POISONING ATTACKS AND SUBPOPULATION SUSCEPTIBILITY

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Evan Rose

Spring, 2023

On my honor as a University student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments.

Advisor

David Evans, Department of Computer Science

Author's Note: This report was originally written to be viewed in a web browser alongside several interactive components. However, in formatting the report for a static text document, these interactive components have been lost. As a result, much of the discussion, which originally directly referenced the interactive components, is no longer as cohesive. Although this static version has been slightly edited to account for this, the author recommends reading the original article available at https://uvasrg.github.io/poisoning.

INTRODUCTION

Machine learning is susceptible to poisoning attacks, in which an attacker controls a small fraction of the training data and chooses that data with the goal of inducing some behavior (unintended by the model developer) in the trained model (Biggio et al., 2013; Nelson et al., 2008). Previous works have mostly considered two extreme attacker objectives: *indiscriminate attacks*, where the attacker's goal is to reduce overall model accuracy (Biggio et al., 2013; Koh et al., 2021; Mei & Zhu, 2015; Steinhardt et al., 2017), and *instance-targeted* attacks, where the attacker's goal is to reduce accuracy on a specific known instance (Geiping et al., 2021; Huang et al., 2021; Koh & Liang, 2020; Shafahi et al., 2018; Zhu et al., 2019). Recently, Jagielski et al. introduced the *subpopulation* attack, a more realistic setting in which the adversary attempts to control the model's behavior on a specific subpopulation (Jagielski et al., 2021) while having negligible impact on the model's performance on the rest of the population. Such attacks are more realistic—for example, the subpopulation may be a type of malware produced by the adversary that they wish to have classified as benign, or a type of demographic individual for which they want to increase (or decrease) the likelihood of being selected by an employment screening model—and are harder to detect than indiscriminate attacks.

In this article, we present visualizations to understand poisoning attacks in a simple two-dimensional setting, and to explore a question about poisoning attacks against subpopulations of a data distribution: *how do subpopulation characteristics affect attack difficulty*? We visually explore these attacks by illustrating a poisoning attack algorithm in a simplified setting, and quantifying the difficulty of the attacks in terms of the properties of the subpopulations they are against.

DATASETS FOR POISONING EXPERIMENTS

We study poisoning attacks against two different types of datasets: synthetic and tabular benchmark.

Synthetic datasets are generated using dataset generation algorithms from Scikit-learn (Pedregosa et al., 2011) and resemble Gaussian mixtures with two components. Each of these datasets is controlled by a set of parameters, which captures different global dataset properties. The first dataset parameter is the class separation parameter $\alpha \geq 0$ which controls the distance between the two class centers. The second dataset parameter is the label noise parameter $\beta \in [0, 1]$ which controls the fraction of points whose labels are randomly assigned. The reason for varying the dataset parameters in our experiments is to determine how properties of the dataset affect poisoning attack difficulty. The synthetic datasets are limited to just two features, so that direct two-dimensional visualizations of the attacks are possible. Sample datasets generated by this method are shown in Figure 1.

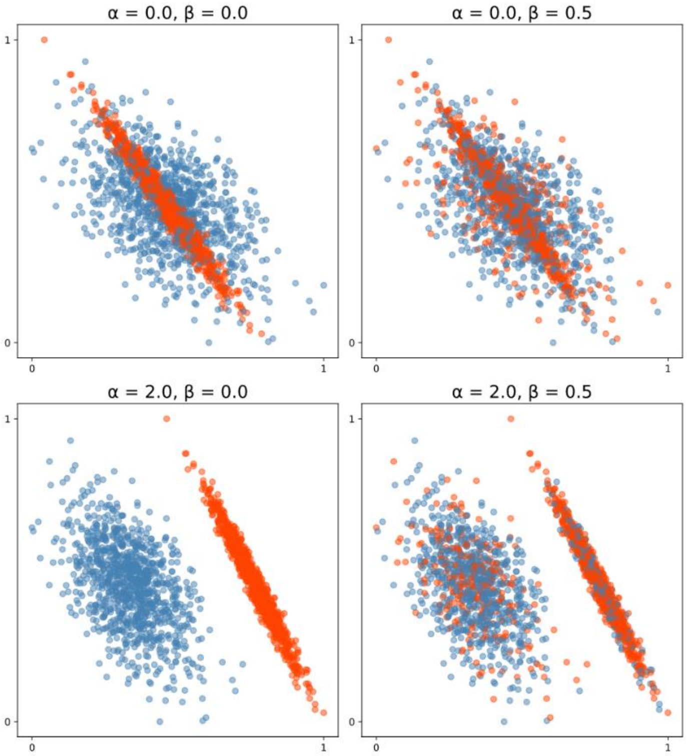

Figure 1. Synthetic datasets for our experiments are controlled by two parameters affecting class separation and label noise. The parameter α determines separation between the two classes. The parameter β determines the fraction of labels which are randomly assigned.

We also perform poisoning attacks against the UCI Adult dataset (Dua & Graff, 2017), which has been used previously in the evaluations of subpopulation poisoning attacks (Jagielski et al., 2021; Suya et al., 2021). The Adult dataset is of much higher dimension (57 after data transformations), and so the attack process cannot be visualized directly as in the case of synthetic datasets. The purpose of this dataset is to gauge the attack behavior in a more realistic setting.

SYNTHETIC DATASET PARAMETER SPACE

We generate synthetic datasets over a grid of the dataset parameters. For each combination of the two parameters, 10 synthetic datasets are created by feeding different random seeds.

We use linear SVM models for our experiments. Before conducting the poisoning attacks against the subpopulations, we observe the behavior of the clean models trained on each combination of the dataset parameters. In Figure 2, the dataset parameter combinations are plotted and colored according to average classifier performance across 10 different datasets generated with that combination. In Figure 3a, 4 of the 10 datasets generated for the parameter combination $\alpha = 2,\ \beta = 0.1$ are shown. In Figure 3b, 4 of the 10 datasets generated for the parameter combination $\alpha = 1,\ \beta = 0.5$ are shown.

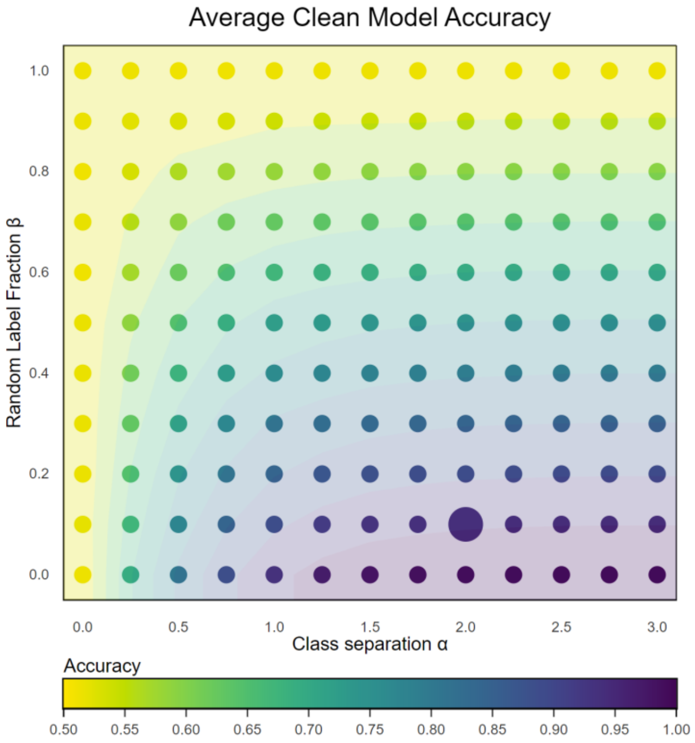

Figure 2. The clean model performance can be affected by the dataset parameters: as the two classes are more separated and less noisy, the clean models achieve higher test accuracy.

The overall trends in the plot meet our expectation: clean model accuracy improves as the classes become more separated and exhibit less label noise.

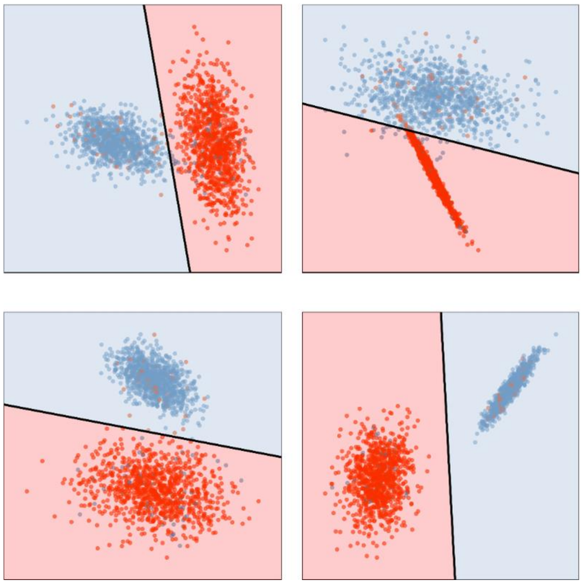

Figure 3a. A sample of datasets generated with the parameter combination $\alpha = 2,\ \beta = 0.1$. The shaded regions indicate clean model classifications.

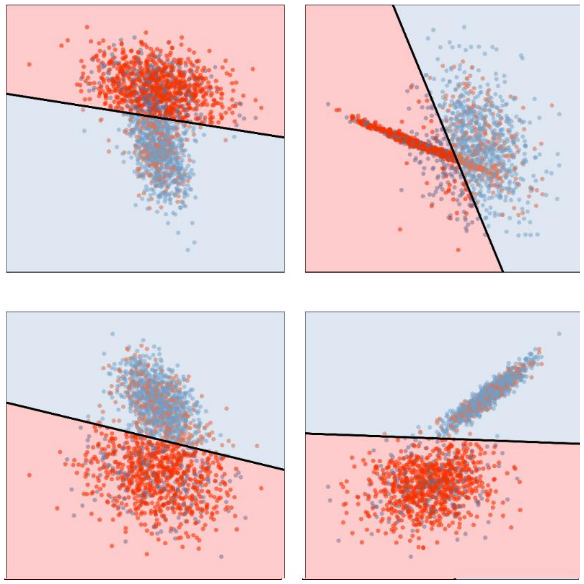

Figure 3b. A sample of datasets generated with the parameter combination $\alpha = 1,\ \beta = 0.5$.

POISONING ATTACKS

We use the *model-targeted* poisoning attack design from Suya et al. (Suya et al., 2021), which is shown to have state-of-the-art performance on subpopulation attacks. In a model-targeted poisoning attack, the attacker's objective is captured in a target model and the attack goal is to induce a model as close as possible to that target model. By describing the attacker’s objective using a target model, complex objectives can be captured in a unified way, thereby avoiding the need for designing customized attacks for individual objectives. Since the attack is online, the poisoning points can be added into the training set until the attack objective is satisfied, giving a natural measure of the attack's difficulty in terms of the number of poisoning points added during the attack. More details about the poisoning attack framework can be found in Appendix A.

Our choice of attack algorithm requires a target model as input. For our attacks, the required target model is generated using the label-flipping attack variant described in Koh et al. (Koh et al., 2021), which is also used by Suya et al. in their experiments. For each subpopulation, we use the label-flipping attack to generate a collection of candidate classifiers (using different fraction and repetition number) which each achieve 0% accuracy (100% error rate) on the target subpopulation. Afterwards, the classifier with the lowest loss on the clean training set is chosen to be the target model, as done in Suya et al.

For our case, the attack terminates when the induced model misclassifies at least 50% of the target subpopulation, measured on the test set. This threshold was chosen to mitigate the impact of outliers in the subpopulations. In earlier experiments requiring 100% attack success, we observed that attack difficulty was often determined by outliers in the subpopulation. By relaxing the attack success requirement, we are able to capture the more essential properties of an attack against each subpopulation. Since our eventual goal is to characterize attack difficulty in terms of the properties of the targeted subpopulation (which outliers do not necessarily satisfy), this is a reasonable relaxation.

Since we want to describe attack difficulties by the (essential) number of poisoning points needed (i.e., the number of poisoning points of the optimal attacks), more efficient attacks serve as better proxy to the (unknown) optimal poisoning attacks. To achieve the reported state-of-the-art performance using the chosen model-targeted attack, it is important to choose a suitable target model, and we ensure this by selecting the target models using the selection criteria mentioned above. These selection criteria are also justified in the analysis of the theoretical properties of the attack framework. Further details of the attack framework's theoretical properties are included in Appendix B.

SYNTHETIC DATASETS

We use cluster subpopulations for our experiments on synthetic datasets. To generate the subpopulations for each synthetic dataset, we run the k-means clustering algorithm ($k = 16$) and extract the negative-label instances from each cluster to form the subpopulations. Then, target models are generated according to the above specifications, and each subpopulation is attacked using the online (model-targeted) attack algorithm.

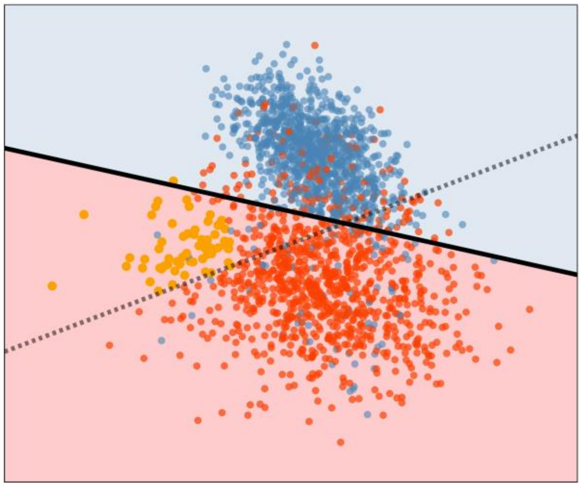

Figure 4a. Before poisoning, the clean model (solid black) correctly classifies the target subpopulation (orange). A target model (dashed gray) is selected which misclassifies the target subpopulation.

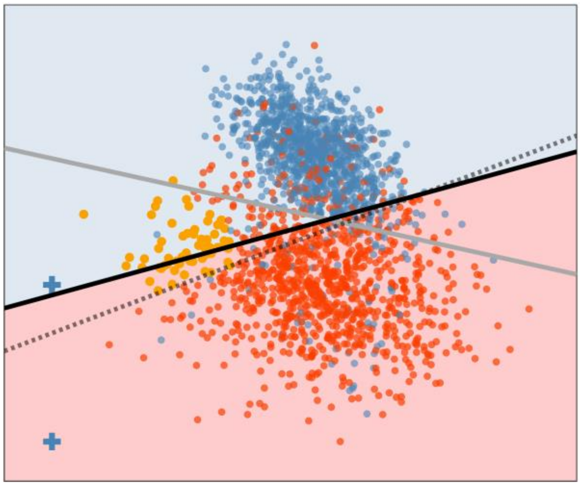

Figure 4b. After sufficiently many poisoning points (shown as crosses colored according to their label) are added into the dataset, the poisoned model (solid black) deviates from its original position (solid gray) and approaches the target model (dashed gray). Multiplicity of duplicated poisoning points is not depicted: in reality 87 non-unique poisoning points were added before attack success.

Figure 4 shows how the attack adds poisoning points into the training set to move the induced model towards the target model. In Figure 4a, a clean dataset is shown which correctly classifies the target subpopulation. In Figure 4b, poisoning points have been added into the training set to induce misclassification of the target subpopulation. Take note of how the poisoning points with positive labels work to reorient the model's decision boundary to cover the subpopulation while minimizing the impact on the rest of the dataset. This behavior also echoes with the target model selection criteria mentioned above, which aims to minimize the loss on the clean training set while satisfying the attacker goal. Another possible way to generate the target model is to push the entire clean decision boundary downwards without reorienting it—that is, the target model differs from the clean model only in the bias term—but this will result in a model with higher loss on other parts of the dataset, and thus a higher (overall) loss difference to the clean model. Intuitively, such an alternative would experience more "resistance" from the dataset, preventing the induced model from moving as swiftly to the target. This distinction is demonstrated in Figure 5.

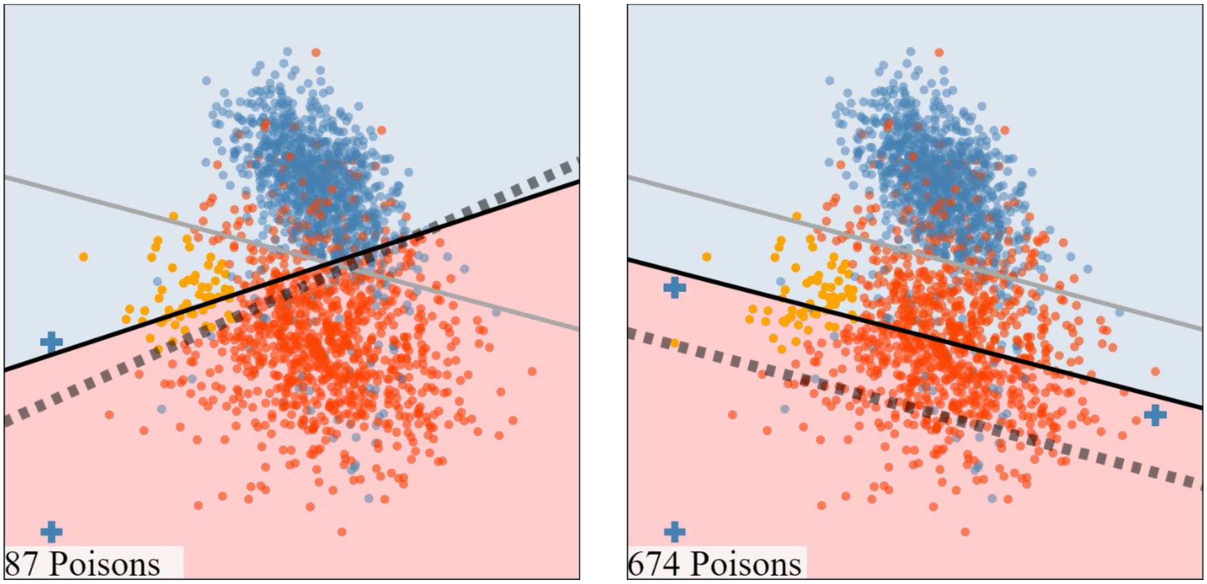

Figure 5a. Selecting a target model according to our selection criteria results in a more efficient attack which uses fewer poisoning points.

Figure 5b. Using a target model which attains higher loss on other parts of the dataset (*collateral damage*) results in a less efficient attack which wastes points on unimportant model behavior.

The comparison in Figure 5 demonstrates the importance of choosing a good target model and illustrates the benefits of the target model selection criteria. By using a target model that minimizes the loss on the clean dataset, the attack algorithm is more likely to add poisoning points that directly contribute to the underlying attack objective. On the other hand, using a poor target model causes the attack algorithm to choose poisoning points that preserve unimportant target model behaviors on other parts of the dataset.

VISUALIZATIONS OF DIFFERENT ATTACK DIFFICULTIES

Now that we have described the basic setup for performing subpopulation poisoning attacks using the online attack algorithm, we can start to study specific examples of poisoning attacks in order to understand what affects their difficulties.

EASY ATTACKS WITH HIGH LABEL NOISE

We first analyze a low-difficulty attack against a dataset with large amounts of label noise. The dataset in Figure 6 exhibits a large amount of label noise ($\beta = 1$), and as a result the clean model performs very poorly. When only a small number of poisoning points are added, the model changes its decision with respect to the target subpopulation.

In the example shown in Figure 6, there is a slight class imbalance due to dividing the dataset into train and test sets (963 positive points and 1037 negative points). As a result of this and the label noise, the clean model chooses to classify all points as the majority (negative) label.

One interesting observation in this example is, despite being easy to attack overall, attack difficulties of different subpopulations still vary somewhat consistently based on their relative locations to the rest of the dataset. Targeting other subpopulations within the same dataset reveals 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.

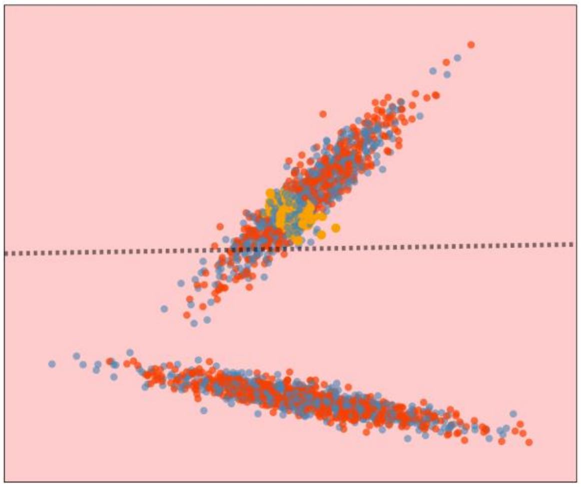

Figure 6a. A dataset with high label noise results in an inaccurate clean classifier. In this case, the classifier classifies the entire input space as the negative (red) label.

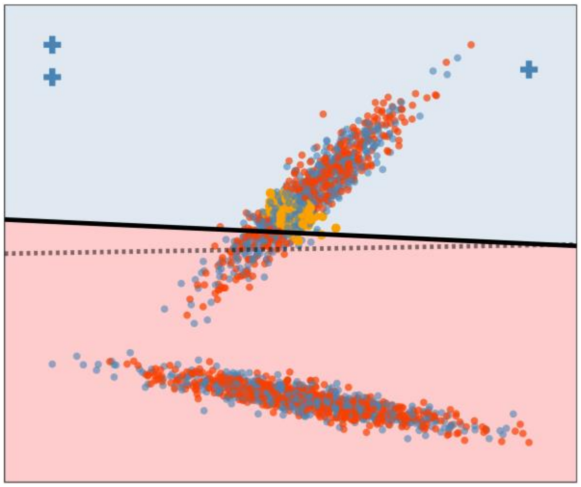

Figure 6b. An attack against a dataset with high label noise is easy, and one possible reason is the clean model may not be heavily influenced by points in the clean training set. Only 23 poisons were required to induce the depicted classifier.

EASY ATTACKS WITH SMALL CLASS SEPARATION

The next example, illustrated in Figure 7, shows an attack result similar to the noisy dataset discussed in the previous section, but now the reason of low attack difficulty is slightly different: although the clean training set now has small label noise, the model does not have enough capacity to generate a useful classification function. The end result is the same: poisoning points can strongly influence the behavior of the induced model. Note that in both of the examples in Figures 6 and 7, the two classes are (almost) balanced; if the labels of the two classes are represented in a different ratio, the clean model may prefer to classify all points as the dominant label, and the attack difficulty of misclassifying target subpopulation into the minority label may also significantly increase.

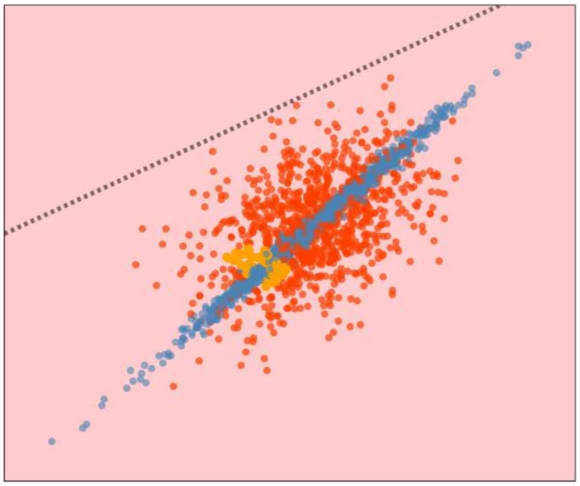

Figure 7a. The linear SVM model does not have sufficient capacity to perform well on a dataset with close class centers.

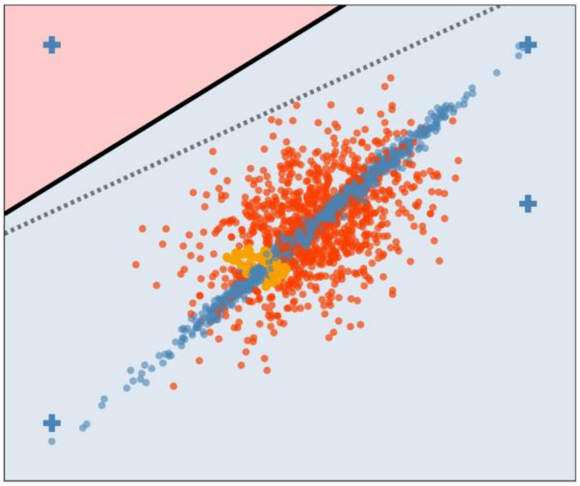

Figure 7b. An attack against a dataset with close class centers is easy, since the clean model exhibits poor performance. The depicted attack used only 39 poisons.

As mentioned earlier, the properties that apply to the two datasets in Figures 6 and 7 should also apply to all other datasets with either close class centers or high label noise. We can demonstrate this empirically by looking at the mean attack difficulty over the entire grid of the dataset parameters, where we describe the difficulty of an attack against a dataset of n training samples that uses p poisoning points using the ratio p/n. That is, the difficulty of an attack is the fraction of poisoning points added relative to the size of the original clean training set to achieve the attacker goal. This relationship is illustrated in Figure 8.

It appears that the attacks against datasets with poorly performing clean models tend to be easier than others, and in general attack difficulty increases as the clean model accuracy increases. The behavior is most consistent with clean models of low accuracies, where all attacks are easy and require only a few poisoning points. As the clean model accuracy increases, the behavior becomes more complex and attack difficulty begins to depend more on the properties of the target subpopulation.

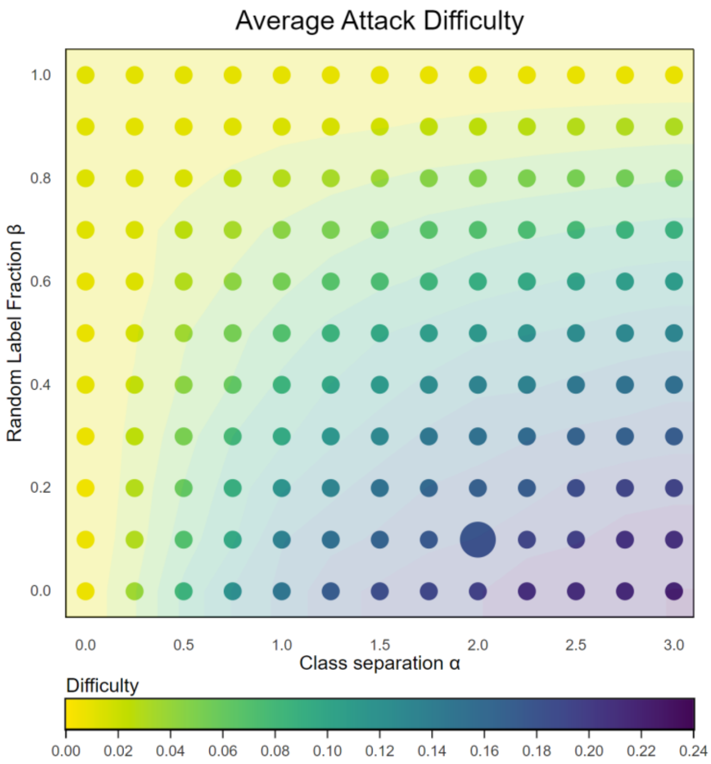

Figure 8. Dataset parameters vs. average attack difficulty. Average attack difficulty is roughly characterized by clean model accuracy.

ATTACKS ON DATASETS WITH ACCURATE CLEAN MODELS

Next, we look at specific examples of datasets with more interesting attack behavior. In the example in Figure 9, the dataset admits an accurate clean model which confidently separates the two classes.

The first attack is easy for the attacker despite being against a dataset with an accurate clean model. The targeted subpopulation lies on the boundary of the cluster it belongs to, and more specifically the loss difference between the target and clean models is small. In other words, the attack causes very little "collateral damage."

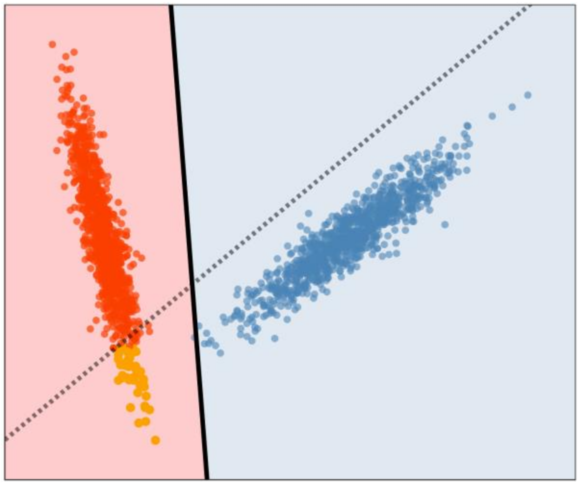

Figure 9a. A linearly separable dataset produces more interesting attack scenarios because many classifiers have large loss difference to the clean model.

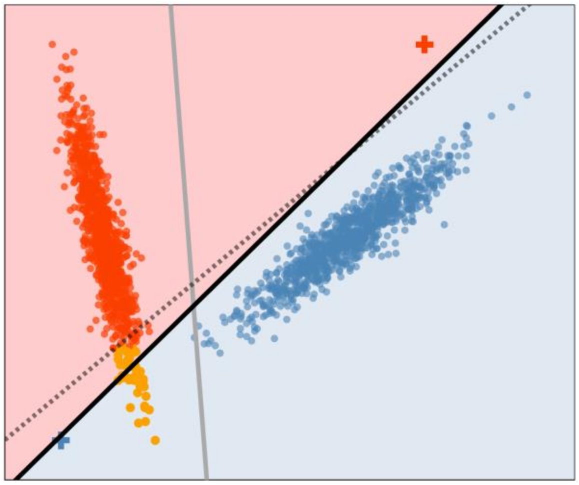

Figure 9b. An attack against the edge of a cluster is easy, especially if it causes little collateral damage. As measured by loss difference, the target model is very close to the clean model. The depicted attack required 60 poisons.

Now, consider the subpopulation depicted in Figure 10, in the middle of the cluster, but within the same dataset as in Figure 9. This is our first example of an exceptionally difficult attack, requiring over 800 poisoning points. The loss difference between the target and clean models is high since the target model misclassifies a large number of points in the negative class. Notice that it seems hard even to produce a target model against this subpopulation which does not cause a large amount of collateral damage. In some sense, the subpopulation is well protected and is more robust against poisoning attacks due to its central location.

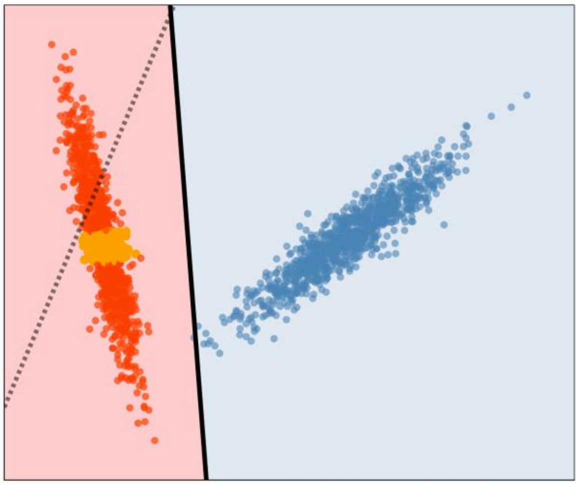

Figure 10a. Selecting a different target subpopulation results in a different target classifier, and thus an attack with a different difficulty.

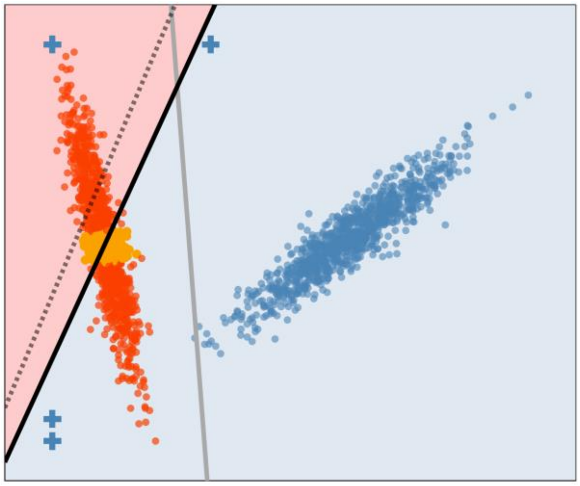

Figure 10b. An attack against the center of a cluster is hard, since the subpopulation is "protected" by the surrounding data points. The depicted attack required 850 poisons.

Quantitative Analysis

Visualizations are helpful, but cannot be directly created in the case of high-dimensional datasets. Now that we have visualized some of the attacks in the lower-dimensional setting, can we quantify the properties that describe the difficulty of poisoning attacks?

In the previous section, we made the observation that attack difficulty tends to increase with clean model accuracy. To be a little more precise, we can see how the attack difficulty distribution changes as a function of clean model accuracy, for our attack setup. This relationship is depicted in Figures 11 and 12.

Figures 11 and 12 empirically verify some of our observations from the earlier sections: attacks against datasets with inaccurate clean models tend to be easy, while attacks against datasets with accurate clean models can produce a wider range of attack difficulties.

Of course, on top of the general properties of the dataset and the clean model, we are more interested in knowing the properties of the subpopulation that affect attack difficulty. One way to do this is to gather a numerical description of the targeted subpopulation, and use that data to predict attack difficulty. Figure 13 shows the correlation between attack difficulty and model loss difference. As expected, model loss difference strongly correlates with attack difficulty. Other numerical factors describing subpopulations, like subpopulation size, do not have clear correlations with the attack difficulty. It is possible that complex interactions among different subpopulation properties might correlate strongly with the attack difficulty, and are worth future investigations.

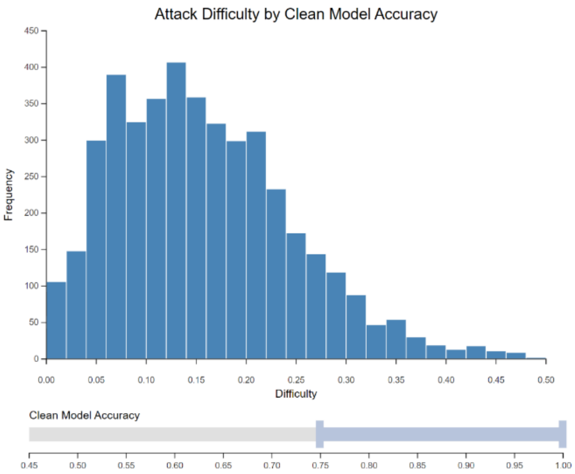

Figure 11. Datasets with accurate clean models tend to produce attacks with a wider range of difficulties. The histogram shows the attack difficulty distribution of all attacks against datasets with at least 75% clean accuracy.

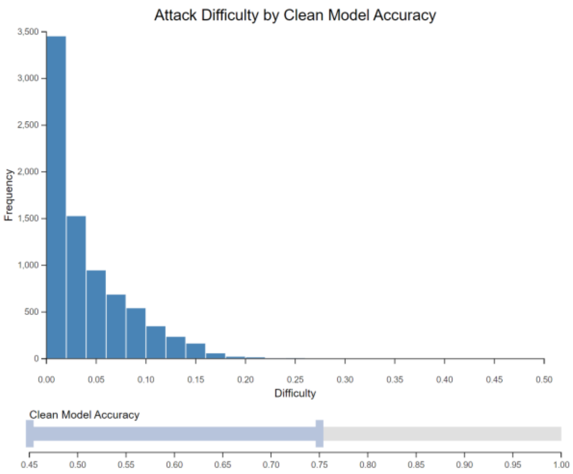

Figure 12. Datasets with inaccurate clean models tend to mostly produce easy attacks. The histogram shows the attack difficulty distribution of all attacks against datasets with at most 75% clean accuracy.

ADULT DATASET

While the visualizations made possible by the synthetic datasets are useful for developing an intuition for subpopulation poisoning attacks, we want to understand how subpopulation susceptibility arises in a more complex and practical setting. For this purpose, we perform subpopulation poisoning attacks against the UCI Adult dataset, based on US Census data, whose associated classification task is to determine whether an adult's income exceeds $50,000 per year based on attributes such as education, age, race, and marital status.

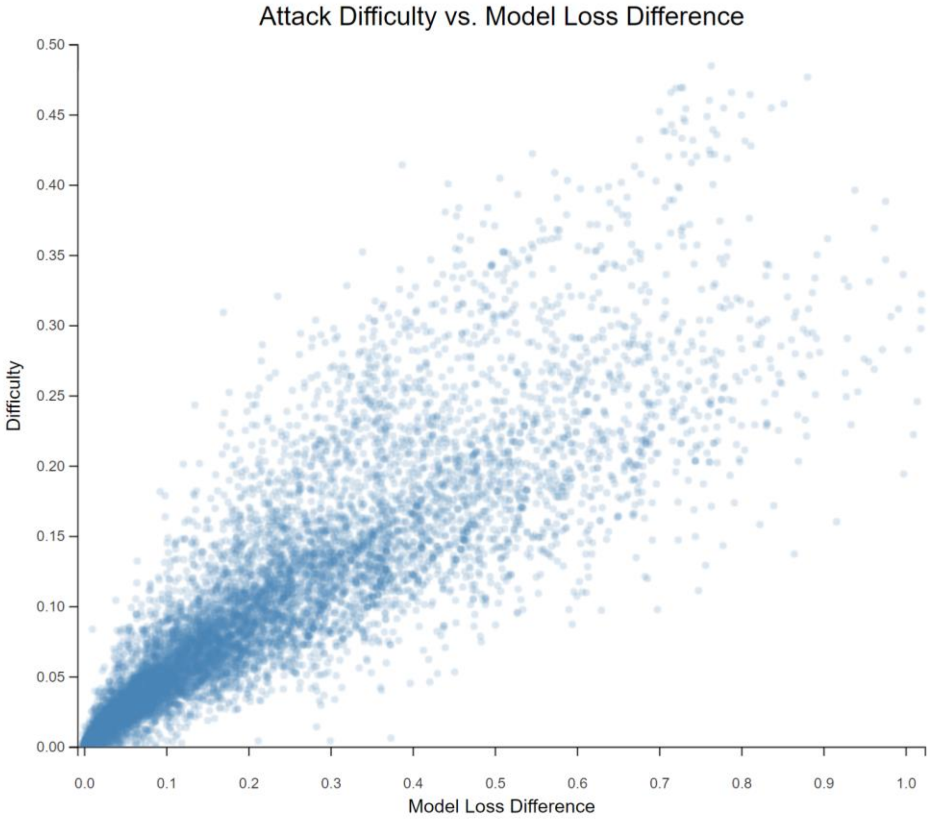

Figure 13. Loss difference between the clean and target models vs. attack difficulty on synthetic datasets. The correlation between loss difference and attack difference empirically demonstrates the target model attack criterion and provides a proxy for attack difficulty as a property of the targeted subpopulation.

SUBPOPULATION FORMATION

Subpopulations for the Adult dataset are selected based on combinations of attributes. To generate a semantic subpopulation, first a subset of categorical features is selected, and specific values are chosen for those features. For example, the categorical features could be chosen to be "work class" and "education level", and the features' values could then be chosen to be "never worked" and "some college", respectively. Then, every negative-label ("≤50K") instance in the training set matching all of the (feature, label) pairs is extracted to form the subpopulation. The subpopulations for our experiments are chosen by considering every subset of categorical features and every combination of those features that is present in the training set. For simplicity, we only consider subpopulations with a maximum of three feature selections.

In total, 4,338 subpopulations are formed using this method. Each of these subpopulations is attacked using the same attack as in the case of synthetic dataset. Of these attacks, 1,602 were trivial (i.e., the clean model already satisfies the attack objective), leaving 2,736 nontrivial attacks.

VISUALIZATION

The Adult dataset is high-dimensional (57 dimensions after data transformations), so attacks against it cannot be directly visualized as in the case of our two-dimensional synthetic datasets. However, by employing dimensionality reduction techniques, we can indirectly visualize poisoning attacks in the high-dimensional setting.

Dimensionality reduction gives us a way to visualize the high-dimensional data in two dimensions, but we still need a way to visualize a classifier's behavior on these points. Previous attempts to visualize high-dimensional classifiers have used self-organizing maps (Hamel, 2006), projection-based tour methods (Caragea et al., 2001), hyperplane intersection techniques (Poulet, 2008), or Voronoi tessellations constructed from projected data points (Migut et al., 2013). We will focus on another type of technique explored separately by Shulz et al. and Rodrigues et al., which attempts to produce a rich depiction of the classifier's behavior by examining the classifier's classification on data points sampled from the high-dimensional space (Rodrigues et al., 2019; Schulz et al., 2014).

The key idea is to find additional points in the high-dimensional space corresponding to points in the lower-dimensional projection space. The visualization process can be described by the following procedure:

1. Given a high-dimensional data space $\mathcal{X}$, a low-dimensional projection space $\mathcal{Z}$, and data points x_i from $\mathcal{X}$, train an embedding map $\pi : \mathcal{X} \to \mathcal{Z}$ to obtain the projected points $z_i = \pi(x_i)$.
2. Sample the projection space $\mathcal{Z}$ to obtain the image sample points z_i'. Determine points x_i' in the data space $\mathcal{X}$ which project to the image sample points $\pi(x_i') \approx z_i$ via some learned inverse mapping $\pi^{-1} : \mathcal{Z} \to \mathcal{X}$.
3. Use the classifier f defined on the data space to determine labels $f(x_i')$ for each of the additional data points x_i'. Visualize the data points and classifier behavior by plotting the projected points z_i on a scatterplot and coloring the background according to the sampled pairs $(z_i', f(x_i'))$.

We apply this technique to our attacks against the Adult dataset, using Isomap embedding as our dimensionality reduction algorithm. The resulting visualization, shown in Figure 14, is a good way to measure the impact of poisoning points on the poisoned classifier. However, it can be difficult to see just how much the classifier's behavior is changing on the targeted subpopulation (never married). We can instead choose to perform the attack visualization technique on only the target subpopulation, in which case we get a much more focused understanding of the model's behavior. Figure 15 illustrates this idea using the same attack from Figure 14.

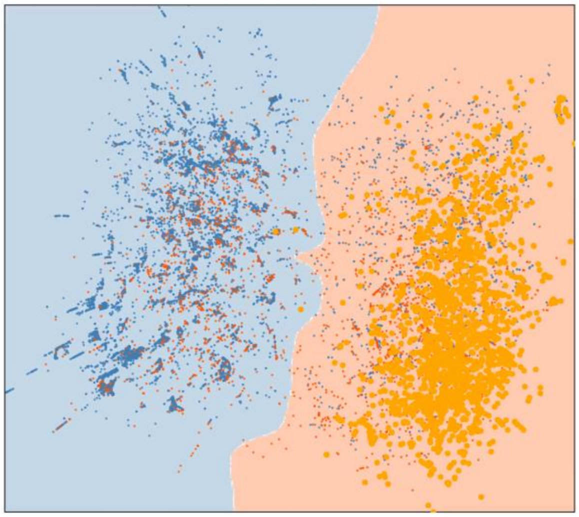

Figure 14a. By projecting points into a lower-dimensional space and shading regions according to newly sampled points, we obtain a visualization of high-dimensional model behavior.

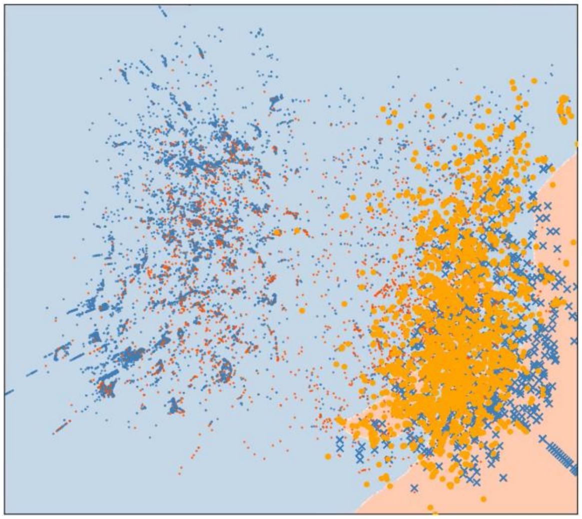

Figure 14b. After the poisons are added, the general behavior of the model (as illustrated in projection space) changes.

These visualizations provide a starting point to understanding subpopulation poisoning attacks in a high-dimensional setting. But, gaining intuitions about high-dimensionality data is difficult, and any mapping to a two-dimensional space involves compromises because of our inability to display high dimensional spaces.

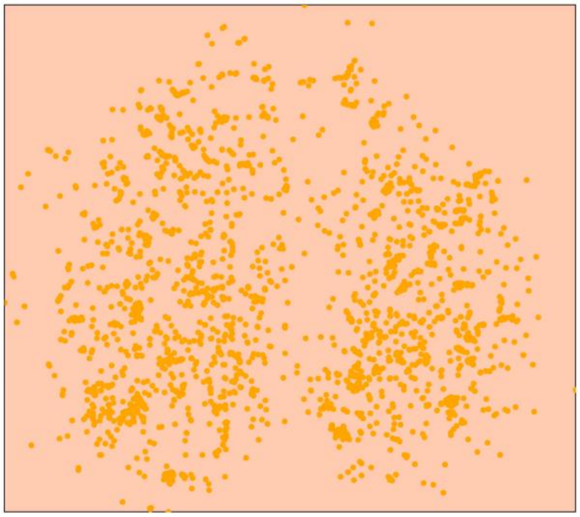

Figure 15a. Projecting only the targeted subpopulation gives a more useful visualization that emphasizes the attack objective.

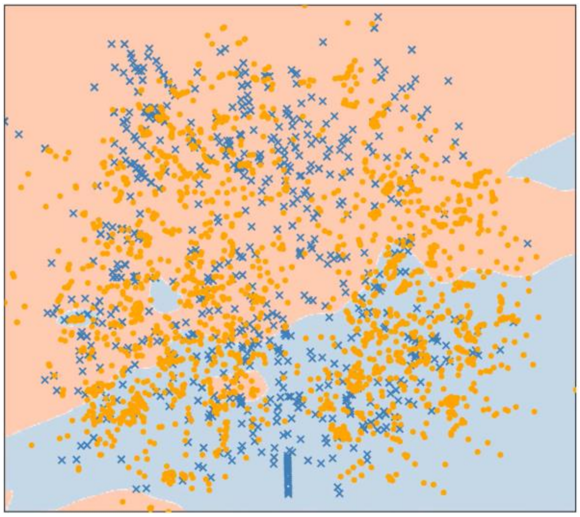

Figure 15b. After poisoning, different regions of the projected target subpopulation have been affected more than others.

QUANTITATIVE ANALYSIS

Figure 16 shows the attack difficulty (measured by the ratio p/n) distribution of all the nontrivial attacks. The range of difficulties as indicated by the histogram is significant: an attack with a difficulty of 0.1 uses 10 times as many points as an attack with a difficulty of 0.01. For a more concrete example, consider the subpopulation consisting of all people who have never married and the subpopulation consisting of divorced craft-repair workers (each subpopulation only taking negative-labeled instances). Both subpopulations are perfectly classified by the clean model, but the attack against the first subpopulation uses 1,490 poisoning points, while the attack against the second uses only 281.

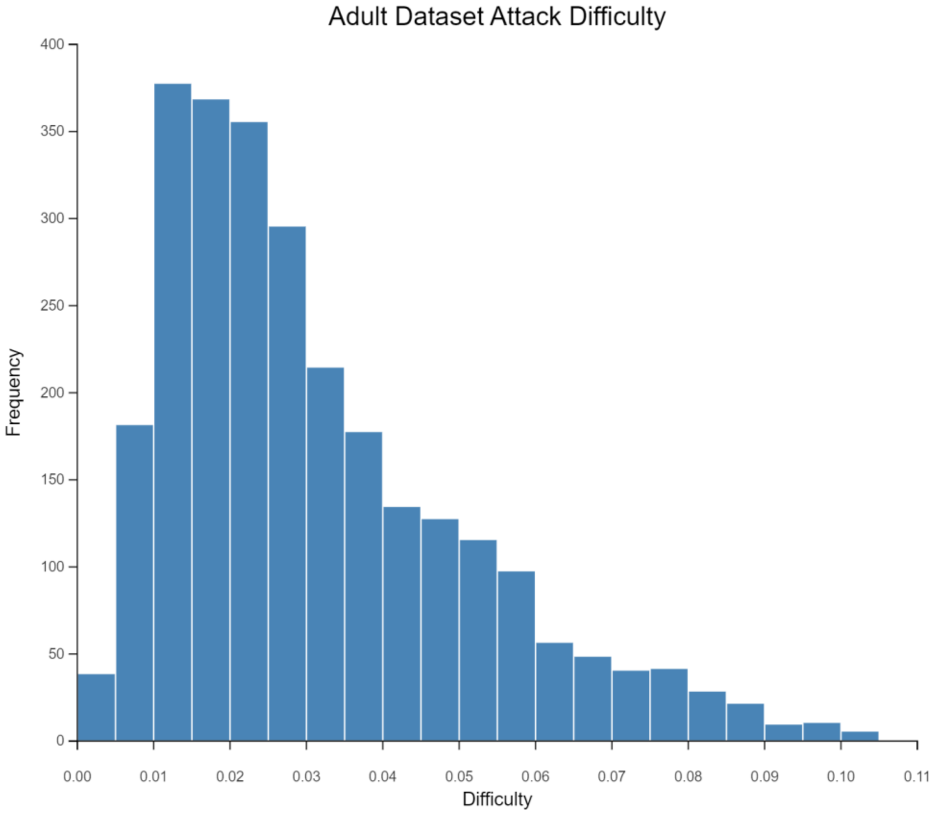

Figure 16. Attacks in a more practical setting still yield an interesting distribution of attack difficulties.

SEMANTIC SUBPOPULATION CHARACTERISTICS

Recall that our goal is to determine how subpopulation characteristics affect attack difficulty. While this question can be posed in terms of the statistical properties of the subpopulation (e.g., relative size to the whole data), it is also interesting to ask which semantic properties of the subpopulation contribute significantly to the attack difficulty. In this section, we explore the relationships between attack difficulty and the semantic properties of the targeted subpopulation.

Ambient Positivity

Figure 17 compares a few attacks from our experiments on the Adult dataset. The subpopulations are all classified with 100% accuracy by the clean model and are of similar sizes (ranges from 1% to 2% of the clean training set size). So, what makes some of the attacks more or less difficult? In the attacks shown in Figure 17, the differences may be related to the points surrounding the subpopulation. In our experiments, we defined the subpopulations by taking only negative-label instances satisfying some semantic properties. If we remove the label restriction, we gain a more complete view of the surrounding points, and, in particular, can consider some statistics (e.g., label ratio) of the ambient points.

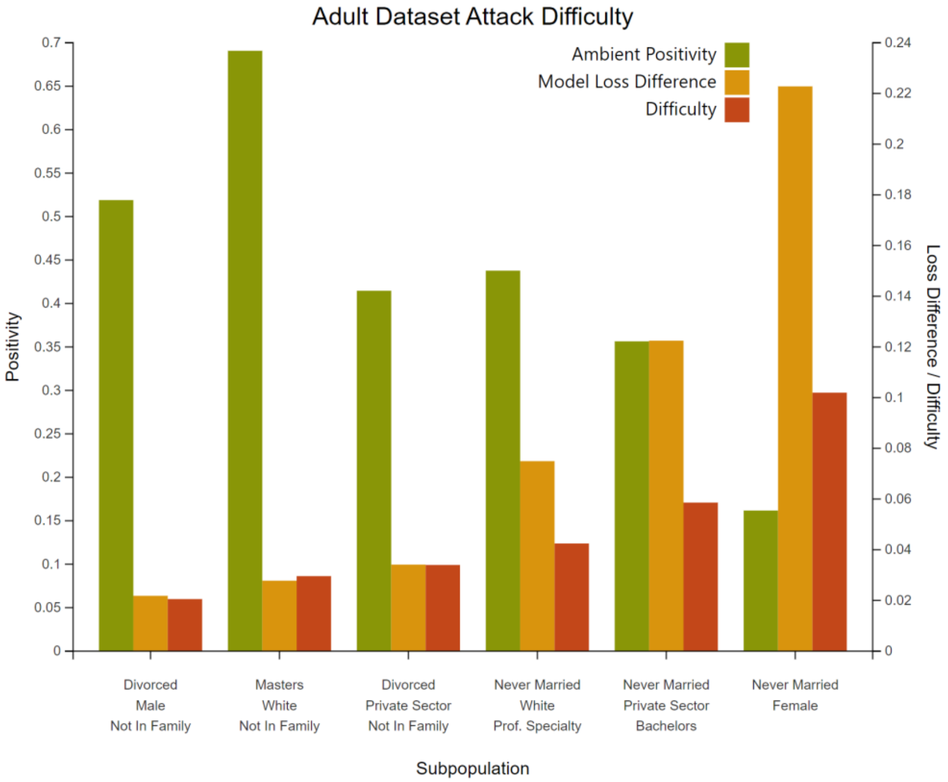

Figure 17. In some cases, ambient positivity is an indicator of the attack difficulty, especially if it relates to the loss difference (on clean training data) between the clean and the target models.

For a subpopulation with a given property P, let us call the set of all points satisfying P the *ambient subpopulation* (since it also includes positive-label points), and call the fraction of points in the ambient subpopulation with a positive label the *ambient positivity* of the subpopulation. In the attacks from Figure 17, attack difficulty is negatively correlated with the ambient positivity of the subpopulation. This makes sense, since positive-label points near the subpopulation work to the advantage of the attacker when attempting to induce misclassification of the negative-label points. Stated in terms of the model-targeted attack, if the clean model classifies the ambient subpopulation as the negative label, then the loss difference between the target and clean models is smaller if there are positive-label points in that region.

But does the ambient positivity of a subpopulation necessarily determine attack difficulty for otherwise similar subpopulations? If we restrict our view to subpopulations with similar pre-poisoning ambient positivity (e.g., between 0.2 and 0.3), we still find a significant spread of attack difficulties, shown in Figure 18. Once again, the subpopulations are all classified with 100% accuracy by the clean model and are of similar sizes. Are these differences in attack difficulties just outliers due to our particular experiment setup, or is there some essential difference between the subpopulations which is not captured by their numerical descriptions? Furthermore, if such differences do exist, can they be described using the natural semantic meaning of the subpopulations?

Pairwise Analysis

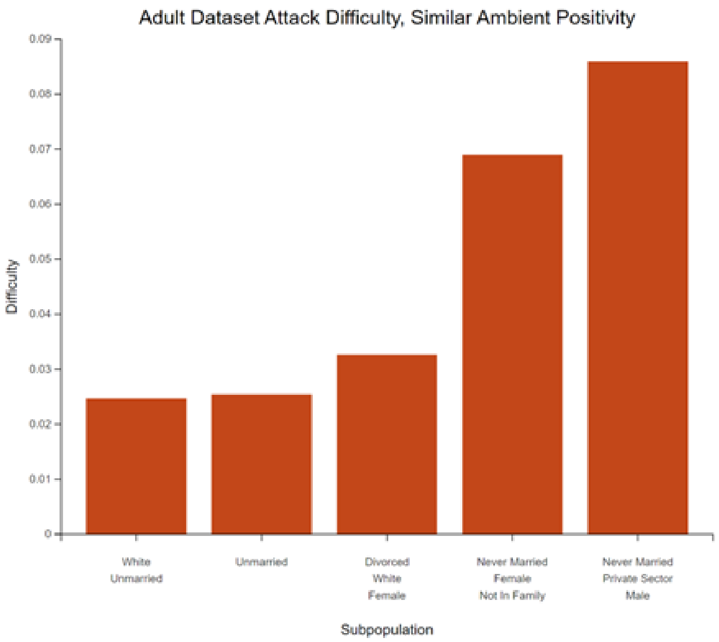

Figure 18. Even with similar size, ambient positivity, and clean model performance, subpopulations still experience significant differences in terms of attack difficulty.

What happens when two semantic subpopulations match on the same features, but differ in the value of only a single feature? For example, consider the two attacks in Figure 19. The subpopulations possess very similar semantic descriptions, only differing in the value for the "relationship status" feature. Furthermore, the subpopulations are similarly sized with respect to the dataset (1.3% and 1.1% of the clean training set, respectively), and each subpopulation is perfectly classified by the clean model. Yet the first subpopulation required significantly more poisoning points, at least for the chosen model-targeted attack.

ADULT DATASET SUMMARY

Our experiments demonstrate that poisoning attack difficulty can vary widely, even in the simple settings we consider. Although we cannot identify all the characteristics that relate to the attack difficulty, we can characterize some of them accurately for certain groups of subpopulations, giving the first attempt in understanding this complicated problem.

DISCUSSION

In this article, we visualize and evaluate the effectiveness of poisoning attacks, in both artificial and realistic settings. The difficulty of poisoning attacks on subpopulations varies widely, due to the properties of both the dataset and the subpopulation itself. These differences in the attack difficulty, as well as the factors that affect them, can have important consequences in understanding the realistic risks of poisoning attacks.

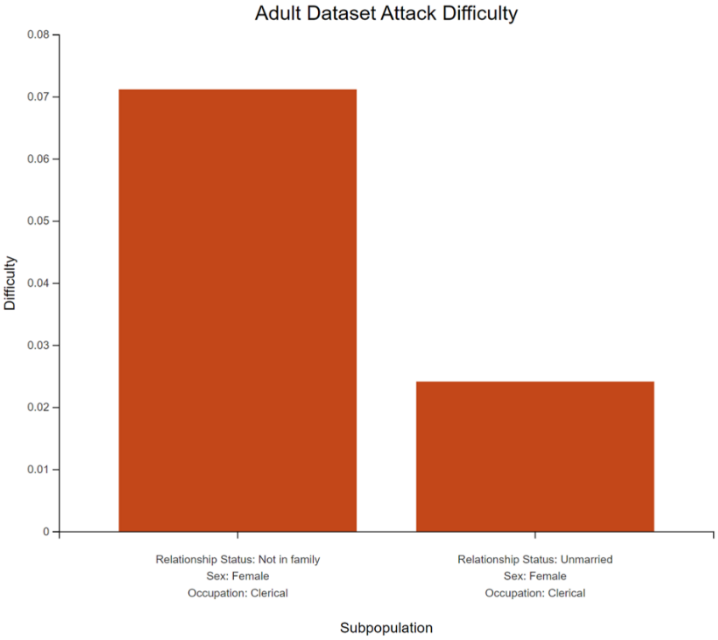

Figure 19. What can we learn about a feature’s impact on the attack difficulty by varying that feature’s value in the target subpopulation? The above subpopulations exhibit significantly different attack values, yet differ only slightly in their semantic descriptions.

Our results are limited to simple settings and a linear SVM model, and it is not yet clear how well they extend to more complex models. However, as a step towards better understanding of poisoning attacks and especially in understanding how attack difficulty varies with subpopulation characteristics, experiments in such a simplified setting are valuable and revealing. Further, simple and low-capacity models are still widely used in practice due to their ease of use, low computational cost and effectiveness (Dacrema et al., 2019; Tramèr & Boneh, 2021), and so our simplified analysis is still relevant in practice. Second, kernel methods are powerful tools to handle non-linearly separable datasets by projecting them into a linearly separable high-dimensional space and are widely adopted in practice. Therefore, if the important spatial relationships among the data points are still preserved after projection, then the same conclusions obtained in our simplified settings still apply to the more complex cases by examining the spatial relationships in the transformed space.

APPENDIX

A. ATTACK ALGORITHM

We use the online model-targeted attack algorithm developed by Suya et al. (Suya et al., 2021). Given the clean training set, target model, and model training parameters, the attack algorithm produces a set of poisoning points sequentially by maintaining an intermediate induced model and choosing the point that maximizes the loss difference between the intermediate model and the target model. The selected poisoning point is then added to the current training set and the intermediate model is also updated accordingly by the attacker using the knowledge of the model training process. Importantly, the online attack provides theoretical guarantees on the convergence of the induced model to the target model as the number of poisoning points increases. Since we have chosen our target model to misclassify the entire subpopulation, this means we have a guarantee that the online attack process will eventually produce a poisoned training set (which may be huge in size) whose induced model can satisfy the attacker objective.

B. ATTACK ALGORITHM THEORETICAL PROPERTIES

We consider a binary prediction task $h : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} = \{+1, -1\}$, and where the prediction model h is characterized by the model parameters $\theta \in \Theta \subseteq \mathbb{R}^d$. We denote the non-negative convex loss on a point $(x, y) \in \mathcal{X} \times \mathcal{Y}$ by $l(\theta; x, y)$, and define the loss over a set of points A as $L(\theta; A) = \sum_{(x,y)\in A} l(\theta; x, y)$. Denote by $\mathcal{D}_c$ the clean dataset sampled from some distribution over $\mathcal{X} \times \mathcal{Y}$.

A useful consequence of using the online attack algorithm is that the rate of convergence is characterized by the loss difference between the target and clean models on the clean dataset.

If we define the *loss-based distance* $D_{l,\mathcal{X},\mathcal{Y}} : \Theta \times \Theta \rightarrow \mathbb{R}$ between two models θ_1, θ_2 over a space $\mathcal{X} \times \mathcal{Y}$ with loss function $l(\theta; x, y)$ by

$$D_{l,\mathcal{X},\mathcal{Y}}(\theta_1, \theta_2) \coloneqq \max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} l(\theta_1; x, y) - l(\theta_2; x, y),$$

then the loss-based distance between the induced model θ_{atk} and the target model θ_p correlates to the loss difference $L(\theta_p; \mathcal{D}_c) - L(\theta_c; \mathcal{D}_c)$ between θ_p and the clean model θ_c on the clean dataset $\mathcal{D}_c$. This fact gives a general heuristic for predicting attack difficulty: the closer a target model is to the clean model as measured by loss difference on the clean dataset (under the same search space of poisoning points), the easier the attack will be. This also justifies the decision to choose a target model with lower loss on the clean dataset.

REFERENCES

Biggio, B., Nelson, B., & Laskov, P. (2013). *Poisoning Attacks against Support Vector Machines* (arXiv:1206.6389). arXiv. https://doi.org/10.48550/arXiv.1206.6389

Caragea, D., Cook, D., & Honavar, V. G. (2001). Gaining insights into support vector machine pattern classifiers using projection-based tour methods. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 251–256. https://doi.org/10.1145/502512.502547

Dacrema, M. F., Cremonesi, P., & Jannach, D. (2019). Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. *Proceedings of the 13th ACM Conference on Recommender Systems*, 101–109. https://doi.org/10.1145/3298689.3347058

Dua, D., & Graff, C. (2017). *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml

Geiping, J., Fowl, L., Huang, W. R., Czaja, W., Taylor, G., Moeller, M., & Goldstein, T. (2021). *Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching* (arXiv:2009.02276). arXiv. https://doi.org/10.48550/arXiv.2009.02276

Hamel, L. (2006). Visualization of Support Vector Machines with Unsupervised Learning. *2006 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology*, 1–8. https://doi.org/10.1109/CIBCB.2006.330984

Huang, W. R., Geiping, J., Fowl, L., Taylor, G., & Goldstein, T. (2021). *MetaPoison: Practical General-purpose Clean-label Data Poisoning* (arXiv:2004.00225). arXiv. https://doi.org/10.48550/arXiv.2004.00225

Jagielski, M., Severi, G., Harger, N. P., & Oprea, A. (2021). *Subpopulation Data Poisoning Attacks* (arXiv:2006.14026). arXiv. http://arxiv.org/abs/2006.14026

Koh, P. W., & Liang, P. (2020). *Understanding Black-box Predictions via Influence Functions* (arXiv:1703.04730). arXiv. https://doi.org/10.48550/arXiv.1703.04730

Koh, P. W., Steinhardt, J., & Liang, P. (2021). *Stronger Data Poisoning Attacks Break Data Sanitization Defenses* (arXiv:1811.00741). arXiv. https://doi.org/10.48550/arXiv.1811.00741

Mei, S., & Zhu, X. (2015). Using machine teaching to identify optimal training-set attacks on machine learners. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2871–2877.

Migut, M., Worring, M., & Veenman, C. (2013). Visualizing multi-dimensional decision boundaries in 2D. *Data Mining and Knowledge Discovery*, *29*. https://doi.org/10.1007/s10618-013-0342-x

Nelson, B., Barreno, M., Chi, F., Joseph, A., Rubinstein, B., Saini, U., Sutton, C., Tygar, J., & Xia, K. (2008, January 1). *Exploiting Machine Learning to Subvert Your Spam Filter.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830.

Poulet, F. (2008). Towards Effective Visual Data Mining with Cooperative Approaches. In S. J. Simoff, M. H. Böhlen, & A. Mazeika (Eds.), *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics* (pp. 389–406). Springer. https://doi.org/10.1007/978-3-540-71080-6_22

Rodrigues, F. C. M., Espadoto, M., Hirata, R., & Telea, A. C. (2019). Constructing and Visualizing High-Quality Classifier Decision Boundary Maps. *Information*, *10*(9), Article 9. https://doi.org/10.3390/info10090280

Schulz, A., Gisbrecht, A., & Hammer, B. (2014). Using Discriminative Dimensionality Reduction to Visualize Classifiers. *Neural Processing Letters*, *42*, 27–54. https://doi.org/10.1007/s11063-014-9394-1

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., & Goldstein, T. (2018). *Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks* (arXiv:1804.00792). arXiv. https://doi.org/10.48550/arXiv.1804.00792

Steinhardt, J., Koh, P. W., & Liang, P. (2017). *Certified Defenses for Data Poisoning Attacks* (arXiv:1706.03691). arXiv. https://doi.org/10.48550/arXiv.1706.03691

Suya, F., Mahloujifar, S., Suri, A., Evans, D., & Tian, Y. (2021). *Model-Targeted Poisoning Attacks with Provable Convergence* (arXiv:2006.16469). arXiv. http://arxiv.org/abs/2006.16469

Tramèr, F., & Boneh, D. (2021). *Differentially Private Learning Needs Better Features (or Much More Data)* (arXiv:2011.11660). arXiv. https://doi.org/10.48550/arXiv.2011.11660

Zhu, C., Huang, W. R., Shafahi, A., Li, H., Taylor, G., Studer, C., & Goldstein, T. (2019). *Transferable Clean-Label Poisoning Attacks on Deep Neural Nets* (arXiv:1905.05897). arXiv. https://doi.org/10.48550/arXiv.1905.05897