2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC) | 979-8-3503-5326-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ETNCC63262.2024.10767521

Impact of White-Box Adversarial Attacks on Convolutional Neural Networks

1st Rakesh Podder
Department of Computer Science
Colorado State University
Fort Collins, CO, USA, 80523
rakesh.podder@colostate.edu

2nd Sudipto Ghosh
Department of Computer Science
Colorado State University
Fort Collins, CO, USA, 80523
sudipto.ghosh@colostate.edu

***Abstract*—Autonomous vehicle navigation and healthcare diagnostics are among the many fields where the reliability and security of machine learning models for image data are critical. We conduct a comprehensive investigation into the susceptibility of Convolutional Neural Networks (CNNs), which are widely used for image data, to white-box adversarial attacks. We investigate the effects of various sophisticated attacks—Fast Gradient Sign Method, Basic Iterative Method, Jacobian-based Saliency Map Attack, Carlini & Wagner, Projected Gradient Descent, and DeepFool—on CNN performance metrics, (e.g., loss, accuracy), the differential efficacy of adversarial techniques in increasing error rates, the relationship between perceived image quality metrics (e.g., ERGAS, PSNR, SSIM, and SAM) and classification performance, and the comparative effectiveness of iterative versus single-step attacks. Using the MNIST, CIFAR-10, CIFAR-100, and Fashion_MNIST datasets, we explore the effect of different attacks on the CNNs performance metrics by varying the hyperparameters of CNNs. Our study provides insights into the robustness of CNNs against adversarial threats, pinpoints vulnerabilities, and underscores the urgent need for developing robust defense mechanisms to protect CNNs and ensuring their trustworthy deployment in real-world scenarios.**

***Index Terms*—convolutional neural networks, image quality metrics, performance metrics, test input generation, white-box adversarial attacks.**

I. Introduction

In the landscape of escalating cyber warfare, adversarial attacks on machine learning (ML) models have emerged as a sophisticated vector for undermining AI-driven systems. The inherent susceptibility of ML algorithms to specially crafted inputs that can lead to incorrect outputs, known as adversarial examples, has introduced a pressing challenge to the field of cybersecurity. The use of ML models in critical applications, such as autonomous vehicles [1], healthcare diagnostics [2], surveillance [3], and XR [4], has become prevalent. The trust placed by end-users in various industry domains, healthcare, and governments on the reliability and security of AI-driven systems is fundamental to their widespread adoption.

Adversarial attacks have rapidly evolved from theoretical considerations to practical threats. These attacks leverage knowledge of the ML model's structure and data processing to introduce subtle perturbations, often imperceptible to humans but catastrophic for the model's decision-making accuracy. The consequences of successful adversarial attacks can range from trivial misclassifications to life-threatening situations [5]. Therefore, understanding and mitigating these attacks are not just academic exercises; they are urgent requirements for the safe deployment of ML in real-world scenarios.

The goal of this paper is to conduct a systematic evaluation of various white-box adversarial attacks [6], where the attacker has complete visibility into the model's architecture, parameters, and training data, on generic neural network models for images. We use a curated set of images such as the MNIST [7], CIFAR-10, CIFAR-100 [8], and Fashion_MNIST [9] datasets processed by a Convolutional Neural Network (CNN). The datasets take into account the variety and complexity required to challenge the CNNs under test. We identify the intrinsic vulnerabilities of CNNs when exposed to white-box attacks such as Fast Gradient Sign Method (FGSM) [10], Basic Iterative Method (BIM) [11], [12], Jacobian-based Saliency Map Attack (JSMA) [13], Carlini & Wagner (C&W) [14], Projected Gradient Descent (PGD) [15] [16], and DeepFool [17].

Through a range of test scenarios that simulate attacks using the Adversarial Robustness Toolbox (ART) library [18], we discern how different performance metrics are affected, specifically focusing on the accuracy and loss incurred by the model under adversarial conditions. We quantify the degradation of performance in CNNs and investigate the robustness of these networks against such exploits. We also evaluate the impact on the image quality and integrity, with a specific focus on the assessment of widely recognized image analysis metrics such as ERGAS [19], PSNR [20], SSIM [21], and SAM [22].

The rest of the paper is organized as follows. Section II provides a brief background on the attacks used in this paper. Section III outlines the evaluation goals, research questions, and metrics. Section IV describes the study design and experimental environment. The results are presented in Section V and discussed in Section VI. Section VII summarizes related work. Section VIII summarizes our conclusions and outlines directions for future work.

II. Background

Our study selected the following sophisticated white-box adversarial attacks based on their relevance to CNN models

979-8-3503-5326-6/24/$31.00 ©2024 IEEE

and image data, prevalence in the current research literature, and real-world applicability [6], [23], [24].

- **Fast Gradient Sign Method (FGSM)**: New images that are classified incorrectly are created by leveraging the gradients of the loss with respect to the input image [10]. Even though the method is straightforward, it is powerful in demonstrating the vulnerability of neural networks to slight, often imperceptible, changes in the input data.
- **Basic Iterative Method (BIM)**: BIM [11], [12] is an extension of FGSM. It iteratively applies the gradient sign attack with small steps, allowing for finer control over the perturbation process and often results in more effective adversarial examples.
- **Jacobian-based Saliency Map Attack (JSMA)**: JSMA [13] uses the model's Jacobian matrix to determine which pixels in the input image to alter to change the classification outcome. The method is more refined than FGSM and BIM, and attempts to change the least number of pixels, thus making the alterations less detectable.
- **Carlini & Wagner (C&W)**: The C&W attack [14] is an effective method that formulates adversarial example creation as an optimization problem. It aims to find the smallest perturbation that can mislead the CNN model, ensuring that the adversarial examples remain as close as possible to the original images.
- **Projected Gradient Descent (PGD)**: PGD is a well-known variation of the BIM, distinguished by its initialization with uniform random noise [15]. Similarly, the Iterative Least-likely Class Method (ILLC) [11] bears a resemblance to BIM, with a key difference being its targeting of the least likely class to maximize the cross-entropy loss making it more effective than FGSM, JSMA and C&W [25].
- **DeepFool**: This algorithm iteratively perturbs the input image in a way that is intended to cross the decision boundary of the classifier [17]. It aims to be as efficient as possible, resulting in minimal perturbation [15].

III. Evaluation Goals, Questions, and Metrics

The main objective of this evaluation is to measure and understand the impact of white-box adversarial attacks on the performance and reliability of CNNs in the context of image processing. We aim to establish a rigorous testing method for detecting vulnerabilities within CNNs and to quantify the effectiveness of adversarial attacks in degrading model performance. The following goals guide our evaluation:

Goals:

1) Assess the impact of adversarial attacks on the accuracy and integrity of the image classification process.
2) Identify the attack methodologies that result in the most significant degradation of performance metrics.
3) Provide insights into the development of more robust CNN architectures and training processes.

Questions:

1) How do various white-box adversarial attacks affect the classification accuracy of CNNs?
2) Which adversarial attack is most effective in inducing the highest error rates?
3) What is the relationship between perceived image quality and classification performance of CNNs under attack?
4) How does the iterative nature of certain attacks (e.g., BIM, PGD) compare to single-step attacks (e.g., FGSM) in terms of effectiveness?

Metrics:

We use a combination of traditional CNN performance metrics and specialized image quality assessments:

- **Loss**: This is a measure of how well the model performs from an error perspective. Specifically, it represents the *cost* incurred for inaccurate predictions. In the code, the loss is calculated using `sparse_categorical_crossentropy`, which is a common loss function for classification tasks. It compares the predicted probability distribution (output of the `softmax` function in the last layer) with the true distribution, where the true distribution is the label of the class that the input image belongs to. A lower loss indicates better performance of the model, as it means the model's predictions are closer to the true labels.
- **Accuracy**: This is a measure of the proportion of correctly predicted instances out of all predictions made. In a classification task like MNIST (which involves classifying images of handwritten digits into 10 classes, from 0 to 9), the accuracy is calculated by the number of images correctly classified divided by the total number of images classified. Higher accuracy means that the model has better predictive performance.
- **Relative Dimensionless Global Error in Synthesis (ERGAS):** This is a global measure of image fidelity, with lower values indicating better synthesis quality [19].

$$\text{ERGAS} = 100 \cdot \sqrt{\frac{1}{d} \sum_{i=1}^{N} \left(\frac{\text{RMSE}_i}{\mu_i} \right)^2}$$

  - d is the scale factor between the spatial resolutions of the original and the processed image (often set to 1 for images of the same resolution).
  - N is the number of bands.
  - RMSE_i is the Root Mean Square Error of the ith band.
  - μ_i is the mean of the ith band of the original image.
- **Peak Signal-to-Noise Ratio (PSNR):** This is a measure of peak error, with higher values indicating smaller differences between original and perturbed images. PSNR is calculated using the maximum pixel value (L) and the Mean Squared Error (MSE) between the original (I) and corrupted (K) images. M and N are the number of rows

and columns respectively in the images [20].

$$\text{PSNR} = 20 \cdot \log_{10}(L) - 10 \cdot \log_{10}(\text{MSE})$$

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (I(i,j) - K(i,j))^2$$

- **Structural Similarity Index (SSIM):** This is a perception-based model that considers changes in texture, contrast, and luminance [21].

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

  - x, y are the windowed images being compared.
  - μ_x, μ_y are the averages of x and y.
  - σ_x^2, σ_y^2 are the variances of x and y.
  - σ_{xy} is the covariance of x and y.
  - c_1, c_2 are variables to stabilize division with a weak denominator.
- **Spectral Angle Mapper (SAM):** It is a measure of the spectral similarity between two images, with lower values indicating higher similarity [22]. It measures the angle between the spectral vectors $\mathbf{a}$ and $\mathbf{b}$.

$$\text{SAM} = \cos^{-1}\left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}\right)$$

The above metrics are calculated before and after the application of each adversarial attack.

- **Pre-attack Performance:** We establish the baseline values of loss, accuracy, ERGAS, PSNR, SSIM, and SAM.
- **Post-attack Performance:** The same metrics are re-assessed post-adversarial attack to evaluate the impact.
- **Adversarial Success Rate:** We record the rate at which adversarial inputs successfully deceive the CNN.
- **Robustness Threshold:** We identify the minimal perturbation magnitude necessary to compromise the model.

IV. Study Design

Our study incorporated the following steps, which we illustrate using the FGSM attack for lack of space. We developed a python script using the *TensorFlow* and *Adversarial Robustness Toolbox (ART)* libraries to (1) create & train a neural network on the MINST [7], CIFAR-10, CIFAR-100 [8], & Fashion_MNIST [9] datasets, (2) generate adversarial examples, and (3) evaluate the models' performance on the adversarial examples. We loaded and preprocessed (normalized) the image datasets such as MNIST, CIFAR-10, CIFAR-100, and Fashion_MNIST.

For **model selection and preparation**, we used the TensorFlow API to create custom CNN models with different hyperparameter values (e.g., *number of neurons*, *dropout rate*, *number of classes*, and *optimizers*). For example, for the FGSM attack, we created a simple neural network model using *TensorFlow's Keras API*. The model consists of a Flatten layer that converts each 28x28 MNIST type images into a 784 element vector, followed by a Dense layer with 128 nodes, a Dropout layer that randomly sets 20% of the input units to 0 during training, and a final Dense layer with 10 nodes corresponding to the 10 possible digits (0-9). For CIFAR-10 & CIFAR-100, the TensorFlow model is a CNN for 32x32 pixel RGB images, featuring three convolutional layers with ReLU activations for feature extraction—first with 32 filters, followed by two layers with 64 filters each, interspersed with 2x2 max pooling for dimensionality reduction. After the convolutional layers, it employs a flattening step, a dense layer of 64 units (ReLU activation), and concludes with a 10-unit softmax output layer for classifying into 10 categories.

We compiled the model using the *adam* optimizer and the *sparse_categorical_crossentropy* loss function, and then trained on the training images and labels for 5 epochs. We evaluated the trained model on the test images and labels to get the baseline loss and accuracy. Training for 5 epochs is sufficient to achieve a reasonable balance between training time and performance, allowing the model to learn effectively without overfitting. Additionally, running the model for more epochs could lead to only marginal improvements in performance metrics, as the model typically converges within the first few epochs.

After defining and training the model, we wrapped it within an ART classifier, such as *TensorFlowV2Classifier* for TensorFlow models, specifying necessary hyperparameters like the number of classes, input shape, and loss object. For each attack type, we created an instance of the corresponding ART attack class, configuring it with relevant parameters (e.g., *eps* for FGSM, *max_iter* for PGD). For example, for the input to the FGSM attack, we created ART's *TensorFlowV2Classifier* using the trained model. ART's *FastGradientMethod* attack is created using the classifier and an epsilon value of 0.1.

We performed **adversarial example generation** by leveraging state-of-the-art techniques to mislead the CNNs while preserving image quality. We implemented a *generate* method that takes the attack instances and passes the original inputs. This process, although slightly varied in parameters and attack initialization, follows the same basic steps across different adversarial techniques, enabling the evaluation of model robustness under various types of adversarial conditions.

The **evaluation method** calculated the loss and accuracy on both the original and adversarial examples through the model's *evaluate* method. This method computes the loss and accuracy metrics by comparing the model's predictions on the input images against the true labels. This process involves feeding the perturbed images into the model and calculating the metrics to assess how well the model performs on these adversarial inputs. A large decrease in accuracy or an increase in loss indicates that the adversarial attack was successful in degrading the model's performance.

Statistical analysis of the adversarial attack's impact was performed by comparing various metrics between the original and adversarial images. Using the *sewar* [26] library, metrics such as ERGAS, PSNR, SSIM, and SAM were computed. The analysis was encapsulated in a DataFrame, providing a

structured view of the impact across different metrics, thereby facilitating an understanding of the adversarial attack's effectiveness in degrading image quality and model performance.

$$\text{EF} = \begin{bmatrix} \text{loss} \\ \text{accuracy} \\ \text{ERGAS} \\ \text{PSNR} \\ \text{SSIM} \\ \text{SAM} \end{bmatrix} \cdot \left(\begin{bmatrix} \text{FGSM} \\ \text{JSMA} \\ \text{C\&W} \\ \text{PGD} \\ \text{DeepFool} \\ \text{BIM} \end{bmatrix} \times \begin{bmatrix} \text{MNIST} \\ \text{CIFAR-10} \\ \text{CIFAR-100} \\ \text{Fashion\_MNIST} \end{bmatrix} \right)$$

The evaluation framework (EF) enables a comprehensive analysis by evaluating how each attack affects model performance across different types of data, quantified through metrics for aspects such as accuracy, error, and image quality.

We used Input Space Partitioning (ISP) and Base Choice Coverage (BCC) [27] in our test method to systematically explore the model's vulnerability across various configurations and adversarial scenarios. ISP facilitates a detailed examination of the model's input space, partitioning it down into multiple blocks for a nuanced analysis of vulnerability using various characteristics. Partitioning allows us to uncover a wider range of weaknesses by examining how different types of inputs can influence the model. BCC extends this analysis by first identifying the base choice for each characteristic that was used to partition the domain of an input variable, and then creating combinations of the input partitions, starting with all the base choices and then by varying one choice at a time.

V. Results

We ran our evaluation on a 3.1 GHz Dual-Core Intel Core i5 processor, with 8 GB 2133 MHz, and LPDDR3 memory.

A. Performance and Image Quality Metrics

For the FGSM attack described in Section IV, the script trains a simple neural network on the various datasets, generates adversarial examples using the FGSM attack, evaluates the model's performance on the adversarial examples as shown in Table I, and uses *matplotlib* to display an original and adversarial image side-by-side as shown in Figure 1.

TABLE I: Metrics From FGSM Adversarial Attacks on MNIST, CIFAR-10, CIFAR-100, & Fashion_MNIST

Metric	MNIST	CIFAR-10	CIFAR-100	Fashion _MNIST
Accuracy	0.10	0.12	0.04	0.19
loss	3.08	6.35	6.96	6.48
ERGAS	27.08	88.32	52.68	14.94
PSNR	22.27	18.60	8.99	4.28
SSIM	(0.882, 0.945)	0.71	0.38	(0.123, 0.114)
SAM	0.28	0.25	0.49	1.08

Table II details the outcomes of applying several adversarial techniques on the CNN metrics. Figure 2 illustrates the practical effects of adversarial manipulations by presenting side-by-side comparisons of original and compromised images using the DeepFool, PGD, JSMA, and BIM attacks.

B. Input Space Partitioning and Base Choice Coverage

The input variables analyzed using ISP are:

- Number of Neurons: Numeric value.
- Dropout Rate: Numeric percentage.
- NB_Classes: Numeric value.
- Optimizer: Alphanumeric value.
- Dataset Type: Image type data.

Table III displays the results of ISP analysis in terms of the variables, the characteristics chosen to partition the input domains, the blocks in each partition, and representative values. For each input variable, the base choice is the block numbered 1 (e.g., a1, b1, c1, d1, and e1) as shown in Table IV. BCC leverages these partitions to construct test cases, each designed to probe different combinations of input conditions. These parameters are crucial in a CNN because they directly influence the model's capacity to learn, generalize, and maintain robustness against adversarial attacks by affecting network complexity, regularization, classification capability, and optimization efficiency.

C. Testing

Tables V to VIII present the results of evaluating the resilience of Convolutional Neural Networks (CNNs) to the FGSM attack under various conditions. Each table represents a distinct experiment focusing on altering one specific model parameter–number of neurons, dropout rate, number of classes, and optimizer–while keeping the others constant to observe its impact on the model's loss and accuracy.

Table V varies the number of neurons, indicating how an increase in the model complexity impacts its vulnerability to adversarial examples, with a trend suggesting that more neurons slightly improve the resistance to FGSM attacks, as shown by decreased loss and increased accuracy.

Table VI explores different dropout rates, a technique for preventing overfitting. The results demonstrate that both very low and very high dropout rates make the model more susceptible to FGSM, with an optimal range providing better defense.

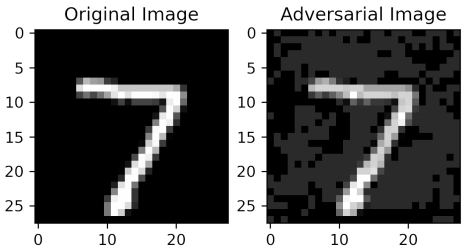
Table VII adjusts the number of classes, testing the model's ability to handle FGSM attacks with varying degrees of classification complexity. However, the impact on model performance does not linearly correlate with the number of classes.

Table VIII examines the effect of different optimizers on model robustness against FGSM attacks. The choice of optimizer significantly affects the model's defense capability, with some optimizers leading to higher susceptibility.

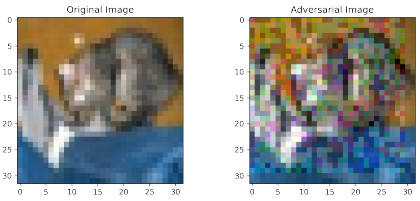
The emphasis on loss and accuracy metrics in our evaluation framework is pivotal for gauging the effectiveness of adversarial attacks on CNN. These metrics reflect the impact of attacks on model performance, offering a clear picture of how well the network withstands manipulation. As we enhance the model's accuracy through various adjustments, such as optimizing the number of neurons or tweaking the dropout rate, we concurrently observe a improvement in image quality metrics like ERGAS or PSNR. For instance, in Table V, increasing the number of neurons leads to a slight improvement in model

TABLE II: Metric Evaluation Table for all Adversarial Attacks

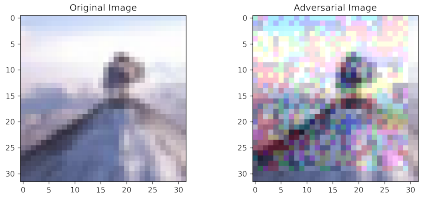
Metric	FGSM	DF	C&W	PGD	JSMA	BIM
Accuracy	0.10	0.00977	0.977	0.0135	0.075	0.02
loss	3.08	2782.88	0.074	80.89	0.975	16.5
ERGAS	27.08	3254.48	78.336	79.73	29.72	26.56
PSNR	22.5	2.617	13.407	13.52	23.56	22.55
SSIM	(0.89, 0.94)	(-0.13, -0.24)	(0.632, 0.71)	(0.62, 0.71)	(0.93, 0.934)	(0.896, 0.95)
SAM	0.28	1.365663	0.748	0.749	0.236	0.27



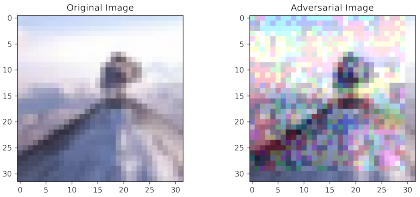
(a) On MNIST



(b) On CIFAR-10

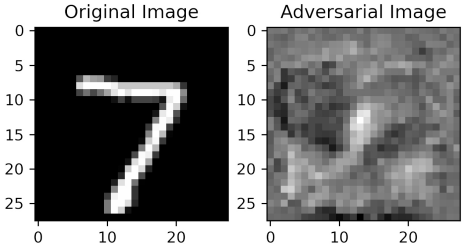


(c) On CIFAR-100

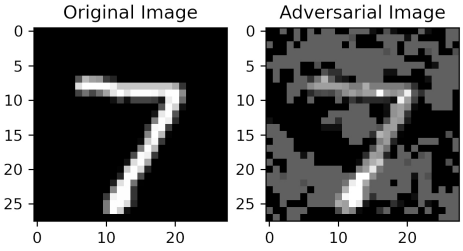


(d) On Fashion_MNIST

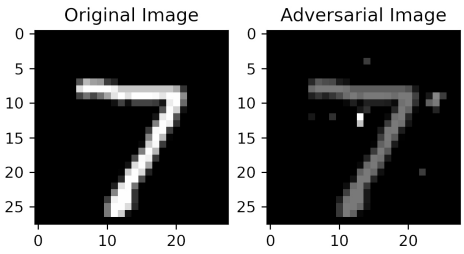
Fig. 1: Original and Compromised Images Generated from FGSM Attacks



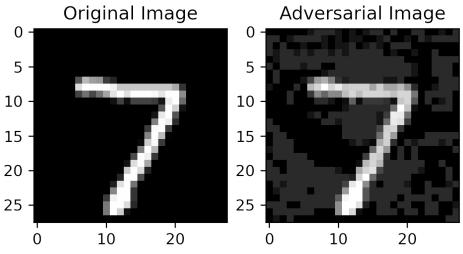
(a) DeepFool attack



(b) PGD attack



(c) JSMA attack



(d) BIM attack

Fig. 2: Original MNIST and Compromised Images Generated Using DeepFool, PGD, JSMA, and BIM.

TABLE III: Input Space Partitioning on the Hyperparameters

Variables	Characteristics	Partitions	Values
Number of Neurons (N)	Numeric: $0 < N < finitevalue$	a1: true	N = 128
		a2: false	N = NULL
Dropout rate (R)	Numeric: $0 < R <= 1$	b1: true	R = 0.2
		b2: false	R = - 0.3
NB_classes (nb)	Numeric: $0 < N < finitevalue$	c1: true	$nb \geq 2$
		c2: false	$nb < 2$
Optimizer (O)	Alphanumeric	d1: true	O = "Adadelta "
		d2: false	O = NULL (None)
Dataset type (val)	Image	e1: nonEmpty	val = "MNIST"
		e2: Empty	val = NULL/Non-Image type

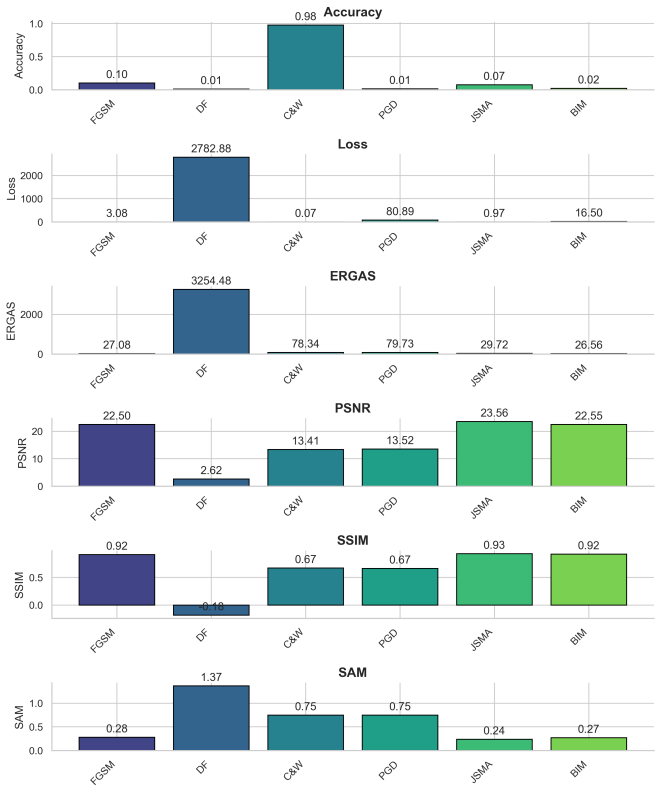


Fig. 3: Effects of Adversarial Attacks on CNN

accuracy, which correlates with enhancements in image quality metrics, indicating a more robust model against FGSM attacks. This relationship suggests that strategies improving accuracy against adversarial examples also contributes to preserving the integrity of image quality post-attack. Statistical evaluation, using a *paired t-test*, reveals that the improvements in Accuracy, Loss, ERGAS, PSNR, and SSIM are statistically significant ($p < 0.05$), while the improvement in SAM shows promising trends but did not reach statistical significance ($p > 0.05$).

Beyond FGSM, we extended our testing to include other adversarial attack types, DeepFool, PGD, C&W, and BIM. We observed consistent patterns across these attacks. For

TABLE IV: Base Choice Coverage Table

Test	Block 1	Block 2	Block 3	Block 4	Block 5
T_1 (base)	a1	b1	c1	d1	e1
T_2	a2	b1	c1	d1	e1
T_3	a1	b2	c1	d1	e1
T_4	a1	b1	c2	d1	e1
T_5	a1	b1	c1	d2	e1
T_6	a1	b1	c1	d1	e2
T_7	a2	b2	c1	d1	e1
T_8	a2	b1	c2	d1	e2
T_9	a1	b2	c1	d2	e2
T_{10}	a1	b2	c2	d2	e1

example, models with higher dropout rates or those employing certain optimizers like *RMSProp* or *Adam* tended to exhibit more significant performance degradation, as highlighted in Tables VI and VIII. This degradation manifested not only in increased loss and decreased accuracy but also in deteriorated image quality metrics, reinforcing the intertwined relationship between model accuracy and image fidelity in the context of adversarial resilience. The consistent observation of these patterns across different types of attacks validates our approach of using FGSM as a representative example in our evaluation. It demonstrates that the insights gained from FGSM tests offer a reliable indication of how CNN might respond to a broader spectrum of adversarial strategies.

VI. Discussion

Based on the empirical evaluation data, we observe that different adversarial attacks on CNNs demonstrate different impacts when applied to the MNIST or other image type (CIFAR-10, CIFAR-100, and Fashion_MNIST) datasets. After testing with new hyper-parameter settings such as optimizer = *Adadelta*, $N = 1000$, $R = 0.2$ and $nb = 200$, we improved the accuracy from Table I's 0.10 to 0.377, and loss from 3.08 to 1.56 for MNIST. By performing the FGSM attack on a CNN trained on CIFAR-10, CIFAR-100 & Fashion_MNIST data sets, we found:

- For CIFAR-10, using optimizer *Adadelta*, $esp = 0.01$, and $nb = 500$, improved the accuracy from 0.12 to 0.187, and loss from 6.35 to 2.24.

TABLE V: Testing the ISP on FGSM varying Number of Neurons (N).

Test Case	N	R	nb	O	val	loss	accuracy
TC_1	a1 = 128	b1 = 0.2	c2 = 10	d1 = Adadelta	e1 = MNIST	3.08	0.105
TC_2	a1 = 100	b1 = 0.2	c2 = 10	d1 = Adadelta	e1 = MNIST	3.10	0.09
TC_3	a1= 150	b1 = 0.2	c2 = 10	d1 = Adadelta	e1 = MNIST	3.04	0.108
TC_4	a1 = 500	b1 = 0.2	c2 = 10	d1 = Adadelta	e1 = MNIST	2.9	0.12
TC_5	a1=1000	b1 = 0.2	c2 = 10	d1 = Adadelta	e1 = MNIST	2.5	0.13

TABLE VI: Testing the ISP on FGSM varying Dropout Rate (R).

Test Case	N	R	nb	O	val	loss	accuracy
TC_1	a1 = 128	b1 = 0.2	c2 = 10	d1 = Adadelta	e1 = MNIST	3.08	0.105
TC_2	a1 = 128	b1 = 0.02	c2 = 10	d1 = Adadelta	e1 = MNIST	3.19	0.099
TC_3	a1 = 128	b1 = 0.001	c2 = 10	d1 = Adadelta	e1 = MNIST	3.8	0.06
TC_4	a1 = 128	b1 = 0.5	c2 = 10	d1 = Adadelta	e1 = MNIST	2.88	0.11
TC_5	a1 = 128	b1 = 0.2	c2 = 10	d1 = Adadelta	e1 = MNIST	2.75	1.24

TABLE VII: Testing the ISP on FGSM varying Number of Classes (nb).

Test Case	N	R	nb	O	val	loss	accuracy
TC_1	a1 = 128	b1 = 0.2	c2 = 10	d1 = Adadelta	e1 = MNIST	3.08	0.105
TC_2	a1 = 128	b1 = 0.2	c2 = 2	d1 = Adadelta	e1 = MNIST	3.10	0.09
TC_3	a1 = 128	b1 = 0.2	c2 = 50	d1 = Adadelta	e1 = MNIST	3.04	0.108
TC_4	a1 = 128	b1 = 0.2	c2 = 100	d1 = Adadelta	e1 = MNIST	2.9	0.12
TC_5	a1 = 128	b1 = 0.2	c2 = 200	d1 = Adadelta	e1 = MNIST	2.5	0.13

TABLE VIII: Testing the ISP on FGSM varying Optimizer (O).

Test Case	N	R	nb	O	val	loss	accuracy
TC_1	a1 = 128	b1 = 0.2	c2 = 10	d1 = Adadelta	e1 = MNIST	3.08	0.105
TC_2	a1 = 128	b1 = 0.2	c2 = 10	d1 = adam	e1 = MNIST	8.6	0.074
TC_3	a1 = 128	b1 = 0.2	c2 = 10	d1 = sgd	e1 = MNIST	3.85	0.104
TC_4	a1 = 128	b1 = 0.2	c2 = 10	d1 = Adagrad	e1 = MNIST	3.29	0.092
TC_5	a1 = 128	b1 = 0.2	c2 = 10	d1 = RMSProp	e1 = MNIST	8.48	0.073

- For CIFAR-100, using optimizer *sgd*, $esp = 0.01$, and $nb = 200$, improved the accuracy from 0.04 to 0.214, and loss from 6.96 to 3.13.
- For Fashion_MNIST, using optimizer *Adadelta*, $esp = 0.01$, and $nb = 500$, improved the accuracy from 0.19 to 0.597, and loss from 6.48 to 1.49.

This variety in impact is shown by significant fluctuations in key performance indicators and image quality metrics. These findings underscore the susceptibility of CNNs to sophisticated adversarial methods. It is evident that the development of more robust defense mechanisms is crucial to ensure the reliability and security of CNN applications across different domains.

Table II shows that the DeepFool attack results in the lowest accuracy among all the attacks, indicating a substantial amount of data loss. In contrast, the FGSM attack shows a comparatively better synthesis quality as it records the lowest ERGAS value among the tested attacks. The JSMA attack stands out with the highest peak error. However, it is interesting to note that in terms of SSIM values, which reflect changes in texture, contrast, and luminance, the impacts are quite similar across all attacks on the MNIST dataset. A particularly noteworthy observation is that despite JSMA's high peak-to-peak error, it has the lowest SAM value, suggesting that it maintains higher similarities between the attacked image and the original image. This characteristic of JSMA could be critical for understanding and countering adversarial attacks on CNN models.

The evaluation extends to different test cases for the FGSM attack as detailed in Tables V through VIII, which illustrate how the accuracy and loss metrics of a CNN are influenced by varying factors such as the number of neurons, the dropout rate, the choice of optimizer, and the number of classes.

A. Answers to Research Questions

The answers to our research questions are as follows.

Q1: The analysis revealed a variable impact of different adversarial attacks on the classification accuracy of CNNs. Specifically, the DeepFool attack was identified as significantly reducing accuracy due to its effective exploitation of substantial data modifications.

Q2: The DeepFool attack stood out as the most effective in inducing the highest error rates, demonstrated by its

notably low accuracy scores, highlighting its proficiency in degrading performance metrics.

Q3: There was a discernible relationship between image quality metrics and the classification performance of CNNs under attack. For instance, attacks like FGSM, which exhibited lower ERGAS values, suggest a higher quality of image synthesis despite adversarial modifications, indicating an inverse relationship between image quality metrics and classification vulnerability.

Q4: Iterative attacks such as BIM and PGD proved more effective than single-step attacks like FGSM. This effectiveness is attributed to the iterative approach's ability to apply perturbations in a refined manner, enhancing the attack's potency.

B. Threats to validity

External Validity: The generalizability of the results is a concern. The study effectively demonstrates the impact of adversarial attacks on CNNs using the MNIST dataset. Moreover, the same results hold forthe datasets like CIFAR-10, CIFAR-100 and Fashion-MNIST. The study provides valuable insights into the vulnerabilities of CNNs, which can inform further research in more varied contexts.

Internal Validity: The causal relationship between the treatment (adversarial attacks) and the observed effects (changes in performance metrics) needs careful examination. Other factors, such as the specific architecture of the CNN or the nature of the dataset, might also influence the outcomes. Ensuring that the observed effects are solely due to the adversarial attacks is crucial for accurate conclusions.

Construct Validity: Construct validity is essential in ensuring that the chosen performance metrics, such as accuracy, loss, ERGAS, PSNR, SSIM, and SAM, accurately depict the impact of adversarial attacks on CNNs. The main challenge lies in whether these metrics comprehensively represent the nuanced effects of such attacks, raising concerns about potential misinterpretations that could skew perceptions of CNN vulnerability and resilience. Adopting domain-specific metrics tailored to evaluate adversarial robustness, combined with a multi-dimensional analysis approach that encompasses a broader range of performance indicators, is considered the best option to mitigate these issues. This would offer a more holistic view of a model's behavior under adversarial conditions. Moreover, benchmarking the CNN's performance against baseline models under a variety of attack scenarios, coupled with the use of standardized adversarial robustness testing frameworks, can provide deeper insights into the network's strengths and weaknesses. Ensuring construct validity, therefore, involves a continuous process of validating and updating the assessment methods to align with evolving adversarial techniques, thereby maintaining the accuracy and relevance of conclusions drawn about CNN robustness.

VII. Related Work

Past research in the field of adversarial machine learning has made significant strides. Carlini et al. [28] provide a linearity-based theory for adversarial examples, proposes fast adversarial training, and refutes some alternative hypotheses. The view of adversarial examples as a fundamental property of linear models in high dimensions sparked significant subsequent research into understanding and improving model robustness. Xue et al. [29] provides a comprehensive analysis of contemporary threats to machine learning systems and defenses across the system lifecycle. It highlights open challenges like physical attacks and efficient privacy preservation. The review of evaluations and future directions makes this a wide-ranging resource for security in machine learning.

Xu et al. [30] provide a comprehensive review of adversarial attacks and defenses across multiple modalities including images, graphs, and text. The analysis of various attacks and security testing methods provides a foundation for choosing CNN hyperparameters that enhance resilience to adversarial attacks in image processing. Goodfellow et al. [10] outline a set of principles for evaluating the robustness of machine learning defenses against adversarial examples, emphasizing the importance of a well-defined threat model and skepticism towards one's own results. It advocates for rigorous testing using adaptive attacks, caution against security through obscurity, and the necessity of public code and model release for reproducibility. Additionally, they provide a checklist to avoid common pitfalls in such evaluations, encouraging comprehensive testing and comparison with existing work.

Wu and Zhu [31] provide useful insights into factors influencing adversarial transferability and proposes a simple but effective smoothed gradient attack to enhance it. The attack has implications on evaluating model robustness. Recent advancements, such as a cluster-based approach with a dynamic reputation system for Flying Ad hoc Networks [32] and a weighted, spider monkey-based optimization method for Vehicular Ad hoc Networks [33], have shown significant improvements in performance metrics, security, and reliability for CNN.

Our work aims to expand upon these foundations by specifically focusing on the impact of white-box adversarial attacks on CNN performance metrics. Unlike previous studies that broadly addressed adversarial threats in machine learning, our research delves into the detailed analysis of how these attacks affect CNNs, providing a more focused understanding of their vulnerabilities and potential defenses. This specificity in studying the direct effects of attacks on CNNs sets our work apart and underscores its importance in the broader context of machine learning security.

VIII. Conclusions and Future work

We showed that Convolutional Neural Networks exhibit significant vulnerability to a range of adversarial attacks, which lead to notable degradation in performance metrics like accuracy, loss, and image quality. The research underscores the importance of developing more resilient CNN architectures and defense mechanisms to counteract these vulnerabilities, particularly in critical applications where CNN reliability is paramount. The findings provide valuable insights for future

research aimed at enhancing the security and robustness of CNNs against sophisticated adversarial threats.

These test scenarios plays a crucial role in the development and refinement of defense mechanisms for Convolutional Neural Networks (CNNs). By subjecting models to a wide range of attack scenarios, we can observe the specific ways in which adversarial inputs manipulate model behavior. This insight is invaluable for devising defense strategies that directly counteract the observed vulnerabilities. For instance, testing can reveal if a model is particularly sensitive to slight perturbations in certain input features, leading to the development of input preprocessing or feature squeezing techniques as countermeasures. Similarly, the effectiveness of adversarial training can be assessed and optimized through iterative testing, by incorporating a diverse set of adversarial examples generated from the latest attack methods. Moreover, testing helps in evaluating the practicality of defense mechanisms under real-world conditions, ensuring that they do not unduly compromise model accuracy or performance. Through this iterative process of attack simulation, vulnerability assessment, and defense implementation, testing fosters a deeper understanding of adversarial threats and guides the creation of more robust and resilient systems.

ACKNOWLEDGMENT

This project was supported in part by the US National Science Foundation under award number OAC 1931363, and Grant No. 1822118 and 2226232.

REFERENCES

[1] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, "Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 998–1026, 2020.

[2] K. Kalaiselvi and M. Deepika, "Machine learning for healthcare diagnostics," *Machine Learning with Health Care Perspective: Machine Learning and Healthcare*, pp. 91–105, 2020.

[3] J. A. Cameron, P. Savoie, M. E. Kaye, and E. J. Scheme, "Design considerations for the processing system of a cnn-based automated surveillance system," *Expert Systems with Applications*, vol. 136, pp. 105–114, 2019.

[4] K. Židek, J. Pitel', M. Balog, A. Hošovskỳ, V. Hladkỳ, P. Lazorík, A. Iakovets, and J. Demčák, "Cnn training using 3d virtual models for assisted assembly with mixed reality and collaborative robots," *Applied Sciences*, vol. 11, no. 9, p. 4269, 2021.

[5] R. Haffar, N. M. Jebreel, J. Domingo-Ferrer, and D. Sánchez, "Explaining image misclassification in deep learning via adversarial examples," in *International Conference on Modeling Decisions for Artificial Intelligence*. Springer, 2021, pp. 323–334.

[6] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for text classification," *arXiv preprint arXiv:1712.06751*, 2017.

[7] Y. LeCun, C. Cortes, C. Burges *et al.*, "Mnist handwritten digit database," 2010.

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

[9] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[11] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[12] K. Alexey, "Adversarial examples in the physical world," *arXiv preprint arXiv: 1607.02533*, 2016.

[13] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[14] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. Ieee, 2017, pp. 39–57.

[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] M. S. Ayas, S. Ayas, and S. M. Djouadi, "Projected gradient descent adversarial attack and its defense on a fault diagnosis system," in *2022 45th International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, 2022, pp. 36–39.

[17] 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.

[18] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig *et al.*, "Adversarial robustness toolbox v1. 0.0," *arXiv preprint arXiv:1807.01069*, 2018.

[19] Q. Du, N. H. Younan, R. King, and V. P. Shah, "On the performance evaluation of pan-sharpening techniques," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 518–522, 2007.

[20] I. Alimuddin, J. T. S. Sumantyo, H. Kuze *et al.*, "Assessment of pan-sharpening methods applied to image fusion of remotely sensed multi-band data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 18, pp. 165–175, 2012.

[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[22] L. Alparone, L. Wald, J. Chanussot, C. Thomas, P. Gamba, and L. M. Bruce, "Comparison of pansharpening algorithms: Outcome of the 2006 grs-s data-fusion contest," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 10, pp. 3012–3021, 2007.

[23] Z. Wu, S.-N. Lim, L. S. Davis, and T. Goldstein, "Making an invisibility cloak: Real world adversarial attacks on object detectors," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 2020, pp. 1–17.

[24] K. Roshan, A. Zafar, and S. B. U. Haque, "Untargeted white-box adversarial attack with heuristic defence methods in real-time deep learning based network intrusion detection system," *Computer Communications*, vol. 218, pp. 97–113, 2024.

[25] L. Ye and S. M. Hamidi, "Thundernna: a white box adversarial attack," *arXiv preprint arXiv:2111.12305*, 2021.

[26] Sewar: A python package for image quality assessment. [Online]. Available: https://pypi.org/project/sewar/

[27] P. Ammann and J. Offutt, *Introduction to Software Testing*. Cambridge University Press, 2008. [Online]. Available: https://books.google.com/books?id=BMbaAAAAMAAJ

[28] 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.

[29] M. Xue, C. Yuan, H. Wu, Y. Zhang, and W. Liu, "Machine learning security: Threats, countermeasures, and evaluations," *IEEE Access*, vol. 8, pp. 74 720–74 742, 2020.

[30] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *International Journal of Automation and Computing*, vol. 17, pp. 151–178, 2020.

[31] L. Wu and Z. Zhu, "Towards understanding and improving the transferability of adversarial examples in deep neural networks," in *Asian Conference on Machine Learning*. PMLR, 2020, pp. 837–850.

[32] S. Gupta and N. Sharma, "Scfs-securing flying ad hoc network using cluster-based trusted fuzzy scheme," *Complex & Intelligent Systems*, pp. 1–20, 2024.

[33] D. Gupta and R. Rathi, "A novel spider monkey optimization for reliable data dissemination in vanets based on machine learning," *Sensors*, vol. 24, no. 7, p. 2334, 2024.