Applicability issues of Evasion-Based Adversarial Attacks and Mitigation Techniques

Kishor Datta Gupta
Computer Science Department
University of Memphis
Memphis, TN, USA
kgupta1@memphis.edu

Dipankar Dasgupta
Computer Science Department
University of Memphis
Memphis, TN, USA
dasgupta@memphis.edu

Zahid Akhtar
Computer Science Department
State University of New York Polytechnic Institute
Newyork, USA
akhtarz@sunypoly.edu

***Abstract*—Adversarial attacks are considered security risks for Artificial Intelligence-based systems. Researchers have been studying different defense techniques appropriate for adversarial attacks. Evaluation strategies of these attacks and corresponding defenses are primarily conducted on trivial benchmark analysis. We have observed that most of these analyses have practical limitations for both attacks and for defense methods. In this work, we analyzed the adversarial attacks based on how these are performed in real-world problems and what steps can be taken to mitigate their effects. We also studied practicability issues of well-established defense techniques against adversarial attacks and proposed some guidelines for better and effective solutions. We demonstrated that the adversarial attacks detection rate and destruction rate co-related inversely, which can be used in designing defense techniques. Based on our experimental results, we suggest an adversarial defense model incorporating security policies that are suitable for practical purposes.**

I. Introduction

Deep learning has shown great success in classifying audio and video data. Advanced and complex neural network models have been developed to solve classification problems. In recent years, it has been perceived that all of these architectures remain vulnerable to evasion types of adversarial attacks. Adversarial attacks/examples on Artificial Intelligence (AI) are performed in ways that cause miss-classification, which humans can properly classify. Based on NIST [1] definition, "Adversarial attack is the manipulation of training data, ML model architecture, or manipulate testing data in a way that will result in wrong output from Machine Learning Model (ML)". Perturb based adversarial attacks fall in the criteria of manipulation of test data. It is also known as the evasion attack [2]and it is transferable to any ML model and we only considered evasion based attacks in our work. A mathematical formulation of an adversarial attack can be written in the following way. Let M, A, C_R, ϵ be a learning model, non adversarial input data, aright class output and noise, respectively. Now consider $A_x = A + \epsilon$, where A_x classified by M as class C_W and $(C_W \neq C_R)$. But, for human eyes $A_x \approx A$ and A_x classify as C_R then A_x is an adversarial example and $L_{(.)} = |A - A_x|$, here $L_{(.)}$ is the measurement of difference between adversarial and non adversarial sample.

Various types of evasion attack exists. The more subtle the noise, the more advanced the attacks. When someone wants to devise a defense technique, they may need to prioritize the attacks based on attack frequency, feasibility, damage capability, etc. In the same way, when a learning/ML model developers want to select a defense technique for their model they may need more information (e.g., cost, time, etc.) other than their accuracy rates. A comprehensive analyses of prior works on evasion attacks and observed practicability show that while most of those works reported detection rate (whether noises are very easily detectable using common algorithms (e.g., Fourier transform, Wavelet transform [3–5], etc), only few of them also observed the destruction rate (the rate of adversarial samples failing to remain adversarial under environmental constraints, e.g., resize, rotate, etc[6]). These studies did not scientifically detail the reasons for higher or lower destruction rates or how to counteract such issues. Moreover, we observed that most of the defense techniques and their efficiency were evaluated against a diverse set of attacks, while ignoring other major key factors such as attacks applicability, platform independence, cross-domain support, cost of computation, security policy compliance, and etc.

More significantly, in the current state-of-the-art adversarial defense techniques, the basics of cyber-security principals are not being addressed, as these techniques need to manipulate the AI internal structure or require information about AI. These defense techniques sometimes reduce the efficiency of the AI model it is protecting. Also, researchers are expanding existing adversarial attacks and providing more advanced attack types, which are extremely difficult to guard against. Several benchmarks [7–12] have been developed to evaluate an adversarial defense, but none of these are numerically quantifiable for practical use.

In this work, we aimed to address some of these limitations by focusing on two sides. First, we tried to analyze the applicability of adversarial/evasion attacks under environmental constraint and provide a solution for it. Our second focus was to analyze the defense technique's applicability and provide a quantifiable measurement. For these objectives, we experimented with destruction rates of well-known adversarial attacks under different environmental constraints. We tried to provide feasibility rated adversarial attacks, which derived from detection rate and destruction rates. Then, we proposed a solution to improve these attacks. These experiments proved that advanced attacks are more vulnerable than others. We experimented with some detection filters and showed that basics adversarial attacks are easier to detect than advanced attacks. More specifically, we tried to provide a comparable indicator

978-1-7281-2547-3/20/$31.00

based on our studies. When a model developer wants to pick a defense technique for their model security, they can use this scale to choose the defense technique needed. We proposed that the defense against adversarial attacks should consist of two modules. The first module would detect adversarial attacks, and the second module would deflect the remaining adversarial attacks, which failed to detect in the first module. We provided a quantifiable scale to measure the efficiency of defense techniques.

In short, our research contributions are

- We observed that the destruction rate of adversarial attacks is inversely proportional to their detection rate, (section II) based on that we provided an attack feasibility score.
- We provided a minimum threshold value for perturbing to decrease destruction rates. (Section II-C)
- We observed that defense techniques are evaluated mostly based on their accuracy against attacks while ignoring the applicability of these defense techniques. Based on that we provided a quantifiable scale system that inclusive of practicability factors.(section III-C)
- Based on the nature of the destruction rate and detection rate we proposed a defense techniques considering practical factors. (section III-D)

In section 2, we presented the limitations of adversarial attacks and our observation's including our proposed solution to improve the destruction rate of adversarial attacks. In section 3, we provided a prototype defense model and scoring system for defense techniques. In section 4, we provided a summary of our findings and in the last section, we concluded our study with the limitations of our study and our future improvement plan.

Adversarial Attacks	High Detection	Low destruction	High Destruction	Low Detection	Feasibility
LAVAN	85	30	30	85	30
DPATCH	85	30	30	85	30
FGSM	75	45	45	75	55
BIM	75	50	50	75	70
JSMA	60	60	60	60	65
HOPSKIPJUMP	55	75	75	55	55
CW2	30	85	85	30	30
Deepfool	25	85	85	25	30

TABLE I
VARYING DEGREE OF DESTRUCTION AND DETECTION RATES OF DIFFERENT ADVERSARIAL ATTACKS. HERE GREEN IS DESIRABLE QUALITY AND RED IS NON DESIRABLE FROM ATTACK PERSPECTIVE; COLOR INTENSITY IDENTIFIED BY INDICATE APPROX PERCENT RANGE. THIS REPRESENTATION CAN BE USED AS A VISUALIZATION OF AN 'ATTACK METHOD' STANDING COMPARE TO OTHER METHODS.

II. ADVERSARIAL ATTACKS

In this section we described adversarial attacks, applicability issues and proposed minimum perturb value to improve these attacks. We provided experimental results demonstrating that our provided minimum perturb attacks make current attack types more sustainable in real world. We also show the destruction rate of different attack types under various environmental factors and their detection rate based on SNR (Signal to Noise ratio) values. At the end of this section, we ranked attack types based on their feasibility score.

A. Background

1) Attacks we studied: In 2014 Ian Goodfellow [13] proposed the first gradient sign method (FGSM) which accomplishes the attack by attaching the sign of the gradient to the input, steadily building the magnitude until the input is misclassified. Another perturb method known as the Basic iterative method (BIM)[14] works as a straightforward extension of FGSM. Over the succeeding years, many developments and variations of BIM attacks have been published by researchers. In 2017, Madry et al.[15] added a random start before BIM and renamed it as projected gradient descent (PGD). Adam used optimization on BIM attack to further improve this method[15]. DeepFool [16] attack types generate an adversarial examples by By iteratively linearizing the decision boundary. Carlini-Wagner (CW)[17] introduced an effective optimization objective for iteratively finding the adversarial examples with the smallest perturbations. The local search attack[18] was introduced in 2016. In 2018, the EAD attack which is elastic-net regularized optimization-based attack [19] and in the same year another PGD attack which improves speed known as decoupling based attack [20]. Another attack known as sparse attack was [21] introduced in 2019, which is based on multiple perturbations. The adversarial patch attacks were introduced by Brown et al.[22]. These attacks are different from the other attack types, as here noises are easily spotted by humans; many of these attacks are very effective in the real world. Some variations of these attacks are LAVAN[23] and DPATCH[24] which shows very strong performances in the practical field.

2) Detection Rate: Adversarial attack is possible to distinguish from the non-adversarial samples by different detection methods. Different techniques have different efficiency to detect the adversarial samples. This efficiency to detect adversarial samples or adversarial image (AI) can be measured by the detection rate. In short, our detection rate can simply put as:

$$Detection_{Rate} = \frac{\sum AI_{detected} + \sum Non_{AI} - \sum False_{detected}}{\sum AI + \sum NonAI}\% \quad (1)$$

3) Destruction Rate: Most of the adversarial attack studies ignore the destruction rate, but we prioritize the destruction rate to evaluate practicability. There could be many reasons that an adversarial attack sample fails to perform as adversarial samples. The rate of this failure can be represented by the destruction rate. Kurakin and Yan Goodfellow [14] provided an equation when adversarial images failed to identified as successful adversarial attack after they converted to PNG or printed on a paper. We used the same equation to provide destruction rates of attack samples. In short, our destruction rate can simply put as:

$$Destruction_{Rate} = \frac{\sum FailedAdversarialImages}{\sum AdversarialImages}\% \quad (2)$$

Kurakin[14] represented destruction rate phenomena by the following equation

$$d = \frac{\sum_{k=1}^{n} C(X^k, y^k{}_{True})\overline{C(X^k_{adv}, y^k{}_{True})}C(T(X^k_{adv}), y^k{}_{True})}{\sum_{k=1}^{n} C(X^k, y^k{}_{True})C(X^k_{adv}, y^k{}_{True})} \quad (3)$$

Here, destruction rate is the fraction of adversarial images which are no longer misclassified in real world scenario.

4) Feasibility Score: We came up with a feasibility score to determine severity of an adversarial attack. In short, our feasibility score can simply put as:
let, Adversarial Image Detection Rate (measuring noise values) (DT)
Adversarial Image Destruction Rate(DS)

$$FeasibilityScore = \frac{1}{|DT - DS|} \tag{4}$$

B. Applicability issue

It is well observed, the adversarial images generated do not remain as adversarial when converted to the visual format of images such as 'PNG'. Deep learning models typically use floating values instead of using integer values which are present in image RGB format. This is because it helps better convergence. Real numbers have infinite range and depend on the precision, where as integers have finite range. The activation functions perform better in achieving global optima, like sigmoid activation/Tanh works better with floating values. If we use integer values there is a chance we will miss many local optima and there is a chance we never get a global optimum. Common hardware is equipped to run deep learning with floating-point values. This tendency of preferring floating conversion from integer values of RGB creates the practicability problem for adversarial images. Due to the conversion of float to integer, added noise/perturb disappears. In the paper, Kevin Eykholt et al.[26] reported that the effect of adversarial example get minimized due to several factors in real world scenario. They are environmental conditions, spatial constraints, physical limits on imperceptibility and fabrication Error. They provided a sticker based adversarial attack which works in real world from different angle and distances. However, this attack seems easily identifiable by the human eye due to the shape and intensity of embedded noise. Sharif et al.[27] used adversarial perturbations on the lens of eyeglasses to attack the face recognition system. However, such attack didn't mention destruction rates of adversarial perturbation also their experiment was in a constrained environment. In 2019, Zeng et al.[28] added perturb in 3D models instead of 2D images and showing successful adversarial attacks but there was also no discussion about destruction rate, which makes it difficult to reproduce with the same accuracy. In 2017, Kurakin et al.[14], showed that in the digital version and printed version destruction rate exist, and for advanced attack methods the destruction rate gets higher. They tried to justify their argument with FGSM, BIM, and least likely iterative methods. Lu et al.[29] in their paper experimented with FGSM, BIM, and LBFGS methods and showed the destruction rate can be achieved up of 100% based on distances that invalidating these attacks. Pierazzi et al.[30] shows that, "it is feasible to create adversarial examples in the problem space (realizable attacks) and that it seems there is no correlation between the ability of the classifier to detect such attacks and the disruption of the adversarial examples". However, the problem with Pierazzi et al.[30]. works is that they restricted the problem space which are not applicable for real situation when other environmental factors are present.

As an example, an image of class label 'A', pixel values are $\begin{pmatrix}127 & 243 & 47\end{pmatrix}$ will divided by 255(8bit range) and converted to $\begin{pmatrix}0.4980392157 & 0.9529411765 & 0.1843137255\end{pmatrix}$. The perturbs ϵ are $\begin{bmatrix}0.00000000007 & -0.00000000004 & 0.000000000089\end{bmatrix}$, which makes the image values as $\begin{bmatrix}0.4980392164 & 0.9529411761 & 0.1843137268\end{bmatrix}$. Now it is an adversarial image that will classify as label B. But when we convert it by multiplying 255 will return the $\begin{bmatrix}127 & 243 & 47\end{bmatrix}$which has class label A. This adversarial operation does not exist in real world due to noise value being too small. As all adversarial attack types aim to reduce the epsilon value the practicability issue rises more in advanced attack types. In table I, We provided a comparison of the degree of detection and destruction rates from attacker perspectives. This table can use an approximation chart to understand visually where an 'Attack method' stands to compare to other attack methods.

C. Determining Minimum perturbation values

We proposed a method of minimum threshold of perturbing value which guarantees that noises will affect when the adversarial sample is converted to any image format. However, it does not differentiate whether the perturbed image will be an adversarial or not, it assures that this noise will affect when the image will be an input for an ML model. We tested, attack samples with minimum threshold value's. We determined the destruction rate and compare the result with attacks sample's conventional destruction rates.

1) Minimum Threshold for Perturb: Assuming, vector spaces of pixel values are converted between 0 to 1. If the standard floor and the ceil math function are used to reconvert vector values to the image, we can have the below formulation for the single color channel.
Let, a pixel non floating value X for a single color channel. Image each channel are N bit. Thus perturb value ϵ need to bigger than a threshold value, which can be derived from below equation.

$$(\frac{x}{2^N - 1} + \epsilon) \times (2^N - 1) \geq x + 0.5 \tag{5}$$

From Equation 5, we get

$$\epsilon \geq \frac{0.5}{2^N - 1} \tag{6}$$

For, 8 bit single channel color, the minimum threshold value will be $T \approx \frac{0.5}{2^8 - 1} \approx 0.00196078431$

From printed version if an average accuracy drop is δ, for a printed version the threshold needs to increase by

$$T_p \approx \epsilon + \frac{\epsilon \times \delta}{100} \tag{7}$$

As an example, we observed 20% drop of FGSM method for 8bit MNIST grey channel adversarial, So here the minimum

Types	Normal	Resize			Rotate			Motion			Illumination		
		2x	0.5x	4x	5'	15'	45'	15x	20x	50x	5	15	25
Dpatch [24]	**10**	0	5	0	0	5	5	35	35	80	0	0	5
Lavan [23]	**15**	0	5	0	0	5	5	35	35	85	0	0	5
Patch [22]	**18**	3	6	3	0	6	6	55	60	70	0	0	5
FGSM[13]	**43**	20	18	20	0	10	15	60	77	85	5	10	12
BIM [15]	**32**	28	16	28	5	12	18	66	76	86	10	15	20
JSMA [25]	**28**	22	10	22	0	10	25	60	75	85	15	25	30
CW [17]	**61**	55	75	55	60	75	90	100	100	100	70	80	100
DF [16]	**85**	90	95	90	72	90	95	100	100	100	85	90	100

TABLE II

DESTRUCTION RATES OF VARIOUS ATTACK TYPES UNDER DIFFERENT ENVIRONMENTAL CONDITIONS. THE VALUES IN COLUMN "NORMAL" INDICATE DESTRUCTION RATES UNDER RAW IMAGE AND FOLLOWING COLUMNS REPRESENT WHERE THE SUCCESSFUL ATTACK TYPES IN NORMAL CONDITIONS ARE EXPERIMENTED WITH OTHER CONDITIONS. IN RESIZE, WE RE-SCALED THE IMAGE AND TURN BACKWARD TO ORIGINAL SIZES. TO SIMULATE MOTION EFFECT WE USED GAUSSIAN BLUR WITH DIFFERENT SIGMA VALUE. FOR ILLUMINATION EFFECT WE INCREASED THE BRIGHTNESS. WE CAN SEE THE DESTRUCTION RATES GETS HIGHER WHEN ROTATE AND MOTIONS ARE HIGHER. ADVERSARIAL PATCH'ES HAVE LOWER DESTRUCTION RATE.

	Detection accuracy rate by SNR value using simple transformation			
	Fourier	**Census**	**Wavelet**	**Gabor**
FGSM	89	99	99	70
BIM	85	80	95	65
JSMA	75	70	85	55
HSJ	55	55	65	0
CW	30	10	15	0
DEEPFOOL	15	15	15	0

TABLE III

USING FOUR COMMON TRANSFORMATION TECHNIQUE TO DISTINGUISH BETWEEN ADVERSARIAL SAMPLES AND COMMON SAMPLES (FROM MNIST). WE USED 20000 CLEAN IMAGES AVG SIGNAL TO NOISE RATIO(SNR) AS THRESHOLD. ATTACK IMAGES WHICH HAVE HIGHER SNR THAN THE THRESHOLD ARE IDENTIFIED AS ADVERSARIAL IMAGES . HERE WE PROVIDED THE DETECTION RATES. WE CAN OBSERVE THAT CW AND DEEPFOOL DETECTING IS HARDER.

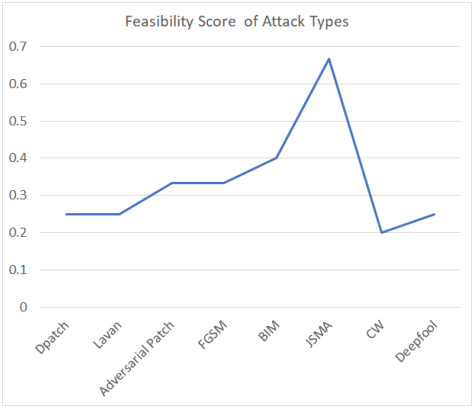

Fig. 2. Feasibility score of different attack types. In X- axis different adversarial attack types mentioned and Y-axis corresponding attack types Feasibility score provided.

threshold should be $T_p \approx 0.00196078431 + 0.00039215686 \approx 0.00235294117$ Any noise ϵ need to be greater than T to have a chance of becoming a digital adversarial image and be greater than T_p to have any chance in becoming a printed adversarial image.

D. Experiments and Analysis on Evasion attack

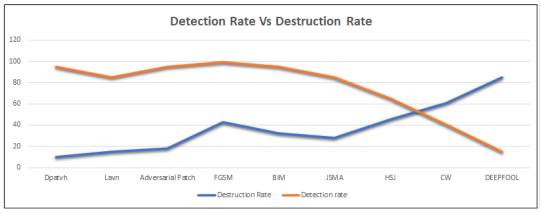

Fig. 1. Relation of destruction rate and detection rate of different attack types. In X-axis different adversarial attack types are mentioned and Y-axis corresponding attack types detection and destruction rate are provided

1) Experimental Data sets: Carlini et al.[7] suggested that testing with one dataset is not enough when we are evaluating adversarial phenomena, therefore we chose two datasets (i.e., MNIST and CIFAR-10). MNIST are all black background images with written digits in white colors consisting of 60000 training and 10000 testing data of single digits zero to nine. When a linear classifier is used, we achieved 83% accuracy and by applying CNN, its efficiency is beyond 99.6%. The CIFAR dataset has images of planes, cars, birds, deer, dogs, frogs, horses, cats, ships, and trucks which include 60000 32x32 color images in 10 distinct classes (6,000 image's of each class). The usual CNN has near 80% accuracy in this dataset, and ResNet has accuracy near 98%, and some more efficient DNN methods have an accuracy of 99% [33]. Adversarial examples are from PGD[15], BIM[15], MBIM[34], FGSM[13], JSMA, DeepFool[16], HopSkipJump[32], Localsearch [18], and CW[35] attack methods in MNIST and CIFAR. We produced FGSM, JSMA, CW, and Deepfool using Pytorch [36]. We used BIM, MBIM, PGD, local Search attack and HopSkipJump using IBM-ART-Toolbox[37] and Cleverhans adversarial library[38]. For our experiments, we generated an average 500 of adversarial examples of each attack type. We considered a standard distortion ϵ = 0.3 for MNIST and ϵ = 8/255 for CIFAR-10 and ImageNet, as the current standard[7] suggested. We also experimented with three types of adversarial patch's on ImageNet dataset. We used 10-20 adversarial patches for our experiment using different adversarial patch attack like DPATCH [24], LAVAN[23].

2) Empirical Experiment: We experimented with basic attack types such as FGSM, JSMA also as and advanced attack types such as CW, Deepfool, etc and used MNIST, CIFAR, and ImageNet data sets. At first we converted all adversarial samples to PNG and observed destruction rates. The successful attack PNG's are printed and scanned as PNG again (similar as [14]. Then we cropped and resized them as original training samples and tried again with the ML model again, we calculating which images were correctly classified as an adversarial attack. Using equation 5, we

	Destruction Rate				Destruction Rate(with threshold)				Destruction Rate(threshold+30)%			
Dataset ->	**MNIST**		**CIFAR**		MNIST		CIFAR		MNIST		CIFAR	
Attack types	**Digital**	**Print**	**Digital**	**Print**	Digital	Print	Digital	Print	Digital	Print	Digital	Print
FGSM [13]	**43**	**22**	**53**	**24**	2	33	6	24	1	23	3	24
BIM [15]	**32**	**26**	**68**	**26**	1	21	4	26	1	26	4	26
MBIM [31]	**46**	**15**	**-**	**-**	2	12	-	-	1	15	-	-
JSMA [25]	**28**	**31**	**76**	**31**	2	22	11	21	1	31	10	-
CW [17]	**61**	**25**	**92**	**32**	12	29	17	27	8	25	26	-
Deepfool [16]	**85**	**22**	**96**	**32**	15	14	32	30	10	16	28	-
HSJ [32]	**45**	**23**	**-**	**-**	5	12	-	-	4	15	-	-

TABLE IV

IN FIRST FOUR COLUMNS UNDER THE DESTRUCTION RATE WITHOUT THRESHOLD, WE SHOW PERCENTAGES OF ADVERSARIAL SAMPLES FAILED TO REMAIN AS ADVERSARIAL. NEXT FOUR COLUMNS SHOW THE ADVERSARIAL RATE AMONG THE ADVERSARIAL SAMPLES WHICH SATISFIED OUR THRESHOLD VALUE. IN THE LAST FOUR COLUMNS WE PROVIDED THE ADVERSARIAL SAMPLES DESTRUCTION RATE WHEN THRESHOLD VALUE INCREASES 30% MORE.

calculated the destruction rates under different environment constraints(example: Rotate, Resize, Illumination, Motion). We simulated motion by blurring the image and we increased the brightness for illumination effect. The results are presented in table II. Here for adversarial patch attack, we used ImageNet and for other attacks, we used average destruction rate of CIFAR and MNIST combination. In the 6th row, FGSM has a 43% destruction rate when it converted to image format. The remaining images have a destruction rate of 10% when their brightness increased 15%. This table shows that Adversarial patch-based attacks have higher tolerance and CW, Deepfool has lower tolerance from the environmental factor.

We used equation 2 with δ value 20% and considered image's with at least 70% of noise above the calculated threshold. We converted the images in PNG and measured the destruction rates. After that, we printed the successful PNG to print and measure the destruction rates. The experimented results are shown in table IV. We used an image difference (Pixel value difference) technique to calculate the δ value as Gupta et al.[5] shows in their paper. We applied 4 transformation techniques such as Fourier transform [39], Census Transform[40], Gabor Transform[41] and Wavelet Transform[42] on adversarial images and also applied them on clean data set. We calculated average SNR value from all clean images and use that as threshold value. We observed SNR values are higher in adversarial images and using that threshold value we can detect basic adversarial attacks. In the image, SNR values are calculated by using the mean of pixels as a signal (S) and std deviations of pixels as noise (N) with below equation

$$SNR = 10 \times \log_{10} \frac{S}{N} \tag{8}$$

We presented these results in table III for MNIST data-set. We not conducted detection experiment for adversarial patch attacks but as they are visible to human eye and the experiments of researcher Chiang et al.[43] shows that adversarial patch attacks are also detectable, we set the detection rate higher as FGSM.

3) Result analysis: From table II, we can see motion and rotation has more effects. We observed that CW and Deep-Fool attacks had a 100% destruction rate if they rotated too much. It is also observable that patch-based attacks have good performance in real world conditions. These results proved that if we use some reflecting technique we can avoid advanced perturb based attacks.

From table IV, we can see that when only considering adversarial images with noise above our threshold, the drop of accuracy sharply declines from the previous result in the tableII where all the adversarial samples destruction rate were shown. From the table, it appears destruction in the printed version does not change much based on the attack type. Destruction rate is pretty consistent with any normal clean image detection failure rate, as well, when we observed the detection rates of different attacks in table III. Here in the1st row, if FGSM samples transformed using Fourier transform and calculate its SNR value that 89% of FGSM images are outside of Clean images SNR value range. We can see the CW, Deep-Fool attacks are hard to detect from SNR values. From the results of the above experiments we draw the graph in figure 1. Here, In the graph, we can see the destruction rate and detection rate are co-related each other. Based on these rates we calculated feasibility scores of different attack types for MNIST dataset. From figure 2 we can see JSMA attacks are more practical for the the MNIST dataset.

III. ADVERSARIAL DEFENSES

In this section, we analyzed existing adversarial defense techniques and their limitations and proposed a framework to mitigate these limitations. Also we proposed a benchmark method to score or rank all defense techniques, so it is easy to compare when a learning model developer wants to select a defense technique.

A. Related Works

In some adversarial defense techniques well-known robust recognition models are trained on adversarial inputs proactively [13], performing defensive distillation [44], training the network with enhanced training data all to create a protection against adversarial example [45]. To detect adversarial inputs Image histogram-based [46] methods are also used. In 2018 Akhter et al. [47] proposed an adversarial attack detection scheme based on image quality related features in order to detect various adversarial attacks. In 2017, Carlini et al. [10] tested ten defense techniques and showed that by using pre-processing techniques attack can be easily bypass. These defense techniques are evaluated based on several benchmark system, some of which we discussed in our proposed benchmarks.

Defense Techniques	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F_score
Adversarial Training [48] [49] [50]	10	5	5	5		5			10	10	10		0.6
Image Prepossessing [35]	10	5				5		10	10	10	0		0.5
Input Reconstruction[51]	10	5	5			5			10	10	0		0.45
Distillation Techniques [44],[52]	10	5	5	5		5		10	10			10	0.6
Transform Function [3]	10	5				5		10	10	10	10		0.6
Defense GAN[53]	10	5	5	5	10	5							0.4
Model Robustifying[54][55]	10	5	5	5		5			10	10	10		0.6

TABLE V

HERE, WE SCORED BASED ON EQUATION 9 FROM OUR STUDY OF DIFFERENT DEFENSE TECHNIQUES. (SOME OF THESE SCORES ARE AN APPROXIMATION BASED OUR UNDERSTANDING) IN THE LAST COLUMN, THE OVERALL SCORE IS PRESENTED. BASED ON THE OVERALL SCORE WE CAN SEE SOME OF THE TECHNIQUES HAVE BETTER USABILITY THAN OTHERS.

B. Limitations of Existing Benchmarking

Researchers have suggested several benchmarks [7–12] and use these benchmark to evaluate defense techniques. They also suggested guidelines on how to develop efficient defense techniques. Most of their benchmark focused on how many attacks a defense technique can defend. Some of them prioritized testing against adaptive attacks. As these benchmarks were mostly dependent on dataset and attack set, how these defense techniques compared to other techniques, such as compliance, computational cost, cybersecurity practices, etc are mostly overlooked. Also, these evaluations did not provide a metric for measurement. We concluded that these benchmarks are very good to evaluate defense techniques' performances and novelty but these benchmarks do not help a learning model developer to pick a defense technique appropriate for specific problems. We tried to answer these limitations by our benchmarking system.

C. Proposed Benchmarking for Defense Techniques

We simulated some use cases where a Machine Learning model requires a defense technique to protect against adversarial attacks.

- First case: we considered that we will protect a Pytorch[56] Resnet model for Predicting handwriting digit tool for android devices.
- Second case: we considered that we will protect a Pytorch Resnet model for Predicting handwriting tools running on an Amazon Web service[57] using the Django framework[58].
- Third case: We considered protecting a MNIST TensorFlow project using ML.NET project[59] as a console application.
- Fourth Case: We considered that we will employ an MNIST dataset using Deeplearning4j[60] from java library as a desktop project.
- Fifth Case: We considered protecting Close circuit camera used in a parking lot for reading car number-plates using deep-learning.

We tried to secure the learning model for the above-mentioned cases with standard cybersecurity practices such as keeping data privacy, secure authentication, and access policy. We considered most of the mentioned defense techniques discussed in section III-A. We observed that many defense techniques were not suitable for some of our cases. For example, image prepossessing based defense is not suitable for mobile apps and adversarial training reduces the accuracy for number plate recognition. Adversarial training based defenses were not suitable if training data is sensitive to share. GAN based defense requires high computation cost, which seems very unreasonable for simple AI tasks where the higher error rate is tolerable. From these case studies, we understood which defense techniques were better suited and what factors were more important than others. We weighted them based on our observations. These factors are:

- F1:Tested against multiple data set. $W = 10$
- F2:Tested against black-box -white box attack. targeted-non targeted, gradient based-non gradient based attacks $W = 5$
- F3: Have low computational overhead cost.$W = 5$
- F4: Cross-models and multi domain applicability .$W = 5$
- F5: Tested against adaptive attack.$W = 10$
- F6: No machine learning involved in the defense technique.$W = 5$
- F7: Randomness exist to answer obscurity.$W = 10$
- F8: No training data needed.$W = 10$
- F9: No Knowledge of the learning model needed to know.$W = 10$
- F10:No modification of the learning model needed.$W = 10$
- F11:No accuracy drop of learning model needed.$W = 10$
- F12:No adversarial knowledge needed to generate defense.$W = 10$

F1 is very important for understanding the effectiveness of an adversarial attack. We weighted these by 10. Number F2 is about the diverseness of attack types. We weighted this by 5. Learning models usually have large computational complexity; thus if the protecting tool requires higher computational overhead it will make full system impractical to use due to both time and cost. We weighted it by 5. Some defense techniques can only work for a specific domain; for instance our proposed methods are only for computer vision domain, and we tested on against attack samples from Resnet and VGG-16. As adversarial samples have transferability to another learning model. It is expected that defense methods will supports cross-models and multi-domains. We weighted this factor only by 5 as defense techniques are easily customize-able as per requirement. Defending against an adaptive attack is very important, as the attacker can try continuously until succeeding. So F5 is important as F2. If there is a Machine learning involved in the defense technique than that technique can be also susceptible to adversarial attack. So involving defending Machine learning with another machine learning will not improve security rather it creates another door way of the same vulnerabilities. We weighted F6 as 5. We have to assume that our defense technique details are known to the attacker. That's why randomness is needed which will make it hard for predict what defense configuration will be set for each attack time. We weighted this as 10. F8, F9, F10, and F11 are related to CIA models of information security. All information security policy measures try to address three goals known as confidentiality (protect

the confidentiality of assets), integrity(preserve the integrity of assets), and availability (promote the availability of assets for authorized users). These goals form the CIA model[61] which is the basis of all security programs[62]. Here, ML is considered as a digital asset, we ensure its confidentiality as we are not consuming or accessing any architectural information of ML. The same way integrity is preserved as we don't need to modify or tune anything in ML architectures. If training data or learning models are needed to generate defense it will violate confidentiality if modification of the learning model is needed it will violate integrity and if the accuracy drop it will violate the availability. Because of this dilemma, we weighted F8, F9, F10, F11 by 10. Zero-day vulnerability can also be present in adversarial defenses if there are no safeguard against unknown adversarial attack methods. There is high probability chance of unknown attacks that's why we give this factor 10 weight value. We disregarded several factors such as easy maintenance or update facility, time to implement, etc, as they also depend on the learning model itself.

The total score of a defense technique can be measured by below equation

$$F_{score} = \frac{\sum F_{i=1..12}}{100} \tag{9}$$

The Maximum score is possible up to 1 and the lowest score is possible as low as 0. Based on these factors, we create a radar map as shown in figure 3 with 3 defense techniques. Here, the effectiveness represents by the factor F1 and F2. Computation feasibility represents by factor F3. F4 represents by platform-independent. Vulnerability represents factor F5,F6,F7. These factors cover potential vulnerability such as a zero-day attack, advanced persistent attack, and insider attacks. F8-F12 are represented by cybersecurity Compliance. We further analyzed other defense methods and presented their benchmark result in table V. We can see that none of the defense technique have a better score than 0.6. As the max possible score is 1, there are opportunities to improve. In section III-D, we tried to provide a defense technique based on these factors.

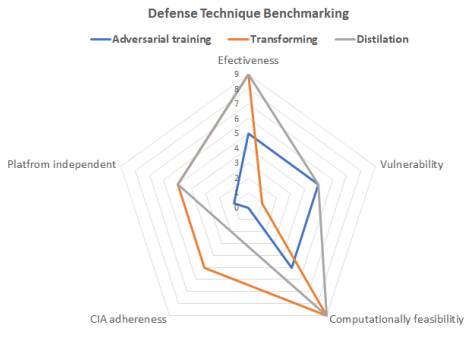

Fig. 3. Our Proposed Bench-marking Process with sample data. Here we tried to represent some common defense techniques in a radar map.

D. Our Proposed Defense Technique

From our experimental results, we observed detection rate and destruction rates are co-related inversely as seen in the table 1. It proved that some attack types are detectable and some attacks can deflect easily. So we assume a defense technique that has two modules, with one for detection and one for destruction rate. In figure 4, we show the basic block diagram of our defense technique.

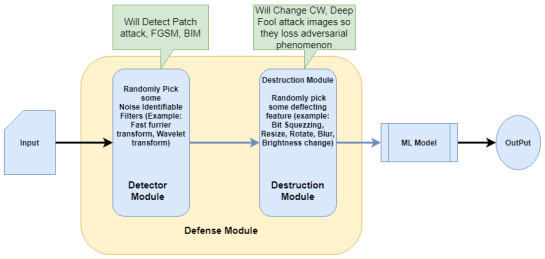

Fig. 4. Our Proposed Defense Module

Destruction Algorithm	Variable	Detection Algorithm	Indicators
Adaptive Smoothing	Kernel size 5x5	Distance transform	SNR
Bilateral Smoothing	Kernel size 3x3	Census Transform	Histogram
Resize	Algorithm variation	Fourier Transform	Euclidean Distance
Sharpen		Wavelet Transform	Loss function
Shrink		Gaussian Noise	
Dilation		Gabor wavelet	
Erosion		Dithering	
Contrast		Gabor Transform	

TABLE VI
SOME EXAMPLES OF DESTRUCTION AND DETECTION ALGORITHM ARE PROVIDED IN THIS COLUMN, IN THE THIRD COLUMN THE TRANSFORM TECHNIQUES WERE APPLIED ON THE INPUT IMAGE AND WE CALCULATED THE SNR VALUES FROM THE TRANSFORMED IMAGES. IN THE FIRST COLUMN, SOME ALGORITHM ARE MENTIONED WHICH REDUCED THE ADVERSARIAL EFFECTS. RANDOMLY PICKING A SET OF FILTERS FROM BOTH GROUPS CAN BE USED TO MAKE THE DETECTION MODULE AND DESTRUCTION MODULE.

In the detection module, there will be different detection techniques such as image processing and transforming and check the detection indicators (Example: signal to noise values, histogram natures, loss function, etc). We proposed that we will randomly pick a set of detection techniques and their algorithmic variables (example: kernel size). Each technique of this set will detect input separately and if one of the techniques detects input as adversarial, the final result will be adversarial from this module.

In the destruction module, we will destroy adversarialness by resizing, blurring, brightening the image. We will also randomly pick the destruction method and its variables (examples: the scale of resizing, Level of brightness, etc).

If there are N_d number of destruction algorithm, N_v number of algorithm variables, N_{dt} number of detection methods total possible combinations of defense technique could exist can calculated using equation 10.

$$PossibleVariation(p_v) = N_d \times {}^{N_{dv}}C_{k_{dv}} \times {}^{N_{dt}}C_{k_{dt}} \tag{10}$$

Here K_{dv}, K_{dt} denoted the number of detection algorithm applied in a sequence. The number of variations in the

algorithm and the probability of predicting which combination was picked can be expressed by equation 11.

$$Probabilities(P) = \frac{1}{p_v} \tag{11}$$

As an example, table VI, where 8 types of destruction and detection algorithm and 3 types of their variation are mentioned. If we consider only 1 types of detection is using in detection module then the probability of correct prediction by an attacker will be $\frac{1}{192} = 0.005$ as $K_{dv} = K_{dt} = 1$

This will make impossible for an attacker to develop an adaptive attack against our defense technique, as each time our technique will have a different defense algorithm.

	MNIST		CIFAR-10		Image Net	
	DT	DS	DT	DS	DT	DS
Clean Samples	2	10	6	10	15	25
FGSM [13]	99	30	90	45	80	50
BIM [15]	95	45	88			
MBIM [31]	92	43				
JSMA [25]	82	33	85	55	65	50
CW [17]	25	90	15	90		
DEEPFOOL [16]	28	95	15	100		
Hopskipjump [32]	50	65	35	85		
BPDA(with BIM)[63]	60	75	55	65		

TABLE VII

DIFFERENT ATTACK TYPES AND CLEAN SAMPLES DETECTED(DT) AND DESTRUCTED(DS) IN OUR DEFENSE FRAMEWORK. BY DESTRUCTED HERE MEANS IT WRONGLY CLASSIFIED FOR CLEAN SAMPLES AND CORRECTLY CLASSIFIED FOR OTHER ADVERSARIAL SAMPLES. IN THE DETECTION MODULE, WE USED COMBINATION OF WAVELET TRANSFORM, FOURIER TRANSFORM, DISTANCE TRANSFORM, AND CENSUS TRANSFORM SNR VALUES FOR RANGE. IN THE DESTRUCTION MODULE, WE USED NORMAL ENVIRONMENT WITH COMBINATION OF RE-SCALING FEATURES. RESULTS ARE SHOWN AS PERCENTS OF ACCURACY.

1) Experiment with our defense technique: For our experiments, we used three data sets ImageNet, CIFAR, and MNIST. We ran different attack types including adaptive attack types BPDA. In the BPDA attack, we used BIM attack types which can pass image resizing effects as defense. We used 100 to 500 attack samples for each attack type with ResNet and Vgg-16 models. Some of the experiments were not able to be conducted due to not being able to produce enough attack samples. In future work, we will do further study by extending our attack dataset. In the table VII the results of our experiments are shown. We calculated the score of defense techniques in table V. Here adversarial training requires training data manipulations so it violates some of the factors. In this table, we can see each defense technique has some strong side and weak side. Most of them are well evaluated against a diverse set of attacks and data-set. But many of them require the learning models information or need to modify, for defense purpose, which we consider as a negative factor as it violates the basic principles of a security system.

2) Analysis and Comparison: From table VII we can see that our proposed models work against a diverse set of attack types but it also detects some of the clean samples as adversarial samples, reducing the accuracy of learning model by destroying some clean samples. We suggested that better tuning of different detection and destruction techniques will improve these drawbacks. Our models identified the basic attacks(Higher perturbs) and deflecting the advanced attack (lower perturbs). Our defense technique does not require to knowing how the learning model works, and there is no need to modify it. Its destruction module requires no training or any data set requirement. In the detection module, we need to knowing the clean images of SNR values, so it will require a set of training data samples. This is a drawback as in this way training data needs to accessible by the security module. But our defense technique has several advantages over other defense techniques; for instance, in this technique we do not need to generate adversarial samples. We are generating detection modules from our training data. We are also not modifying or accessing learning models. One of the most important features is that we are randomly picking detection and destruction techniques, which is very effective against adaptive attacks. GAN based defense techniques need high computational power which in this defense technique is not required. Most importantly, our defense technique is very practical and easy to implement in different frameworks or platforms. This requires very little computation power and less knowledge about attack methods or learning models. In table VIII, we presented a result comparison with other defence technique in terms of Accuracy. Using equation 9, we calculate our F_{score} which has 0.65 and it is better score than other defense techniques.

Detection Method	**MNIST**				**CIFAR**				**Avg**
	FGSM	**JSMA**	**DF**	**CW**	**FGSM**	**JSMA**	**DF**	**CW**	
RF Learning[64]	0.96	0.84	0.98	0.66	0.64	0.63	0.60	0.72	0.77
KNN learning [64]	0.98	0.80	0.98	0.6	0.56	0.52	0.52	0.69	0.73
SVM learning [64]	0.98	0.89	0.98	-	0.69	0.69	0.64	0.77	0.81
Feature Squeez[35]	**1.00**	**1.00**		-	0.20	0.88	0.77	-	0.77
Ensemble [65]	0.99	-	0.45	-	0.99	-	0.42	-	0.71
Decision match[66]	0.93	0.93	0.91	-	0.93	**0.97**	0.91	-	0.93
Image features [47]	**1.00**	0.90	**1.00**	-	0.72	0.70	0.68	-	0.83
Proposed Method	0.99	0.99	0.98	**0.98**	0.97	0.98	**0.97**	**0.98**	**0.98**

TABLE VIII

COMPARISON WITH OTHER ADVERSARIAL INPUT DETECTION TECHNIQUES BASED ON ACCURACY. TO BE NOTED, OUR SCORES ARE A COMBINATION OF DETECTION AND DESTRUCTION RATES. WHICH ARE NOT CONSISTENT WITH OTHER METHODS RESULTS AS THEY ONLY PROVIDED THE DETECTION RATE.

IV. OUR SUGGESTIONS

We suggest that when researchers generate adversarial attacks they maintain the minimum threshold as equation 7 so attacks has a lower destruction rate. Also, they should provide the feasibility scores of attacks under different detection methods and environmental constraints as we mentioned in tableII. Based on our study, reducing the perturbs/noises in an adversarial image made it harder to detect but also easy to destroy. Keeping that in mind, new attacks should be generated in a way that its detection rates and destruction rates both are low as possible. In our study, we noted that higher accuracy of a defense technique does not necessarily make it a better technique, if we consider other aspects. Defense techniques should consider a threat model that consists of all the factors defined in section III-C and provided us a F_score using equation 9. Defense techniques with more close to Fscore 1 are more practical to implement. Also, detailed score distribution of Fscore will help learning model developers

to choose which criteria of defense techniques will be more useful for their learning models. In our table V, the factors of different defense techniques are detailed, indicating that deep learning researchers have not prioritized cybersecurity or platform independence as they care more about accuracy. Improving accuracy is important for a security system but there is always another trade-off to consider. It opens a door of opportunities for researchers to develop more deploy-able defense techniques for learning models.

V. Conclusion

The paper summarized applicability issues of advanced adversarial attacks and reported our experimental results on benchmark datasets. It is to be noted that we used limited datasets and learning models in our experiments, further study be needed for rigorous validation of conducted observations. While experimental results within different techniques may slightly vary, we believe the overall trend of results will be aligned with the reported results and similar to other works[6][14]. In this paper, we proposed a metric to calculate minimum noise/perturb value which will have an effect when an adversarial image is converted to any visual form. As mentioned earlier, Researchers who are designing new attacks with increase subtle noise should consider a high destruction rate. Our threshold value in equation 7 may help them to avoid having a higher destruction rate. We also provided a benchmark which is more practical in evaluating defense techniques. Our results highlighted the need for a more usable defense system for learning models. We showed that a defense technique can be developed by the combination of detecting and deflecting, which are more practical from usability perspective. We provided a proof-of-concept defense technique and compared our results with other works. In further work, we will develop a more comprehensive library of destruction and detection modules to improve the efficiency of defense modules.

References

[1] E. Tabassi, K. Burns, M. Hadjimichael, A. Molina-Markham, and J. Sexton, "A taxonomy and terminology of adversarial machine learning," 2019.

[2] D. Dasgupta, Z. Akhtar, and S. Sen, "Machine learning in cybersecurity: a comprehensive survey," *The Journal of Defense Modeling and Simulation*, p. 1548512920951275, 2020.

[3] U. Shaham, J. Garritano, Y. Yamada, E. Weinberger, A. Cloninger, X. Cheng, K. Stanton, and Y. Kluger, "Defending against adversarial images using basis functions transformations," *arXiv preprint arXiv:1803.10840*, 2018.

[4] S. Soleymani, A. Dabouei, J. Dawson, and N. M. Nasrabadi, "Defending against adversarial iris examples using wavelet decomposition," *arXiv preprint arXiv:1908.03176*, 2019.

[5] K. D. Gupta, D. Dasgupta, and Z. Akhtar, "Determining sequence of image processing technique (ipt) to detect adversarial attacks," *arXiv preprint arXiv:2007.00337*, 2020.

[6] H. H. Nguyen, M. Kuribayashi, J. Yamagishi, and I. Echizen, "Detecting and correcting adversarial images using image processing operations and convolutional neural networks," *arXiv preprint arXiv:1912.05391*, 2019.

[7] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," *arXiv preprint arXiv:1902.06705*, 2019.

[8] A. Athalye and N. Carlini, "On the robustness of the cvpr 2018 white-box adversarial example defenses," *arXiv preprint arXiv:1804.03286*, 2018.

[9] N. Carlini and D. Wagner, "Magnet and efficient defenses against adversarial attacks" are not robust to adversarial examples," *arXiv preprint arXiv:1711.08478*, 2017.

[10] ——, "Adversarial examples are not easily detected: Bypassing ten detection methods," *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017.

[11] N. Carlini, "Lessons learned from evaluating the robustness of defenses to adversarial examples," 2019.

[12] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," *arXiv preprint arXiv:2002.08347*, 2020.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014.

[14] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[15] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.

[17] N. Carlini and D. Wagner, "Defensive distillation is not robust to adversarial examples," *arXiv preprint arXiv:1607.04311*, 2016.

[18] N. Narodytska and S. P. Kasiviswanathan, "Simple black-box adversarial perturbations for deep networks," *arXiv preprint arXiv:1612.06299*, 2016.

[19] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "Ead: elastic-net attacks to deep neural networks via adversarial examples," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[20] J. Rony, L. Gustavo, R. Sabourin, and E. Granger, "Decoupling direction and norm for efficient gradient-based l2 adversarial attacks," 2018.

[21] F. Tramèr and D. Boneh, "Adversarial training and robustness for multiple perturbations," in *Advances in Neural Information Processing Systems*, 2019, pp. 5858–5868.

[22] T. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch. arxiv e-prints (dec. 2017)," *arXiv preprint cs.CV/1712.09665*, vol. 1, no. 2, p. 4, 2017.

[23] D. Karmon, D. Zoran, and Y. Goldberg, "Lavan: Localized and visible adversarial noise," *arXiv preprint arXiv:1801.02608*, 2018.

[24] X. Liu, H. Yang, Z. Liu, L. Song, H. Li, and Y. Chen, "Dpatch: An adversarial patch attack on object detectors," *arXiv preprint arXiv:1806.02299*, 2018.

[25] R. Wiyatno and A. Xu, "Maximal jacobian-based saliency map attack," *arXiv preprint arXiv:1808.07945*, 2018.

[26] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[27] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 acm sigsac conference on computer and communications security*, 2016, pp. 1528–1540.

[28] X. Zeng, C. Liu, Y.-S. Wang, W. Qiu, L. Xie, Y.-W. Tai, C.-K. Tang, and A. L. Yuille, "Adversarial attacks beyond the image space," in *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition, 2019, pp. 4302–4311.

[29] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "No need to worry about adversarial examples in object detection in autonomous vehicles," *arXiv preprint arXiv:1707.03501*, 2017.

[30] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing properties of adversarial ml attacks in the problem space," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 1332–1349.

[31] Y. Dong, F. Liao, T. Pang, X. Hu, and J. Zhu, "Discovering adversarial examples with momentum," *arXiv preprint arXiv:1710.06081*, 2017.

[32] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," *arXiv preprint arXiv:1904.02144*, 2019.

[33] M. Wistuba, A. Rawat, and T. Pedapati, "A survey on neural architecture search," *arXiv preprint arXiv:1905.01392*, 2019.

[34] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.

[35] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[36] Gongzhitaao, "gongzhitaao/tensorflow-adversarial," Jan 2018. [Online]. Available: https://github.com/gongzhitaao/tensorflow-adversarial/tree/v0.2.0

[37] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.1.1," *CoRR*, vol. 1807.01069, 2018. [Online]. Available: https://arxiv.org/pdf/1807.01069

[38] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long, "Technical report on the cleverhans v2.1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2018.

[39] D. Yin, R. G. Lopes, J. Shlens, E. D. Cubuk, and J. Gilmer, "A fourier perspective on model robustness in computer vision," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 255–13 265.

[40] B. Froba and A. Ernst, "Face detection with the modified census transform," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.* IEEE, 2004, pp. 91–96.

[41] D. Spina, C. Valente, and G. Tomlinson, "A new procedure for detecting nonlinearity from transient data using the gabor transform," *Nonlinear Dynamics*, vol. 11, no. 3, pp. 235–254, 1996.

[42] A. F. Pérez-Rendón and R. Robles, "The convolution theorem for the continuous wavelet tranform," *Signal processing*, vol. 84, no. 1, pp. 55–67, 2004.

[43] P.-Y. Chiang, R. Ni, A. Abdelkader, C. Zhu, C. Studor, and T. Goldstein, "Certified defenses for adversarial patches," *arXiv preprint arXiv:2003.06693*, 2020.

[44] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[45] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *arXiv preprint arXiv:1605.07725*, 2016.

[46] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8571–8580.

[47] Z. Akhtar, J. Monteiro, and T. H. Falk, "Adversarial examples detection using no-reference image quality features," in *2018 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2018, pp. 1–5.

[48] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[49] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.

[50] S. Latif, R. Rana, and J. Qadir, "Adversarial machine learning and speech emotion recognition: Utilizing generative adversarial networks for robustness," *arXiv preprint arXiv:1811.11402*, 2018.

[51] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 135–147.

[52] M. Soll, T. Hinz, S. Magg, and S. Wermter, "Evaluating defensive distillation for defending text processing neural networks against adversarial examples," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 685–696.

[53] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-gan: Protecting classifiers against adversarial attacks using generative models," *arXiv preprint arXiv:1805.06605*, 2018.

[54] J. Bradshaw, A. G. d. G. Matthews, and Z. Ghahramani, "Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks," *arXiv preprint arXiv:1707.02476*, 2017.

[55] T. Strauss, M. Hanselmann, A. Junginger, and H. Ulmer, "Ensemble methods as a defense to adversarial perturbations against deep neural networks," *arXiv preprint arXiv:1709.03423*, 2017.

[56] N. Ketkar, "Introduction to pytorch," in *Deep learning with python*. Springer, 2017, pp. 195–208.

[57] S. Narula, A. Jain *et al.*, "Cloud computing security: amazon web service," in *2015 Fifth International Conference on Advanced Computing & Communication Technologies*. ieee, 2015, pp. 501–505.

[58] J. Forcier, P. Bissex, and W. J. Chun, *Python web development with Django*. Addison-Wesley Professional, 2008.

[59] Y. Lee, A. Scolari, B.-G. Chun, M. Weimer, and M. Interlandi, "From the edge to the cloud: Model serving in ml. net." *IEEE Data Eng. Bull.*, vol. 41, no. 4, pp. 46–53, 2018.

[60] V. Kovalev, A. Kalinovsky, and S. Kovalev, "Deep learning with theano, torch, caffe, tensorflow, and deeplearning4j: Which one is the best in speed and accuracy?" 2016.

[61] R. D. Alexander and S. Panguluri, "Cybersecurity terminology and frameworks," in *Cyber-Physical Security*. Springer, 2017, pp. 19–47.

[62] S. Samonas and D. Coss, "The cia strikes back: Redefining confidentiality, integrity and availability in security." *Journal of Information System Security*, vol. 10, no. 3, 2014.

[63] M. Naseer, S. Khan, and F. Porikli, "Local gradients smoothing: Defense against localized adversarial attacks," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1300–1307.

[64] J. Hayes and G. Danezis, "Machine learning as an adversarial service: Learning black-box adversarial examples," *arXiv preprint arXiv:1708.05207*, vol. 2, 2017.

[65] A. Bagnall, R. Bunescu, and G. Stewart, "Training ensembles to detect adversarial examples," *arXiv preprint arXiv:1712.04006*, 2017.

[66] J. Monteiro, Z. Akhtar, and T. H. Falk, "Generalizable adversarial examples detection based on bi-model decision mismatch," *arXiv preprint arXiv:1802.07770*, 2018.