Analyzing Adversarial Vulnerabilities in Deep Learning: Unveiling Shapley Insights for Fine-Grained Pixel Analysis

Arooj Arif

***Abstract*—Deep learning researchers are currently facing challenges related to transparency and interpretability in AI models, especially in the context of adversarial attacks. Previous studies have explored the utility of SHAP signatures, but our research takes a fresh approach by utilizing these tools in a unique way. We aim to identify the essential pixels in images that play a crucial role in AI models' decision-making, particularly in scenarios where the models are under deceptive attack. Our method offers a new perspective on Explainable AI (XAI), and to the best of our knowledge, it is the first of its kind. By focusing on these critical pixels that we identify through SHAP analysis, we gain unprecedented insights into the vulnerabilities of deep learning models. This method enables us to understand and enhance the robustness of AI models against sophisticated manipulations. Our research opens up new avenues in the AI field and significantly contributes to the development of more secure, transparent, and reliable AI systems. This advancement is particularly critical in areas where the precision and dependability of AI decisions are crucial. Our work not only addresses the existing challenges in AI security but also establishes a precedent for future explorations in making AI systems more interpretable and trustworthy.**

I. Introduction

As artificial intelligence continues to advance quickly, protecting deep learning models from adversarial attacks has emerged as a key area of concern. These adversarial attacks, which change the raw data in small but deliberate ways, are a big problem for the security and dependability of AI systems. Deep learning models, especially Convolutional Neural Networks (CNNs), are easy targets for these attacks, even though they are very powerful in other ways. Because of this weakness, strong defences need to be built to protect these systems from these kinds of threats. [1] We present a new method that combines the best features of pre-trained CNNs with advanced adversarial simulations, SHAP (SHapley Additive exPlanations) analysis, and Explainable AI (XAI) signatures in our work. The goal of this multifaceted approach is to not only make deep learning models more reliable, but also to show how they make decisions. In this way, it meets the very important need for systems that are safe from malicious risks and also clear about how they work [3], [4], [5]. Putting SHAP analysis and XAI together is very important in this case. These techniques make it possible to understand complex models more nuancedly, making them clearer. This is becoming more and more important in high-stakes situations, like healthcare, where AI choices must be accurate and easy to understand, as well as in self-driving systems, where safety and dependability are very important [6]. Recent study by [6], which looks at adversarial attacks in the context of IoT network intrusion detection, is very similar to what we're doing because it focuses on AI systems that are strong and easy to understand. In addition, our suggested method is rigorously tested using a number of case studies. It is clear from these studies that the method works to find and reduce adversarial risks, which is a big step forward in the field of AI security. Our study helps us understand how to make AI systems both safe and easy to understand, which is a balance that is hard to achieve but necessary for AI technologies to be widely used. It does this by pointing out the inherent link between model robustness and explainability. A summary of this research shows that it not only addresses the problems that [1] and [2] pointed out in AI security, but it also builds on recent progress in the field, like the work of [4] and [5] , to suggest a new, useful way to make deep learning systems more reliable and clear.

II. Proposed Methodology:

The goal of our suggested model, which is shown in Figure 1, is to improve deep learning systems' resistance to adversarial assaults. Using the MNIST dataset as a training dataset, a pre-trained Convolutional Neural Network is used to focus on extraction of correct examples. The robustness of the model is then evaluated by subjecting it to exhaustive adversarial assault simulations through a variety of methodologies. Subsequently, a comprehensive SHAP analysis is conducted to gain insights into the specific impacts of individual pixels on the model's decision-making process. In the last phase, XAI signatures are used to identify critical pixel for attack generation nd detection. The implementation

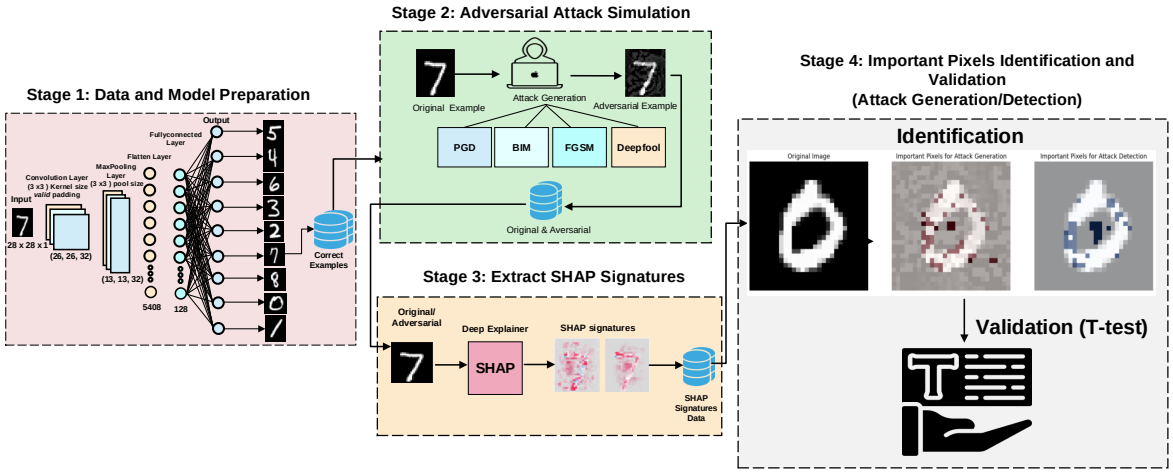


Fig. 1: Overview of the Proposed Model

of this multi-stage strategy guarantees a thorough and resilient protection against adversarial threats in machine learning models.

III. Stage 1: Data and Model Preparation

During the initial stage of our inquiry, we employed a convolutional neural network (CNN) that had been pre-trained on the MNIST dataset to classify digits. CNN's well-developed expertise in extracting features from picture data was essential for our investigation. One crucial preprocessing step entailed the selection of cases that were effectively classified by the model, resulting in the creation of a dataset consisting of precisely anticipated instances. The objective of this strategy was to separate the impacts of adversarial perturbations on a model that is otherwise behaving appropriately.

The dataset, after selection, exhibited a diverse class distribution: 973 instances with label 0, 1133 with label 1, 1016 with label 2, 989 with label 3, 969 with label 4, 882 with label 5, 937 with label 6, 1005 with label 7, 946 with label 8, and 984 with label 9. Although not perfectly balanced, this distribution mimics the natural frequency of digit occurrences in the actual world, providing a degree of scientific reliability to our study. The quality of our research depended on our acknowledging this little imbalance. Instead of testing the model in an artificially balanced environment, it enabled us to seriously evaluate its resilience to adversarial attacks in a real-world situation.

By focusing on correctly classified cases, we made sure that the next study was all about how adversarial attacks affect a model that is already good at what it does. The systematic methodology employed in the process of data selection established a well-defined and practical foundation for our inquiry. This allowed us to thoroughly examine the ability of neural networks to withstand adversarial manipulations in a manner that closely mirrors real-world scenarios.

IV. Stage 2: Adversarial Attack Simulation

By making adversarial examples, the goal of this step was to thoroughly test the model's stability. In order to learn how the neural network model reacts when exposed to intentionally misleading settings, this evaluation is crucial. The choice of epsilon numbers $\varepsilon = [0.03, 0.04, 0.05, 0.1, 0.2, 0.25]$ was a key part of our method. These numbers were picked to show a range of disturbance intensities:

- **Lower Epsilon Values** ($\varepsilon = 0.03, 0.04, 0.05$):Represent subtle but potentially effective perturbations, testing the model's sensitivity to minimal adversarial modifications.
- **Moderate Epsilon Value** ($\varepsilon = 0.1$): Reports the model's performance in a moderately strong attack and does so in a fair way.
- **Higher Epsilon Values** ($\varepsilon = 0.2, 0.25$): Check to see how well the model can handle more direct and damaging attempts to change it.

Choosing these values let us fully study how the model behaved under different levels of adversarial intensity, which is important for figuring out how robust it is overall. Utilized Foolbox, a Python library, to facilitate the generation of a wide range of adversarial attacks. Executed a series of adversarial attacks, including LinfBasicIterativeAttack (BIM), LinfFastGradientAttack (FGSM), LinfDeepFoolAttack, and LinfProjectedGradientDescentAttack (PGD). Systemat-

ically collected and analyzed the adversarial examples generated from these attacks.

A. Inclusion of the UMAP visualization:

In the analysis, we use a UMAP projection shown in Figure 2 to evaluate the performance of our neural network model. This includes its ability to handle adversarial examples generated by Projected Gradient Descent (PGD) attacks. The projection efficiently separates the ten-digit classes into distinct color-coded clusters (digits 0 to 9), indicating the model's pattern recognition capabilities. Figure 2 also reveals the strategic placement of adversarial examples. These examples are positioned within and around the digit clusters in a dispersed pattern, unlike the well-defined groupings of genuine digits. This pattern suggests that adversarial examples can deceive the model and lead to classification errors. The inclusion of Figure 2 in our analysis has a dual purpose. Firstly, it visually represents the deceptive nature of adversarial examples, highlighting how they can seamlessly blend with genuine data and mislead the model. Secondly, it emphasizes the need to enhance our model's ability to differentiate between authentic and adversarial data points. This visual evidence supports our findings and underscores the urgency of fortifying neural network architectures against adversarial intrusions. Ultimately, this can help in advancing security protocols within deep learning applications.

V. Stage 3: Extraction of XAI Signatures

A. Exploring Model Decision-Making

The main objective of this stage is to comprehend how individual input features, particularly pixels, affect the model's predictions. This process is crucial in order to distinguish the model's responses to both normal and adversarial inputs. Our approach involved:

- **Dataset Segmentation:** We selected subsets from our dataset, comprising normal examples and those modified by adversarial attacks, specifically focusing on an epsilon value of 0.2.
- **SHAP Deep Explainer Utilization:**We utilized SHAP Deep Explainer to generate XAI signatures, which expose the importance of each pixel in the model's predictions.
- **SHAP Value Calculation:** SHAP values were calculated for the selected subsets to measure the impact of each input feature on the model's output in normal and adversarial scenarios.

The SHAP values computed showed notable differences in influence patterns between normal and adversarial examples. This analysis clarifies how the model's decision-making process is altered under various input conditions.

Aspect	Attack Generation	Attack Detection
Objective	Identify model vulnerabilities to change predictions.	Recognize changes in interpretation from original to adversarial instances.
Focus	On influential features of adversarial examples.	On feature contribution differences between normal and adversarial examples.
Methodology	Analyze SHAP values of adversarial examples.	Compare SHAP values between normal and adversarial examples.
SHAP Values Used	From adversarial examples, indicating impactful features.	The difference in SHAP values between normal and adversarial examples.
Purpose	Craft adversarial examples exploiting vulnerabilities.	Detect and understand adversarial attacks' effects.
Insight Gained	Identifies model weaknesses and modification strategies.	Reveals decision-making changes due to attacks.

TABLE I: Key Differences Between Attack Generation and Detection Using SHAP Values

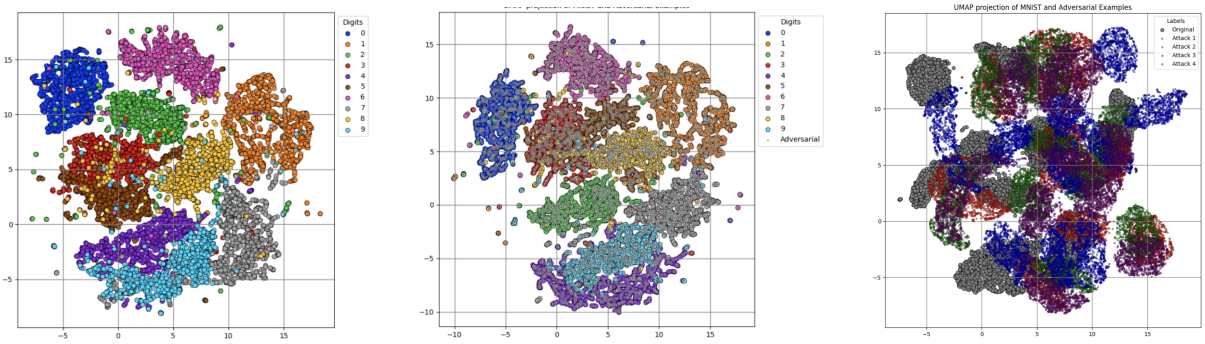
VI. Stage 4: Identification and Validation of Critical Pixels

A. SHAP Values for Critical Pixel Analysis

In Stage 4, we use the SHAP values that we extracted earlier to pinpoint and confirm the pixels that have a significant impact on the model's predictions. This step is essential for gaining a deeper understanding of the model's behavior and developing effective strategies to make it more resilient to adversarial attacks. The process involves:

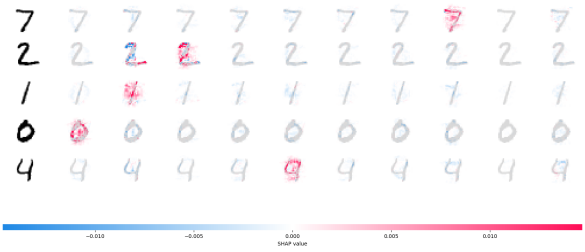
1) Attack Generation Pixel Analysis: We first analyze the SHAP values for adversarial image pixels. This step is crucial to identify which pixels, when altered, are most effective in generating successful adversarial attacks. By understanding the pixels that significantly influence the model's erroneous decisions, we can uncover how adversarial perturbations are guided and how they can be strategically crafted.

2) Attack Detection Pixel Analysis: Subsequently, we focus on detecting these adversarial manipulations by examining the differences in SHAP values between normal and adversarial images. This comparison highlights the pixels where the largest deviations occur, signaling potential areas of the image being exploited by adversarial attacks. This analysis is vital for developing robust detection mechanisms that can identify and counteract these manipulative changes.

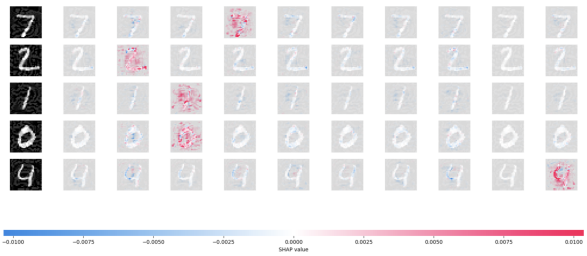


(a) Original MNIST digit classes (b) Original and PGD attack (c) Original and 4 attacks

Fig. 2: UMAP projections illustrating the separation of digit classes within the MNIST dataset and the integration of adversarial examples. The left panel (2a) shows the natural clustering of MNIST digits, the middle panel (2b) overlays PGD adversarial examples, and the right panel (2c) displays additional adversarial examples for comparison.



(a) SHAP heatmap for correct classifications highlighting influential pixels for the model's accurate predictions.Red indicates pixels increasing model confidence, and blue shows those decreasing it in correct classifications.



(b) SHAP heatmap for adversarial samples showing how attacks distort the model's focus, leading to potential misclassifications.

Fig. 3: SHAP heatmaps contrast the pixel influence on the model's decisions between normal and adversarial examples, underlining the adversarial impact.

3) Original Image Pixel Importance: Alongside these analyses, we also scrutinize the SHAP values of original, unmanipulated images. This examination is essential to establish a baseline of the model's interpretation of unperturbed images. Understanding the importance of pixels in the original context provides a reference point, helping us to better interpret the changes observed in the adversarial context.

Through this comprehensive methodology, we aim to provide a deep understanding of how pixel-level manipulations affect AI decision-making, both in generating and detecting adversarial attacks, as well as understanding the inherent behavior of the model under normal conditions.

B. Statistical Validation of Critical Pixels

To substantiate the distinction between critical and non-critical pixels identified in our analysis, we conducted a statistical t-test on the SHAP values. This test helps validate the significance of the differences observed in the SHAP values between critical and non-critical pixels. The implementation of a t-test ensures that the observed disparities are not due to random chance, thereby lending statistical rigor to our findings and confirming the reliability of our pixel importance assessments.

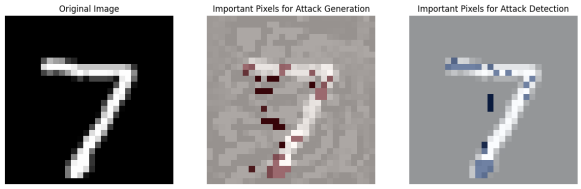


Fig. 4: Visualization of critical pixels in adversarial samples.

VII. Case Studies:

This section outlines various case studies to validate the methodology.

SHAP Value Proximity to Zero	Interpretation
Close to Zero	• Feature has minimal impact on the model's prediction. • Changes in the feature value do not significantly affect the prediction. • Feature is considered neutral or weak in terms of prediction influence.
Far from Zero (Positive or Negative)	• Feature has a substantial impact on the model's prediction. • Variations in the feature value lead to significant changes in the prediction. • Feature is a strong influencer of the predicted outcome. • The magnitude of the SHAP value indicates the strength of the feature's influence.

TABLE II: Interpretations of SHAP Values Based on Proximity to Zero

A. Case Study 1: Evaluating Model Robustness Against Different Adversarial Attacks

The main objective of this case study is to evaluate the robustness of the MNIST classification model against different kinds of adversarial attacks such as Basic Iterative Method (BIM), Fast Gradient Sign Method (FGSM), DeepFool, and Projected Gradient Descent (PGD). The focus of the study is to compare the performance of the MNIST model before and after being subjected to these adversarial attacks. The performance will be measured by the number of misclassifications that occur at different levels of perturbation, as measured by the epsilon values. The study aims to provide insights into the vulnerabilities of the MNIST model to different types of adversarial attacks and to determine the effectiveness of these attacks at varying intensities. Figure 5 represents the model's accuracy against various types of attacks. The graph shows how the accuracy of the model decreases with the increasing intensity of the perturbation, which is measured by epsilon. This trend of decreasing robust accuracy is observed for all types of attacks.

In Figure.5, it is clear that the robust accuracy for FGSM and DeepFool attacks declines slower at lower epsilon values but drops significantly at higher values. This indicates that these attacks are less effective at lower perturbations but can still compromise the model eventually. On the other hand, BIM and PGD attacks show a more consistent and gradual decrease in robust accuracy, indicating a steady effectiveness at all levels of perturbation. Table III provides a numerical representation of the model's vulnerabilities by listing the total number of misclassifications for each type of attack at various epsilon values. Together, the table and the figure give a comprehensive view of the model's performance under adversarial conditions.

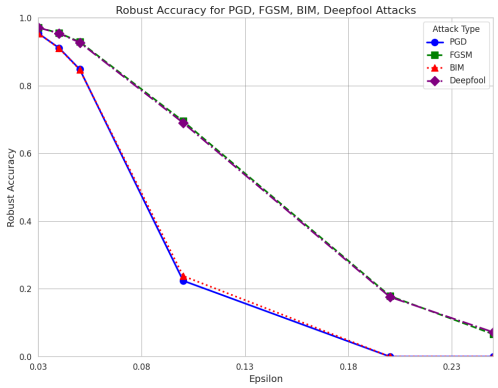


Fig. 5: Robust Accuracy for PGD, FGSM, BIM, Deepfool Attacks

TABLE III: Summary of Total Misclassifications for Various Adversarial Attacks

Attack Type	Epsilon					
	0.03	**0.04**	**0.05**	**0.10**	**0.20**	**0.25**
BIM	451	881	1508	7490	9834	9834
FGSM	284	440	702	2998	8070	9186
Deepfool	289	449	713	3045	8104	9130
PGD	451	881	1483	7616	9834	9834

The findings emphasize the importance of implementing advanced defensive measures to enhance the robustness of machine learning models against adversarial attacks. The observations made in this case study can help in the creation of such defenses, especially in addressing the weaknesses exposed by

BIM and PGD when dealing with smaller perturbations, and the slower but eventual impact of FGSM and DeepFool when dealing with higher perturbations.

B. Critical Analysis of Class-Specific Metrics

In the field of adversarial machine learning, it's important to evaluate how well a model performs when faced with attacks. While overall metrics like robust accuracy and misclassification rates give a general idea of a model's performance, it's important to also examine class-specific metrics like F1-Score, Precision, and Recall. These metrics help to identify specific vulnerabilities in the model, especially in a security context where false positives and false negatives can have different implications. To get a more detailed understanding of the model's performance, we've compiled a table Table. IV that shows the F1-Score, Precision, and Recall for each class under different types of attacks and perturbations. This table not only demonstrates the model's overall resilience but also highlights specific weaknesses in certain classes that may be overlooked in more general analyses.

As an example, high precision rate during an attack indicates that when a model predicts a specific class, it is likely accurate, although it may miss out on identifying all true instances (resulting in lower recall). On the other hand, high recall with low precision suggests a model that is prone to false alarms by classifying non-members as belonging to the targeted class. The F1-Score metric balances precision and recall to provide a single measure.

Upon careful analysis of the table, it has been observed that certain classes are more susceptible to specific types of attacks. This highlights the necessity of implementing targeted defensive strategies for these classes. For instance, classes that exhibit a significant drop in F1-Score when attacked with FGSM are particularly vulnerable to the perturbations caused by this method. This knowledge can inform the creation of tailored training data or the implementation of defense mechanisms specific to these classes. Furthermore, the variation in robustness between classes can guide the allocation of defense resources. Classes with lower robustness metrics represent weak points in the model's defenses and could benefit from focused defense measures. In conclusion, the comprehensive evaluation of the model's robustness provided by the detailed metrics, robust accuracy trends, and misclassification insights in the table is essential in developing effective defenses against adversarial attacks. This multifaceted assessment ensures that the model remains accurate and trustworthy in the face of ever-evolving adversarial challenges.

Epsilon	Class	Metrics											
		F1-Score BIM	F1-Score Deepfool	F1-Score FGSM	F1-Score PGD	Precision BIM	Precision Deepfool	Precision FGSM	Precision PGD	Recall BIM	Recall Deepfool	Recall FGSM	Recall PGD
0.03	0	0.9842	0.9872	0.9872	0.9842	0.9777	0.9817	0.9817	0.9777	0.9908	0.9928	0.9928	0.9908
0.03	1	0.9690	0.9788	0.9792	0.9664	0.9454	0.9601	0.9609	0.9444	0.9938	0.9982	0.9982	0.9894
0.03	2	0.9446	0.9675	0.9669	0.9442	0.9405	0.9694	0.9693	0.9396	0.9488	0.9656	0.9646	0.9488
0.03	3	0.9612	0.9778	0.9783	0.9627	0.9588	0.9749	0.9749	0.9589	0.9636	0.9808	0.9818	0.9666
0.03	4	0.9558	0.9717	0.9722	0.9573	0.9519	0.9692	0.9712	0.9548	0.9598	0.9742	0.9732	0.9598
0.03	5	0.9557	0.9715	0.9726	0.9544	0.9349	0.9571	0.9582	0.9357	0.9773	0.9864	0.9875	0.9739
0.03	6	0.9693	0.9796	0.9801	0.9693	0.9793	0.9860	0.9870	0.9783	0.9594	0.9733	0.9733	0.9605
0.03	7	0.9455	0.9682	0.9682	0.9459	0.9498	0.9682	0.9663	0.9517	0.9413	0.9682	0.9701	0.9403
0.03	8	0.9233	0.9464	0.9480	0.9234	0.9731	0.9807	0.9830	0.9720	0.8784	0.9144	0.9154	0.8795
0.03	9	0.9289	0.9550	0.9560	0.9301	0.9351	0.9619	0.9619	0.9335	0.9227	0.9482	0.9502	0.9268
0.04	0	0.9708	0.9826	0.9821	0.9718	0.9664	0.9766	0.9757	0.9693	0.9753	0.9887	0.9887	0.9743
0.04	1	0.9432	0.9708	0.9716	0.9367	0.9071	0.9464	0.9479	0.9026	0.9823	0.9965	0.9965	0.9735
0.04	2	0.8994	0.9461	0.9471	0.8980	0.8840	0.9424	0.9434	0.8822	0.9154	0.9498	0.9508	0.9144
0.04	3	0.9192	0.9596	0.9611	0.9185	0.9124	0.9577	0.9597	0.9081	0.9262	0.9616	0.9626	0.9292
0.04	4	0.9098	0.9568	0.9579	0.9116	0.8984	0.9529	0.9539	0.9028	0.9216	0.9607	0.9618	0.9205
0.04	5	0.8995	0.9546	0.9546	0.9010	0.8550	0.9320	0.9320	0.8614	0.9490	0.9785	0.9785	0.9444
0.04	6	0.9373	0.9698	0.9703	0.9402	0.9587	0.9804	0.9814	0.9579	0.9167	0.9594	0.9594	0.9232
0.04	7	0.9073	0.9512	0.9513	0.9042	0.9141	0.9521	0.9494	0.9162	0.9005	0.9502	0.9532	0.8925
0.04	8	0.8493	0.9163	0.9202	0.8527	0.9491	0.9704	0.9751	0.9447	0.7685	0.8679	0.8710	0.7770
0.04	9	0.8548	0.9304	0.9314	0.8586	0.8729	0.9371	0.9390	0.8721	0.8374	0.9238	0.9238	0.8455
0.05	0	0.9532	0.9750	0.9750	0.9559	0.9546	0.9686	0.9686	0.9549	0.9517	0.9815	0.9815	0.9568
0.05	1	0.9095	0.9534	0.9542	0.9031	0.8583	0.9169	0.9177	0.8538	0.9673	0.9929	0.9938	0.9585
0.05	2	0.8318	0.9189	0.9198	0.8318	0.8061	0.9118	0.9136	0.8105	0.8593	0.9262	0.9262	0.8543
0.05	3	0.8496	0.9311	0.9321	0.8532	0.8215	0.9260	0.9279	0.8292	0.8797	0.9363	0.9363	0.8787
0.05	4	0.8547	0.9300	0.9325	0.8589	0.8390	0.9143	0.9180	0.8481	0.8710	0.9463	0.9474	0.8700
0.05	5	0.8297	0.9208	0.9214	0.8325	0.7735	0.8825	0.8827	0.7838	0.8946	0.9626	0.9637	0.8878
0.05	6	0.8966	0.9533	0.9555	0.9047	0.9361	0.9701	0.9724	0.9402	0.8602	0.9370	0.9392	0.8719
0.05	7	0.8319	0.9297	0.9294	0.8313	0.8443	0.9320	0.9285	0.8494	0.8199	0.9273	0.9303	0.8139
0.05	8	0.7301	0.8671	0.8700	0.7372	0.8955	0.9566	0.9592	0.8739	0.6163	0.7928	0.7960	0.6374
0.05	9	0.7467	0.8838	0.8847	0.7568	0.7680	0.9069	0.9089	0.7691	0.7266	0.8618	0.8618	0.7449
0.1	0	0.5609	0.8876	0.8888	0.5961	0.7279	0.9032	0.9078	0.7781	0.4563	0.8726	0.8705	0.4830
0.1	1	0.3045	0.8518	0.8484	0.1693	0.3256	0.7599	0.7550	0.1887	0.2860	0.9691	0.9682	0.1536
0.1	2	0.2387	0.6409	0.6450	0.2272	0.2046	0.6216	0.6259	0.1887	0.2864	0.6614	0.6654	0.2854
0.1	3	0.2356	0.6673	0.6743	0.2389	0.2168	0.6361	0.6414	0.2248	0.2578	0.7017	0.7108	0.2548
0.1	4	0.3187	0.7195	0.7239	0.3181	0.2971	0.6844	0.6897	0.2896	0.3437	0.7585	0.7616	0.3529
0.1	5	0.2421	0.6843	0.6859	0.2167	0.2023	0.5861	0.5885	0.1805	0.3016	0.8220	0.8268	0.2799
0.1	6	0.3469	0.7821	0.7838	0.3541	0.3495	0.7862	0.7897	0.3592	0.3445	0.7781	0.7816	0.3487
0.1	7	0.2639	0.7425	0.7388	0.2636	0.2443	0.7002	0.6962	0.2442	0.2850	0.7899	0.7849	0.2852
0.1	8	0.1718	0.5624	0.5684	0.1669	0.1545	0.5486	0.5554	0.1630	0.1908	0.5767	0.5836	0.1823
0.1	9	0.1869	0.6064	0.6067	0.1825	0.1791	0.5892	0.5903	0.1749	0.2007	0.6307	0.6312	0.1960
0.15	0	0.1935	0.7615	0.7528	0.1575	0.1092	0.4804	0.4791	0.1029	0.4565	0.9307	0.9198	0.0896
0.15	1	0.0709	0.6661	0.6512	0.0337	0.0471	0.4160	0.4070	0.0240	0.0512	0.9329	0.9247	0.0160
0.15	2	0.0679	0.4026	0.3763	0.0367	0.0379	0.2362	0.2215	0.0195	0.0349	0.7530	0.7172	0.0295
0.15	3	0.0761	0.5076	0.4773	0.0381	0.0407	0.2717	0.2545	0.0238	0.0371	0.6646	0.6317	0.0250
0.15	4	0.1186	0.5514	0.5331	0.0652	0.0781	0.2929	0.2822	0.0485	0.1094	0.6452	0.6279	0.0863
0.15	5	0.0650	0.4053	0.3756	0.0365	0.0364	0.2327	0.2200	0.0190	0.0345	0.7612	0.7270	0.0501
0.15	6	0.1199	0.5737	0.5454	0.0700	0.0716	0.3386	0.3220	0.0521	0.1091	0.6117	0.5798	0.0883
0.15	7	0.0862	0.4992	0.4735	0.0435	0.0453	0.2627	0.2486	0.0272	0.0418	0.6993	0.6636	0.0306
0.15	8	0.0392	0.2970	0.2852	0.0233	0.0235	0.1733	0.1641	0.0142	0.0213	0.5872	0.5625	0.0186
0.15	9	0.0418	0.2780	0.2645	0.0248	0.0247	0.1542	0.1460	0.0136	0.0224	0.6013	0.5675	0.0203
0.2	0	0.0703	0.6920	0.6672	0.0430	0.0385	0.4683	0.4532	0.0262	0.0273	0.9781	0.9657	0.0159
0.2	1	0.0247	0.4637	0.4494	0.0125	0.0140	0.3394	0.3267	0.0065	0.0073	0.9782	0.9684	0.0035
0.2	2	0.0262	0.2206	0.2057	0.0121	0.0126	0.1392	0.1307	0.0062	0.0066	0.8852	0.8351	0.0029
0.2	3	0.0277	0.3064	0.2876	0.0141	0.0147	0.1912	0.1797	0.0071	0.0074	0.8694	0.8184	0.0036
0.2	4	0.0414	0.3921	0.3736	0.0219	0.0226	0.2323	0.2189	0.0107	0.0111	0.8492	0.8054	0.0052
0.2	5	0.0237	0.2087	0.1929	0.0115	0.0119	0.1276	0.1192	0.0058	0.0061	0.8889	0.8524	0.0030
0.2	6	0.0464	0.4479	0.4291	0.0242	0.0250	0.2796	0.2648	0.0118	0.0122	0.8348	0.7989	0.0061
0.2	7	0.0330	0.2947	0.2750	0.0166	0.0171	0.1900	0.1779	0.0084	0.0087	0.8596	0.8159	0.0042
0.2	8	0.0167	0.1629	0.1540	0.0083	0.0085	0.1239	0.1170	0.0054	0.0057	0.7417	0.7070	0.0036
0.2	9	0.0149	0.1434	0.1339	0.0076	0.0077	0.1121	0.1052	0.0050	0.0053	0.7470	0.7096	0.0029
0.25	0	0.0175	0.3467	0.3191	0.0090	0.0088	0.2839	0.2619	0.0049	0.0047	0.9914	0.9714	0.0024
0.25	1	0.0092	0.1963	0.1750	0.0049	0.0048	0.1289	0.1161	0.0025	0.0024	0.9883	0.9725	0.0013
0.25	2	0.0076	0.1060	0.0940	0.0032	0.0031	0.0708	0.0633	0.0017	0.0016	0.9912	0.9881	0.0010
0.25	3	0.0082	0.1311	0.1175	0.0034	0.0033	0.0900	0.0804	0.0018	0.0017	0.9849	0.9747	0.0010
0.25	4	0.0142	0.1885	0.1677	0.0059	0.0057	0.1228	0.1106	0.0029	0.0028	0.9634	0.9715	0.0015
0.25	5	0.0065	0.1045	0.0923	0.0030	0.0029	0.0702	0.0620	0.0016	0.0015	0.9906	0.9888	0.0010
0.25	6	0.0176	0.2197	0.1954	0.0075	0.0073	0.1170	0.1040	0.0027	0.0026	0.9635	0.9732	0.0014
0.25	7	0.0134	0.1649	0.1473	0.0057	0.0056	0.0943	0.0842	0.0023	0.0022	0.9705	0.9731	0.0013
0.25	8	0.0044	0.0710	0.0645	0.0020	0.0019	0.0429	0.0387	0.0010	0.0010	0.9621	0.9569	0.0005
0.25	9	0.0040	0.0644	0.0576	0.0018	0.0017	0.0382	0.0343	0.0009	0.0008	0.9648	0.9571	0.0004

TABLE IV: Class-Specific Performance Metrics under Adversarial Attacks: This table delineates the F1-Score, Precision, and Recall for each MNIST class (0 through 9) across a spectrum of adversarial attacks at varying levels of epsilon perturbations. The metrics provide insights into the differential impact of each attack (BIM, Deepfool, FGSM, and PGD) on the classifier's ability to robustly identify each digit, underscoring the nuanced vulnerabilities and resilience of the model on a class-by-class basis.

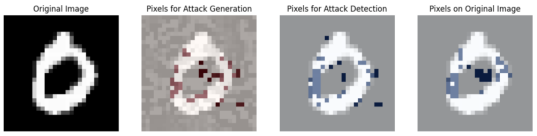
C. Pixel-Level Analysis for Enhancing AI Security Against Adversarial Attacks

1) Analysis of Individual Pixels: We first applied the SHAP values analysis to individual examples, focusing on the difference between normal and adversarial images. By examining a specific instance (index 3 in our dataset), we gained detailed insights into how individual pixels influenced the AI model's decision-making process. The visualization of these critical pixels (Figure 6a) reveals that certain pixels have a more pronounced impact on the model's response to adversarial attacks. This individual analysis allows us

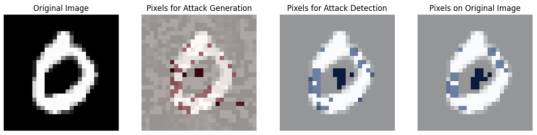
to pinpoint exact vulnerabilities in the AI model for a given input.

2) Aggregate Analysis across Multiple Examples: In contrast, our aggregate analysis involved averaging SHAP values across a set of images to identify commonly influential pixels. This approach provided a broader view, highlighting general patterns and shared vulnerabilities across multiple examples. The resulting heatmaps (Figure 6b) depict a more generalized perspective, essential for understanding overarching weaknesses in the model's interpretability regarding adversarial attacks.

3) Implications of the Findings: The comparison between individual and aggregate analysis underscores the multifaceted nature of AI vulnerabilities. While individual analysis offers a deep dive into specific instances, providing precise insights for tailored defenses, aggregate analysis exposes widespread trends, guiding more general strategies for enhancing AI robustness. These complementary perspectives are crucial in developing comprehensive defense mechanisms against adversarial attacks in AI systems.



(a) Individual Pixel Analysis. This image demonstrates the critical pixels identified in a single example, highlighting their influence on the AI model's decision-making process in response to an adversarial attack.



(b) Aggregate Pixel Analysis. This heatmap represents the average influence of pixels across multiple examples, revealing common patterns and vulnerabilities in the AI model's interpretation of images.

Fig. 6: Comparative Visualization of Individual and Aggregate Pixel Analysis.

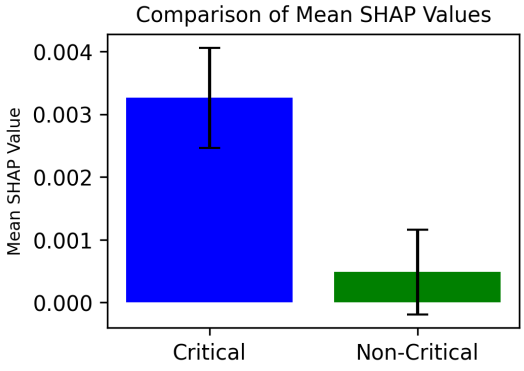
D. Case Study 3: Validation of Identified Critical Pixels

In Section VI, we discussed the critical pixels identified through our methodology for both attack generation and detection. To determine their significance, we conducted a rigorous statistical validation using a two-sample t-test. This validation step is crucial to highlight the critical role played by these pixels in shaping the decision-making process of our machine learning model.

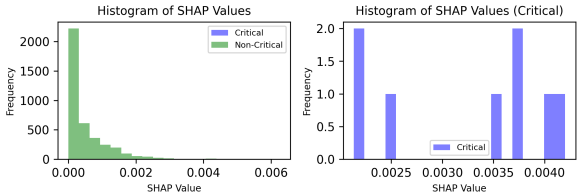
Our approach involved a comprehensive examination of SHAP values, with specific emphasis on those related to attack detection. SHAP values are useful in understanding the inner workings of our model, and they shed light on the pixels that significantly influence its predictions. To streamline the analysis, we categorized these SHAP values into two distinct groups: critical pixels and non-critical pixels. We based this categorization on a set of indices derived from our methodology. Figure 7a displays a bar chart that compares the means of two groups, highlighting the difference in mean SHAP values for 'Critical' and 'Non-Critical' pixels. This visual representation is critical to our analysis. Figure 7b shows histograms of SHAP values that provide a closer look at the distribution of SHAP values for both 'Critical' and 'Non-Critical' pixels. These histograms help us to understand the data distribution and its impact on statistical tests. The validation of our study primarily depends on the comparison of two groups using statistical analysis. We used the two-sample t-test, a well-known method for identifying significant differences between two sets of data. Our results were significant with a computed t-statistic of 11.5968 and an extremely low p-value of 1.34×10^{-30}. Figures 7a and 7b show the statistical indicators that confirm the crucial role of critical pixels in our model's behavior. These pixels influence both the generation and detection of adversarial attacks.

VIII. Conclusion:

This research conducted a thorough investigation into adversarial attack simulations, analyzing the vulnerabilities of deep learning models at the pixel level. By analyzing these attacks in detail, the researchers gained insights into their intricacies. Afterwards, the power of SHAP (SHapley Additive exPlanations) analysis was used to extract Shapley values, providing a deep understanding of how each pixel contributes to model predictions. With these Shapley signatures, critical pixels within the model's decision-making process were identified. This sequential approach, starting with attack simulations, followed by Shapley analysis, and culminating in identifying critical pixels, has enriched our comprehension of adversarial threats and laid the foundation for enhancing the robustness of AI models. In conclusion, this study significantly advances our knowledge in this domain. It offers a comprehensive framework for strengthening AI models against adversarial intrusions, thereby contributing to the broader field of deep learning security.



(a) Comparison of Mean SHAP Values. The bar chart illustrates a comparison between the mean SHAP values for 'Critical' and 'Non-Critical' groups. Error bars represent standard deviations. This graph is essential for assessing the significance of the differences between these groups.



(b) Histograms of SHAP Values. In the left subplot, the histogram depicts the distribution of SHAP values for both 'Critical' (in blue) and 'Non-Critical' (in green) groups. The right subplot zooms in on the 'Critical' group's SHAP values. These histograms provide insights into the data distribution and its impact on statistical tests

Fig. 7: SHAP values into critical and non-critical groups

REFERENCES

[1] Akhtar, N., Mian, A. (2018). Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. IEEE Access. Link

[2] Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J. (2017). Boosting Adversarial Attacks with Momentum. IEEE/CVF Conference on Computer Vision and Pattern Recognition. Link

[3] Lee, K., Lee, K., Lee, H., Shin, J. (2018). A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. ArXiv. Link

[4] Morris, J. X., Lifland, E., Yoo, J. Y., Grigsby, J., Jin, D., Qi, Y. (2020). TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. Conference on Empirical Methods in Natural Language Processing. Link

[5] Croce, F., Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. International Conference on Machine Learning. Link

[6] Zhou, X., Liang, W., Li, W., Yan, K., Shimizu, S., Wang, K. (2022). Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System. IEEE Internet of Things Journal. Link

[7] Entezari, N., Al-Sayouri, S. A., Darvishzadeh, A., Papalexakis, E. (2020). All You Need Is Low (Rank): Defending Against Adversarial Attacks on Graphs. Web Search and Data Mining. Link

[8] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," ArXiv e- prints, Feb. 2018.