symmetry

MDPI

Article

Effects of Adversarial Training on the Safety of Classification Models

Handong Kim and Jongdae Han *

Department of Computer Science, Sangmyung University, Seoul 03016, Korea; aggsae@gmail.com
* Correspondence: elvenwhite@smu.ac.kr

Abstract: Artificial intelligence (AI) is one of the most important topics that implements symmetry in computer science. As like humans, most AI also learns by trial-and-error approach which requires appropriate adversarial examples. In this study, we prove that adversarial training can be useful to verify the safety of classification model in early stage of development. We experimented with various amount of adversarial data and found that the safety can be significantly improved by appropriate ratio of adversarial training.

Keywords: requirements engineering; nonfunctional requirements; safety; artificial intelligence; adversarial training

1. Introduction

Artificial intelligence (AI) is one of the most important topics that implements symmetry in computer science. As humans learn how to recognize objects through a trial-and-error approach, AI that utilizes supervised learning techniques also learns using a similar approach. However, sometimes it is overlooked that we need to supply not only correct data but erroneous data to make use of such symmetry.

Software requirements are one of the most important factors for software quality. Verifying the requirements in an early stage of the software development life cycle induces higher software quality and lower cost. Requirements can be divided into functional requirements and non-functional requirements (NFR). Functional requirements are fulfillments of functional user expectations, while other requirements are considered NFRs. Typical NFRs include performance (whether software performs the desired behavior, ex. accuracy), fairness [1–6], security [7–9], safety (whether the execution result of the software is safe) [10–12], and transparency (whether software can be trusted) [13–15]. It is essential to verify not only functional requirements but also NFRs for high-quality software.

Traditionally, there are well-known methods to verify requirements, including thoughtful documentation of requirement specifications, model-based verification, etc. These methods focus on making detailed documentation and testing at each stage of the software development lifecycle. Hereafter, developers can directly check the operation of code; thus, functional requirements and NFRs can be verified through codes.

Nevertheless, with the advent of deep neural networks, which have different characteristics from existing common software, it is difficult to verify requirements through traditional methods. Performance, which is one of the most widely applied NFRs, can be verified to some extent with accuracy or F1 score. Most data-driven deep neural networks have some kind of accuracy as a critical criterion. However, other NFRs such as safety are difficult to verify because of the deep neural network's black-box property.

It is difficult to apply traditional verification methods to deep neural networks because they operate as a black box model. In traditional software, NFRs can be verified through codes, but it is difficult to verify a deep neural network with program codes because the model is not a explainable representation of logical flow. Meanwhile, the verification of

check for updates

Citation: Kim, H.; Han, J. Effects of Adversarial Training on the Safety of Classification Models. *Symmetry* **2022**, *14*, 1338. https://doi.org/10.3390/sym14071338

Academic Editor: José Carlos R. Alcantud

Received: 5 April 2022
Accepted: 20 June 2022
Published: 28 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

NFRs is very important for deep neural networks, especially if they are applied to mission- or safety-critical systems, such as autonomous driving [16,17], medical diagnosis [18–24], and finance [25]. If NFRs such as safety, security, and transparency are not guaranteed, the application of newer AI techniques will be risky due to liability issues. While quality assurance of AI is performed only on the performance corresponding to how well AI through metrics such as accuracy and F1 score, there should be more considerations upon other NFRs to apply those models to a variety of domains.

Several studies have been conducted in this regard, and those studies can be categorized into two. The first category directly deals with various types of AI-inducing NFRs.

- Studies have defined the fairness of AI, addressed problems that may arise when fairness is not verified, and proposed methods for verifying fairness [2–6].
- In [7–9], security problems that can arise due to the vulnerability of AI were discussed. Additionally, attack methods threatening the security of AI were explained and methods to prevent them were proposed.
- In [13], the authors proposed a method for understanding the behavior of AI by visualizing the activation of each layer of neural network models.
- Some researchers [10–12] defined the safety of AI, proposed safety methods, and studied safety-violation cases.

The other category consists of studies investigating problems related to the NFRs of AI, although they do not directly address them. Explainable AI (XAI) and adversarial training are two of the topics of such studies. XAI is expected to enable people to understand the process of training and deriving the results of AI. XAI is expected to validate AI models by providing more transparency [14,15,20,26,27]. On the other hand, the concept of adversarial training has been introduced, which makes trained models more robust against adversarial examples [28–33].

Although the main purpose of these studies is to maintain or improve the performance of models against adversarial examples, they can be interpreted as a verification method to properly classify adversarial examples that threaten the safety of AI models. As those studies explain the learning process of models with adversarial examples, they do not address the application of adversarial training for verification of AI NFRs. Therefore, in our study, on the basis of the above-cited studies, we will verify NFRs of AI, especially the safetyof classification models using *adversarial training*. The details of our research are discussed in Section 3.

In this study, we will show the safety of classification models can be verified using adversarial examples as a part of data preparation for training (adversarial training). To prove this, we set up the following research questions (RQs).

RQ1. Can classification models appropriately classify adversarial examples by adversarial training?

RQ2. Do they show same results as RQ1 for safety-related datasets?

RQ3. Does adversarial training affect safety regardless of the model used for training?

RQ4. At what rate should adversarial examples be included with normal data to archive a balance between safety and accuracy?

With experiments, we could find out the following:

1. Adversarial training can improve safety. Depending on the experimental results, safety, which is defined as the successful classification of adversarial examples, is improved by 20–40%.
2. Adversarial training can be used to verify the safety of AI. Experimental results show that adversarial training can affect safety improvement, which means it can be used as a verification method to ensure safety by applying a certain number of adversarial examples to the training dataset.

3. Adversarial training can be utilized as a factor to verify safety regardless of the model structure. Improvement in safety is observed regardless of the model structure.
4. We propose a process of adjusting the adversarial-to-normal ratio in training datasets according to the preferential requirements. Experimental results with different adversarial example inclusion ratios show that adversarial inclusion ratios play a key role in a tradeoff between accuracy and safety.

The rest of this paper is structured as follows. In Section 2, we review previous studies on verification methods for AI NFRs. In Section 3, we define the safety of software and show the experimental design applied in this study. In Section 4, we explain experiments conducted according to the designed experimental methods, and analyze the results in Section 5. We conclude the paper and recommend future work in Section 6.

2. Related Work

Prior studies have directly and indirectly addressed the NFRs verification of AI.

2.1. Studies That Directly Addressing the NFRs of AI

2.1.1. Fairness of AI

With current supervised learning techniques, AI learns the features of data and derives the results of input data based on the learned features. Because of these characteristics, if there is a bias in the training dataset, discriminatory results should be derived by AI.

As explained in [2,6], bias in training datasets can lead to discriminatory results according to features such as race, gender, and age of each individual. The authors proposed a statistical method to find out discriminating elements in training datasets and argued that it is important to maintain the fairness of AI as data bias is directly related to privacy.

Feldman et al. [3] addressed the problem of unfairness that is caused by discriminatory features of datasets and devised a method to solve it.

Zhang et al. [5] showed that such imbalances originate from not only data but also data features, and lead to worse performance. To improve the fairness of AI [1,3,4] ensured the fairness of AI by removing or finding factors, such as bias in datasets.

2.1.2. Security of AI

AI is recently applied to areas even closely related to human privacy. Due to such trend, the more it demands security for sensitive, private information.

Studies have been conducted to identify and classify various security problems occurring because of vulnerabilities in AI and attacks that can cause security problems [9]. There are studies that suggest attack methods that can threaten security through these vulnerabilities and defense methods to defend them.

Mei et al. [8] proposed an attack method that can threaten the security of tools widely used in actual data analysis. Another study by the same authors [7] proposed the method to attack security through injecting malicious data into training data and technique to prevent such attacks.

Aryal et al. [34] covers various adversarial attack methods that utilize machine learning. Although our study is not directly addressing specific attack methods, we expect our method is effective against black-box adversarial attacks.

A series of studies [35–37] suggest CNN-based machine learning can be useful to detect malware, resulting in improved security. They say careful data preparation for training leads to an improvement in security.

2.1.3. Transparency of AI

Most of the AI currently applied operates in a black-box manner, therefore users cannot explain how certain results can be derived from models. Lack of transparency in AI due to black-box operations can lead to complex issues, including legal issues regarding who is responsible when problems arise due to AI's decisions

A study [13] proposed two tools to visualize layers activated in the process of deriving results for convolutional network-based deep learning models. By analyzing visualized layers, users can make sure whether the corresponding result is appropriately derived and prevent problems in advance.

2.1.4. Safety of AI

Safety is usually defined as a state or a place where you are safe and not in danger or at risk. To apply the definition to AI, AI decisions should not endanger humans when used in various domains. Safety is especially important in domains such as autonomous driving, where AI decisions have a significant impact on humans. If safety is not properly verified throughout software development using AI, it can lead to serious results.

Amodei addressed safety issues that occur in existing AI studies [10]. In the study, examples showed safety can be violated often by prioritizing performance.

There are other studies [11,12] described the concept and definition of safety in AI and proposed general methods for achieving safety.

2.2. Studies That Indirectly Addressing NFRs of AI

In the previous section, we discussed studies that directly address the NFRs of AI. There are other studies addressing the NFRs of AI. For example, XAI and adversarial training are closely related to transparency and safety. In this section, we discuss studies regarding XAI and adversarial training and how they indirectly address the NFRs of AI.

2.2.1. Explainable AI (XAI)

Although the main purpose of XAI is not to verify AI NFRs, there are some studies explaining how transparent learning and result derivation processes in AI can affect some NFRs. Due to the black-box characteristics of emerging deep neural networks, derived results from such models cannot be trusted without proper explanation. In particular, it is particularly important to solve these problems in fields where results have a large impact on humans, such as autonomous driving and medical diagnosis. To solve this problem, XAI applied to enable humans to look into the AI learning process. There are well-known studies which defined the concept of XAI and introduced the methodology used [14,15,20,26,27].

2.2.2. Adversarial Examples and Attacks

The main purpose of adversarial examples and attacks are not verifying AI NFRs. However, in [28], it was noted that

> "a good response to adversarial examples is an important safety issue in AI".

Therefore, adversarial examples can be a crucial part of the safety of AI. Many studies have shown that the characteristics of AI learning algorithms are vulnerable to adversarial examples, which behave as adversarial examples regardless of the structure and training data of the models [29,30].

Goodfellow et al. [30] proposed a fast gradient sign method (FGSM) that uses gradients from the learning process to generate adversarial examples. We chose FGSM as our primary adversarial attack method in our experiment.

Another study [31] demonstrated that data from the real world can serve as adversarial examples for models trained in a "laboratory" setting. In their experiment, they made some pictures with a smartphone camera and applied them to an established dataset, resulting in erroneous results.

Papernot et al. [32] proposed a method for adversarial attacks in a black box environment, with no information regarding the network or the training dataset, without knowing the structure of the neural network and the properties of the training dataset. The study shows the possibility of adversarial attacks in a production environment.

The authors in [33] summarized and classified various methods for generating adversarial examples. Although we have not used all of the methods in our experiment, our experiment shows a general tendency rather than a method-by-method comparison.

Several studies [38–40] proposed methods to utilize adversarial training while training classification models. While these methods definitely improve the training process, they redefine performance metrics regarding adversarial examples, whereas our proposed process lets a human supervisor set a balance between multiple requirements, including performance.

These studies show that adversarial attacks can be threats to certain NFRs. Therefore, we conducted experiments to show NFR verification can be done by appropriate use of adversarial examples.

3. Overview of Experiments

In this section, we define safety and present an overview of experiments to prove our RQs.

3.1. Definition of Safety

AI must ensure minimum safety regardless of its purpose. In a field where AI is applied, the actual input data may not exist in the dataset used for training. Additionally, AI suffers from multiple threats in real-world applications, such as adversarial attacks (adversarial examples).

Examples of adversarial attacks are shown in Figure 1. Both signs look the same in human eyes. Nevertheless, the right sign is slightly skewed with an adversarial attack and is classified as a different sign from the left one. If only the left sign is used in a training autonomous driving model and the right sign is given as real world input, it can cause fatal safety problems. Therefore, it is important to consider adversarial attacks while training models to ensure safety. Therefore, we define safety as follows:

> "a measure of whether a model responds appropriately to data with untrained features or to data that has been adversarially attacked to obtain incorrect results from the model".

Practically, it is difficult to collect data exhaustively with consideration of every situation. In addition, even if they are able to be collected, a lot of resources are required to process and train them. Therefore, we need to prioritize data requirements.

Figure 1. Examples of adversarial examples.

3.2. Overview of Our Experiments

We assume that adversarial examples are given the highest priority and are the data that should affect model behavior. This is because adversarial examples are data designed to cause problems or confuse the results of the model. Figure 1 shows an adversarial example of traffic signs for autonomous driving. As shown in the figure, traffic signs with certain natural conditions (sunlight, fog, etc.) might lead to erroneous decisions for the driving system. Creating and configuring datasets can also cause other problems. Generating corresponding data for every possible condition is difficult because there are too many possibilities in the environment in which the model is being applied. For these reasons, adversarial training can be a way to ensure AI safety, which has been proven through experiments. If adversarial training allows a model to better respond to

adversarial examples to improve safety performance, it can be used to verify whether safety is guaranteed in advance by preparing adversarial examples before training and mixing them with the training data.

To prove this, we first prepared different kinds of datasets and used them to generate adversarial datasets with different strengths of artificial transform. We define it as the degree of transformation ϵ. Several degrees of ϵ were prepared to find meaningful ϵ for generating adversarial examples. If ϵ is too small, transformed data will have little difference from the original data. Therefore, it can not deliver meaningful change. If ϵ is too large, there are too many perturbations in adversarial examples for humans to find the differences. Thus, it cannot be used as a meaningful adversarial example either. To maintain consistency, the size of the training dataset is fixed regardless ϵ to control the effect of the size of the training dataset over performance.

Furthermore, to find out at what ratio adversarial examples should be included for safety verification while preparing data, we prepared multiple datasets by setting different mixing ratios of the original data and the adversarial examples. Because the former study shows that adversarial examples tend to decrease [41], we intend to find a balance between safety and accuracy. In order to maintain the accuracy of the model learned with the original dataset with limited resources, we populated the original data with adversarial examples at an appropriate ratio while preserving the size of the training dataset.

Accuracy represents how well a model classifies its original validation data, whereas safety represents how well a model classifies its adversarial validation data, which is deemed threatening to a model by the definition given above. With the belief that results may differ according to the characteristics of the model's structure, we prepared learning models with different structures. Detailed information regarding the dataset, adversarial attack method, accuracy, safety, models, and experimental methods is discussed in Section 4.

4. Methodology

We conducted experiments under different conditions to answer formulated RQs. To control the effects of model structure and data characteristics, we prepared three datasets and three different models. Section 4.1 introduces datasets and models used in our experiment and the reason they were selected. Section 4.2 presents a brief introduction to the FGSM algorithm used for generating adversarial examples and the reason it was selected. Section 4.3 describes the detailed experimental processes designed in Section 3.

4.1. Datasets and Models

4.1.1. Datasets

For the experiments, three datasets, including CIFAR-10, CIFAR-100 [42], and the German traffic sign recognition benchmark (GTSRB) [43], were used. CIFAR-10 comprises 60,000 images in 10 classes, each containing 6000 images with three channels and 32×32 in size. We used 5000 images per class for training and the remaining were used for evaluation. Figure 2 shows examples from CIFAR datasets.

CIFAR-100 comprises 60,000 images in 100 classes, each of which contains 600 images with three channels and a size of 32×32. We used 500 images per class for training, and the remaining were used for evaluation. CIFAR-10 and CIFAR-100 were used for the experiment to answer RQ1, which corresponds to the most basic hypothesis established herein. There is a difference between CIFAR-10 and CIFAR-100. CIFAR-10 has a large amount of data for a small number of classes, which means it is likely for the model to successfully learn features for each class. Conversely, CIFAR-100 has a small amount of data for many classes, so the model is less likely to learn features for each class. If it is possible to identify the improvement of safety through adversarial training for datasets with these opposite characteristics, we can reliably expect that improvement of safety will occur regardless of data characteristic. Therefore, we used both CIFAR datasets for the experiment.

Figure 2. Examples from CIFAR datasets.

The German Traffic Sign Recognition Benchmark (GTSRB) is a German traffic sign dataset. It comprises more than 50,000 images of traffic signs under 43 classes. Figure 3 shows some examples from GTSRB data. The consistency of the number of data corresponding to each class is one of the differences between GTSRB and CIFAR datasets. By adding adversarial examples, the number of images per class can be different. Therefore, we selected GTSRB because we assume if it is possible to identify the safety improvement of a model through adversarial training for datasets in which the number of images is not uniform for each class, we can get valid results for other datasets as well. Another important reason for selecting GTSRB is that it is closely related to safety.

Figure 3. Examples from GTSRB dataset.

4.1.2. Models

Three models with different structures were applied to understand the effect of adversarial training through data augmentation on safety performance regardless of the network structure. LeNet [44] is a classic convolutional neural network-structured model developed in 1998 with a simple structure. LeNet has the simplest structure among the models used in the experiment here. It was used to understand the effect of a simpler model. ResNet18 [45] and VGG16 [46] are the most well-known and widely used models for classification, and their performance has also been proven through the ImageNet large-scale visual recognition challenge. Although VGG16 came second in the 2014 competition, it is more popular than the winning model, GoogLeNet, because of its simplicity and ease of use. ResNet18 won the 2015 competition with the addition of Residual blocks to VGGNet. We selected two models with well-known classification performance. For LeNet, we directly implemented the model by referring to the paper and using it.

PyTorch's default model was used for ResNet18 and VGG16. Fine-tuning for the models was intentionally omitted to show the side-effect of adversarial examples over accuracy. If we fine-tune these models more, there is a possibility they can reach some "golden" status regardless of how much we put into adversarial examples, effectively

reducing their side-effects. Because it is hard to measure and control resources used for fine-tuning, we decided to use models with their default conditions.

4.2. FGSM

We define the safety of AI as how well it classifies the adversarial examples mentioned in Section 3. To create adversarial examples for the experiment, we used FGSM [30] to generate adversarial examples. Because FGSM generates adversarial examples using gradients in the training process, it is necessary to train at least once with the dataset that we want to generate adversarial examples from. Therefore, we used LeNet, which has a relatively simple structure and does not require much time to learn in our paper.

Adversarial examples were generated using the gradient change that occurred during the training process. At this time, ϵ was used to determine the degree of adversarial attack by reflecting the slope value. We tried ϵ of 0.05 and 0.1, because ϵ above 0.1 makes the transformation too obvious to even human observation. Such examples are not able to serve as adversarial examples. Therefore, two types of ϵ are used to generate adversarial examples. In [30], adversarial examples were applied only to the validation data to confirm the effect of adversarial examples. However, because our goal is to confirm that adversarial examples are also useful to verify safety in AI, we applied adversarial examples to training data also.

As shown in Figure 4, adversarial examples were generated for CIFAR-10, CIFAR-100, and GTSRB data to be used for experiments. In this paper, the training and evaluation data to which the adversarial attack is applied are referred to as adversarial training data and *adversarial evaluation data*. The generated adversarial training data were extracted at random and used to populate the original training data at a specific ratio for training, whereas the adversarial evaluation data were used for experiments to evaluate the safety.

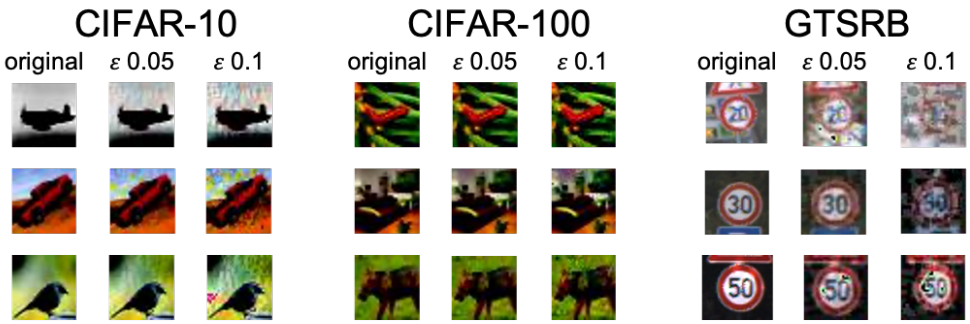

Figure 4. Examples of generated adversarial examples.

4.3. Methodology

We conducted the experiments using a PC with Ubuntu 18. 04 OS, 4. 20 GHz Intel (R) i7-7700K CPU, 64 GB RAM, and NVIDIA Titan XP 12GB GPU. Our code can be found at a github repository (https://github.com/JongdaeHan/Adversarial_validation, accessed on 5 June 2022).

The models used in the experiment have hyperparameters set to a batch size of 64, a learning rate of 0.001, a momentum of 0.9, CrossEntropy as loss function, and SGD as optimization function. As described above, these models are not fine-tuned to make the side-effect of adversarial training visible.

Figure 5 shows the experimental procedure.

The experiment is composed of two stages. The first step is data preparation. We generated the adversarial dataset by applying the FGSM algorithm to the original dataset. The adversarial examples were applied to both evaluation and training data. We populated the training data by randomly extracting the data from the original training data and the adversarial training data according to the specified ratio while preserving the original total number of training data. The mixing ratio is varied from 10:0 to 6:4 (Figure 5 (1–2)).

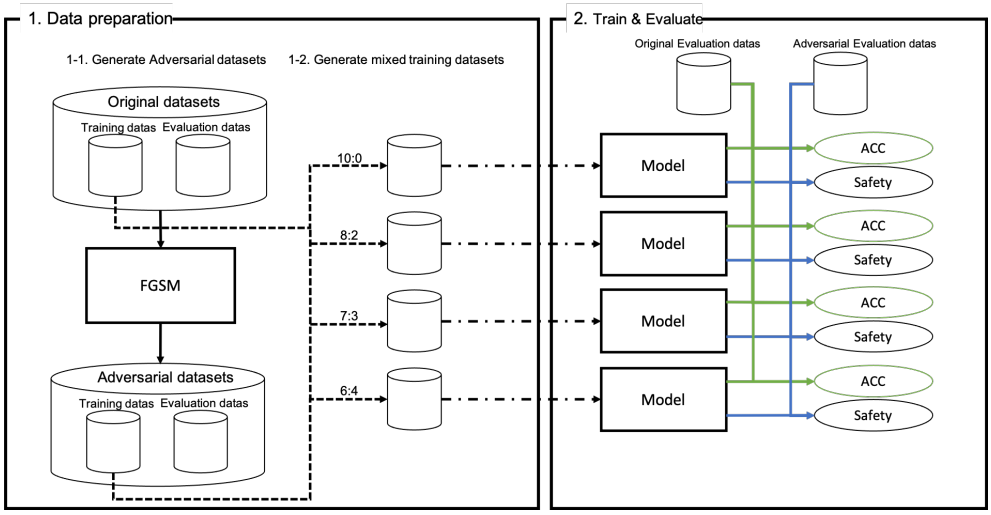

Figure 5. Brief structure of our experiments.

The second step involves training and evaluating the models. Using populated training data for the model set with the aforementioned hyperparameters, the training proceeded to the epoch in which the accuracy measured during the training process exceeded 90%. If the training accuracy does not reach 90% even after 30 epochs, training is stopped and the model is stored without more training. Although the models we used have far more capability by being trained for more epochs, we limit their capability to showing the effect of adversarial training drastically. Trained models are evaluated with original (not populated with adversarial examples) validation data and adversarial evaluation data for accuracy and safety, respectively.

5. Results and Discussion

We hypothesized that adversarial training could significantly improve safety while having a minimal impact on performance, and we designed experiments to test our hypotheses. In this section, we summarize and discuss the results of our experiments.

In our experiment, a dataset without adversarial examples is designated as *orig*. The training data and evaluation data in *orig* are designated as *orig_train* and *orig_eval*, respectively. A dataset with adversarial examples, is designated as *adv*, and the training data and evaluation data in *adv* are designated as *adv_train* and *adv_eval*, respectively. These designations refer to each dataset and the amount of data contained in each dataset. The *N-model* is a model trained with populated training data containing N% of adversarial examples in the training data. For example, a model trained without any adversarial training data is called a "0-model", and a model trained with populated training data containing 20% of adversarial training data is called a "20-model". After *N-model* is evaluated with evaluation data, the number of successfully classified cases is designated by adding *correct* in front of each evaluation data. For example, for *orig_eval* evaluation data, the number of successfully classified data is referred to as *correct_orig_eval*, whereas for *adv_eval* evaluation data, the number of successfully classified data is referred to as *correct_adv_eval*.

We used accuracy and safety to evaluate the experimental results. Accuracy is a measure of how well the trained model classifies *orig_eval* and is expressed as:

$$acc(\%) = correct\_orig\_eval / orig\_eval \times 100 \quad (1)$$

Safety is a measure of how well the trained model classifies *adv_eval*, and it can be expressed as follows:

$$safety(\%) = correct\_adv\_eval / adv\_eval \times 100 \quad (2)$$

Although the same equation is applied, we distinguish between the two parameters depending on whether they are in *orig_eval* or *adv_eval*.

5.1. Results from Experiments with CIFAR-10

Table 1 lists the CIFAR-10 results according to ϵ for each model.

Table 1. Results from experiments of CIFAR-10 with ϵ.

Population Ratio	Matrix	$\epsilon = 0.05$			$\epsilon = 0.1$		
		LeNet	**ResNet18**	**VGG16**	**LeNet**	**ResNet18**	**VGG16**
10:0	Acc	63	53	78	63	53	78
	Safety	18	33	33	11	27	27
8:2	Acc	60	51	73	60	50	74
	Safety	45	35	56	38	31	53
7:3	Acc	58	49	72	57	49	69
	Safety	45	36	56	41	34	54
6:4	Acc	58	48	70	58	47	70
	Safety	47	36	56	45	34	56

The results for the evaluation using adversarial examples produced with ϵ of 0.05 show significant difference between accuracy and safety. In the case of a 0-model, the difference is 45% for LeNet, 20% for ResNet18, and 45% for VGG16. These results mean the 0-model has a severe weakness against adversarial attack. The 0-model and 20-model showed a minor decrease in accuracy by 3% (LeNet), 2% (ResNet18), and 5% (VGG16) for each model, while significantly increasing safety by 27% (LeNet), 2% (ResNet18), and 23% (VGG16). Although there is a difference between the 30-model and the 40-model, the accuracy slightly decreased compared with that of the 0-model, and the safety was significantly improved. Compared to the 20-model, 30-model, and 40-model, lower accuracy and higher safety were observed in the model with a higher proportion of adversarial examples, but the differences were not that significant.

In the results for the adversarial examples produced with ϵ of 0.1, as in ϵ of 0.05, the accuracy of the 20-, 30-, and 40-models was slightly reduced when compared with the 0-model, but the safety was significantly improved.

These results are shown in Figures 6 and 7. Figure 6 confirms the results for *adv_eval* with ϵ of 0.05. As shown in Table 1, when adversarial examples were populated with a higher rate, the accuracy was slightly decreased while the safety was significantly improved.

Figure 7 shows the results for *adv_eval* made with ϵ of 0.1. Similar to Figure 6, the accuracy and safety change as the ratio of populated adversarial examples increases.

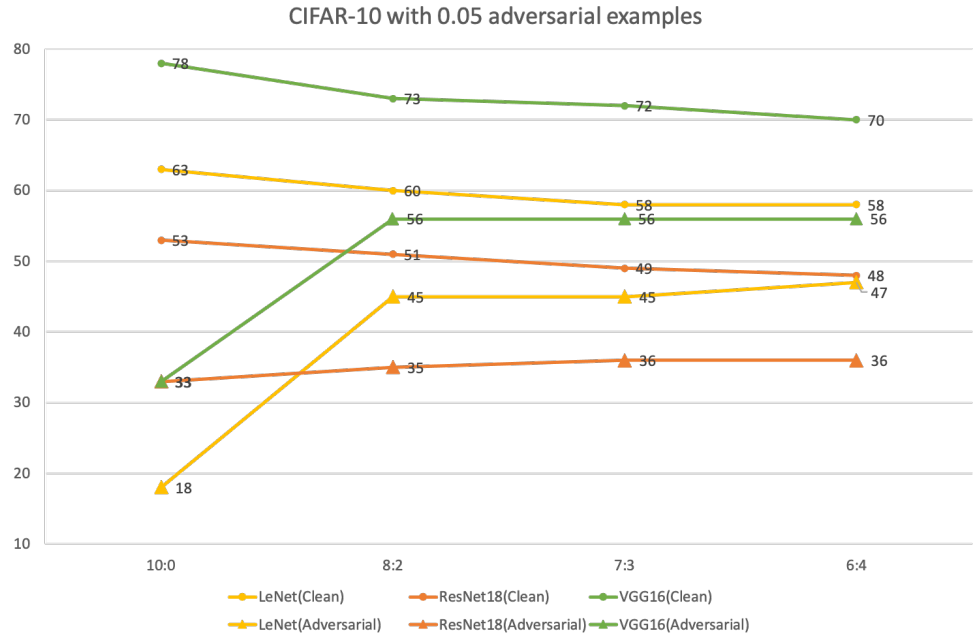

Figure 6. Graphs for experiment with CIFAR-10 using ϵ of 0. 05.

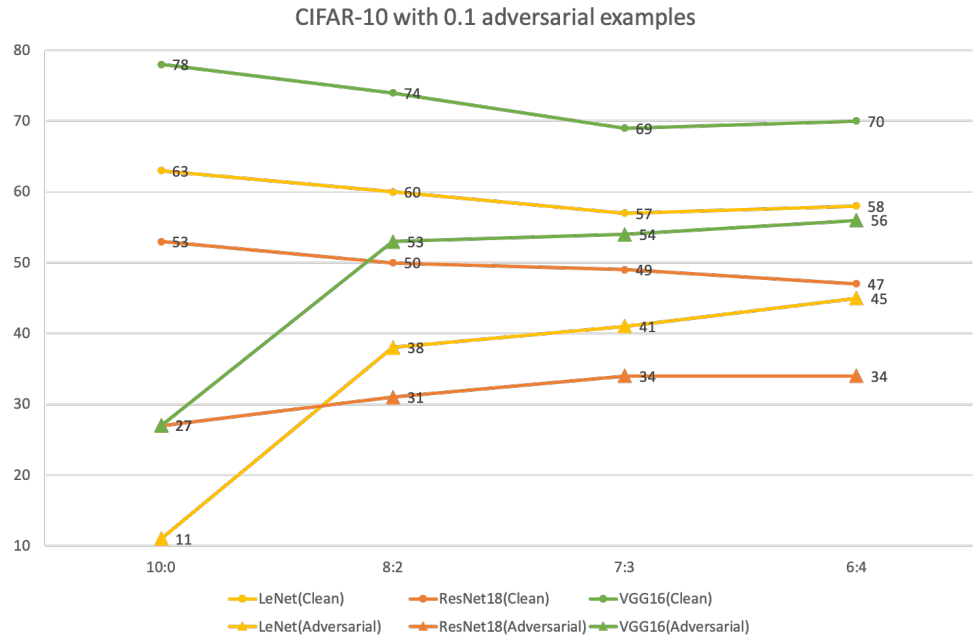

Figure 7. Graphs for experiment with CIFAR-10 using ϵ of 0.1.

5.2. Results from Experiments with CIFAR-100

Table 2 lists the CIFAR-100 results according to ϵ for each model.

As shown above, accuracy and safety for CIFAR-100 are much lower than those for CIFAR-10 because CIFAR-100 has more classes than CIFAR-10 but less data corresponding to the classes. Since the amount of data for training was small, the models were not sufficiently trained within the 30 epochs-limit, unlike in other datasets. Nevertheless, as in the experimental results of CIFAR-10, as the population ratio of adversarial examples in the training data increased, the accuracy decreased slightly while the safety improved significantly. This is confirmed by the results of 0-model and 20-model, where the range of performance change in accuracy and safety can be observed the most. The decrease in accuracy and improvement in safety were the same in ResNet18, which shows the

lowest accuracy and safety due to insufficient training. Also, the improvement in safety was larger than the decrease in accuracy in LeNet and VGG16, which were well trained when compared with ResNet18. Similar to the CIFAR-10 results, even in a situation where training is insufficient, safety could be improved with a small impact on accuracy through adversarial training.

Table 2. Results from experiments of CIFAR-100 with ϵ.

Population Ratio	Matrix	$\epsilon = 0.05$			$\epsilon = 0.1$		
		LeNet	**ResNet18**	**VGG16**	**LeNet**	**ResNet18**	**VGG16**
10:0	Acc	27	19	40	27	19	40
	Safety	15	7	10	4	6	7
8:2	Acc	26	17	33	25	17	35
	Safety	15	9	20	12	8	21
7:3	Acc	26	17	33	21	16	29
	Safety	15	10	22	12	9	20
6:4	Acc	24	15	30	22	16	26
	Safety	18	10	22	15	10	20

These results are depicted in Figures 8 and 9. Figure 8 shows the results for the CIFAR-100 *adv_eval* with ϵ of 0.05. In some models, the improvement in safety was smaller than the decrease in accuracy because the models were not properly trained. However, even without proper training, safety increased by 3%, and accuracy decreased by 2% for the 30- and 40-models of LeNet, whereas safety increased by 1% and accuracy remained unchanged in the 20- and 30-models of ResNet18.

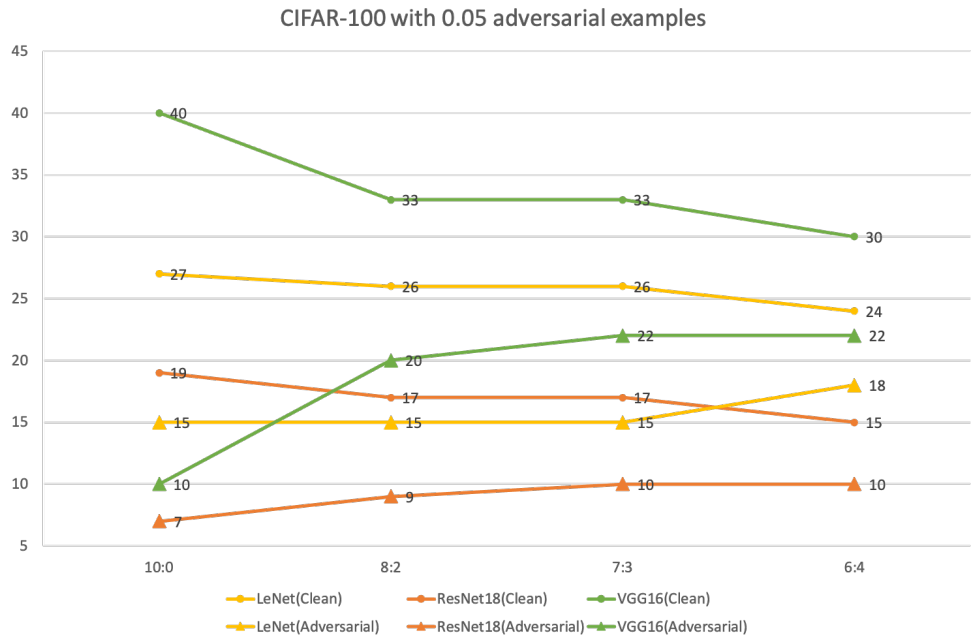

Figure 8. Graphs of CIFAR-100 with ϵ of 0.05.

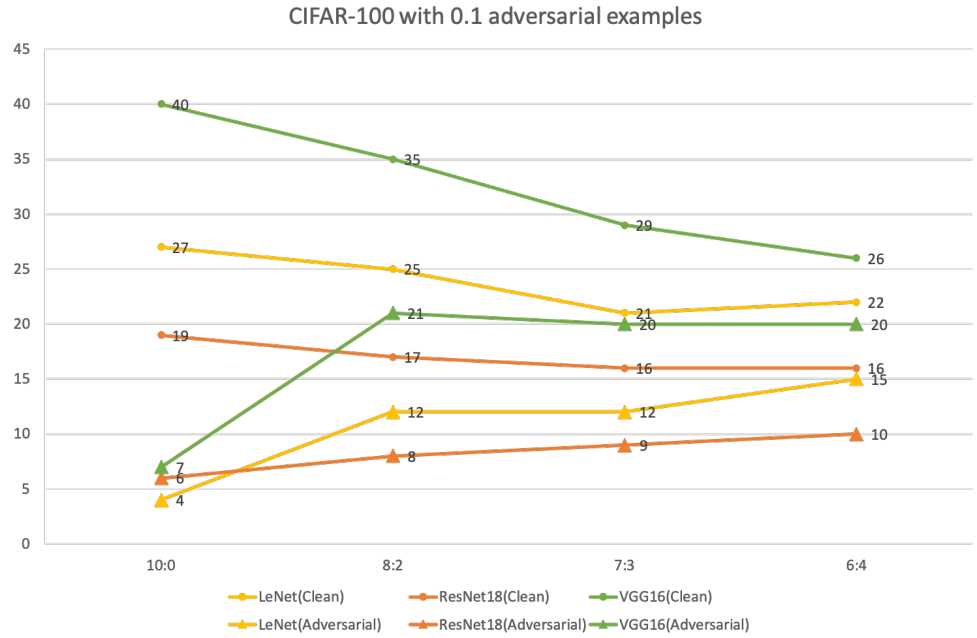

Figure 9. Graphs of CIFAR-100 with ϵ of 0.1.

Figure 9 depicts the CIFAR-100 *adv_eval* results with ϵ of 0.1. In ResNet18, which was the most insufficiently trained, the decrease in accuracy and improvement in safety is the same. In LeNet and VGG16, the improvement in safety was generally greater than the decrease in accuracy. CIFAR-100 results show that adversarial training can significantly improve safety with little impact on accuracy in most, if not all, cases, even for models that are not properly trained.

5.3. GTSRB Results

Table 3 shows the GTSRB results with ϵ for each model. The 0-model result shows that GTSRB had higher accuracy than the previous dataset's 0-model, but with lower safety. The accuracy and safety significantly differ by 68% (LeNet), 38% (ResNet18), and 53% (VGG16). This difference shows GTSRB has a significant weakness against adversarial attack.

Table 3. Results of GTSRB with ϵ.

Population Ratio	Matrix	$\epsilon = 0.05$			$\epsilon = 0.1$		
		LeNet	**ResNet18**	**VGG16**	**LeNet**	**ResNet18**	**VGG16**
10:0	Acc	85	71	93	85	71	93
	Safety	17	33	40	17	33	40
8:2	Acc	76	63	90	73	60	85
	Safety	57	49	75	58	42	70
7:3	Acc	75	60	86	70	58	81
	Safety	63	50	76	65	45	72
6:4	Acc	75	58	87	65	54	77
	Safety	66	52	79	65	46	73

We compared the GTSRB 0- and 20-models and discovered that GTSRB benefits the most from adversarial training. Calculating the average of differences in accuracy and safety between 0- and 20-model for each network, in the case of CIFAR-10, the accuracy of LeNet and ResNet18 decreased by an average of 3% and 2.5%, and the safety increased by

an average of 27% and 35%, respectively. The accuracy of VGG16 decreased by an average of 4.5%, and the safety increased by an average of 24.5%. In the case of CIFAR-100, the accuracy of LeNet, ResNet18, and VGG16 decreased by an average of 1.5%, 2%, and 6%, and their safety increased by an average of 4%, 2%, and 12%, respectively. In the case of GTSRB, the accuracy of the three models decreased by an average of 10.5%, 9.5%, and 5.5%, and their safety increased by an average of 39.5%, 12.5%, and 32.5%, respectively.

These results are depicted in Figures 10 and 11. Figure 10 shows the GTSRB *adv_eval* results obtained with ϵ of 0.05. In most models, except ResNet18, adversarial training could significantly improve safety with a slight decrease in accuracy.

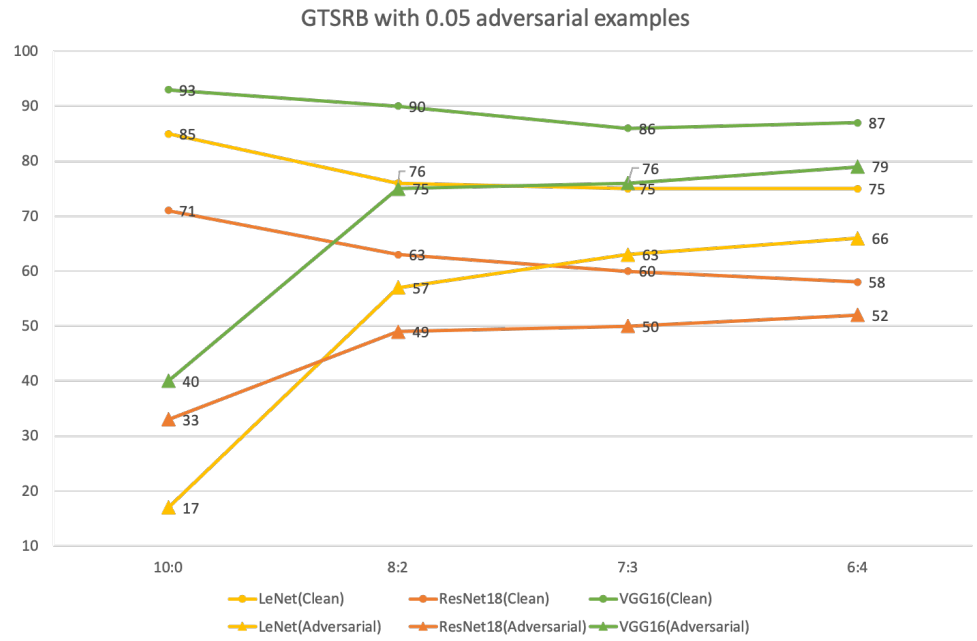

Figure 10. Graphs of GTSRB with ϵ of 0.05.

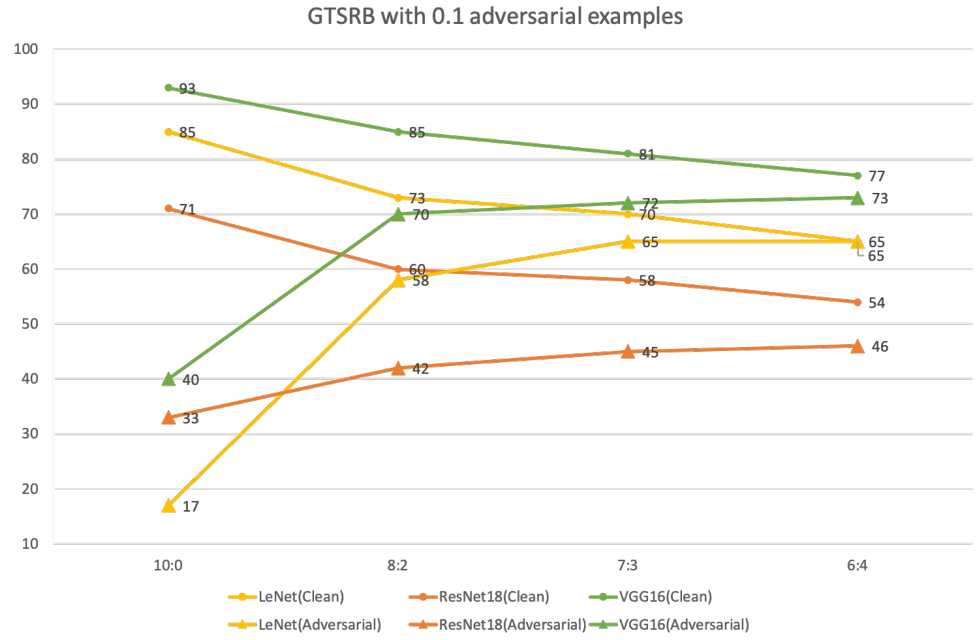

Figure 11. Graphs of GTSRB with ϵ of 0.1.

Figure 11 shows the GTSRB *adv_eval* results with ϵ of 0.1. In both Figures 10 and 11 as with the CIFAR datasets, the improvement in safety is much larger than the decrease in accuracy, and there is a slight difference among models with adversarial examples.

5.4. Discussion

The experimental results are discussed according to three different datasets and models and various experimental conditions, and the answers to the RQs are provided as follows. For all datasets, the safety of a 0-model is largely different from the accuracy. This confirms that the adversarial examples were properly generated and that models are vulnerable to adversarial attacks.

RQ1 is a question of whether adversarial training affects the evaluation of adversarial examples. In experiments using CIFAR datasets, the safety of the 20-, 30-, and 40-models, including the adversarial examples, was significantly improved when compared with that of the model without adversarial examples. This confirms that adversarial training can improve the safety performance of the classification model. However, without sufficient training, the improvement in results from CIFAR-100 is limited but the tendency could still be observed in a similar way.

RQ2 is an extension of RQ1. It asks whether the dataset used in the experiment can benefit from adversarial training even if it is a dataset related to safety. Although it varies depending on the size and structure of data, the safety of GTSRB is more than the CIFAR datasets. At the same time, as in CIFAR datasets, the degree of improvement in safety is greater than the degree of decrease in accuracy when adversarial training was performed. This confirms that adversarial training can improve safety with a small effect on accuracy, even in safety-related datasets. The result shows the benefit from adversarial training can vary depending on the characteristics of the data.

RQ3 asks whether adversarial training affects safety regardless of the structure of the model. To prove this, models with three different structures were applied in the experiment. The results from adversarial training show that safety was improved for all models, although the benefit might differ. Although some models used in the experiment may have a greater loss in accuracy when compared with the safety improvements, the overall experimental results show that they can improve safety performance without largely impacting accuracy regardless of the model structure.

RQ4 is a question of what ratio would be the best performance to mix adversarial examples with original data to form a populated training dataset. This depends on the dataset used and the priority between accuracy and safety. Experiments showed that it is essential to mix adversarial examples from the training data, at least 20% when constructing the training dataset to verify its safety. This is because the greatest improvement in safety could be obtained compared to the 0-model when the 20-model was used in all experimental environments. Even when adversarial training data was included at a higher percentage, the degree of safety improvement was small. Thus, it can be considered as the minimum requirement to configure 20% of the total training dataset as adversarial examples. If accuracy has a higher priority, the proportion of adversarial examples can be reduced and vice versa. Generally speaking, the "best proportion" of adversarial examples might differ depending on the characteristics of data, requirement priority, number of examples, and so on. Therefore, there should be careful assessment and verification of AI models before deploying them.

Additionally, the shape of the graphs shows the same tendency for all datasets and ϵ values. This shows that regardless of the proportion of the adversarial example, adversarial training can improve safety.

In summary, adversarial training can improve the safety of classification models regardless of the structure of the model, and is applicable to safety-related datasets. To verify safety, it is recommended that at least 20% of the training data comprise adversarial examples, and safety can be verified by adjusting the population ratio of adversarial examples according to the priority between accuracy and safety. On the basis of these

results, we propose a new classification model for the training process that reflects factors that can verify safety.

As shown in Figure 12, the performance, one of the major AI NFRs, has been verified by training the model using the original dataset and evaluating the trained model. If the performance does not reach the standard, the same process is repeated until the target performance is achieved through processes such as modifying the structure of the model, tuning hyperparameters, or modifying the dataset. Due to the black-box property of deep neural networks, it is hard to guarantee safety will remain intact during such a process.

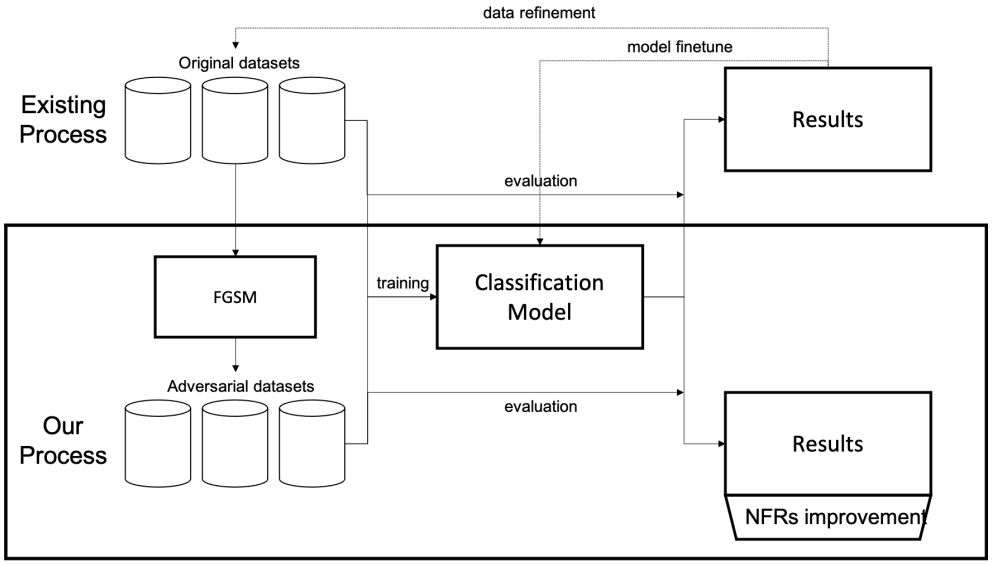

Figure 12. Comparison between existing processes and our process.

We propose a process regarding performance and safety enhancement here. First of all, an adversarial dataset is created using the original dataset. The ϵ must be decided under human supervision. Then, using the generated adversarial dataset, a populated training dataset is constructed with a suggested population ratio of 20% and the model is trained with the dataset. The trained model is evaluated for accuracy and safety through the original and adversarial evaluation datasets, respectively. If safety is not achieved, add more adversarial examples from the pre-generated dataset and repeat the process. With the proposed method, safety can be used to obtain additional information on the coordination of data by comparing priorities between accuracy and safety. If accuracy is more important, the population ratio of the adversarial examples in the training dataset can be decreased, and vice versa. This does not provide information regarding the model structure or hyperparameter finetuning, but provides additional information regarding what augmentation should be applied to the minimal dataset.

6. Conclusions

Through experiments, we found that adversarial training can improve the safety performance of models with little impact on accuracy. This means that generating adversarial examples using the dataset obtained before proceeding with learning and including the adversarial examples in the training dataset can be applied to verify the safety of AI. Furthermore, we propose a method to achieve a balance between accuracy and safety for the datasets used in experiments by generating a populated dataset of different ratios. Using the proposed method, a balance between accuracy and safety can be achieved by performing the experiment several times according to the important requirements and adjusting the population ratio of adversarial examples in the training dataset. Thus, the required resources can be reduced, at least for the training process. Moreover, the cost of training models that include verifying other NFRs can be reduced by adjusting the ratio

of inclusion of adversarial examples for higher-value requirements through accuracy and safety after one training and evaluation.

In future studies, we will verify the validity of the proposed process for more NFRs, like privacy or fairness. Different model structures and various adversarial attack methods can also be applied to validate our process.

Author Contributions: Conceptualization, J.H.; methodology, J.H.; software, H.K.; validation, J.H. and H.K.; formal analysis, J.H. and H.K.; resources, J.H.; data curation, H.K.; writing—original draft preparation, H.K.; writing—review and editing, J.H.; visualization, H.K.; supervision, J.H.; project administration, J.H.; funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data used in this study can be acquired by contacting authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bird, S.; Dudík, M.; Edgar, R.; Horn, B.; Lutz, R.; Milan, V.; Sameki, M.; Wallach, H.; Walker, K. *Fairlearn: A Toolkit for Assessing and Improving Fairness in AI*; Microsoft Tech Report; MSR-TR-2020-32; 2020; pp. 142–149.
2. Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; Zemel, R. Fairness through awareness. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, Cambridge, MA, USA, 8–10 January 2012; pp. 214–226.
3. Feldmen, M.; Friedler, S.A.; Moeller, J.; Scheidegger, C.; Venkatasubramanian, S. Certifying and removing disparate impact. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 259–268.
4. Tramer, F.; Atildakis, V.; Geambasu, R.; Hsu, D.; Hubaux, J.P.; Humbert, M.; Lin, H. Fairtest: Discovering unwarranted associations in data-driven applications. In Proceedings of the IEEE European Symposium on Security and Privacy, Paris, France, 26–28 April 2017.
5. Zhang, J.; Harman, M. Ignorance and Prejudice. In Proceedings of the 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), Madrid, Spain, 22–30 May 2021.
6. Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; Dwork, C. Learning fair representations. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 325–333.
7. Mei, S.; Zhu, X. The security of latent dirichlet allocation. In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, San Diego, CA, USA, 9–12 May 2015; pp. 681–689.
8. Mei, S.; Zhu, X. Using machine teaching to identify optimal training-set attacks on machine learners. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
9. Barreno, M.; Nelson, B.; Joseph, A.D.; Tygar, J.D. The security of machine learning. *Mach. Learn.* **2010**, *2*, 121–148. [CrossRef]
10. Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; Mané, D. Concrete problems in AI safety. *arXiv* **2016**, arXiv:1606.06565.
11. Juric, M.; Sandic, A.; Brcic, M. AI safety: State of the field through quantitative lens. In Proceedings of the 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 28 September–2 October 2020; pp. 1254–1259.
12. Leike, J.; Martic, M.; Krakovna, V.; Ortega, P.A.; Everitt, T.; Lefrancq, A.; Legg, S. AI safety gridworlds. *arXiv* **2017**, arXiv:1711.09883.
13. Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; Lipson, H. Understanding neural networks through deep visualization. *arXiv* **2015**, arXiv:1506.06579.
14. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
15. Murdoch, W.J.; Singh, C.; Kumbier, K.; Abbasi-Asi, R.; Yu, B. Interpretable machine learning: Definitions, methods, and applications. *arXiv* **2019**, arXiv:1901.04592.
16. Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Zieba, K. End to end learning to self-driving cars. *arXiv* **2016**, arXiv:1604.07316.
17. Levinson, J.; Askel, J.; Becker, J.; Dolson, J.; Held, D.; Kammel, S.; Thrun, S. Towards fully autonomous driving: Systems and algorithms. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 June 2011; pp. 163–168.
18. Vieira, S.; Pinaya, W.H.; Mechelli, A. Using deep learning to investigate the neuroimaging correlates of psychiatric and neurologicla disorders: Methods and applications. *Neurosci. Biobehav. Rev.* **2017**, *74*, 58–75. [CrossRef]
19. Ramsundar, B.; Kearnes, S.; Riley, P.; Webster, D.; Konerding, D.; Pande, V. Massively multitask networks for drug discovery. *arXiv* **2015**, arXiv:1502.02072.

20. Holzinger, A.; Biemann, C.; Pattichis, C.S.; Kell, D.B. What do we need to build explainable AI systems for the medical domain? *arXiv* **2017**, arXiv:1712.09923.
21. Krause, J.; Pere, A.; Ng, K. Interacting with predictions: Visual inspection of black-box machine learning models. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, 7–12 May 2016; pp. 5686–5697.
22. Tan, J.; Ung, M.; Cheng, C.; Greence, C.S. Unsupervised feature contruction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders. In *Pacific Symposium on Biocomputing Co-Charis*; World Scientific: Singapore, 2014; pp. 132–143.
23. Pesapane, F.; Volonté, C.; Codari, M.; Sardanelli, F. Artificial intelligence as a medical device in radiology: Ethical and regulatory: Ethical and regulatory issues in Europe and the United States. *Insights Into Imaging* **2018**, *9*, 743–753. [CrossRef]
24. Miller, D.D.; Brown, E.W. Artificial intelligence in medical practice: The question to the answer? *Am. J. Med.* **2018**, *131*, 129–133. [CrossRef]
25. Fu, K.; Cheng, D.; Tu, Y.; Zhang, L. Credit card fraud detection using convolutional neural networks. In *International Conference on Neural Information Processing*; Springer: Cham, Switzerland, 2016; pp. 483–490.
26. Samek, W.; Wieg, T.; Müller, K.R. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv* **2017**, arXiv:1708.08296.
27. Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
28. Kurakin, A.; Goodfellow, I.; Bengio, S.; Dong, Y.; Liao, F.; Liang, M.; Abe, M. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*; Springer: Cham, Switzerland, 2018; pp. 195–231.
29. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
30. Goodfellow, I.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
31. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2016**, arXiv:1607.02533.
32. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017; pp. 506–519.
33. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial examples: Attacks and defenses for deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2805–2824. [CrossRef]
34. Aryal, K.; Gupta, M.; Abdelsalam, M. A Survey on Adversarial Attacks for Malware Analysis. *arXiv* **2021**, arXiv:2111.08223.
35. Kimmell, J.C.; Abdelsalam, M.; Gupta, M. Analyzing Machine Learning Approaches for Online Malware Detection in Cloud. In Proceedings of the 2021 IEEE International Conference on Smart Computing (SMARTCOMP), Irvine, CA, USA, 23–27 August 2021; pp. 189–196.
36. McDole, A.; Abdelsalam, M.; Gupta, M.; Mittal, S. Analyzing CNN Based Behavioural Malware Detection Techniques on Cloud IaaS. *Cloud Comput.* **2020**, *2020*, 12403.
37. Kimmel, J.C.; Mcdole, A.D.; Abdelsalam, M.; Gupta, M.; Sandhu, R. Recurrent Neural Networks Based Online Behavioural Malware Detection Techniques for Cloud Infrastructure. *IEEE Access* **2021**, *9*, 68066–68080. [CrossRef]
38. Poon, H.-K.; Yap, W.-S.; Tee, Y.-K.; Lee, W.-K.; Goi, B.-M. Hierarchical gated recurrent neural network with adversarial and virtual adversarial training on text classification. *Neural Netw.* **2019**, *119*, 299–312. [CrossRef]
39. Terzi, M.; Susto, G.A.; Chadhari, P. Directional adversarial training for cost sensitive deep learning classification applications. *Eng. Appl. Artif. Intell.* **2020**, *91*, 103550. [CrossRef]
40. Dong, X.; Zhu, Y.; Zhang, Y.; Fu, Z.; Xu, D.; Yang, S.; Melo, G. Leveraging adversarial training in self-learning for cross-lingual text classification. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 11–15 July 2020; pp. 1541–1544.
41. Ajunwa, I.; Friedler, S.; Scheidegeer, C.E.; Venkatasubramanian, S. Hiring by Algorithm: Predicting and Preventing Disparate Impack. 2016. Available online: http://tagteam.harvard.edu/hub_feeds/3180/feed_items/2163401 (accessed on 17 January 2022).
42. Krizhevskey, A.; Hinton, G. *Leawrning Multiple Layers of Features from Tiny Images*; University of Toronto: Toronto, ON, Canada, 2009.
43. Stallkamp, J.; Schlipsing, M.; Salmen, J.; Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Netw.* **2012**, *32*, 323–332. [CrossRef]
44. LeCun, Y.; Bottou, L.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1995**, *86*, 2278–2324. [CrossRef]
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
46. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.