Benchmarking the Effect of Poisoning Defenses on the Security and Bias of the Final Model

Anonymous Author(s)
Affiliation
Address
email

Abstract

Machine learning models are susceptible to a class of attacks known as adversarial poisoning where an adversary can maliciously manipulate training data to hinder model performance or, more concerningly, insert backdoors to exploit at inference time. Many methods have been proposed to defend against adversarial poisoning by either identifying the poisoned samples to facilitate removal or developing poison agnostic training algorithms. Although effective, these proposed approaches can have unintended consequences on other aspects of model performance, such as worsening performance on certain data sub-populations, thus inducing a classification bias. In this work, we evaluate several adversarial poisoning defenses. In addition to traditional security metrics, i.e., robustness to poisoned samples, we propose a new metric to measure the potential undesirable discrimination of sub-populations resulting from using these defenses. Our investigation highlights that many of the evaluated defenses trade decision fairness to achieve higher adversarial poisoning robustness. Given these results, we recommend our proposed metric to be part of standard evaluations of machine learning defenses.

1 Introduction

Machine learning (ML) is used in numerous critical applications including healthcare, finance, and the Internet-of-Things. However, the sensitivity of these applications also motivates a need to develop secureML algorithms to avoid safety and security incidents. In particular, *adversarial poisoning attacks* on machine learning has received significant attention [18]. In a poisoning attack, an attacker modifies a portion of the training data to influence and/or degrade the performance of the trained model. Often, the goal is to encode a backdoor in a few training samples, which the attacker can later trigger at inference time. Despite the presence of poisoned training samples, the overall performance on benign (i.e., non-trigger) inputs is often satisfactory, thus avoiding suspicion the model has been poisoned. However, when a sample appears containing a backdoor trigger (e.g., image patch), the desired erroneous behavior occurs (e.g., a targeted misclassification of a malicious input).

To mitigate the effect of adversarial poisoning, a variety of defense techniques have been proposed: flagging the data that is “suspicious" [3, 20], segmenting training data using model ensembles [15, 23], abstaining from predicting [22], or by manipulating the training data before feeding it to the training process to neutralize potential malicious modifications [4, 9, 25, 26]. An ideal defense would fully mitigate or remove the effect of poisoning *without any undesirable side-effects*.

The side-effects of adversarial poisoning defenses have not been studied by prior works. Prior work primarily focused on reducing the poisoning attack success rate, while maintaining accuracy on benign samples. Although both of these metrics are useful in evaluating the quality of a prospective defense, they do not provide a complete picture of the final “defended” model’s behavior. For example, we discovered most filtering defenses remove a consistent fraction of the training data regardless of

Submitted to 2022 Trustworthy and Socially Responsible Machine Learning (TSRML 2022). Do not distribute.

whether the removed samples are poisonous or not. In general, as poisoned samples are a small subset of the training data containing a unique set of features, i.e. the trigger, we hypothesize that prior defenses appeared effective as they (possibly unintentionally) mitigated the influence outlier features. However, outliers are frequently under-represented *benign* sub-populations that occur naturally in the data and can be harmed by outlier mitigation techniques, which may result in potential decision bias against benign sample. Typically, a sub-population within a class may include samples collected under specific conditions (e.g., a winter storm or lightening), a particularly under-represented sample (e.g., very uncommon type of airplane), or an under-represented minority (e.g., women). An example of benign sub-populations within a class is shown in Appendix A.1.

In this paper, we study the side-effects of existing poisoning defenses and show many defenses are non-ideal as they negatively affect the decision fairness of the defended model on under-represented benign sub-populations. We benchmark existing defenses against dirty and clean label poisoning attacks using traditional security metrics and a new metric that measures the side-effects of prospective defenses on the model's decision. Borrowing from fairness literature, we use the *Statistical Parity Difference (SPD)* [5], to measure the effect a defense has on *minority* sub-populations in the data (e.g., images collected with poor visibility in a driving dataset). Our extensive experimental results shed new light on the effects that existing poisoning defenses have on decision fairness.

2 Background: Poisoning Attacks and Defenses

2.1 Poisoning Attacks

Poisoning attacks manipulate a small percentage of the data used during training to achieve some adversarial goal, often backdoor injection. They can be broadly classified into *dirty label* and *clean label* attacks. In dirty label attacks [10, 16], the adversary's goal is to induce a misclassification into a target class through use of a backdoor trigger. The adversary generates a trigger (e.g. image patch) and poisons a percentage of the training data by adding the trigger, $X_p \subseteq X$, as well as modifying the labels of those samples, $Y_p \subset Y$, to the target class. At inference time, the adversary adds the trigger to induce the backdoor behavior, targeted misclassification. When the backdoor is not present, the model behaves normally.

In contrast to dirty label attacks, clean label attacks **do not modify** the labels and rely on more inconspicuous modifications. As poisoned inputs appear consistent with their labels, human inspection is unlikely to detect the attack. A large variety of clean label attacks have been proposed in literature [19]. Two distinct attacks are the original *Clean Label Backdoor Attack (CLBD)* [21] and *Witches' Brew* (also known as *Gradient Matching*) [7]. These attacks often solve optimization problems to generate poison images close to the original without changing the labels which inject the backdoor to the model if used for training. For example, a crafted cat image still looks like a cat to a human, but the model sees the image as if it is a truck in terms of features or training gradients. Turner et al. leverage a GAN and adversarial example approach to generate poison images while ensuring the perturbation is bounded. In Witches' Brew, the attacker chooses one or few images in the test data as *trigger images* and aims to make the model classify them as the target class(es). This attack applies bounded perturbations to the poison data by aligning the training gradient of the poison data with the correct labels, and that of the trigger samples with the target labels, using a surrogate model.

2.2 Poisoning Defenses

To mitigate poisoning attacks, multiple defenses have been proposed recently. We provide a high-level overview of the specific defenses that we evaluate in Section 4.

Activation Defense [3] is a filtering-based defense, which analyzes the training set and filters out samples that are deemed "too different" with respect to the rest of the data. After training, the training data is passed again through the model and the last layer activations are recorded and clustered. Samples associated with either *small* clusters or isolated clusters are removed from the training set. In our evaluation, potentially poisonous data was marked based on the smallest activation cluster(s).

Spectral Signatures [20] is a filtering-based defense where the activations of the network for each training sample are analyzed using singular value decomposition (SVD). Samples with unusual SVD are removed from the training set. This defense has an additional hyperparameter to define how much

poison is expected to be in the training data. In our evaluation, we use the best case scenario where the expected poisoning hyperparameter exactly matches the true poisoning percentage.

Deep Partition Aggregation (DPA) [15] is an ensemble-based defense that creates multiple weak classifiers and performs inference by voting. The training samples are split in k disjoint partitions $P_1, ..., P_k$ and each partition P_i is used to train a different model to create an ensemble. During inference, the models are ensembled and a prediction is made based on a majority vote.

Finite Aggregation [23] is an extension of DPA and includes two hyper-parameters to guide the ensembling process, k and d. They are defined as the inverse sensitivity and the spreading factor, respectively. First, the defense partitions the training dataset into kd disjoint partitions. Then, each data partition is assigned to d of the kd submodels in the partition. The kd submodels models are trained on their assigned partitions. During inference, a prediction is made based on a majority vote.

Inverse Self-Paced Learning (ISPL) [13] is a filtering defense that relies on identifying "compatible or homogeneous sets" in the training data. The defense defines the notion of "self-expanding sets" and propose an iterative approach, which results in groups of homogeneous sets, i.e. all of the samples in the set belong to either the primary or noisy distribution. In their scenario, the noisy distribution, which is assumed to be the minority, contains the poisoned data. One the data has been segmented, they train a model on each partition and use each model to classify data from all of the other partitions. Using a majority voting scheme based on the misclassification rate on the other partitions, the primary and poison distributions can be identified.

Adversarial Training [17] was originally used as a defense against evasion attacks, but has been examined in some works as a poisoning defense (e.g. Geiping et al. [8] used it as a baseline defense). In this defense, adversarial inputs are generated on-the-fly using a known evasioning attack so as to imrpove the model's generalization performance on the adversarial distribution. In our evaluation, we use the Projected Gradient Descent (PGD) attack, as is traditional, with the same hyperparemeters used by Geiping et al. [8] for consistency.

Data Augmentation techniques such as Mixup [26], CutOut [4], and CutMix [25] use data augmentation to improve the model's generalization. Maxup [9] applies a set of these data augmentation techniques multiple times and selects the worst-case input for training to further improve generalization. As existing poisoning attacks relied on precise and sometimes large input manipulations, random data augmentation can introduce variability that the attacks are not prepared to address. Borgnia et al. [2] propose this approach, using Maxup with CutOut, as a poisoning defense.

3 Metrics to Evaluate Poisoning Defenses

Security and Accuracy Metrics: Traditionally, defenses are evaluated by measuring their performance in terms of the clean accuracy and final attack success rate. The clean accuracy (also known as benign accuracy) is the accuracy of the model evaluated on the test set with no poisoned samples. The attack success rate is the percentage of poisoned samples in the test set that were successfully misclassified. We use both of these metrics in our evaluations.

Model Quality Metric: In this paper, we introduce a new metric to help determine what effect various defenses have on different benign sub-populations in the dataset. Ideally, applying a defense should not result in a model that incorrectly predicts benign inputs coming from sub-populations. Samples in these sub-populations are typically uncommon or "difficult" to predict.

We utilize the statistical parity difference (SPD) [5] to determine how a particular defense treats benign samples from different sub-populations. Let us denote a feature set by X, the corresponding label set by Y and the cardinality by $|\cdot|$. Given a class $y \in Y$, two populations within this class $P_1 := (X_1, Y_1)$ and $P_2 := (X_2, Y_2)$ consisting of solely *benign data samples* and a model $\mathcal{M} : X \to Y$ trained with a defense, the corresponding SPD for this given class $y \in Y$ can be computed as follows:

$$SPD = \frac{|\{(x,y) \in P_1 : \mathcal{M}(x) = y\}|}{|P_1|} - \frac{|\{(x,y) \in P_2 : \mathcal{M}(x) = y\}|}{|P_2|} \tag{1}$$

We interpret the above metric based on the range of the value following standard conventions [1]:

- $|SPD| \leq 0.1$: An acceptable range, where none of the populations is disproportionately misclassified compared to the other. We refer to this as a fair outcome.

- $|SPD| > 0.1$: An unfair outcome where the evaluated model is biased towards P_1 (if $SPD > 0.1$) or P_2 (if $SPD < -0.1$).

Ideally, applying a defense should not disproportionately reduce the model performance for a particular benign population. Hence, an $|SPD| \leq 0.1$ is desirable. Clearly, applying a defense should not exacerbate the misclassification rate of these types of benign sub-populations.

4 Benchmarking of Poisoning Defenses

4.1 Experimental Setup

We evaluate the **seven** popular defenses presented in Section 2 under multiple poisoning attacks. We also use a modified combination defense where we apply both Data Augmentation and Adversarial Training (D.A. + A.T.), which is expected to mitigate the drop in clean accuracy when doing adversarial training alone. Specifically, we apply Cutmix to augment the data and perform adversarial training on a fraction of the data; for our experiments we choose 75% as the hyperparameter. We test against a dirty-label backdoor attack [10] (DLBD) and a clean-label backdoor attack [21] (CLBD). Under both attacks, only samples from one of the classes (the target class) are poisoned. To fully understand the effect that an adversary may have over the model, we vary the percentage of poison for each defense. For DLBD, we use 0%, 1%, 5%, 10%, 20%, and 30% poison. For CLBD, we use 0%, 20%, 50%, and 80% poison. Note that this poison is only applied to the target class. In addition, for both types of attacks, we used two different triggers: a *bullet hole* and a *peace sign*. For more information on the triggers used in our experiments, refer to Appendix A.3. We present the results where the defenses are evaluated using the MicronNet model [24] trained on the GTSRB dataset [12].

We only present the evaluations using the default hyperparameters for each defense in their respective paper as described in Section 2. For additional evaluations using different hyperparameters refer to Appendix A.4. To further test generalizability, we also evaluate these defenses using the CIFAR-10 dataset [14] against the dirty label backdoor [10] and Witches' Brew [7] attacks. Refer to Appendix A.5 for these results.

Baselines: We assess the performance of multiple popular defenses and compare them with a variety of baselines to understand their effect on the security and fairness metrics. As baselines, we use the following models.

1. *Undefended:* This defense mimics the scenario where a model is poisoned and no corrective measure is applied, creating a worst case scenario.
2. *Perfect filter:* This is an oracle filtering defense that fully removes the poisoned data from the training set. This is the best case scenario.
3. *Random filter:* This baseline randomly removes a percentage of the data, allowing us to determine how simply removing some data compares with popular defenses. We include this baseline given that our preliminary experimental results suggest that popular filtering defenses remove around 10% of the training data regardless of whether it is benign or poisonous.

Sub-population Generation: To determine different sub-populations within the benign training data, we make use of an explanatory model called BEAN regularization [6]. Then, given a test set, for each sample within a class, the explanatory model predicts whether the input sample belongs to a well-represented group (population 1) or out-of-distribution group (population 2). This labeling process is performed on each class in the dataset and fed to the SPD metric previously defined in (1). For more information on the BEAN model and why we choose this, refer to Appendix A.2.

4.2 Experimental Results

We now present the results for the **three** baselines and **eight** defenses we evaluated. Every baseline or defense, except Finite Aggregation, was evaluated for a total of three trials[1]. We report the average metrics among all trials.

[1]For Finite Aggregation, we only ran one trial as it requires training a large number of models (e.g., with the default parameters of $k = 50$, $d = 10$, 500 models MicronNet models need to be trained), requiring 3 days to run a single experiment using a 32GB V100 GPU under GTSRB.

Security Assessment: We first evaluate the clean accuracy and attack success rate for the defenses and baselines. In Figure 1, we report these metrics across the varying poisoning percentages for the two attacks using the bullet holes and peace sign triggers. We omit 0% poison for the attack success rate plots since there will never be a successful attack.

We first notice that there is no clear correlation between poison percentage and clean accuracy for either attack type or trigger. We do observe, however, that most of the defenses have lower clean accuracy than the baselines. Particularly, Adversarial Training and the D.A. + A.T. combo tend to have the lowest clean accuracy. This is consistent with the fact that training on adversarial examples usually hurts the model performance [17]. Data Augmentation is an exception where it often performs even better than the baselines. Most of the defenses have similar accuracy for both DLBD and CLBD; the exception is Activation Defense which performs worse against CLBD.

For the attack success rate, we observe that in DLBD, nearly most of the defenses are more robust than the Undefended and Random Filter baselines. This does not extend to CLBD as many of the defenses are actually less robust than the Undefended baseline. We also notice that the choice of trigger can affect some of the defenses such as ISPL for CLBD, with the peace sign being a more nefarious trigger. DPA and Finite Aggregation consistently offer very high levels of robustness for both attacks, and even outperform the Perfect Filter baseline during CLBD.

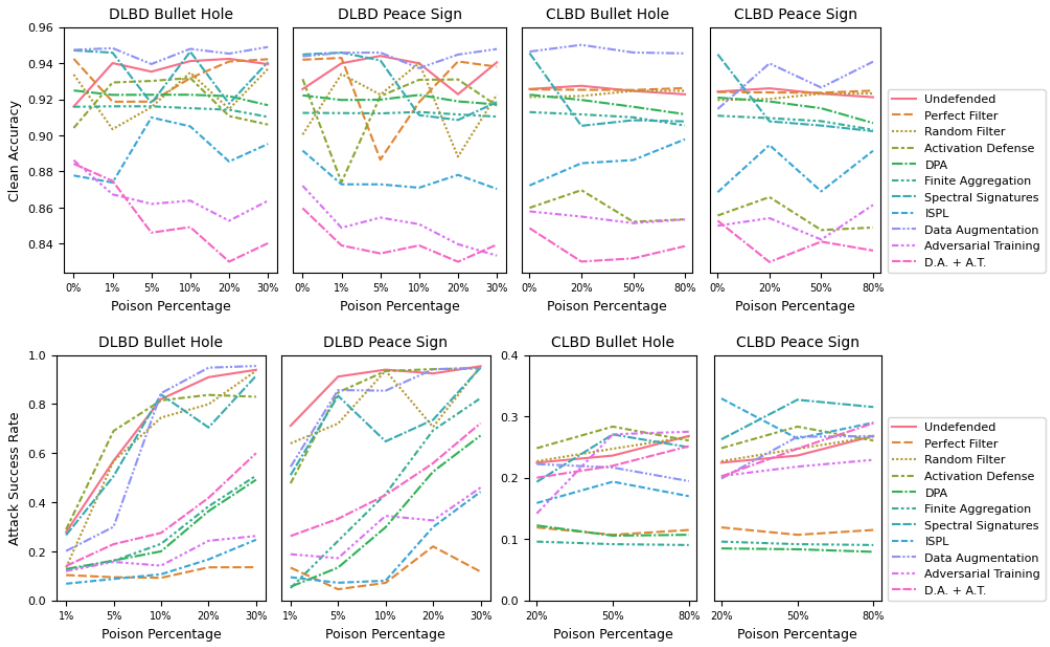

Figure 1: The clean accuracy (top) and attack success rate (bottom) for the baselines and defenses across varied poison. Each model is evaluated against the dirty label backdoor (DLBD) and clean label backdoor (CLBD) attacks using the bullet hole and peace sign triggers on the GTSRB dataset.

Fairness Assessment: We visualize the resulting SPD metric of the models using heatmaps by aggregating the fairness of classes across different models for scenarios with different poison percentages. To generate this heatmap, for each model and poison percentage, we first categorize the resulting SPD values associating with each class into SPD intervals described in Section 3. The classes belonging to $SPD < -0.1$ were considered as negative classes and $SPD > 0.1$, were considered as positive classes and $|SPD| \leq 0.1$ were considered as fair classes. The counts of classes belonging to each of these three categories were taken across different poison percentages and averaged for all trials. This was performed individually for all defenses or models under consideration and were plotted against each other. In Figure 2, we visualize the number of classes where $|SPD| \leq 0.1$ across the varying poisoning percentages and the percentage of classes for each of the three intervals averaged across all poison percentages for the two attacks using the bullet holes and peace sign triggers.

In the heatmaps for the class counts of the $|SPD| \leq 0.1$ (first row of Fig. 2), *darker blue* reflects that more classes fall into the acceptable range, and thus, the model is fairer over all. The different

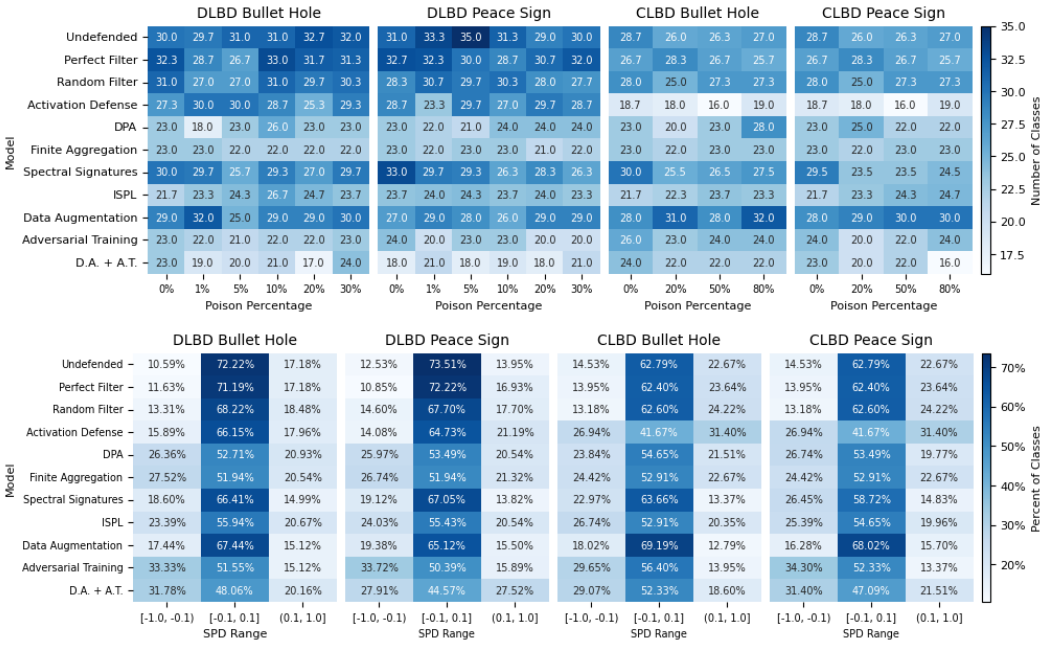

Figure 2: The number of classes within the acceptable $|SPD| \leq 0.1$ range across varied poison (top) and the percentage of classes within each of the three SPD ranges averaged across all poison percentages (bottom) for each baseline and defense. Each model is evaluated against the dirty label backdoor (DLBD) and clean label backdoor (CLBD) attacks using the bullet hole and peace sign triggers on the GTSRB dataset.

baselines are also shown for relative comparison purposes in each row. We first notice that the poison percentage does not have a large effect on the fairness as the class counts stay relatively consistent. The baseline models are the most fair while the evaluated defenses tend to be less fair. Interestingly, Activation Defense exhibits high levels of fairness during DLBD but the lowest fairness during CLBD. However, we observe that all of the filtering and adversarial training-based defenses are always less fair than the baselines and other defenses. Intuitively, this reflects the mechanics of the defenses, which remove outliers or try to minimize the effect of minority data. In contrast, data Augmentation consistently shows the highest fairness for both types of attacks.

In the heatmaps for the class percentages of the three intervals (second row of Fig. 2), we notice that most defenses are not especially unfair as the majority of classes always lie within the $|SPD| \leq 0.1$ range. The exceptions are Activation Defense and the D.A. + A.T. combo which in some cases drop below 50% indicating they may be biased. We also observe that each baseline or defense tends to be skewed towards the either the negative $SPD < -0.1$ or positive $SPD > 0.1$. The baselines and Activation Defense tend to be skewed towards the positive range with the percentage of positive range larger that the negative range, while the other defenses tend to be skewed towards negative range. This pattern is consistent for both DLBD and CLBD and for both triggers. This shows that most of the defenses are biased towards a specific sub-population regardless of the type of poisoning attacks.

5 Conclusion

Machine learning algorithms, although used for critical tasks, are susceptible to adversarial attacks. Poisoning attacks, in particular, pose a large risk to these ML models. Many defenses have been proposed to protect against poisoning attacks. Traditionally, these defenses have been evaluated using attack success rate and benign accuracy. However, these metrics do not show the complete way in which a defense may influence the model. To uncover potential side-effects of defenses, we introduced the use of a fairness metric to understand how different sub-populations can be affected.

In our evaluations, we found that some defenses that produce robust models with a low attack success rate can actually yield unfair and biased models with a low amount of classes in the acceptable

Statistical Parity Difference (SPD) range. This was particularly true for the filtering and adversarial training defenses. Overall, our work highlights that creating robust models may have unintended consequences on the final model quality and certain sub-populations. We encourage future evaluations of adversarial defenses to use metrics outside of the traditional clean accuracy and attack success rate such as SPD to measure additional qualities of the model including its resulting fairness.

References

[1] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilović, et al. Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63(4/5):4–1, 2019.

[2] Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021. URL `https://doi.org/10.1109/icassp39728.2021.9414862`.

[3] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

[4] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[5] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference*, ITCS '12, page 214–226, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311151. doi: 10.1145/2090236.2090255. URL `https://doi.org/10.1145/2090236.2090255`.

[6] Yuyang Gao, Giorgio A. Ascoli, and Liang Zhao. Bean: Interpretable and efficient learning with biologically-enhanced artificial neuronal assembly regularization. *Frontiers in Neurorobotics*, 15, 2021. ISSN 1662-5218. doi: 10.3389/fnbot.2021.567482. URL `https://www.frontiersin.org/articles/10.3389/fnbot.2021.567482`.

[7] Jonas Geiping, Liam Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276*, 2020.

[8] Jonas Geiping, Liam H Fowl, Gowthami Somepalli, Micah Goldblum, Michael Moeller, and Tom Goldstein. What doesn't kill you makes you robust(er): How to adversarially train against data poisoning. *arXiv preprint arXiv:2102.13624*, 2021.

[9] Chengyue Gong, Tongzheng Ren, Mao Ye, and Qiang Liu. Maxup: Lightweight adversarial training with data augmentation improves neural network training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2474–2483, 2021. doi: 10.1109/CVPR46437.2021.00250.

[10] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[12] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

[13] Charles Jin, Melinda Sun, and Martin Rinard. Defending against backdoor attacks using ensembles of weak learners. 2022.

[14] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

[15] Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defense against general poisoning attacks. *arXiv preprint arXiv:2006.14768*, 2020.

[16] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2018.

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[18] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P. Wellman. Sok: Security and privacy in machine learning. In *IEEE European Symposium on Security and Privacy*, pages 399–414, 2018. doi: 10.1109/EuroSP.2018.00035.

[19] 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. In *International Conference on Machine Learning*, pages 9389–9398. PMLR, 2021.

[20] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

[21] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. 2018.

[22] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.

[23] Wenxiao Wang, Alexander J Levine, and Soheil Feizi. Improved certified defenses against data poisoning with (deterministic) finite aggregation. In *International Conference on Machine Learning*, pages 22769–22783. PMLR, 2022.

[24] Alexander Wong, Mohammad Javad Shafiee, and Michael St Jules. Micronnet: A highly compact deep convolutional neural network architecture for real-time embedded traffic sign classification. *IEEE Access*, 6:59803–59810, 2018.

[25] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision*, 2019.

[26] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

A Appendix

A.1 Sub-population Examples

Many adversarial poisoning defenses attempt to find poisoned samples by searching for sub-populations that are different from the rest of the samples in the class. However, it is possible for naturally occurring low-represented benign sub-populations to be falsely identified in this process. In different samples, we observed difference in properties like brightness, contrast, blurriness, zoom-in or zoom-out etc., can create a sub-population that may be potentially marked as poison. One such example is Figure 3 where the difference in brightness can create a sub-population that may potentially be marked as poison.

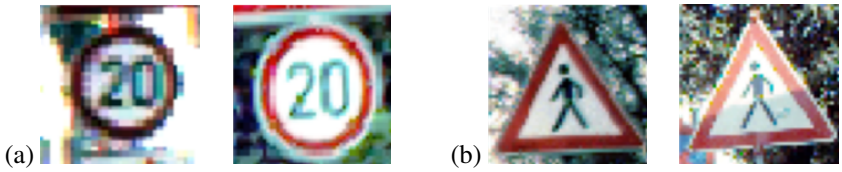

Figure 3: Sample images in the GTSRB dataset where sub-populations can naturally occur. Here, the (a) speed limit and (b) pedestrian crossing signs appear to be different sub-populations due to differences in brightness.

A.2 BEAN Regularization

We used the BEAN regularization explanatory model to determine the different sub-populations within the training data. BEAN allows for semantic interpretability due to its layer-wise regularization rules that are biologically motivated. By applying these learning constraints which adjust the weight space (enforcing modularity) BEAN ultimately allows sparsifying connections to disentangle learned concepts into distinct groups. Our qualitative tests reproduced this same behavior on multiple datasets (including GTSRB and CIFAR-10) and architectures. While the original paper, tests BEAN for better generalization and zero-shot learning, in this study, we leveraged the trait of explanatory data characterization in detecting out-of-distribution subpopulations within a single "learned" class of the pretrained BEAN model. The pretrained BEAN model uses the same architecture as the baseline defense (Activation defense) but solely for the purpose of evaluating the defense model behavior on non-poisoned inputs only.

A.3 Trigger Images

In our experiments, we evaluated the defenses and baselines against the dirty label and clean label backdoor attacks using the bullet hole and peace sign triggers for the GTSRB dataset. We also ran additional experiments on the CIFAR-10 dataset using the copyright and watermark triggers. All of these are shown in Figure 4.

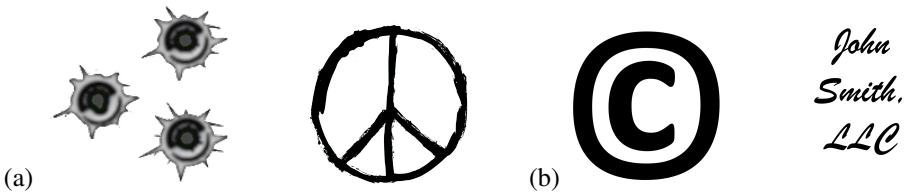

Figure 4: Poisoning attack triggers for (a) the bullet hole and peace sign used for the GTSRB dataset and (b) the copyright and watermark used for the CIFAR-10 dataset.

A.4 Alternative Hyperparameter Evaluations

In our evaluations, we used the default hyperparameters for all of the defenses based on their respective paper. We now present results for the Spectral Signatures, Data Augmentation, Adversarial Training, and the Data Augmentation and Adversarial combination (D.A. + A.T.) defenses where we use alternative hyperparameters. For Spectral Signatures, we evaluated using the best case scenario where the expected poison is the actual poison amount; now we use a fixed 30% poison for the dirty label attacks and fixed 20% poison for the clean label attacks. For Data Augmentation, Borgnia et al. [2] used Maxup with Cutout; now we use the CutMix, a different data augmentation method. For Adversarial Training, Geiping et al. [8] used a 7-step PGD attack with $\epsilon = 0.1$ and a step size of 0.02; we now use a weakened PGD attack with the same hyperparamaters that Madry et al. [17] used, a 10-step PGD with $\epsilon = 0.03$ and a step size of 0.007. For the D.A. + A.T. combo, we originally adversarially perturbed 75% of the training samples; we now use a weakened version where we only perturb 50% of the training samples.

Security Assessment: The clean accuracy and attack success rate for these alternative hyperparameter defenses are shown in Figure 5. The undefended baseline and original hyperparameter variations

are also included as reference. We report these metrics across the varying poisoning percentages for the two attack using the bullet holes and peace sign triggers. We omit 0% poison for the attack success rate plots since there will never be a successful attack. We observe that the choice of expected poison for Spectral Signatures does not affect the clean accuracy but does have a significant effect on the attack success rate. The choice of Data Augmentation method (Maxup or CutMix) does not have a very large difference for the clean accuracy or attack success rate. For both Adversarial Training and the D.A. + A.T., the weakened variant has a higher clean accuracy but also higher attack success rate.

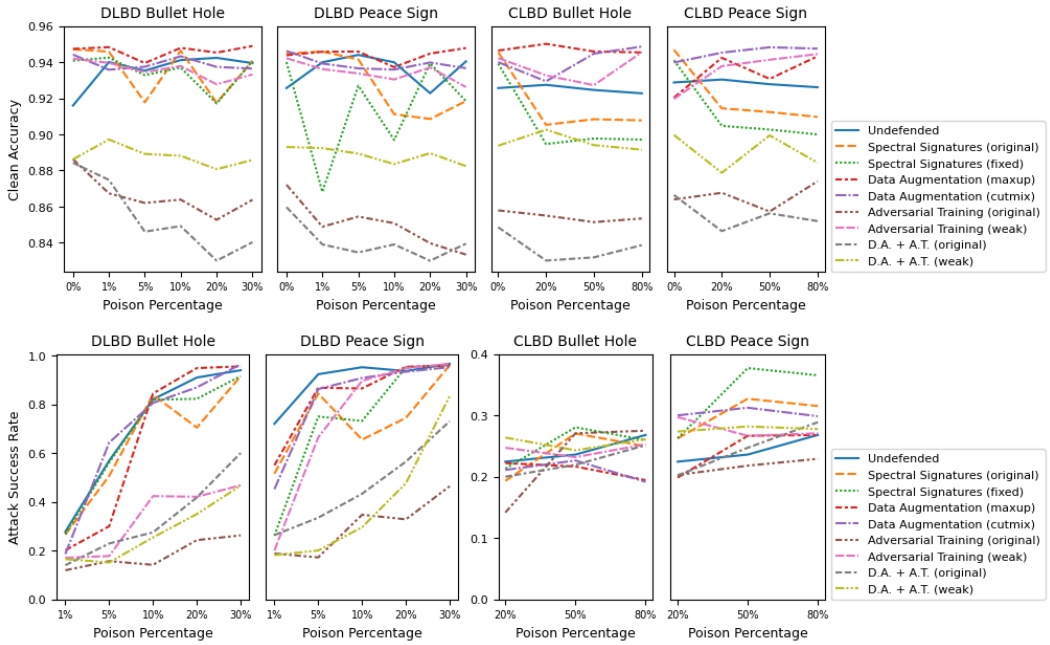

Figure 5: The clean accuracy (top) and attack success rate (bottom) for the additional hyperparameter modified defenses across varied poison. Each model is evaluated against the dirty label backdoor attack (DLBD) and clean label backdoor attack (CLBD) using the bullet hole and peace sign triggers on the GTSRB dataset.

Fairness Assessment: The fairness class counts for the acceptable $|SPD| \leq 0.1$ range and the percent distribution for all three SPD ranges are shown in Figure 6. We observe that the hyperparameters for Spectral Signature and the choice of Data Augmentation do not affect the fairness of the model as the SPD remains very similar. For both Adversarial Training and the D.A. + A.T. combo, however, we notice a large increase in fairness for the weakened variants. The same trend where the defense tends to be skewed towards the either the negative $SPD < -0.1$ or positive $SPD > 0.1$ remains consistent regardless of the hyperparameters.

A.5 Alternative Dataset and Attack Evaluations

In addition to our experiments using the MicronNet model trained on the GTSRB dataset, we also use a different model architecture and dataset to properly judge each defense's generalizability. For each of the baselines and defenses, except DPA and Finite Aggregation [2], we trained a ResNet-18 [11] on the CIFAR-10 [14]. Each model was trained for 100 epochs. We evaluate these defenses against the dirty label backdoor [10] and Witches' Brew [7] attacks. We use the Adam optimizer for models evaluated against the DLBD and the SGD optimizer for models evaluated against the Witches' Brew attack. The CIFAR-10 dataset uses the copyright and watermark triggers (see Appendix A.3) for the DLBD. The Witches' Brew attack does not use a trigger and we use 0%, 1%, 5%, 10%, 20%, and 30% poison.

[2]We omit these defenses as we lacked the time and resources to train on this choice of architecture and dataset. The results will be available for the full paper.

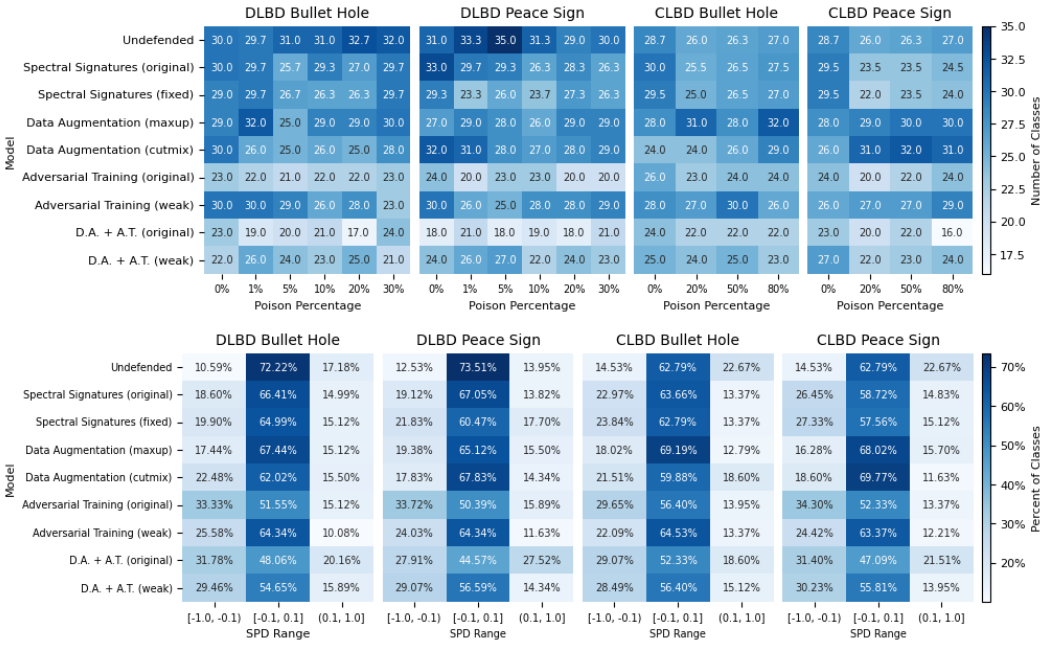

Figure 6: The number of classes within the acceptable $|SPD| \leq 0.1$ range across varied poison (top) and the percentage of classes within each of the three SPD ranges averaged across all poison percentages (bottom) for the additional hyperparameter modified defenses. Each model is evaluated against the dirty label backdoor attack (DLBD) and clean label backdoor attack (CLBD) using the bullet hole and peace sign triggers on the GTSRB dataset.

Security Assessment: In Figure 1, we report the clean accuracy and attack success rate across the varying poisoning percentages for the two attacks (and triggers where applicable). We omit 0% poison for the attack success rate plots since there will never be a successful attack. We immediately notice that nearly all of the defenses have a lower clean accuracy than the baselines. Adversarial Training and the D.A. + A.T. combo in particular perform extremely poorly, but this may be due to the choice of hyperparameters for the PGD attack. For the attack success rate, the defenses perform better than the undefended baseline for the DLBD. However, for the Witches' Brew attack, all of the defenses perform worse than the baseline. This may demonstrate that these defenses are brittle against this type of attack.

Fairness Assessment: The fairness class counts for the acceptable $|SPD| \leq 0.1$ range and the percent distribution for all three SPD ranges are shown in Figure 6. We observe that since the CIFAR-10 dataset tends to have balanced classes, the baselines and most of the defenses are very fair and tend to exhibit no bias. Adversarial Training and the D.A. + A.T. combo show some form of bias, but is still very low. This shows that against a balanced dataset, models tend to stay fair and do not exhibit much bias against sub-populations. For this reason, we used the GTSRB dataset for our main results as this dataset is unbalanced which is more realistic.

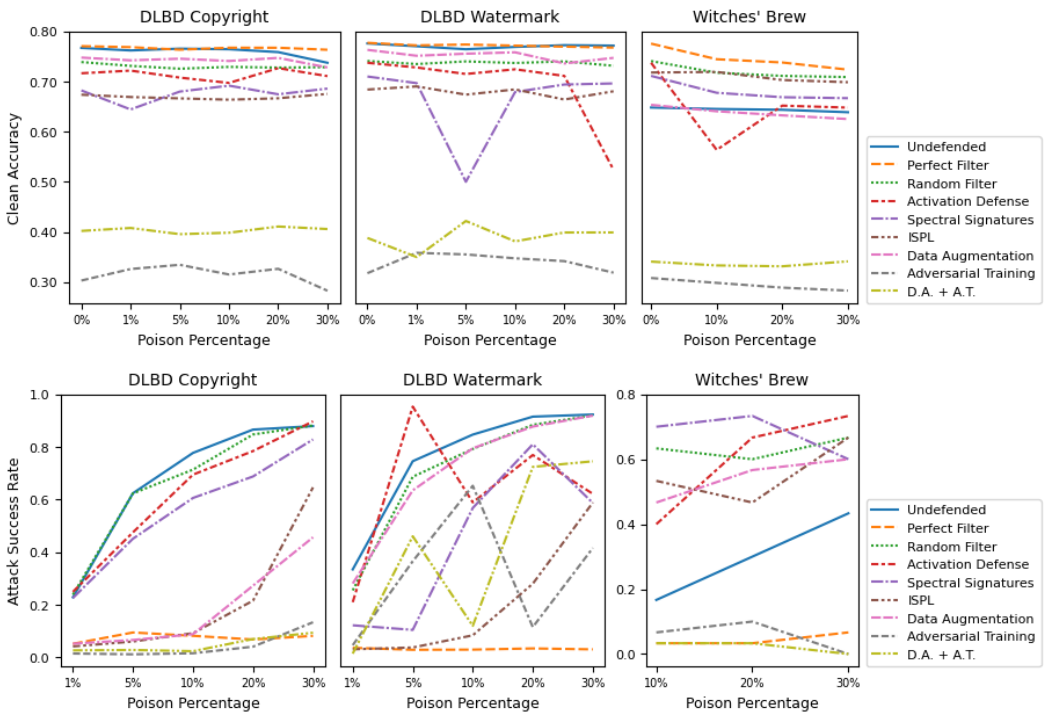

Figure 7: The clean accuracy (top) and attack success rate (bottom) for the baselines and defenses across varied poison. Each model is evaluated against the dirty label backdoor attack (DLBD) attack using the copyright and watermark triggers and the Witches' Brew attack on the CIFAR-10 dataset.

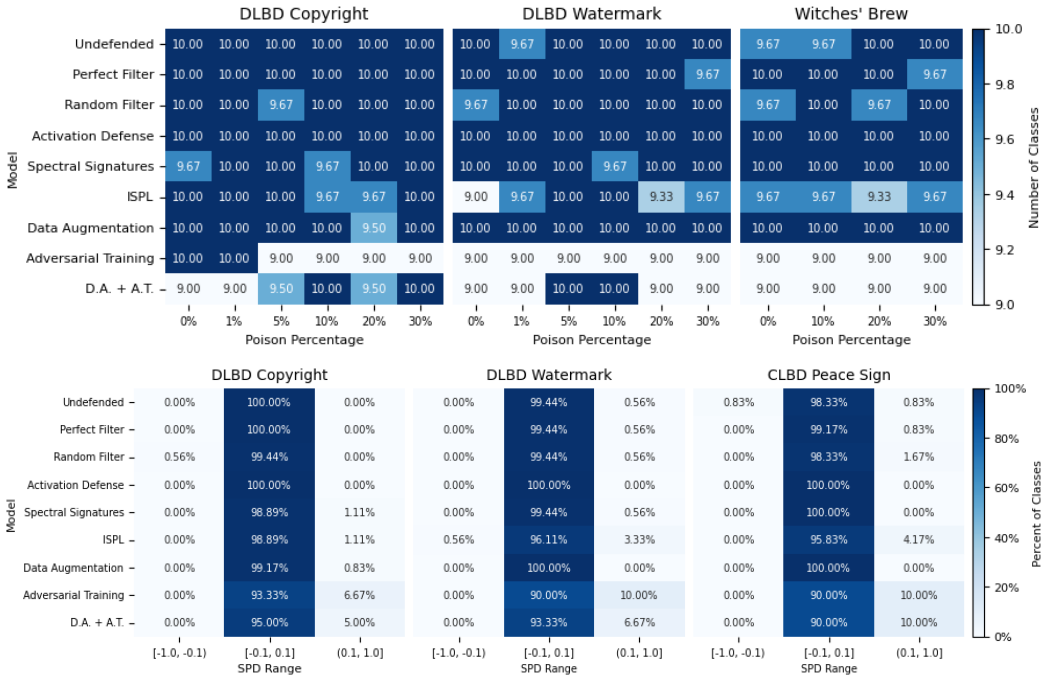

Figure 8: The number of classes within the acceptable $|SPD| \leq 0.1$ range across varied poison (top) and the percentage of classes within each of the three SPD ranges averaged across all poison percentages (bottom) for each baseline and defense. Each model is evaluated against the dirty label backdoor attack (DLBD) attack using the copyright and watermark triggers and the Witches' Brew attack on the CIFAR-10 dataset.