Impact of Data Duplication on Deep Neural Network-Based Image Classifiers: Robust vs. Standard Models

Alireza Aghabagherloo*§, Aydin Abadi**, Sumanta Sarkar‖, Vishnu Asutosh Dasu¶ and Bart Preneel*§
* *COSIC, Department of Electrical Engineering, KU Leuven, 3001 Leuven, Belgium*
§{*Alireza.aghabagherloo, bart.preneel*}*@esat.kuleuven.be*
** *Newcastle University, aydin.abadi@ncl.ac.uk*
‖ *University of Essex, Sumanta.Sarkar@essex.ac.uk*
¶ *Pennsylvania State University, vdasu@psu.edu*

Abstract—**The accuracy and robustness of machine learning models against adversarial attacks are significantly influenced by factors such as training data quality, model architecture, the training process, and the deployment environment. In recent years, duplicated data in training sets, especially in language models, has attracted considerable attention. It has been shown that deduplication enhances both training performance and model accuracy in language models. While the importance of data quality in training image classifier Deep Neural Networks (DNNs) is widely recognized, the impact of duplicated images in the training set on model generalization and performance has received little attention.**

In this paper, we address this gap and provide a comprehensive study on the effect of duplicates in image classification. Our analysis indicates that the presence of duplicated images in the training set not only negatively affects the efficiency of model training but also may result in lower accuracy of the image classifier. This negative impact of duplication on accuracy is particularly evident when duplicated data is non-uniform across classes or when duplication, whether uniform or non-uniform, occurs in the training set of an adversarially trained model. Even when duplicated samples are selected in a uniform way, increasing the amount of duplication does not lead to a significant improvement in accuracy.

Index Terms—**Adversarial training, deduplication, convolutional neural networks, deep learning**

1. Introduction

Machine learning (ML) is a data-driven process that heavily relies on training datasets to develop efficient and robust models. During training, an ML model learns patterns and relationships within the dataset. High-quality data contributes to more accurate and reliable models, whereas poor-quality data can introduce errors or biases [1]. Numerous studies [1], [2], [3], [4] indicate that vulnerabilities often stem from the dataset itself rather than the model. These works emphasize that issues such as poor data quality, bias, sparsity, or misrepresentation can significantly affect model performance and robustness, leading to vulnerabilities independent of the model's architecture.

Thus, conducting a thorough analysis and evaluation of dataset quality is crucial for ensuring the development of effective data *preprocessing* methods. Preprocessing involves cleaning raw data through various steps, including removing redundancies and outliers. The recently proposed DeepSeek language model [5], which has garnered significant attention, also highlights the importance of high-quality data in building efficient models. One concern that may hinder the efficiency of a language model is the presence of duplicate entries in text datasets, as discussed in [6], which emphasizes the importance of removing duplicates before training.

Similarly, duplicate images frequently appear in real-world datasets. Consequently, a similar conclusion can be drawn that duplicates may affect ML models for image processing, particularly in image classification tasks. However, the impact of duplication on image classification has received limited attention. Previous studies on image classification have primarily examined the effect of training set images being duplicated on the test sets [7], [8].

We hypothesize that the existence of duplicates within the training set in image classifier DNNs will also prohibit general learning and will negatively affect efficiency, accuracy, and robustness against attacks like adversarial attacks [9], [10]. However, existing literature does not examine the impact of duplicates in training data, which leaves their effects unexplored.

In this work, we address these gaps in the literature by conducting a comprehensive study on the impact of duplicates in image classification. We examine the impact of varying levels of duplication on model generalization and training efficiency by considering both uniform and non-uniform duplication scenarios. We also evaluate the impact of repetitive data within the dataset of an adversarially trained model [11], shedding light on its influence on robustness and accuracy.

1.1. Our Contribution

The previous works studying the effect of repetitive data within datasets indicated that duplication can increase accuracy and result in better generalization while being susceptible to memorization. In this paper, our primary goal is to obtain a complete understanding of the impact of image duplication in the training set on generalization. We show that duplication in image classifier datasets does not always yield better generalization and can negatively affect it, especially in non-uniform duplications and adversarial settings.

Our primary goal is to present a comprehensive analysis of how duplicates affect the accuracy of image classification. We will first conduct a theoretical analysis of the impact of duplication in different scenarios. Assuming data is selected from two Gaussian distributions, we will evaluate the impact of adding duplicate data. Selecting both non-uniform and uniform duplication with regard to duplicated data point labels, our analysis shows that non-uniform duplication leads to biased decision boundaries and reduces model generalization to unseen data. Our empirical results on the CIFAR-10 dataset, when randomly selected images have been duplicated, show no considerable accuracy improvement. A similar experiment on the adversarially trained model on the CIFAR-10 dataset shows an accuracy drop when adding duplicated images. The implementations are publicly available [12]. The main contributions of this paper include:

- A theoretical analysis on the effect of duplication in generalization in standard and adversarially trained models will be given in Section 3.
- An experiment-based analysis of the impact of repetitive data, focusing on data collected from two Gaussian distributions, examining the effects of duplication when the duplicated data is either uniform or non-uniform, will be given in Section 4.1.
- An experimental-based analysis of the impact of duplication when data is selected from the CIFAR-10 dataset in both standard models trained on CIFAR-10 and an adversarially trained model on CIFAR-10 will be given in Section 4.2.

The paper is structured as follows: Section 2 reviews the related work on the impact of duplicates on the generalization error. Section 3 presents the theoretical perspective on how data duplication affects generalization error. Section 4 provides an experimental-based analysis of the impact of repetitive data. Section 5 concludes the paper.

2. Related Work

Data duplication, which often occurs in the real world, poses a critical concern in training datasets intended for the training of an ML model [13]. As mentioned earlier, despite its importance, this issue has received relatively less attention in image classification than in other areas, such as Large Language Models (LLMs). In the context of LLMs, much research has been conducted on the impact of duplicate data [13], [14], [15]. Lee et al. [15] show that duplication in training data of language models results in the memorization of duplicated data. Carlini et al. argue that memorization has adverse effects on the privacy and fairness of LLMs. Memorization makes LLMs vulnerable to membership inference [16] and data extraction [17] attacks. Abadi et al. [13] address the duplication issue in Federated Learning (FL) based LLMs across clients, introducing a privacy-preserving data de-duplication mechanism. This body of work emphasizes how duplicates can skew model training and performance in LLMs, leading to overfitting and reduced generalization. In LLMs, it has also been shown that deduplicated datasets reduce the memorization frequency and improve generalization [15].

In contrast, the impact of duplication in image classification has not been systematically studied. Only a limited set of studies have focused on this issue. For example, Barz and Denzler [7] highlight a considerable volume of exact duplicates and near-duplicates between the test and training sets in CIFAR-10 and CIFAR-100 [18] datasets.

Their studies show that when they tested on ciFAIR (the duplicate-free version of CIFAR), the models experienced a substantial drop in classification accuracy; in particular, in CIFAR-10, there is a 9%–14% drop in accuracy when they removed repeated images between test set and train set. In contrast, with LLM, where memorization resulting from data duplication harms generalization [13], [15], limited research on image classifier DNNs suggests that duplication may actually enhance generalization [7], [19].

Similarly, [20] explores whether memorization resulting from duplication of training data in a test set is necessary for achieving generalization in models, especially in the deep learning context. The main limitation of their observation is that they mainly examine the effect of the duplication of train sets in the test set rather than evaluating the impact of having duplicated data when training. A limitation that we aim to address in this work.

Another study by Li et al. [8] introduces a framework called CE-Dedup, which evaluates the impact of near-duplicate images on Convolutional Neural Networks (CNN) training performance. The authors propose a hash-based image deduplication method to balance the trade-off between dataset size and model accuracy. Their experiments demonstrate that removing redundant images can reduce dataset size by up to 75% with an accuracy loss, leading to more efficient training. However, this work mainly focuses on a deduplication method rather than providing a detailed analysis of the effects of duplication on model performance.

In this paper, we present a comprehensive analysis of the impact of data duplication across various scenarios, demonstrating that while duplication in image classifiers always sacrifices efficiency, it does not necessarily lead to improved accuracy. More focus is on the effect of duplication in training sets on direct generalization. Our experiments on the CIFAR-10 dataset reveal that exceeding a certain threshold of duplicated randomly selected images leads to overfitting. In adversarially trained models, duplication further degrades accuracy. This shows that, above the mentioned threshold in standard training or an adversarial training setting, deduplication increases efficiency (as discussed in [8]) and results in better generalization.

3. Theoretical Perspective on the Impact of Duplicates on Generalization Error

3.1. Generalization Error

Generalization refers to a model's ability to perform well on unseen data, balancing between bias and variance. The generalization error can be decomposed into three components: bias, variance, and irreducible error. Geman et al. [21] decomposed the Mean Squared Error (MSE) loss function in terms of a prediction model's bias and variance, as stated in Theorem 1 [21].

Theorem 1. *For a prediction model $\hat{f}(x)$ trained on a dataset D to estimate the target function $f(x)$ using an MSE loss function, the bias-variance trade-off is given by:*

$$\begin{aligned}\mathbb{E}_{x,D,\gamma}\left[(y-\hat{f}(x))^2\right] &= \mathbb{E}_{x,D}\left[\left(\mathbb{E}_D[\hat{f}]-f(x)\right)^2\right] \\ &\quad + \mathbb{E}_{x,D}\left[\left(\hat{f}-\mathbb{E}_D[\hat{f}]\right)^2\right] \\ &\quad + \sigma_{\gamma}^2. \qquad (1)\end{aligned}$$

This can be simplified as:

$$e = \text{Bias}^2[\hat{f}] + \text{Var}[\hat{f}] + \text{Irreducible error}. \qquad (2)$$

In this framework, bias decreases as the model complexity increases, allowing it to learn more patterns. However, this often leads to higher variance (overfitting), where the model becomes sensitive to the specific training data, potentially reducing generalization performance. In other words, the risk of overfitting typically decreases as the number of samples increases or the complexity of the model is reduced. However, reducing model complexity or increasing the number of samples can lead to higher bias. The trade-off between bias and variance is critical for improving model generalization without overfitting, particularly in repeated patterns or data duplication scenarios.

Proposition 1. *In a typical scenario, increasing the amount of data generally reduces variance. However, in the case of a duplicated dataset, the variance of the classes that do not have duplicated data in the training set may increase, while the variance of the classes with more duplicated data progressively decreases. Regarding bias, duplication amplifies bias in favor of the duplicated class. This dynamic underscores that generalization can either improve or deteriorate depending on the balance between variance and bias in the context of duplication.*

In Section 4, we validate the bias-variance trade-off in a model trained on data with duplication established in Proposition 1.

3.2. Generalization Error in Adversarial Settings

Typically, models exhibit susceptibility to adversarial attacks [9], [10], [22], with adversarial training serving as a common defense mechanism. It involves augmenting the training dataset with adversarial examples (perturbed inputs) so the model learns to recognize and correctly classify them. This improves the model's resilience to adversarial perturbations by making the model robust to a wider range of potential attacks. As we will discuss, duplicated data might benefit adversaries. Thus, it is vital to analyze the effect of data duplication in the adversarial settings.

Assume $\bar{f}(x) = \mathbb{E}_D[\hat{f}(x)]$ and the target function is $f(x)$. The bias-variance trade-off for the mean squared error (MSE) loss function becomes more complex when noise γ and adversarial perturbations $\beta(x)$, generated by an adversarial algorithm, are introduced. In adversarial machine learning, generalization becomes more challenging due to adversarial perturbations. These perturbations aim to exploit vulnerabilities in the model, often leading to a trade-off between robustness and generalization.

According to [23], the vulnerability of ML models under adversarial attacks can be attributed to the bias-variance trade-off. The relationship between generalization and adversarial vulnerability can be analyzed using the bias-variance decomposition in the context of adversarial settings, as stated in Theorem 2 [23].

Theorem 2. *Assume $\bar{f}(x) = \mathbb{E}_D[\hat{f}(x)]$ represents the expected model prediction over the dataset D, and $f(x)$ is the true target function. When adversarial perturbations $\beta(x)$ are introduced, the MSE loss function for a model $\hat{f}(x)$ trained on D with noise γ is given by:*

$$\mathbb{E}_{x,D,\gamma}\left[(y-\hat{f}(x+\beta(x)))^2\right] \approx \qquad (3)$$

$$\mathbb{E}_{x,D}\left[(f(x)-\bar{f}(x)-c_x)^2\right] + \mathit{Var}[\gamma] + \mathit{Var}[\hat{f}] + \mathbb{E}_{x,D}[c'_x]$$

Where:

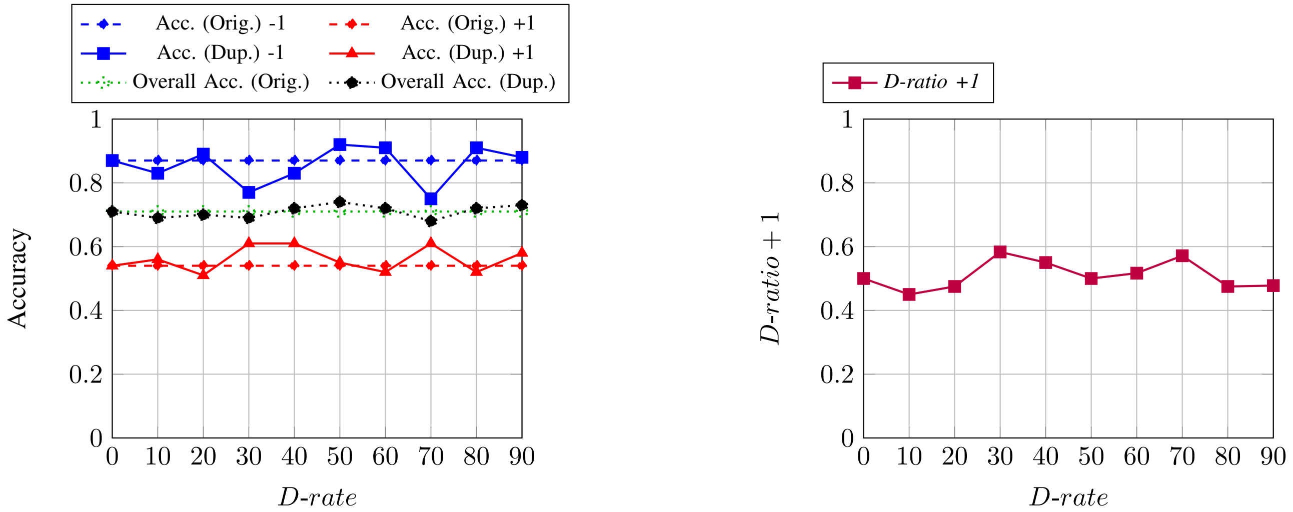
$$c_x = \nabla\bar{f}(x)^T\beta(x),$$

and

$$c'_x = 2\left(\hat{f}(x)-\bar{f}(x)\right)\left(\nabla\hat{f}(x)-\nabla\bar{f}(x)\right)^T\beta(x).$$

In this equation, c_x represents the interaction between the model's gradient and the adversarial perturbation, while c'_x captures the variability in the model's predictions under adversarial conditions. These terms illustrate how adversarial perturbations can increase both bias and variance, thus impacting generalization in adversarial settings. This equation highlights that in an adversarial-trained model, beyond the typical trade-off between bias and variance, there is an additional term $\mathbb{E}_{x,D}[c'_x]$.

Proposition 2. *The impact of the additional term $\mathbb{E}_{x,D}[c'_x]$ becomes significant when dealing with duplicated data. Specifically, duplicated data exacerbates the variability introduced by adversarial perturbations, leading to a compounding effect on bias and variance. This makes a model*



(a) Effect of duplication rate on accuracy results.

(b) *D-ratio +1* vs. *D-rate*.

Figure 1: Comparison of accuracy results (left subfigure) and the percentage of duplications from class +1 to the total number of duplications (right subfigure, denoted as *Duplication Ratio for +1 (D-ratio +1)*) with varying levels of data duplication (denoted as *Duplication Rate (D-rate)*) in a uniform duplication setting.

overly sensitive to specific perturbations within the duplicated data, reducing its ability to generalize effectively in adversarial settings to clear data.

Section 4.2 discusses our experimental results on the accuracy and robustness of a model in an adversarial setting when we have duplicated data in the training set, validating the assertion in Proposition 2.

4. Experimental Analysis of the Impact of Duplication in the Training Set

In this section, we validate the theoretical analysis presented in the previous section using experimental results. First, we conduct theoretical experiments using data sampled from two Gaussian distributions. Then, we perform experiments on the CIFAR-10 dataset [24] in both standard and adversarial settings. Notably, in our paper, the standard setting refers to training a model, known as the standard model, without any adversarial training or robustification. In contrast, the adversarial setting denotes training a model that has been made robust using adversarial training [9].

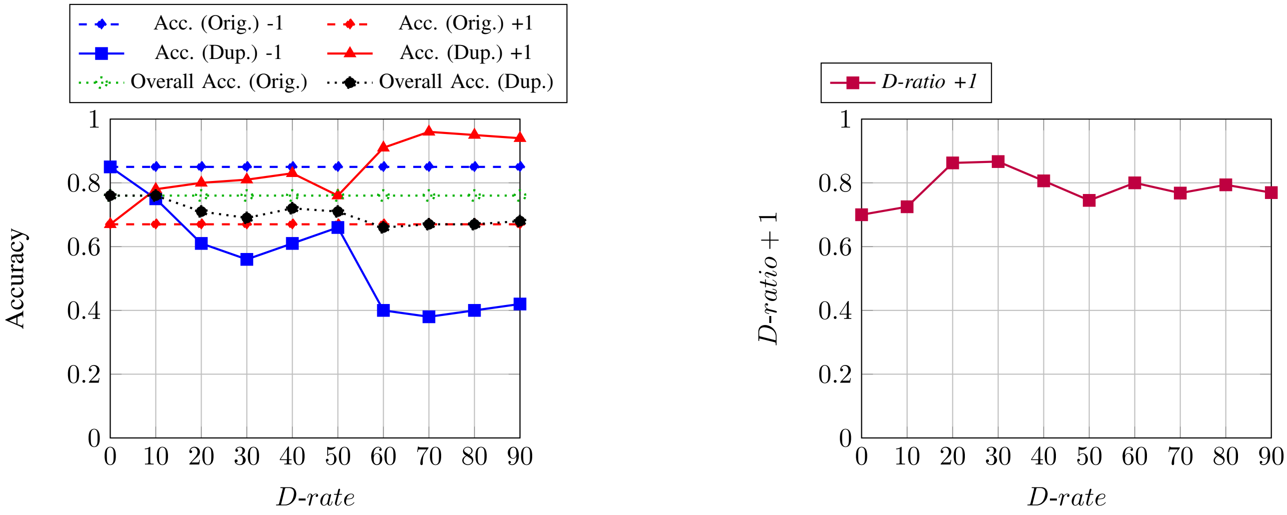
In our paper *duplication* refers to the process of repeating and adding certain data points to the training set. In our experiments, we selected duplications using two types of methods. The first method, *uniform duplication*, includes uniformly selecting data from the dataset at random and adding them to the dataset. The second method, *non-uniform (biased) duplication*, selects data points from one class more favorably than other classes.

4.1. Generalization Error with Gaussian-Distributed Data

This section analyzes the effects of repetitive data across various scenarios, using two-dimensional Gaussian data generated with class means of $[0, 0]$ and $[1, 1]$, a shared covariance matrix of $\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$, and 100 samples per class. Then, duplicated data with varying *Duplication Rates (D-rate)*, ranging from 10% to 90% of the dataset, have been added to the dataset to analyze the effect of duplication. Additionally, the Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel is used to compute decision boundaries in the experiments. The definitions of the RBF kernel, the optimization problem, and the mathematical details of the probability density functions for the two distributions are presented in Appendix A.

The impact of duplication on the decision boundary and generalization appears to be influenced by two key factors. The first factor is the randomly selected duplicates' pattern and proximity to the decision boundary. The second factor is the proportion of duplicated data points from a single class relative to the total dataset size, referred to as *D-ratio +1*, which will be examined in Section 4.1.1 and Section 4.1.2.

4.1.1. Uniform Duplication. When uniform duplicates are added, the SVM's objective function experiences an increase in the effective density of data points near the mean vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. This can cause the decision boundary to shift slightly towards regions with more duplicates. As the data is selected randomly (not favoring any specific class), this can, by chance, cause *D-ratio +1* to deviate from 50% but will generally remain close to it. The detailed results are shown in Figure 1. We refer to the original dataset



(a) Effect of duplicates on accuracy results.

(b) *D-ratio +1* vs. *D-rate*.

Figure 2: Comparison of accuracy results (left subfigure) and the percentage of duplications from class +1 to the whole number of duplications (right subfigure) with varying levels of data duplication in a non-uniform setting.

as data without any duplicates. **Acc. (Orig.) -1** and **Acc. (Orig.) +1** refer to the accuracy of class -1 and class +1 in the original dataset respectively, which means the rate of correctly identifying elements from class -1 and class +1, respectively. **Acc. (Dup.) -1** and **Acc. (Dup.) +1** represent the accuracy for these classes after duplicates are added to the original dataset. **Overall Acc. (Orig.)** and **Overall Acc. (Dup.)** denote the overall accuracy for the original and dataset with duplication, respectively.

A comparison of Figure 1a and Figure 1b reveals a correlation between *D-ratio +1* and accuracy. Initially, class +1 has an accuracy of 87%, while class -1 has 54%. When *D-ratio +1* is exactly 50% (e.g., at *D-rate* = 50), duplication increases accuracy, resulting in a 3% increase. However, when *D-ratio +1* deviates from 50% (e.g., at *D-rate* = 30), it benefits one class while harming the other, ultimately reducing overall accuracy. Specifically, at *D-rate* = 30, *D-ratio +1* is 58%, increasing class +1 accuracy from 54% to 61% but decreasing class -1 accuracy from 87% to 77%, leading to a 2% drop in total accuracy. The other important observation is that the effect of uniform duplicates on the accuracy of individual classes and the overall dataset remains within approximately the range of up to 10% around the original accuracies.

This demonstrates that deviation from uniform duplication can introduce class bias, as described in Equation 2 and Proposition 1, ultimately degrading accuracy. Another key observation from Figure 1a is that increasing the number of duplications does not necessarily enhance generalization.

Therefore, uniform duplication generally has a positive effect on accuracy, except when there is a deviation from uniformity. To better understand how such deviations impact accuracy, we conducted an experiment where the number of duplications was biased toward a specific class.

4.1.2. Non-uniform Duplication. Here, a non-uniform duplication strategy is applied by assigning different selection probabilities to the classes. Samples with label 1 are selected with a probability of 0.7, while those with label -1 are selected with a probability of 0.3. As shown in Figure 2 when duplicates are added from one class, say $\mathcal{C}_1$, the optimization problem's constraints are more heavily influenced by $\mathcal{C}_1$, resulting in a decision boundary that skews towards $\mathcal{C}_2$. This non-uniformity in the distribution of duplication increases the classifier's bias towards $\mathcal{C}_1$ and negatively impacts generalization.

This figure clearly illustrates how non-uniform duplication introduces bias, leading to an increase in accuracy for one class while decreasing accuracy for the other. For instance, at *D-rate* = 40, the original accuracy of Class -1 is 0.85, while Class +1 starts at 0.67. After duplication, the accuracy of Class -1 drops to 0.61, whereas the accuracy of Class +1 increases to 0.83. Consequently, the overall accuracy decreases from 0.76 to 0.72. Another key observation is that, in this setting, increasing the number of duplications further exacerbates bias and leads to a decrease in overall accuracy (Figure 2a).

4.2. Generalization Error with CIFAR-10 Dataset

In this section, we present an experimental-based analysis of the impact of data duplication when training models on the CIFAR-10 dataset. Our primary focus is to systematically analyze how randomly selected repetitive data influences classification accuracy in both standard models and an adversarially trained model, which is trained under Projected Gradient Descent (PGD) attacks constrained by an ℓ_2 norm with $\epsilon = 0.5$. We introduce varying levels of duplicated data during training and evaluate model performance under each duplication setting.

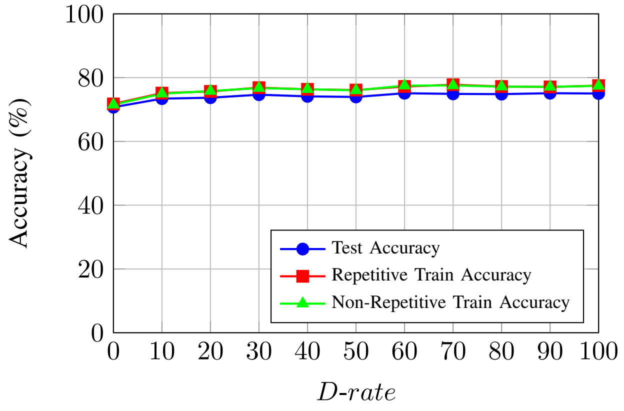


Figure 3: The effect of adding repetitive samples to the training set on test accuracy, the accuracy of the model on the training set including duplicated samples (repetitive accuracy), and the accuracy of the model on the training set excluding duplicated samples (non-repetitive accuracy).

4.3. Results and Observations in Standard Model

Since the dataset is selected randomly, the distribution of duplicated samples from CIFAR-10 does not favor any particular class. As previously discussed in Section 4.1.1, this theoretically may lead to improved classification accuracy. However, the actual impact will be examined here.

Our key findings are summarized in Figure 3, which shows the test and training accuracy for models trained with varying levels of data duplication. The experiments reveal that without duplication, the test accuracy is 70.72%. With 30% duplication (D-rate = 30), the test accuracy improves to 74.12%, reflecting a modest 3% increase. This suggests that increasing the number of duplications, which adds more data for training and reduces efficiency, does not lead to a substantial improvement in accuracy. Increasing the duplication rate from 30% to 60% results in only small improvements, increasing test accuracy from 74.12% to 75.11%. However, beyond this point, additional duplication yields diminishing returns, and in some cases, even slight performance degradation. This shows that excessive duplication may introduce redundancy without contributing significant diversity to the training process.

4.4. Results and Observations in Robust Model

Adversarially trained models exhibit a more complex response to data duplication, where repetition does not improve robustness. Unlike standard models, which benefit from uniform repetition across classes, adversarially trained models are sensitive even to uniform duplication. Such repetition undermines adversarially trained models' accuracy.

This is illustrated in Figure 4, which shows the impact of duplication on both accuracy (i.e., the ability to correctly classify clean data) and adversarial accuracy (i.e., robustness—the ability to correctly classify data with adversarial perturbations) across different levels of uniform repetition. Without duplication, test accuracy is 41.16%. At 30% duplication, accuracy decreases to 27.39%, and at 60% duplication, it further drops to only 20.90%. This suggests a significant negative effect of duplication on accuracy, without improving adversarial accuracy (or robustness). Furthermore, this supports our claim in Proposition 2 that repetitive data negatively impacts accuracy.

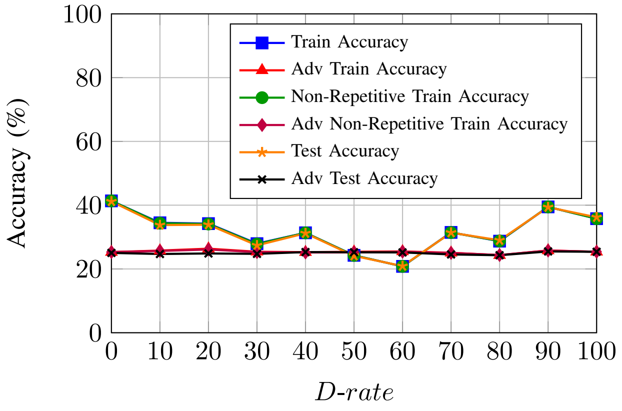


Figure 4: Impact of repetitive data in the training set on training accuracy, test accuracy, and the robustness (adversarial accuracy) of an adversarially trained model.

This suggests that adding duplications in adversarially trained models does not lead to a general improvement in robustness or accuracy, and in fact, it may hinder the model's ability to classify clean and adversarial data correctly.

5. Conclusion and Future Work

This paper investigates the impact of duplicated samples in image classifier DNNs. First, a theoretical analysis explores how duplication influences generalization. Then, experimental studies are conducted. Our findings suggest that while duplication can sometimes aid in refining decision boundaries, it does not always improve generalization. In particular, under non-uniform duplication or adversarial training conditions, duplicated data may negatively impact generalization. The observations from our experiments conclude that duplication not only negatively impacts efficiency—since more data must be trained—but also requires careful handling, as it can harm the model's generalization ability depending on the distribution and nature of the repetitive data.

Our results are based on selecting duplications from a Gaussian distribution and training a DNN on CIFAR-10 as a toy example. Future work can explore the impact of duplication in more practical scenarios, such as FL, where duplication is more likely due to clients lacking knowledge of each other's datasets. Another research direction involves developing methods for private deduplication. Additionally, duplication may have privacy implications, as duplicated samples could be more easily revealed. However, the actual effect on privacy requires further investigation.

Acknowledgment

This work was supported by the Flemish Government through the Cybersecurity Research Program with grant number: VOEWICS02 and in part by the Research Council KU Leuven projects IF/C1 From Website Fingerprinting to App Fingerprinting: Inferring private user activity from encrypted network traffic, and the AIDE project funded by the Belgian SPF BOSA under the programme "Financing of projects for the development of artificial intelligence in Belgium" with reference number 06.40.32.33.00.10." Sumanta Sarkar was supported by UKRI through the EPSRC grants EP/R029563/1 and EP/Y028813/1.

References

[1] L. Budach, M. Feuerpfeil, N. Ihde, A. Nathansen, N. Noack, H. Patzlaff, F. Naumann, and H. Harmouch, "The effects of data quality on machine learning performance," *arXiv preprint arXiv:2207.14529*, 2022.

[2] G. Fenza, M. Gallo, V. Loia, F. Orciuoli, and E. Herrera-Viedma, "Data set quality in machine learning: Consistency measure based on group decision making," *Applied Soft Computing*, 2021.

[3] M. Esposito and D. Falessi, "Validate: A deep dive into vulnerability prediction datasets," *Information and Software Technology*, 2024.

[4] P. Xiong, M. Tegegn, J. S. Sarin, S. Pal, and J. Rubin, "It is all about data: A survey on the effects of data on adversarial robustness," *ACM Computing Surveys*, 2024.

[5] DeepSeek-AI, A. Liu, B. Feng *et al.*, "Deepseek-v3 technical report," 2024.

[6] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini, "Deduplicating training data makes language models better," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[7] B. Barz and J. Denzler, "Do we train on test data? urging cifar of near-duplicates," *Journal of Imaging*, 2020.

[8] X. Li, L. Chang, and X. Liu, "Ce-dedup: Cost-effective convolutional neural nets training based on image deduplication," in *IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking*, 2021.

[9] A. Aghabagherloo, R. Gálvez, D. Preuveneers, and B. Preneel, "On the brittleness of robust features: An exploratory analysis of model robustness and illusionary robust features," in *IEEE Security and Privacy Workshops (SPW)*, 2023.

[10] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Annual Conference on Neural Information Processing Systems 2019, NeurIPS*, 2019.

[11] F. Tramèr and D. Boneh, "Adversarial training and robustness for multiple perturbations," in *NeurIPS*, 2019.

[12] A. Aghabagherloo, "Source code," 2025. [Online]. Available: https://github.com/alireza1375/Duplication.git

[13] A. Abadi, V. A. Dasu, and S. Sarkar, "Privacy-preserving data deduplication for enhancing federated learning of language models," in *The Network and Distributed System Security (NDSS) Symposium*, 2025.

[14] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, "Quantifying memorization across neural language models," in *The Eleventh International Conference on Learning Representations*, 2024.

[15] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini, "Deduplicating training data makes language models better," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[16] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE Symposium on Security and Privacy (SP)*, 2017.

[17] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, and A. Oprea, "Extracting training data from large language models," in *USENIX Security Symposium (USENIX Security)*, 2021.

[18] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[19] V. Feldman and C. Zhang, "What neural networks memorize and why: Discovering the long tail via influence estimation," in *Advances in Neural Information Processing Systems*, 2020.

[20] H. Abdullah, K. Wang, B. Hoak, Y. Wang, S. S. Arora, and Y. Cai, "Is memorization actually necessary for generalization?" 2021.

[21] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, 1992.

[22] A. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, and G. Loukas, "A taxonomy and survey of attacks against machine learning," *Comput. Sci. Rev.*, 2019.

[23] H. Aboutalebi, M. J. Shafiee, M. Karg, C. Scharfenberger, and A. Wong, "Vulnerability under adversarial machine learning: Bias or variance?" *arXiv preprint arXiv:2008.00138*, 2020.

[24] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

Appendix A. Theoretical Model: Gaussian Data Generation and SVM Classification

The data is generated using two Gaussian distributions with mean vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, and a shared covariance matrix Σ. The probability density function for class $\mathcal{C}_k$ where $k \in \{1, 2\}$ is given by:

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right), \tag{4}$$

where $\mathbf{x} \in \mathbb{R}^d$ is a data point in d-dimensional space.

In the experiments, an SVM with a Radial Basis Function (RBF) kernel is used to compute the decision boundaries. The RBF kernel is defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma|\mathbf{x}_i - \mathbf{x}_j|^2\right), \tag{5}$$

where $\gamma > 0$ is a parameter that controls the width of the Gaussian function.

The SVM solves the following optimization problem to find the optimal separating hyperplane:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}|\mathbf{w}|^2 + C\sum_{i=1}^{n} \xi_i, \tag{6}$$

subject to the constraints:

$$y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \tag{7}$$

where $\mathbf{w}$ is the weight vector, b is the bias term, ξ_i are the slack variables, C is the regularization parameter, and $\phi(\mathbf{x})$ is the feature transformation induced by the RBF kernel.