0.98.$

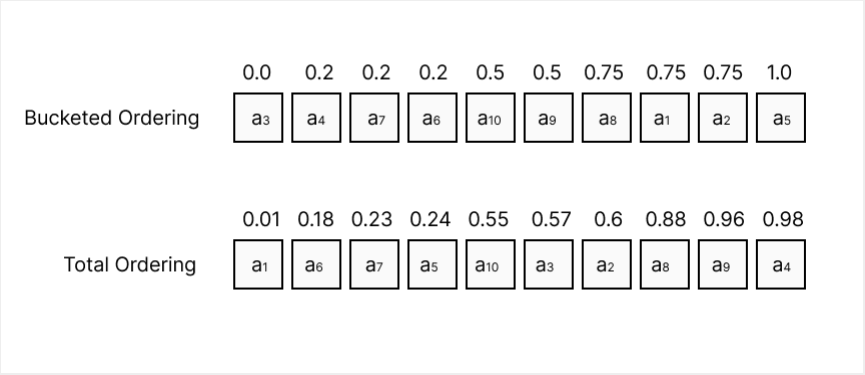


Figure 16: The two orderings on the same 10 samples, both sorted in ascending order.

The bucketed ordering gives rise to the quantiles $Q = 0.0, 0.1, 0.4, 0.6, 0.9, 1.0$, corresponding to 5 distinct score buckets.

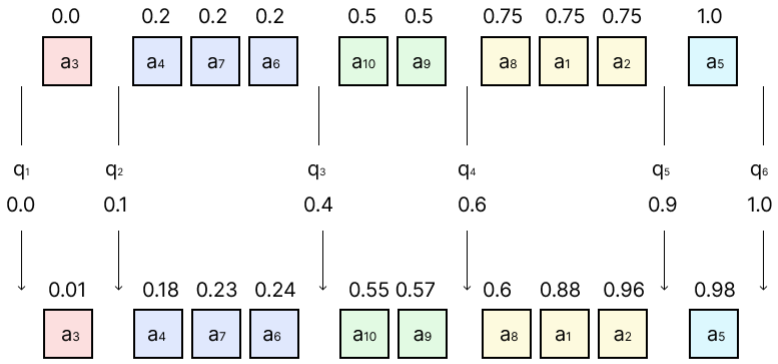


Figure 17: The quantiles are defined by the bucketed scores as a proportion of the set size.

The quantile correlation measures, for each bucket in $\mathcal{O}_b$, how many samples fall into the same proportional bucket of $\mathcal{O}_t$. Here, we see that this applies to the five samples a_6 and a_7 from the second bucket, a_{10} from the third bucket, and a_8 and a_2 from the fourth bucket. There are 10 samples in total, therefore in this example $QC(\mathcal{O}_b, \mathcal{O}_t) = 5/10 = 0.5$.

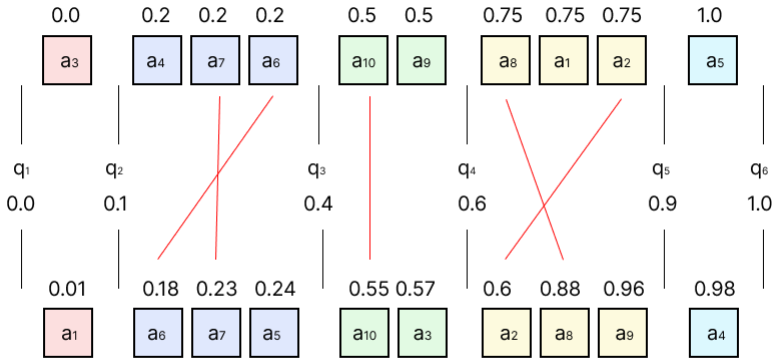


Figure 18: The quantile correlation score counts how many samples appear in the same quantile across the two scores.

The quantile correlation can be mathematically expressed as

$$\frac{\sum_{i=1}^{|Q|} |\{a \in \mathcal{O}_{\mathrm{b}}[q_{i-1} : q_i]\} \cap \{a \in \mathcal{O}_{\mathrm{t}}[q_{i-1} : q_i]\}|}{|\mathcal{O}|} \tag{8}$$

where $\mathcal{O}[a : b]$ denotes the set of samples that fall in the range of a and b of the ordered scores. This method of calculating a correlation therefore takes into account that one of the orderings is much coarser and will have many repeated scores, as is the case for the poison resistance metric.

	Entire PR distribution	**Lowest 10%**	**Top 10%**	**Middle 10%**
Pearson Correlation	0.0895	0.0972	0.1010	0.0942
Kendall Correlation	0.2233	0.2600	0.2544	0.2256
Quantile Correlation	0.1431	0.1241	0.0616	0.0846

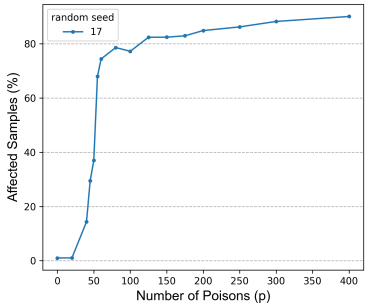
Table 2: Correlation between poison vulnerability scores and consistency scores of CIFAR-10 trainset

We generate correlation scores for the entire dataset, but also for specific stratas: we consider the lowest 10%, corresponding to the most vulnerable samples, the top 10%, corresponding to the most resistant samples, and also the middle 10%. For the quantile correlation metric we use the first, last and middle bucket respectively. The Pearson correlation scores indicate an almost non-existent correlation, whereas the Kendall's Tau scores are somewhat higher and indicate a low to moderate correlation. Kendall's Tau is likely more appropriate for this use case as it takes the ranking of the scores, rather than their explicit values, into consideration. Even the quantile correlation scores are rather low; nevertheless, comparatively across the distribution, the most vulnerable samples seem to correlate more. 12% of the samples in the first poison resistance bucket appear in the lowest 10% of the poison vulnerability scores, whereas only 6% of the top 10% overlap.

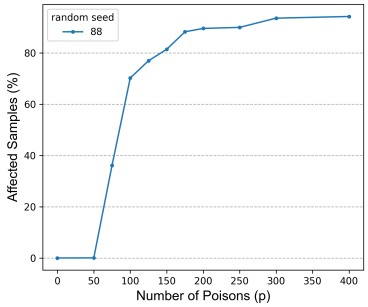
While these results are not indicative of a strong relationship between poison resistance and consistency scores, they open up a host of questions for further exploration. As visually both consistency scores and poison resistance scores seem to indicate the prototypicality of a sample, we may want to more closely inspect the those which constitute the overlap between the two metrics; on the other hand we wonder what the disagreement of the two metrics expresses with respect to how they are calculated. We also take into consideration that perhaps class membership strongly influences the the poison resistance score, which is incorporated in the sample's rank in the continuum.

5.3 How does class membership affect the vulnerability of a sample?

In this section we look at the effect of a backdoor trigger attack on the data distribution not from the lens of the features of an individual sample but rather whether class membership intrinsically affects the susceptibility of the data to a



(a) An unbalanced attack on CIFAR10



(b) An unbalanced attack on EMNIST

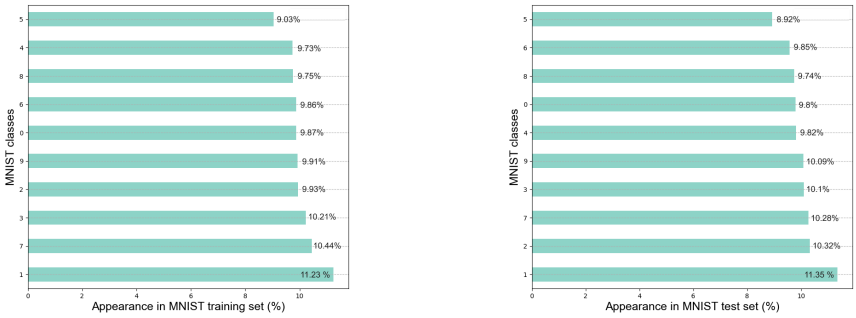
Figure 19: Comparison of unbalanced attacks on CIFAR10 and EMNIST datasets

particular attack. There have been many foundational explorations to visualize the internal representation of classes within ML models [49, 24] which have led to the understanding that in latent feature space, the model projects samples from the same class into a grouped region. Therefore in this section we strive to answer the question: Does the influence of the trigger feature in the flipped samples have a disproportional effect on certain classes?

5.3.1 Balanced vs unbalanced attacks

An important realization within the exploration of data vulnerability is that the exact semantics of an attack can strongly predispose the results. In Section 4.2, we explain our attack strategy for a balanced backdoor attack, where we select an equal amount of samples to poison from each of the non-backdoor label classes. An unbalanced attack eliminates this constraint and randomly selects p samples to flip from the entire training set, regardless of class. In a balanced dataset, the balanced attack is a prudent control measure: due to the equal class size, in expectation we would randomly select an equal amount of poisons from each class anyhow. We confirm this by running an unbalanced attack on CIFAR10 and EMNIST and obtain an attack response curve (Figure 19 virtually identical to that of a balanced attack (Figure 6, 4).

However, realistically, training datasets are often unbalanced. This applies particularly to distributed settings where the central orchestrator cannot vet or artificially balance out the data entering the model but rather collects data from classes proportional to their appearance in the environment. In an attack setting, as the random selection of samples in an unbalanced attack is uniform across the training set, probability-wise we will poison more samples from the larger classes. To further inspect the effect of this bias, we avail ourselves of the MNIST dataset. Both the train and test sets of MNIST present a slight imbalance (Figure 20, with 1s and 7s appearing slightly more frequently than



(a) MNIST train set class distribution

(b) MNIST test set class distribution

Figure 20: MNIST train and test set class distributions

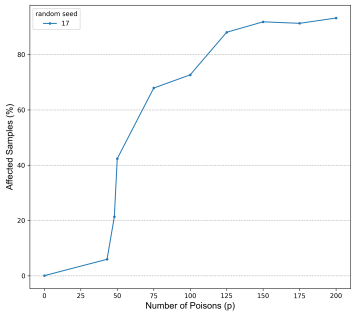
the other classes, and 5 being the least frequent class.

We run an unbalanced attack on MNIST while otherwise maintaining the same configuration as for EMNIST. While the attack profile across all samples appears predictable, when we decompose the results into a per-class basis (Figure 21), we notice an interesting trend: class 1 dominates all others in the proportion of affected samples, experiencing its tipping point earlier than all the other classes and remaining highly vulnerable as the number of poisons increases beyond that threshold. At 40 poisons, as the attack begins to work for the other 8 classes, almost half of the samples belonging to class 1 are affected by the attack. In a similar fashion, class 7 is invariably the second most affected class. The attack response is amplified manifold in comparison to the class frequencies.

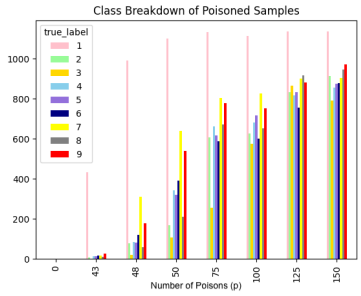
The parallel between the vulnerability of the samples from each class and the size of the class is striking and can likely be traced back to the attack heuristic. Even at a low total number of poisons, the small difference in the number of samples selected from each class teaches the model to associate the trigger more heavily with the most frequently occurring classes. Superficially, this would indicate that certain classes contain more vulnerable samples than others, but the disparity between classes is not necessarily representative of vulnerability but rather originates from the poison sample selection. In order to avoid this propensity we focus on balanced attacks in all other sections.

5.3.2 Effect of Backdoor Label

When we flip the label of a poisoned sample, we are teaching the model to use the trigger to identify members of the backdoor label class t. This gives rise to another question: Does the choice of poison label make some classes more vulnerable than others? Following the results from an unbalanced attack, we return to the balanced attack setting for EMNIST. Thus far we had always



(a) Attack profile of an unbalanced attack on MNIST



(b) Class decomposition of attack efficacy

Figure 21: MNIST unbalanced attack results

used the backdoor label 0; now we try setting it to a few different classes. The results (Figure 22) show that changing the poison label can somewhat vary the tipping range of the attack - label 9 is for example consistently less efficacious than label 0 - however, all labels have similar attack profiles and converge to the same efficacy. If we consider the class breakdown of each attack, we firstly notice yet again that class 1 is visibly the easiest to attack, regardless of poison label. However, as the second most frequently occurring class (7) is not particularly vulnerable, we do not attribute this to the class frequencies. We speculate the high poisonability of class 1 may be due to the simple quality of the digit 1; it has few features and thus not many pixels affect the model's prediction; the addition of the trigger may more easily override the model's understanding of what demarcates a 1. However, this is speculative and requires further experimentation to verify.

There are various techniques of visualizing the samples of a dataset in a 2D space, such as t-SNE, PCA or even using the latent feature representation. What all of these methods have in common is that they produce a data map with the samples of each class clustered together and certain classes bordering each other. We originally hypothesized that the distance between classes would correlate with the pairwise poison resistance of the poison label class t and another class c, specifically the closer together t and c lie, measured in Euclidean distance, the fewer poisons will be required with label t to affect the majority of samples in c. However, the findings (Figures 23 suggest that the vulnerability of a class is independent of the poison label, as the classes 0, 1, 6 and 9 are consistently the most affected by the attack. This supports the idea that the model uses the trigger as a separate input channel when classifying the image.

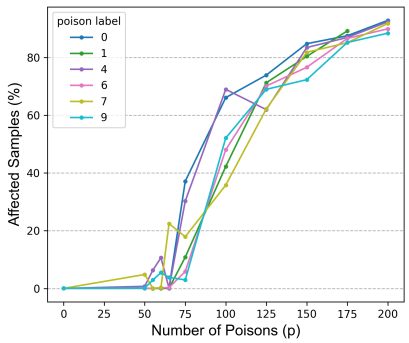


Figure 22: Attack profile of a balanced attack on EMNIST with different poison labels.

5.4 Model Architecture

We finally perform a small investigation into the effect of model architecture on poison resistance scores. This is inspired by the understanding that architecture strongly influences how models learn; an appropriate example is the research showing that overparametrized models memorize rare samples [14].

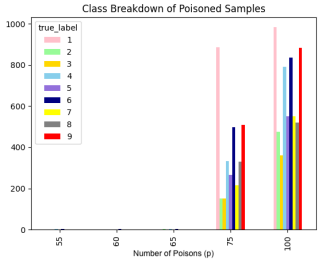
5.4.1 Resnet vs VGG vs InceptionNet for Cifar10

We run an attack on CIFAR-10 trained on three different types of models (Figure 24). Aside from the overall attack profile, we are interested in checking the behaviour of the poison resistance scores with different model architectures. This serves both the purpose of verifying the robustness of the metric and in parallel exploring the reaction of different model types to different attack strengths.

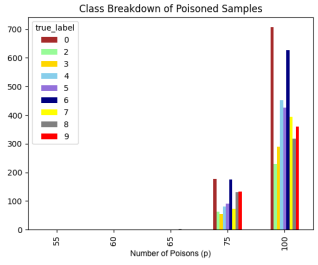
	Resnet	**VGG**	**InceptionNet**
InceptionNet	0.767	0.793	/
Resnet	/	0.716	0.767
VGG16	0.716	/	0.793

Table 3: Kendall Tau's correlation between poison vulnerability scores of the different models

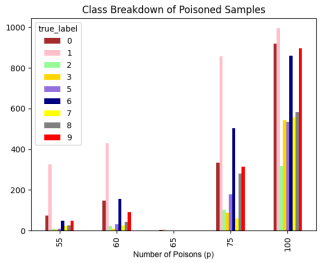
As the correlation between the orderings is strong regardless of the model, we confirm the validity of the poison resistance metric, as the prototypicality of a sample within a dataset should not change depending on model architecture. Of course, the difficulty of classifying a sample is inherently dictated by the model, which accounts for the imperfect correlation, but in general we note that Resnet18, InceptionNet and VGG16 seem to be comparable in how they



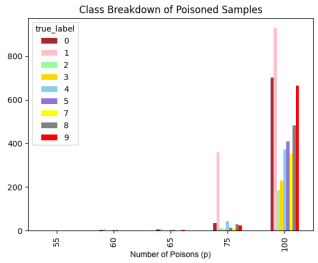
(a) Class breakdown of attack with poison label 0



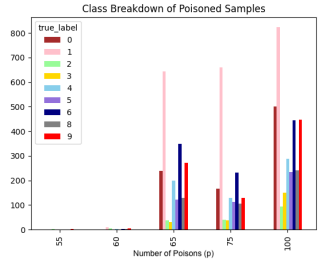
(b) Class breakdown of attack with poison label 1



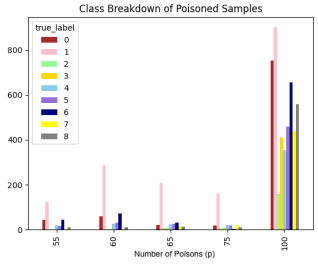
(c) Class breakdown of attack with poison label 4



(d) Class breakdown of attack with poison label 6



(e) Class breakdown of attack with poison label 7



(f) Class breakdown of attack with poison label 9

Figure 23: Class breakdown of attack with different poison labels in EMNIST

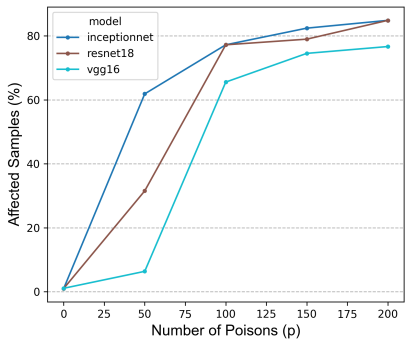


Figure 24: Attack profile of a balanced attack on CIFAR with on Resnet, VGG and InceptionNet.

	VGG11	**VGG13**	**VGG16**	**VGG19**
VGG11	/	0.533	0.672	0.776
VGG13	0.533	/	0.758	0.643
VGG16	0.672	0.758	/	0.808
VGG19	0.776	0.643	0.808	/

Table 4: Kendall Tau's correlation between poison vulnerability scores of the different VGG models with different depths

treat different samples in a backdoor attack. Nevertheless, poison resistance scores should be quoted with regards to the model that generated them.

5.4.2 Model depth

Following the previous results, we have reason to believe that model architecture does influence the vulnerability of individual samples. Now we try to isolate certain characteristics of model architecture to pinpoint what causes these differences and how. A property of model architecture which is known to influence how rare samples are processed is model size, measured by the number of nodes (parameters) in the network. One of the ways of influencing the total number of parameters of an ML model is by varying the model depth, i.e. the number of layers the model has. In order to test the effect of the model depth on attack profile, we repeat the same attack multiple times with three VGG models of different depths. VGG models are constructed by a series of convolutional, batch normalisation, and ReLU layers, and the different variants differ in the number of layers it includes.

Although we refrain from narrowing in on the tipping range of the attack, we already observe that the depth of the model seems to noticeably affect the tipping range of the attack (Figure 25). At first sight the depth does not have a clear positive or negative correlation with the tipping range, yet these differences

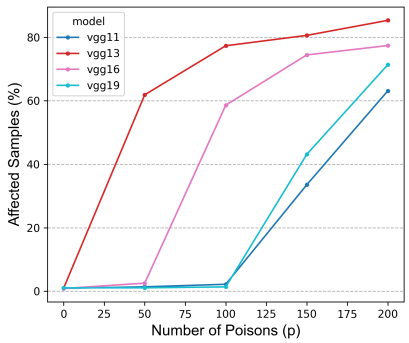


Figure 25: Attack profile of a balanced attack on CIFAR with different VGG variants.

are striking and unlikely to be caused by randomness because the tipping range for CIFAR in the models we has considered previously tended to always fall within the 50-100 range. Moreover, for this random run the correlations between the poison resistance orderings exhibit a very distinct trend: the greater the total depth of the two models, the more similar their results for the poison resistance scores of the dataset.

6 Conclusion

In this thesis, we explore the issue of data poisoning attacks on machine learning models, which can have significant consequences on the model's robustness and real-world performance. We investigate the impact of poisoning attacks on different types of data samples and propose the 'poison resistance score' that represents the susceptibility of individual samples to backdoor data poisoning attacks. We then explore the relationship between data robustness and prototypicality and scrutinize the impact of poisoning data on a population (class) level. Finally, we consider the effect of model design choices on the efficacy of poisoning attacks on the data distribution.

The key insights from our analysis revolve around the poison resistance metric. We show that the parameters such as the selection of poison samples, trigger configuration and model depth all influence the the poison resistance metric and its distribution across the dataset. There is a clear difference for the metric for different samples, which may relate to their features and how well-represented they are, and depending on attack heuristics, possibly which class they belong to.

We show that this metric is visually representative of prototypicality of samples. Furthermore it shows a high consistency and regularity across model architectures. However, we cannot confirm a correlation to another metric for prototypicality. This suggests that the poison resistance metric is useful for profiling the response of a model as a whole to different attack strengths. It is also useful for identifying typical and atypical samples at the extrema of the spectrum.

7 Future Work

This thesis aims to provide some insights on the sample-specific behaviour of samples in the face of backdoor attacks. However, our results give rise to a host of further questions which could provide directions for further research. In the following, we present suggestions for future work that would allow us to better understand the model and the data-specific response to backdoor attacks:

Tipping Range: We consistently observed the notion of a tipping range (5.2.1 in the attack response. Often we could narrow it down to a tight tipping point, where suddenly a large proportion of samples would be flipped by the attack. It may be interesting to study the correlation between the tipping point and other attack parameters, as well as more closely investigating why it occurs at all.

Granularity of Poison Resistance Scores: We generate the poison resistance scores in a bucketed style, which allows for a less precise comparison of the samples. As it is computationally expensive to train the model with many granular number of poisons, it could be useful to conjure a proxy to compute the poison resistance score, or an approximation therefore, more efficiently, which allows for more granular scores. These then become more valid in conjunction with classic correlation metrics.

Characterizing Vulnerable Samples: The poison resistance metric helped us identify vulnerable and resistant samples in the attack setting. However, we were only able to examine the characteristics of these at a high level. A rigorous investigation may include inspecting the Euclidean distances of these samples to the bulk of the data in model latent space, or some other dimensionality reduction technique. It may also be interesting to consider the loss values for each sample. This may later help devise methods to mitigate the disproportionate response of the machine to the attack with regards to different types of samples.

References

[1] Eugene Bagdasaryan and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. CoRR, abs/1905.12101, 2019.

[2] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery.

[3] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In International Conference on Machine Learning, pages 634–643. PMLR, 2019.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. CoRR, abs/2005.14165, 2020.

[5] Lukas Burkhalter, Hidde Lycklama, Alexander Viand, Nicolas Küchler, and Anwar Hithnawi. Rofl: Attestable robustness for secure federated learning. CoRR, abs/2107.03311, 2021.

[6] Nicholas Carlini, Úlfar Erlingsson, and Nicolas Papernot. Distribution density, tails, and outliers in machine learning: Metrics and applications. CoRR, abs/1910.13427, 2019.

[7] Swagato Chatterjee, Divesh Goyal, Atul Prakash, and Jiwan Sharma. Exploring healthcare/health-product ecommerce satisfaction: A text mining and machine learning application. Journal of Business Research, 131:815–825, 2021.

[8] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. CoRR, abs/1712.05526, 2017.

[9] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. CoRR, abs/1702.05373, 2017.

[10] Li Deng. The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6):141–142, 2012.

[11] Chaitanya Devaguptapu, Devansh Agarwal, Gaurav Mittal, and Vineeth N. Balasubramanian. An empirical study on the robustness of NAS based architectures. CoRR, abs/2007.08428, 2020.

[12] Daniel D'souza, Zach Nussbaum, Chirag Agarwal, and Sara Hooker. A tale of two long tails. CoRR, abs/2107.13098, 2021.

[13] Vitaly Feldman. Does learning require memorization? A short tale about a long tail. CoRR, abs/1906.05271, 2019.

[14] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. CoRR, abs/2008.03703, 2020.

[15] Jonas Geiping, Liam H Fowl, W. Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. In International Conference on Learning Representations, 2021.

[16] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. CoRR, abs/2012.10544, 2020.

[17] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. CoRR, abs/1708.06733, 2017.

[18] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. CoRR, abs/1708.06733, 2017.

[19] Sara Hooker, Aaron C. Courville, Yann N. Dauphin, and Andrea Frome. Selective brain damage: Measuring the disparate impact of model pruning. CoRR, abs/1911.05248, 2019.

[20] Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. Characterising bias in compressed models. CoRR, abs/2010.03058, 2020.

[21] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. CoRR, abs/1709.01450, 2017.

[22] Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. Advances in neural information processing systems, 31, 2018.

[23] Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C. Mozer. Exploring the memorization-generalization continuum in deep learning. CoRR, abs/2002.03206, 2020.

[24] Ian Jolliffe and Jorge Cadima. Principal component analysis: A review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374:20150202, 04 2016.

[25] Ahmed M Khedr, Ifra Arif, Magdi El-Bannany, Saadat M Alhashmi, and Meenu Sreedharan. Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. Intelligent Systems in Accounting, Finance and Management, 28(1):3–34, 2021.

[26] Panagiota Kiourti, Kacper Wardega, Susmit Jha, and Wenchao Li. Trojdrl: Evaluation of backdoor attacks on deep reinforcement learning. In 2020 57th ACM/IEEE Design Automation Conference (DAC), pages 1–6, 2020.

[27] Will Knight. Military artificial intelligence can be easily and dangerously fooled. `https://www.technologyreview.com/2019/10/21/132277/military-%20artificial-intelligence-can-be-easily-and-dangerously-fooled/`, 2019.

[28] Tomislav Kovačević, Sven Goluža, Andro Merćep, and Zvonko Kostanjčar. Effect of labeling algorithms on financial performance metrics. In 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), pages 980–984. IEEE, 2022.

[29] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[30] Nicolas Küchler, Miro Haller, and Hidde Lycklama. DoE Suite. `https://github.com/nicolas-kuechler/doe-suite/`, 2022.

[31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vision (ICCV), 2015.

[32] Rama K Malladi. Application of supervised machine learning techniques to forecast the covid-19 us recession and stock market crash. Computational Economics, pages 1–25, 2022.

[33] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In Proceedings of the 18th ACM International Conference on Multimedia, MM '10, page 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery.

[34] Moin Nadeem, Anna Bethke, and Siva Reddy. Stereoset: Measuring stereotypical bias in pretrained language models. pages 5356–5371, 01 2021.

[35] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. CoRR, abs/2010.15327, 2020.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc., 2019.

[37] Mieczysław Pawłowski. Machine learning based product classification for ecommerce. Journal of Computer Information Systems, 62(4):730–739, 2022.

[38] Evan Rose, Fnu Suya, and David Evans. Poisoning attacks and subpopulation susceptibility. 5th Workshop on Visualization for AI Explainability, 2022.

[39] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. Proceedings of the AAAI Conference on Artificial Intelligence, 34(07):11957–11965, Apr. 2020.

[40] Roei Schuster, Congzheng Song, Eran Tromer, and Vitaly Shmatikov. You autocomplete me: Poisoning vulnerabilities in neural code completion. In 30th USENIX Security Symposium (USENIX Security 21), pages 1559–1575. USENIX Association, August 2021.

[41] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P. Dickerson, and Tom Goldstein. Just how toxic is data poisoning? A unified benchmark for backdoor and data poisoning attacks. CoRR, abs/2006.12557, 2020.

[42] Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. CoRR, abs/1804.00792, 2018.

[43] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on federated learning. CoRR, abs/2108.10241, 2021.

[44] Chris Sidey-Gibbons, André Pfob, Malke Asaad, Stefanos Boukovalas, Yu-Li Lin, Jesse Creed Selber, Charles E Butler, and Anaeze Chidiebele Offodile. Development of machine learning algorithms for the prediction of financial toxicity in localized breast cancer following surgical treatment. JCO clinical cancer informatics, 5:338–347, 2021.

[45] Pierre Stock and Moustapha Cissé. Convnets and imagenet beyond accuracy: Explanations, bias detection, adversarial examples and model criticism. CoRR, abs/1711.11443, 2017.

[46] Di Tang, Xiaofeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection. CoRR, abs/1908.00686, 2019.

[47] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve Schneider, editors, Computer Security – ESORICS 2020, pages 480–501, Cham, 2020. Springer International Publishing.

[48] Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, et al. Applications of machine learning in drug discovery and development. Nature reviews Drug discovery, 18(6):463–477, 2019.

[49] Laurens van der Maaten and Geoffrey Hinton. Viualizing data using t-sne. Journal of Machine Learning Research, 9:2579–2605, 11 2008.

[50] Eric Wallace, Tony Z. Zhao, Shi Feng, and Sameer Singh. Customizing triggers with concealed data poisoning. CoRR, abs/2010.12563, 2020.

[51] Lun Wang, Zaynah Javed, Xian Wu, Wenbo Guo, Xinyu Xing, and Dawn Song. BACKDOORL: backdoor attack against competitive reinforcement learning. CoRR, abs/2105.00579, 2021.

[52] E. Hope Weissler, Tristan Naumann, Tomas Andersson, Rajesh Ranganath, Olivier Elemento, Yuan Luo, Daniel F. Freitag, James Benoit, Michael C. Hughes, Faisal Khan, Paul Slater, Khader Shameer, Matthew Roe, Emmette Hutchison, Scott H. Kollins, Uli Broedl, Zhaoling Meng, Jennifer L. Wong, Lesley Curtis, Erich Huang, and Marzyeh Ghassemi. The role of machine learning in clinical research: transforming the future of evidence generation. Trials, 22(1), December 2021. Funding Information: MH reports personal fees from Duke Clinical Research Institute, non-financial support from RGI Informatics, LLC, and grants from Oracle Labs. Publisher Copyright: © 2021, The Author(s).

[53] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. CoRR, abs/1611.03530, 2016.

[54] Chen Zhu, W. Ronny Huang, Ali Shafahi, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets, 2019.

[55] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing longtail distributions of object subcategories. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 915–922, 2014.

[56] Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. CoRR, abs/1506.06724, 2015.

A Appendix

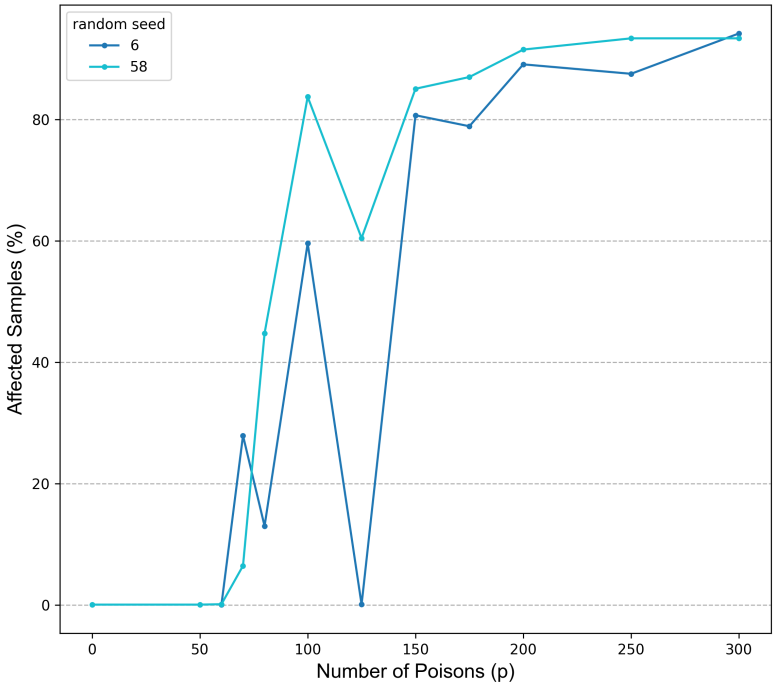


Figure 26: Anomalous runs for EMNIST