AFRL-RY-WP-TP-2023-0074

EXPLORING THE REMOVAL OF BIT-PLANES FOR INCREASED ADVERSARIAL ROBUSTNESS (Preprint)

Alex Hildenbrandt, Ashley Diehl, and Christopher Menart
Decision Sciences Branch
Multi-Domain Sensing Autonomy Division

Robert Canady
Vanderbilt University

Melissa Robertson
Brigham Young University

Hannah Richards
Applied Research Solutions

MAY 2023
Final Report

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
May 2023	Journal Article Preprint	2 May 2023	2 May 2023

4. TITLE AND SUBTITLE

EXPLORING THE REMOVAL OF BIT-PLANES FOR INCREASED ADVERSARIAL ROBUSTNESS (Preprint)

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
N/A	N/A	N/A
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
N/A	N/A	N/A

6. AUTHOR(S)

Alex Hildenbrandt, Ashley Diehl, and Christopher Menart (AFRL/RYAT)
Robert Canady (Vanderbilt University)
Melissa Robertson (Brigham Young University)
Hannah Richards (Applied Research Solutions)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Decision Sciences Branch
Multi-Domain Sensing Autonomy Division
Air Force Research Laboratory, Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command, United States Air Forces

Vanderbilt University 110 21st Avenue South, Suite 110.
Nashville, TN 37240

Brigham Young University.
Provo, UT 84602

Applied Research Solutions
51 Plum St Ste 240
Beavercreek, OH, 45440

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'SREPORT NUMBER(S)
Air Force Research Laboratory, Sensors Directorate Wright-Patterson Air Force Base, OH 45433-7320 Air Force Materiel Command, United States Air Forces	AFRL/RYAT	AFRL-RY-WP-TP-2023-0074

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

PAO case number AFRL-2023-0277, Clearance Date 2 May 23. The U.S. Government is joint author of this work and has the right to use, modify, reproduce, release, perform, display, or disclose the work. Report contains color.

14. ABSTRACT

Machine Learning has been found to be a very valuable and powerful tool, that will almost certainly see an increase in use in the the future. However, it has also been found to have some vulnerabilities that could be exploited. This is concerning overall, but particularly important if the machine learning model is being used for safety-critical, but even for non-safety-critical applications. Many defenses have been worked on to combat this issue, but most defenses seem to be broken shortly after being proposed. These defenses are also typically very resource hungry as well as cause normal performance to drop. In this work, we present several ideas as to how to make robust models using fewer resources and without impacting clean performance. We explore several different combinations of defenses as well as metrics to determine the impact of the attacks on metrics besides just accuracy.

15. SUBJECT TERMS

machine learning, sensor fusion, adversarial machine learning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
Unclassified	Unclassified	Unclassified	SAR	16

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Alexandra Hildenbrandt	N/A

Exploring the Removal of Bit-planes for Increased Adversarial Robustness

Robert Canady[a], Melissa Robertson[b], Hannah Richards[c], Alex Hildenbrandt[d], Ashley Diehl[d], and Christopher Menart[d]

[a]Vanderbilt University, Nashville, TN, USA
[b]Brigham Young University, Provo, UT, USA
[c]Applied Research Solutions, Beavercreek, OH, USA
[d]Air Force Research Laboratory, Wright-Patterson AFB, OH, USA

ABSTRACT

Machine Learning has been found to be a very valuable and powerful tool, that will almost certainly see an increase in use in the the future. However, it has also been found to have some vulnerabilities that could be exploited. This is concerning overall, but particularly important if the machine learning model is being used for safety-critical, but even for non-safety-critical applications. Many defenses have been worked on to combat this issue, but most defenses seem to be broken shortly after being proposed. These defenses are also typically very resource hungry as well as cause normal performance to drop. In this work, we present several ideas as to how to make robust models using fewer resources and without impacting clean performance. We explore several different combinations of defenses as well as metrics to determine the impact of the attacks on metrics besides just accuracy.

Keywords: Machine Learning, Adversarial Machine Learning, Sensor Fusion

1. INTRODUCTION

Machine learning (ML) is being increasingly used in many different application domains because of how well it performs, oftentimes outperforming humans in tasks like image classification or text recognition. With how quickly they are able to process data and give insights, they are becoming more feasible to deploy in different situations.

ML models have a vast array of different application areas ranging from things that make everyday life easier, such as smart home devices and recommendation systems, to systems that protect the US from foreign or domestic threats and autonomous vehicles. Each of these with different levels of security and varying disruptive potential.

These same ML models however, have been shown to be vulnerable to slight perturbations to the input data. This is concerning for several reasons. ML is being deployed in safety-critical and mission-critical situations where a slight error could be disastrous. It could also potentially disrupt business by having false or adversarial data.

Several defenses have been proposed over the years to mitigate the impact of these adversarial examples, which are either proactive or reactive. The problem with most defense strategies is the amount of compute resources that are needed. Especially with adversarial training, the training time and the amount of resources used increases significantly. Also, adversarial training can cause a decrease in performance on clean, non-attacked data.

In this work we wanted to explore some solutions to the issues caused by Adversarial Machine Learning. More specifically, we wanted to explore less resource intensive approaches as well as ways to improve upon existing defense strategies. We describe some background and related works in Section 2. In Section 3, we then discuss our methods and ideas behind how we attempt to solve some of the problems laid out so far. We then end by going over and discussing the results in Section 2 and finish with our conclusions in Section 5.

Further author information: (Send correspondence to Robert Canady)
E-mail: robert.e.canady@vanderbilt.edu or robcanady@gmail.com

2. RELATED WORKS

2.1 Adversarial Machine Learning

Adversarial Machine Learning (AML) has been gaining increased attention. Possibly due to the prevalence of machine learning in everyday life as well as its integration into safety-critical processes. Most of the initial work in this area has been in the computer vision domain, and more specifically the image classification task. The field was kick started with the work done by Christian Szegedy and others.[1] This was then followed by the work[2] that introduced the idea of evasion attacks. Some of the notable attacks that followed included Fast Gradient Sign Method[3] (FGSM), Projected Gradient Descent[4] (PGD), and the Carlini and Wagner[5] (C&W). Each of these attacks have the same goal - perturb an image enough to cause misclassification by a machine learned model, but also keep the perturbation within a limit that is unnoticeable to a human observer. FGSM does this in one step, while PGD and others adopt an iterative approach to creating this perturbation.

These attacks typically use the model's architecture and the trained weights and biases to craft the perturbations. There are several different settings that are normally considered. The first and most popular would be a white box setting. In these, the attacker has full knowledge of the model architecture being used, as well as its trained weights and biases, training data, training parameters, etc. The next setting would be gray box where the attacker is aware of some information, like the model architecture but not the weights and biases. The last setting would then be black box where the attacker has no knowledge about the model or how it was trained. They only have access to whatever data they are trying to poison. AML attacks are also split by whether the attacks are carried out through software channels or whether the attack changed something in the physical world, i.e. placing tape on a stop sign.

When developing these attacks, there are several important hyperparameters that must be set. The first is the attack strength, or perturbation bound, ϵ. The next is the attack space which is denoted in terms of the L_p norm where $p = 0, 1, 2, \infty$. Then for the iterative attacks, we will have to choose how many iterations we want to go through to develop the attack. The perturbation bound, ϵ, controls how much any single pixel can be changed. For PGD attacks, the typical value is $8/255$. The most commonly used norm is the L_∞ norm or the max norm.

Due to the success of these attacks, there has also been a focus on developing defenses to the attacks. These defenses typically fall into two categories: reactive and proactive. An example of a reactive defense would be detecting that an image has been perturbed and then mitigate the effects in some way. A proactive defense would be adversarial training (AT), where the goal of training is to make the model robust to potential attacks by adding data to the training set that has been perturbed using AML techniques. While defenses have seen moderate success, they are plagued by several issues. One is that defensive adversarial techniques typically cause clean accuracy to decrease. Another is that each time a new, promising defense is proposed, an attack is developed that is able to break the defense.

Use of different types of neural networks, like a Bayesian Neural Network (BNN), have also been explored as a possible defense. The idea behind this is that a BNN uses probability distributions for its weights and biases rather than static floating point values. While this increases model size, it has been shown to be an effective defense.[6] This technique allows the model to estimate the uncertainty in their predictions. One of the benefits of this, is that unbalanced datasets tend to cause overfitting to the over-sampled class, and Bayesian NNs can possibly mitigate this problem

While most of the work has been done with image classifiers, there has been some recent work that looks at the adversarial robustness of object detection models, sensor fusion models, and natural language processing models.

2.2 Sensor Fusion

Sensor fusion is an approach where instead of only using one modality of data, i.e. RGB images, to make predictions, we also add another modality such as Synthetic Aperture Radar, Infrared, LiDAR, etc. The idea is that more data will increase the prediction performance. This makes sense due to the fact that IR sensors are able to see in all different weather conditions whereas RGB images are highly affected by weather conditions.

There have been several different approaches used for carrying out sensor fusion. The first is data-level fusion. This is where the raw data from different modalities are combined. For example, if you combined 3 channel RGB images with 1 channel Infrared Radar (IR) data, you would then just have a 4 channel image. Care just must be taken to make sure that the objects line up before fusing. The next kind of fusion would then be feature level fusion. In this approach, different models are trained for each data modality. The fusion occurs at the feature map layer of the network when the data is passed through the network. The decision at the output layer is then made from the fused feature maps. The last type of fusion would be decision level fusion. For this, the data is passed through each network that has been trained on different modalities and then the decision layer, or output layer, are fused and the predictions are made from that.

We mainly have focused on using decision-level fusion, so we will describe some of those approaches further. We tested several different decision-level fusion algorithms, namely: Naive-Bayes, Highest Probability, Borda Count, Generalized Chernoff, and Sandia Probabilistic Fusion.

1. *Naive-Bayes:* In this approach, we assume that the sensors are independent, after multiplying their probabilities together, the largest value is selected as the predicted target.

2. *Highest Probability:* This approach chooses the class with the highest probability from all EO and SAR probabilities.

3. *Borda Count:* For this approach, we rank the probabilities from 1 to n, with n being the highest probability. The class with the highest average rank between EO and SAR is then chosen as the predicted target.

4. *Generalized Chernoff:* This approach, we do not assume that the sensors are independent. Weights are applied to account for the correlation of their probability density functions. Further description can be found in the original report[7] or another work[8] that utilizes these techniques.

5. *Sandia Probabilistic:* This approach is essentially a goodness of fit hypothesis test which compares the distribution of prediction probabilities in the training set with those in the testing set. Further description can be found in the original report[9] or another work[8] that utilizes these techniques.

3. METHODOLOGY

In this section we will try to convey our reasoning behind some of the approaches we took. While adversarial training has seen the most success against AML attacks, we wanted to explore defenses that might not be quite as resource intensive. That is how we initially found the work using bit-planes and eventually built on top of that work.

3.1 Bit-planes

A bit-plane in the context of an image can be described as follows. Suppose we have an nxn 8-bit RGB image. Each pixel can be represented as an 8-bit binary number. Removing a bit-plane from the image is the process of nullifying the m least significant bits (LSB). The thought from the paper was that by removing these LSBs, we will be removing some of the extraneous information that could be a place where most of the perturbation exists. Since the goal of AML is to perturb an image within a certain bound, it makes sense that the attacks would focus on the background information rather than the object meant to be classified.

In the previously mentioned work, they trained models on a combination loss of bit-plane removed images as well as clean/untouched images. While this was shown to be an effective approach, we wanted to see if we could take it further and just train the models on bit-plane dropped images. This way the model will solely focus on the most important part of the image, which lies in the $8\text{-}m$ MSBs. In Figure 1, we have displayed images with no bit-planes removed to images with 7 bit-planes removed.

We also wanted to explore whether removing bit-planes at inference time had a positive effect on robustness. An example showing why this was explored is presented in Figure 2. In this example, we have shown the initial image as well as the initial image after being attacked by FGSM and PGD.

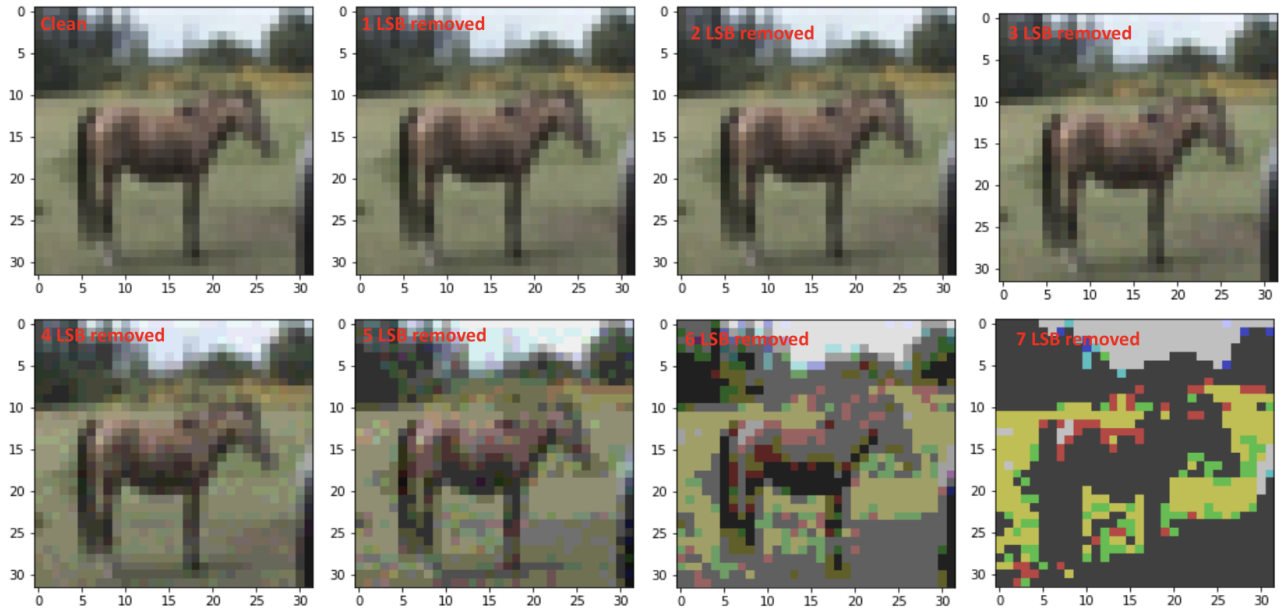

Figure 1. Examples of CIFAR10 Images with Bit-planes Removed

3.2 Combining Defenses

We also explored the idea of combining less resource intensive defenses, like bit-plane removal, with more resource intensive approaches like Adversarial Training or Bayesian Neural Networks. We hypothesized that combining different defenses would increase robustness further than just with a single defense.

3.2.1 Adversarial Training and Bit-plane Removal

Due to the fact that adversarial training has seen the most success at defending against AML, we wanted to see how it would perform when combined with bit-plane removal which is a less resource intensive defense approach.

3.2.2 Bayesian Neural Networks and Bit-plane Removal

Bayesian Neural Networks (BNNs) have seen moderate success in increasing adversarial robustness. A BNN uses a typically Gaussian probability distribution for the weights and biases rather than static floating point values. The idea is that it helps us estimate the uncertainty in predictions. This then will give us insights into how the model is learning and potential ways to create more robust models.

3.3 Sensor Fusion

The last defense approach we explored was sensor fusion. This was described in the previous section. We wanted to explore whether the same attacks were effective against fused models and also see whether the same defenses worked. In Figure 3, we show examples of EO and SAR images attacked by FGSM, PGD and CW. We hypothesized that fusing information from multiple sources would make it more difficult to fool the model rather than if it was just trained on one type of data. We also hypothesized that SAR models would be harder to impact due to the fact that SAR data has only 1-channel rather than a typical 3-channel RGB image. Also, the fact that SAR images are typically noisier in general, made us think that these would be harder to successfully attack.

4. EXPERIMENTAL SETTINGS AND RESULTS

To test our ideas, we focused on 2 datasets within the computer vision community. These were CIFAR10[10] and UNICORN.[11] The first two datasets allowed us to test our ideas on a single type of data while UNICORN has

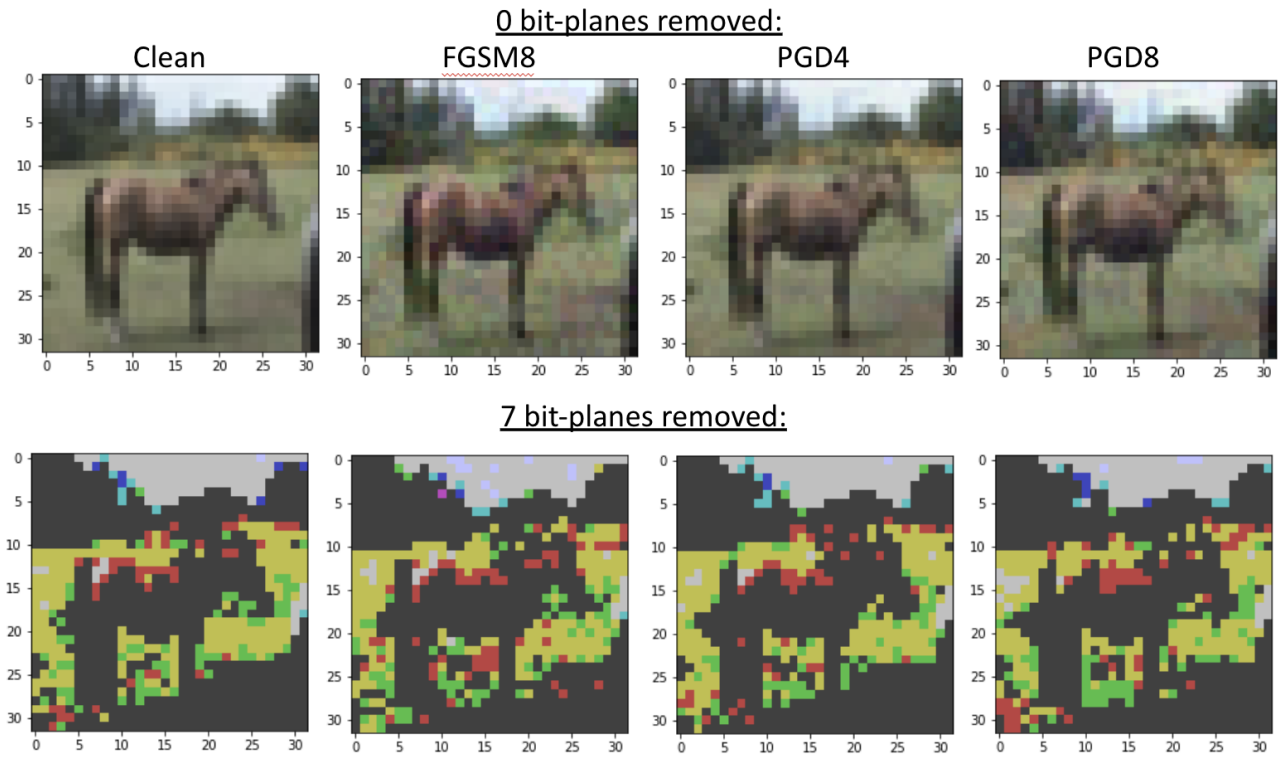

Figure 2. Examples of Attacked CIFAR10 Images with 0 and 7 Bit-planes Removed

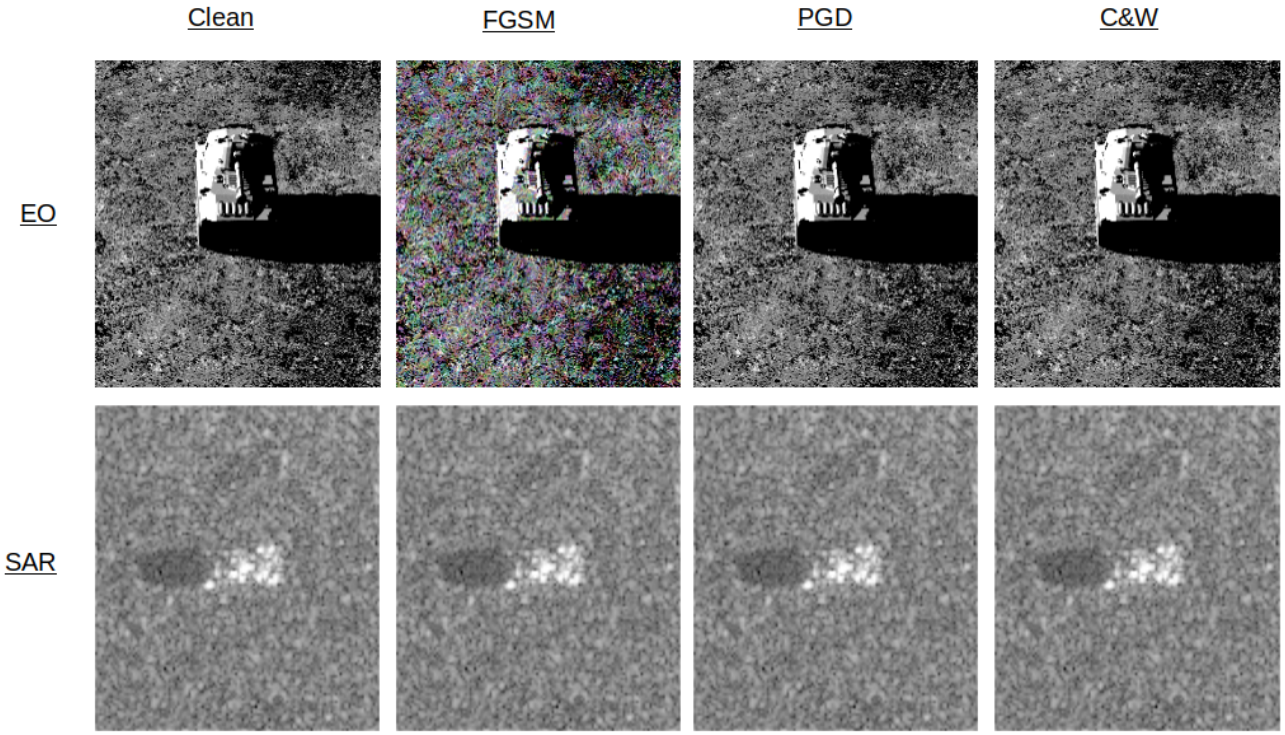

Figure 3. Example EO and SAR Images After Attack

Model Type	Bit-plane(s)	Clean Acc.	FGSM8 Acc.	PGD4 Acc.	PGD8 Acc.
Normal	0	90.81	32.35	12.08	4.03
MSB	1	**91.11**	29.06	8.84	1.99
	2	90.79	29.36	8.79	1.69
	3	90.94	29.05	10.94	4.30
	4	90.56	25.89	10.27	2.05
	5	89.57	24.31	14.09	2.52
	6	87.17	22.73	**19.59**	**4.43**
	7	79.54	14.66	14.90	1.24
Rand MSB	1,3,5,6,7	87.13	20.19	21.87	2.06
	1,2,3,4,5,6,7	86.64	18.67	19.63	1.61
	5,6,7	87.29	22.72	**23.00**	**2.76**
BPFC	1	87.36	19.71	16.21	1.07
	2	85.86	21.73	15.60	2.33
	3	87.86	28.91	23.11	4.80
	4	87.56	37.61	29.95	10.78
	5	86.01	**40.16**	**34.10**	**14.91**

Table 1. Initial Experimentation Results

Electro-optical (EO) and Synthetic Aperture Radar (SAR) data. EO data is most comparable to a 3-channel RGB image whereas SAR data consists of only 1-channel of data.

Throughout experimentation, we have mainly focused on a few architectures - ResNet, DenseNet. This selection gives us the most diversity in terms of model size and performance.

For attacks, we mainly focus on FGSM and PGD. Although, later on we add the CW attack into the evaluation. FGSM and PGD have several different hyperparameters that can be changed, most importantly, the number of iterations, the attack strength and the L_p norm with $p = 1, 2, \infty$. We also tested the attacks after modifying them using the insights from the Expectation over Tranform (EOT) paper. For the purpose of saving space, we only include the L_∞ norm attacks, but we conducted experiments using L_2 and $EOTL_\infty$. The results were basically mirroring each other.

4.1 Initial Experimentation

The initial experimentation into the idea of using bit-plane removal used the Densenet161 architecture. We also looked mainly at the CIFAR10 dataset.

4.1.1 No Bit-planes Removed at Inference

We display some of our initial results training with only bit-plane removed imagery and then compare those with models trained using the initial combination of clean loss and bit-plane removed loss. Our models will be signified using MSB, while the models from the paper that we trained ourselves are signified as BPFC. The number of bit-planes removed during training will also accompany each signifier. We trained our models with 0-7 bit-planes removed while the BPFC models only were trained on data with 1-5 bit-planes removed. We also trained several models with randomly chosen bit-planes removed. These results are presented in Table 1.

As can be seen, the MSB models actually outperformed the normal model and the BPFC models on clean data in most cases. There also appears to be a pattern between FGSM and the PGD attacks. The model's performed better on PGD as the number of bit-planes removed (up to 6) at training increased, while we saw the opposite trend for FGSM models. This is not observed for the BPFC models. With those, the more bit-planes removed (up to 5), the better the model performed against adversarial data, FGSM and PGD.

We also adversarially trained models in combination with bit-plane dropout. These results are presented in Table 2. We used the models we had already trained on bit-plane dropped imagery and finetuned them with adversarial examples for 25 epochs. We used the PGD attack with an attack strength of 8/255.

Model Type	Bit-plane(s)	Clean Acc.	FGSM8 Acc.	PGD4 Acc.	PGD8 Acc.
Normal	0	77.44	35.49	49.78	24.75
MSB	1	77.41	34.95	48.99	**24.96**
	3	77.45	34.91	49.49	24.01
	5	77.41	**36.06**	50.16.	24.56
	6	76.38	34.69	49.2	23.68
	7	76.55	35.05	49.00	24.33
Rand MSB	1,3,5,6,7	76.5	34.63	49.95	24.34
	1,2,3,4,5,6,7	76.76	35.02	49.98	24.66
	5,6,7	**77.56**	35.87	**51.33**	24.9

Table 2. Adversarial Training Initial Experimentation Results

4.1.2 Bit-planes Removed at Inference

For this part of the experimentation, we looked at the impact of removing bit-planes before sending images through for inference. Since the idea behind removing bit-planes was to remove some of the adversarial perturbations, we decided to look at its effect during inference time. We looked at the impact of this on clean images as well as adversarial images.

As can be seen in Figure 4, on clean/untouched data, the MSB models did not see much decrease in performance when bit-planes were removed at inference until 6 or 7 bit-planes had been removed. The normally trained model saw a sharp drop when just a single bit-plane was removed at inference. Once the images were attacked, we saw a different pattern. The models all performed similarly when no bit-planes had been removed, but the normally trained model saw a sharp increase in accuracy once just a single bit-plane had been dropped. The MSB models saw a more gradual increase in accuracy over the number of bit-planes removed. It took until around 6 or 7 bit-planes removed for the MSB models to outperform the normal model.

Figure 5 displays the results of removing bit-planes at inference for the combination of bit-plane removal and adversarial training models. We see very similar trends to the previous figure when it comes to clean data and clean data with bit-planes removed. There is a difference once we start to look at the results after attack. The MSB with adversarial training models do not see as sharp of a drop when attacked. They also perform similarly when bit-planes have been removed, until 6 or 7 bit-planes when they actually see an increase in performance. The model that was just adversarially trained, saw a decrease in performance when bit-planes were removed.

4.1.3 Initial Results Discussion

The results shown and discussed so far, show that bit-plane removal, just by itself is a very promising defense. When combined with adversarial training, it was shown to improve the effectiveness of the adversarial training.

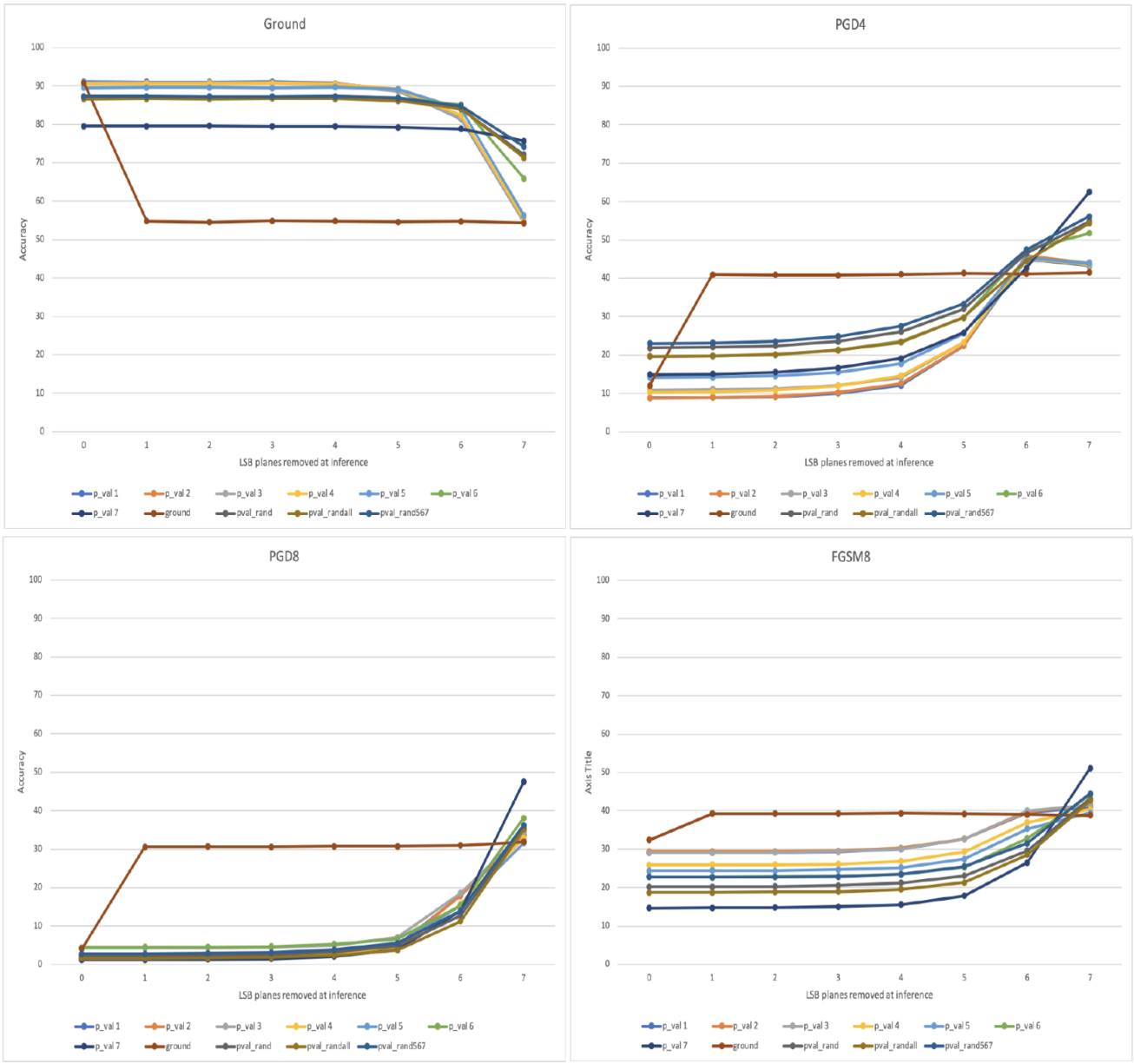

Figure 4. Results with Bit-planes Removed at Inference Time

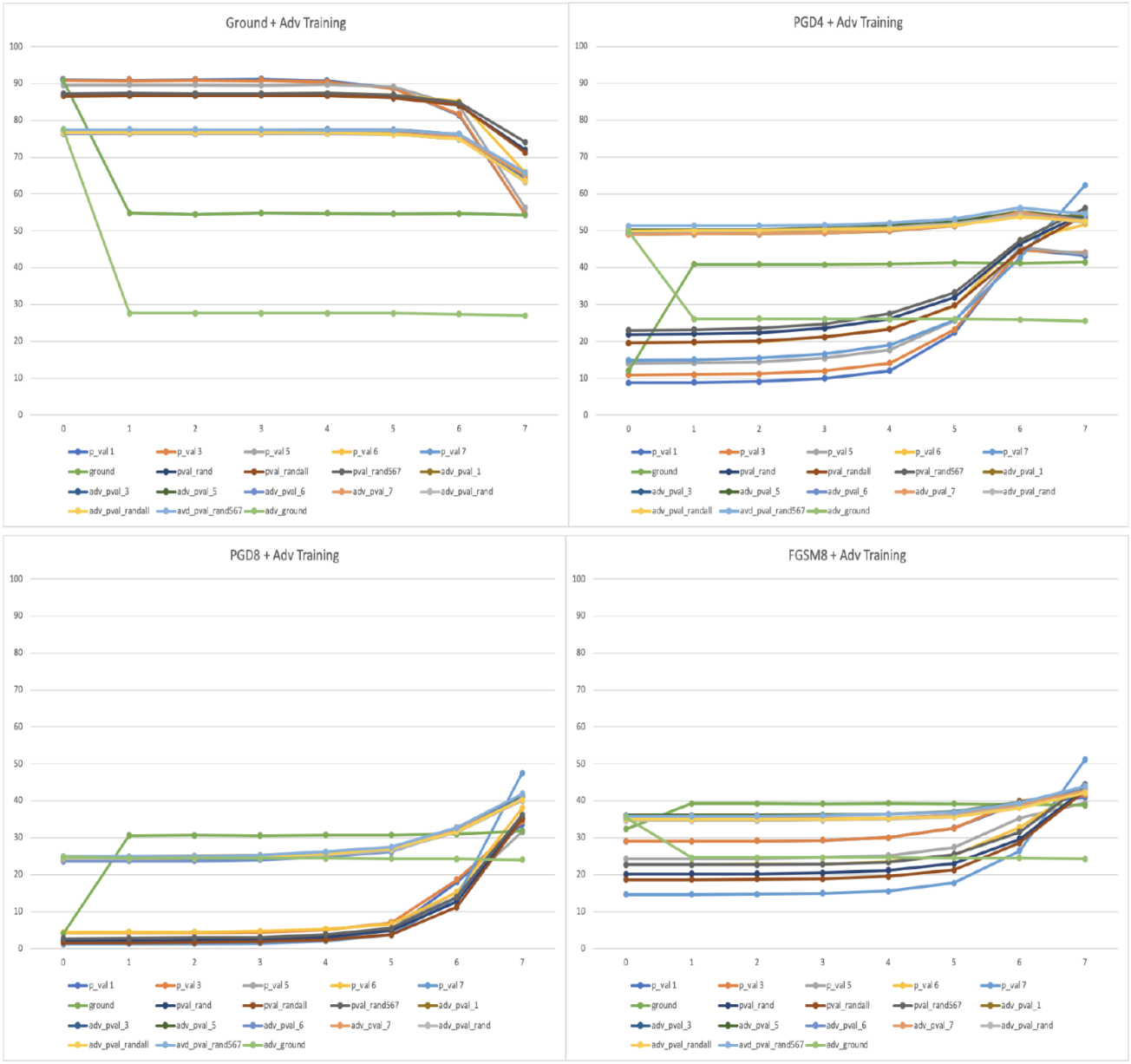

Figure 5. Adversarially Trained Results with Bit-planes Removed at Inference Time

4.2 Combination of Bayesian Neural Networks with Bit-plane Dropout

In this section, we discuss the results, presented in Figures 6 and 7, of combining bayesian neural networks with bit-plane removal. Note: a value of -5 for accuracy was due to some missing data points from failures during experimental evaluation. We do not believe these failures took anything from the big picture takeaways.

For this set of experiments, we switched to the ResNet-18 architecture. This was due to the fact that it is smaller with similar, and sometimes better, than the Densenet architecture. In general, we did not see enough improvement with this combination to warrant the added training time that BNNs induce. Something we noticed during experimentation was that when the BNN models were used for the attacked model, the perturbations were much stronger than with normal and MSB models.

We did however look at something differently from the previous exploration. We explored the impact of removing bit-planes before and after the images were attacked. We wanted to see whether removing bit-planes before the image had the chance to be attacked would have any effect on the overall performance. In this case, we did see a change in how the models performed. When the bit-planes were removed before attack, we saw mainly a decrease in performance when more bit-planes were removed during inference. We saw the opposite pattern, which matches with previous results, of performance increasing when more bit-planes were removed.

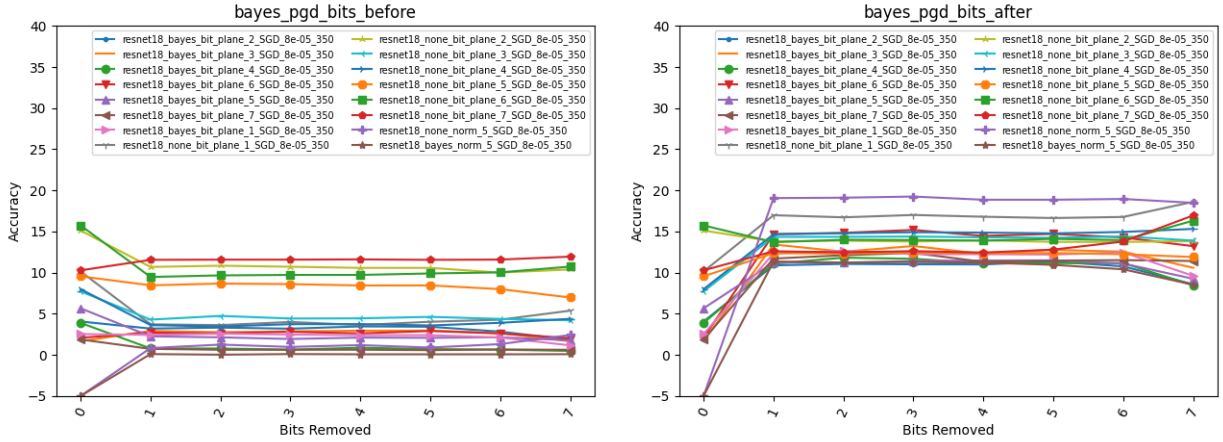

Figure 6. Results of attacking Bayesian Bit-plane models with a Bayesian target model before and after bit-plane removal

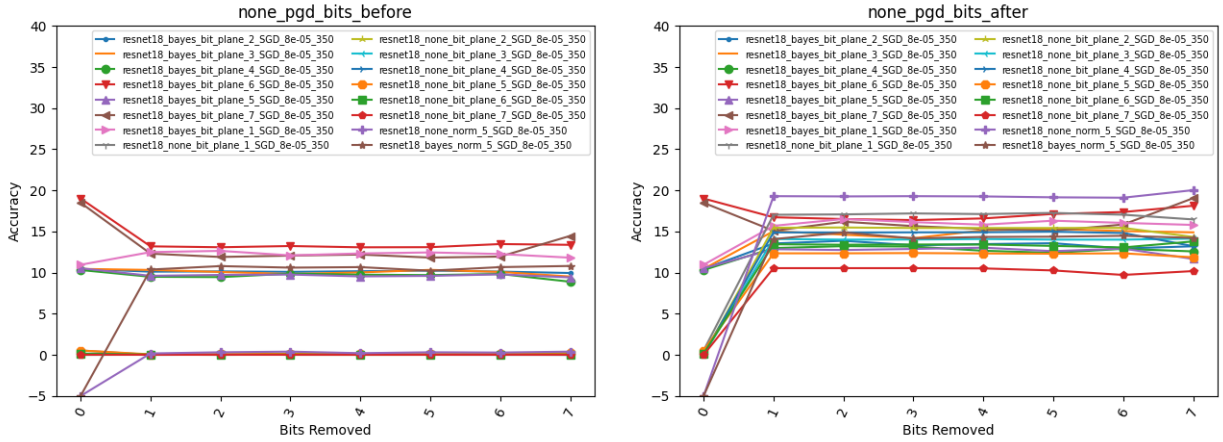

Figure 7. Results of attacking Bayesian Bit-plane models with a normal target model before and after bit-plane removal

Data Type	Defense	Clean Acc	FGSM Acc	PGD Acc	C&W Acc
EO	None	100%	17.7%	31.3%	31.3%
EO	AT	100%	10.1%	12.3%	12.3%
EO	BP	99%	**85%**	**99%**	**99%**
EO	BP,AT	100%	10.2%	11.4%	11.4%
SAR	None	99%	**99%**	**99%**	**99%**
SAR	AT	88%	**88%**	**88%**	**88%**
SAR	BP	99%	**99%**	**99%**	**99%**
SAR	BP, AT	99%	**88.1%**	**88.2%**	**88.1%**

Figure 8. Unfused EO and SAR results

4.3 Sensor Fusion

For this set of experiments, we stuck with the ResNet-18 architecture. Figure 8 shows the results of each EO and SAR model, unfused. The attacks were developed such that they caused a normally trained model to have 0% accuracy on the whole test set. You may also notice that for the normal model in this table, the accuracies are above 0% for the attacked datasets. We believe this has something to do with the normalization of the data. It is something we will look into further. We evaluated four different types of training in this section - normal, bit-plane removal, adversarial training, and bit-plane removal with adversarial training.

EO Model FusionType	Clean Test Data % (SAR model)	FGSM Test Data % (SAR model)	PGD Test Data % (SAR model)	C&W Test Data % (SAR model)
Norm				
Sandia	97.6% (Norm)	**97.6% (Norm)**	**97.6% (Norm)**	**97.6% (Norm)**
Highest Prob	98.2% (Norm)	**98.2% (Norm)**	**98.2% (Norm)**	**98.3% (Norm)**
NaiveBayes	97.9% (Norm)	**97.9% (Norm)**	**97.9% (Norm)**	**97.9% (Norm)**
GenChernoff	98.3% (Norm)	**98.3% (Norm)**	**98.3% (Norm)**	**98.3% (Norm)**
bordaCount	61.6% (Norm)	61.6% (Norm)	61.6% (Norm)	61.7% (Norm)
BP				
Sandia	**99.1% (Norm)**	**99.1% (Norm)**	**99.1% (Norm)**	**99.1% (Norm)**
Highest Prob	**98.4% (Norm)**	**98.4% (Norm)**	**98.4% (Norm)**	**98.4% (Norm)**
NaiveBayes	**99.7% (Norm)**	**99.7% (Norm)**	**99.7% (Norm)**	**99.7% (Norm)**
GenChernoff	**99.6% (Norm)**	**99.6% (Norm)**	**99.6% (Norm)**	**99.6% (Norm)**
bordaCount	**99.8% (Norm)**	**99.8% (Norm)**	**99.8% (Norm)**	**99.8% (Norm)**
AT				
Sandia	29.4% (BPAT)	29.4% (BPAT)	29.4% (BPAT)	29.4% (BPAT)
Highest Prob	47.1% (Norm)	47.1% (Norm)	47.1% (Norm)	47.1% (Norm)
NaiveBayes	26.1% (Norm)	26.1% (Norm)	26.1% (Norm)	26.1% (Norm)
GenChernoff	30.3% (BP)	30.3% (BP)	30.3% (BP)	30.3% (BP)
bordaCount	35.4% (AT)	35.4% (AT)	35.4% (AT)	35.4% (AT)
BPAT				
Sandia	33.0% (BPAT)	33.0% (BPAT)	33.0% (BPAT)	33.0% (BPAT)
Highest Prob	39.3% (BP)	39.3% (BP)	39.3% (BP)	39.3% (BP)
NaiveBayes	28.4% (Norm)	28.4% (Norm)	28.4% (Norm)	28.4% (Norm)
GenChernoff	29.9% (BP)	29.9% (BP)	29.9% (BP)	29.9% (BP)
bordaCount	45.0% (BPAT)	45.0% (BPAT)	45.0% (BPAT)	45.0% (BPAT)

Figure 9. Best fused EO and SAR results - EO model denoted by first column, SAR model is placed in the parentheses next to the accuracy

Figures 9 and 10 show the results of sensor fusion. The first table shows the results when both EO and SAR are faced with the same type of dataset i.e., clean, fgsm, pgd, cnw. The far left column specifies the EO model, and then we took the best fusion combination and record the SAR model in the parentheses next to the accuracy. The top row indicates what dataset the model was evaluated on and a description of how we show the results. We tested 5 different types of decision level fusion for each clean and attacked dataset.

As can be seen in Figure 9, the BP model trained on EO data when fused with either the Normal or AT SAR model, performs the best across all attacks as well as the clean test data. We noticed that the accuracy does not change across attacks for these. We believe this to be caused by the fact that the SAR models were not fooled by the perturbations, so they were much more confident in their predictions. This led to the fusion algorithms being more heavily weighted towards the SAR decision.

Figure 10 shows us that this is still the case when the EO model is presented with clean data, while the SAR model is presented with the different types of attacked data.

EO Model FusionType	FGSM Test Data % (SAR model)	PGD Test Data % (SAR model)	C&W Test Data % (SAR model)
Norm			
Sandia	**97.6% (Norm)**	**97.6% (Norm)**	**97.6% (Norm)**
Highest Prob	**98.3% (Norm)**	**98.3% (Norm)**	**98.3% (Norm)**
NaiveBayes	**97.9% (Norm)**	**97.9% (Norm)**	**97.9% (Norm)**
GenChernoff	**98.3% (Norm)**	**98.3% (Norm)**	**98.3% (Norm)**
bordaCount	62.1% (AT)	62.1% (AT)	62.1% (AT)
BP			
Sandia	**99.1% (Norm)**	**99.1% (Norm)**	**99.1% (Norm)**
Highest Prob	**98.5% (ALL)**	**98.5% (ALL)**	**98.5% (ALL)**
NaiveBayes	**99.8% (AT)**	**99.8% (AT)**	**99.8% (AT)**
GenChernoff	**99.7% (AT)**	**99.7% (AT)**	**99.7% (AT)**
bordaCount	**99.8% (AT)**	**99.8% (AT)**	**99.8% (AT)**
AT			
Sandia	35.9% (BPAT)	35.9% (BPAT)	35.9% (BPAT)
Highest Prob	41.3% (BP)	41.3% (BP)	41.3% (BP)
NaiveBayes	21.2% (Norm)	21.2% (Norm)	21.2% (Norm)
GenChernoff	22.7% (Norm)	22.7% (Norm)	22.7% (Norm)
bordaCount	46.7% (AT)	46.7% (AT)	46.7% (AT)
BPAT			
Sandia	32.9% (BPAT)	32.9% (BPAT)	32.9%% (BPAT)
Highest Prob	38.7% (BP)	38.7% (BP)	38.7% (BP)
NaiveBayes	28.2% (Norm)	28.2% (Norm)	28.2% (Norm)
GenChernoff	29.9% (BP)	29.9% (BP)	29.9% (BP)
bordaCount	46.3% (AT)	46.3% (AT)	46.3% (AT)

Figure 10. Fused EO and SAR results when EO model is presented with clean data and SAR model is presented with adversarial data - EO model denoted by first column, SAR model is placed in the parentheses next to the accuracy

4.3.1 AML Cost

In Figure 11, we present the cost results we collected. We tracked the GPU load throughout training for the different types of training we conducted. The top two images were normal training and bit-plane removal training, respectively. The bottom two images are adversarial training and adversarial training with bit-plane removal, respectively. As can be seen, the normal and bit-plane models use similar resources for a similar amount of time. The adversarially trained model has the highest load for the longest amount of time. The adversarially trained with bit-plane removal model, has a much longer training time than the normal or bit-plane model, but still takes less time than the adversarially trained model. We believe this is due to the fact that some of the data has been removed during training when the bit-planes are removed.

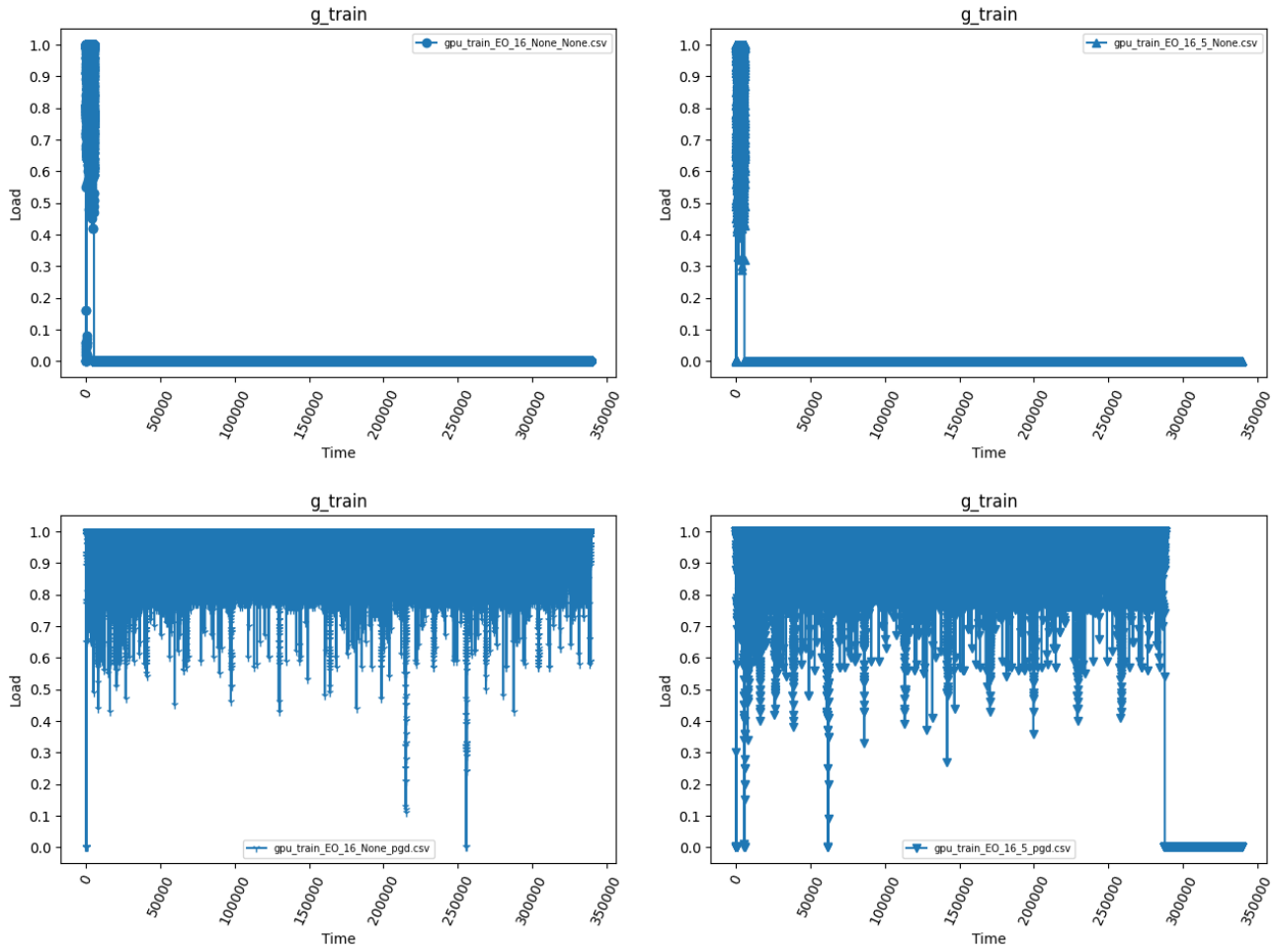

Figure 11. GPU Load vs Training Time

5. CONCLUSION

In this paper, we have provided extensive experimental results from our research into techniques for more robust and secure ML. We explored some existing techniques as well as novel variations of those techniques. We specifically looked into less resource intensive defense techniques to AML. We discovered that while bit-plane removal can increase adversarial robustness, that combining this defense with adversarial training causes a further increase. This combination is actually less resource intensive than traditional adversarial training.

We also explored the impact that AML could have on sensor fusion of EO and SAR data. In this experimentation, we found that SAR models are much more difficult to create a successful attack than for EO models. This allows the fused models to be much more robust than when just using EO data.

ACKNOWLEDGMENTS

This project was supported by Wright State University through the Autonomy Technology Research Center. This internship enabled me to work with mentors from the Air Force Research Lab where I was placed on the Robust and Secure Machine Learning project under the guidance of Alex Hildenbrandt, Ashley Diehl and Christopher Menart.

REFERENCES

[1] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R., "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199* (2013).

[2] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., Giacinto, G., and Roli, F., "Evasion attacks against machine learning at test time," *CoRR* **abs/1708.06131** (2017).

[3] Goodfellow, I. J., Shlens, J., and Szegedy, C., "Explaining and harnessing adversarial examples," (2015).

[4] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A., "Towards deep learning models resistant to adversarial attacks," (2019).

[5] Carlini, N. and Wagner, D. A., "Towards evaluating the robustness of neural networks," *CoRR* **abs/1608.04644** (2016).

[6] Liu, X., Li, Y., Wu, C., and Hsieh, C.-J., "Adv-BNN: Improved adversarial defense through robust bayesian neural network," in [*International Conference on Learning Representations*], (2019).

[7] Farrell, W. J. and Ganesh, C., "Generalized chernoff fusion approximation for practical distributed data fusion," in [*2009 12th International Conference on Information Fusion*], 555–562 (2009).

[8] Ashby, M. and Zelnio, E., "Multi-platform EO and SAR fusion for target ID," in [*Algorithms for Synthetic Aperture Radar Imagery XXIX*], Zelnio, E. and Garber, F. D., eds., **12095**, 1209505, International Society for Optics and Photonics, SPIE (2022).

[9] Simonson, K. M., "Probabilistic fusion of atr results," (8 1998).

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

[11] Leong, C., Rovito, T., Mendoza-Schrock, O., Menart, C., Bowser, J., Moore, L., Scarborough, S., Mindard, M., and Hascher, D., "Unified coincident optical and radar for recognition (unicorn) 2008 dataset," tech. rep. (2019).