An Evolutionary Algorithm Guided Search of Chained Adversarial Attacks on Generative Adversarial Networks

1st Feby Elizabeth John
Department of Computer Science and Engineering
Amrita School of Computing, Coimbatore
Amrita Vishwa Vidyapeetham, India
ORCID: 0009-0003-2288-3270
cb.sc.p2aie24031@cb.students.amrita.edu

2nd Sharath S R
Department of Computer Science and Engineering
Amrita School of Computing, Coimbatore
Amrita Vishwa Vidyapeetham, India
ORCID: 0009-0002-4692-2548
sharathravikumar26@gmail.com

3rd Ritwik Murali
Department of Computer Science and Engineering
Amrita School of Computing, Coimbatore
Amrita Vishwa Vidyapeetham, India
ORCID: 0000-0002-1269-2257
m_ritwik@cb.amrita.edu

***Abstract*—Though Generative Adversarial Networks (GANs) are crucial in medical imaging to address data scarcity, they are vulnerable to adversarial attacks. This work exploits the exploratory capabilities of Evolutionary Algorithms (EAs) to search for optimal white-box adversarial attack chains, applied to the latent space of a medical-image augmentation GAN during its training. As part of the same, three standard white-box attacks are adapted to affect Deep Convolutional GANs (DCGANs) that generate augmented histopathological images. A sensitivity analysis is conducted across six EA configurations, to study the behavior of EA in this experimental setup. Results show that EA behavior depends strongly on hyperparameter choices and the right hyperparameters help establish a balance between exploration and exploitation with EAs. This work highlights the risk of latent-space attack chaining for medical-image GANs and the potential for using EAs to explore the search space and identify optimal chained attack sequences.**

***Index Terms*—Combinatorial Optimization, Adversarial Attacks, Evolutionary Algorithm, Generative Adversarial Networks**

I. Introduction

To address the challenge of data scarcity to train classification models in medical imaging field, since their inception, Generative Adversarial Networks (GANs) [1] have emerged as powerful tools due to their data generation capability. This synthetic data can enhance the performance and robustness of medical classifier models. However, the security and robustness of both GANs and the consequent downstream classifiers trained on their generated outputs are critical in the medical domain. A major threat to security of Deep Neural Networks comes from adversarial attacks [16], [17]. While there has been extensive research focusing on attacking classifier models using several white-box attacks [18]–[20], recent studies have shown that GANs are also susceptible to adversarial manipulation, including attacks on the latent space and backdoor injections [21]–[23].

While many studies have developed various adversarial attacks, this paper explores a new idea of chaining existing white-box adversarial attacks to investigate their impact on the performance of GANs and the consequent classifiers trained on these poisoned GANs. This search for powerful attack sequences is a combinatorial search problem, which becomes computationally expensive as the number of attacks considered grows. This is where Evolutionary Algorithms (EAs) [24]–[26] offer a powerful approach for optimization in complex search spaces [27], [28].

EAs have been used in various studies such as GAN-related optimization [29], malware evolution [30], and particularly to develop new adversarial attacks such as GenAttack [31], showing how effective evolutionary strategies are to fool image classification models in black-box settings. However, the application of EAs to explore the potential of sequentially chaining white-box attacks to directly affect the ability of GANs to produce acceptable augmented medical images remains unexplored.

To address these gaps, this paper makes the following key contributions:

1) Formalize the problem of sequential white-box adversarial attacks against medical-imaging GANs as a search and optimization challenge suitable for evolutionary algorithms, addressing the limitations of combinatorial optimization.
2) Conduct a sensitivity analysis by experimenting with different configurations of EA hyperparameters to explore EA behavior, showing that effective evolutionary search depends on careful parameter tuning.

Note: Both the data & code for the experiments will be made available upon acceptance of this work for presentation.

II. Related Work

The introduction of GANs by Goodfellow et al. [1] was a significant step towards synthetic data generation. Early challenges with training instability and mode collapse [5] led to several improved variants, including Deep Convolutional GANs (DCGANs) [3], Wasserstein GANs (WGANs) [4], and Progressive Growing GANs [7]. In medical imaging, GANs have proven to be crucial for image synthesis, addressing data scarcity challenges [8]. GANs have been used for data augmentation of images across various image modalities, including MRI, CT, and histopathology [9], [10]. Studies have shown that data augmentation by GANs can greatly improve performance of medical image classifiers; for instance, GAN-augmented datasets significantly improve liver lesion classification accuracy [12]. Among the popular GANs used in medical imaging, the DCGAN architecture is known for its stability [11], [14].

However, the discovery of adversarial vulnerabilities in deep neural networks [16] was game-changing in the domain of adversarial learning. Some white-box attacks include the Fast Gradient Sign Method (FGSM) [17], its iterative variants like Projected Gradient Descent (PGD) [18], and more targeted approaches such as the Jacobian-based Saliency Map Attack (JSMA) [20].

While these attacks were originally designed for discriminative classifiers, their applicability to generative models such as GANs remains underexplored. Many studies have explored various attack and defense strategies specifically for generative models [22]. However, most research focuses on single attacks rather than systematic exploration of chained white-box attacks that can target GANs by adapting them to a GAN's latent space. More recently, a study has explored ensemble-based approaches and attack sequences have to improve transferability across models [32], but the focus is on black-box classifier attacks rather than generative models.

The existing literature thus shows various gaps. While there are many standard white-box attacks designed for classifiers, how they can be modified to attack GANs remains underexplored. Existing research on chaining adversarial attacks uses an ensemble of black-box attacks and focuses on transferability, rather than exploring whether chaining of attacks will have a compounding effect on the GAN's performance quality. Moreover, there is huge scope for systematic study of how EAs can optimize the search for powerful sequences of existing white-box attacks that can most harmfully affect the performance of a medical-imaging GAN.

III. Experimental Setup and Methodology

A. Methodology

The proposed study used a DCGAN for medical-image augmentation and a downstream ResNet50 classifier to study the degradation in classification performance when inputted with the poisoned DCGAN's synthetic data. This study used the benchmark histopathological LC25000 dataset [15], containing 25,000 histopathology images of malignant and benign lung and colon tissues.

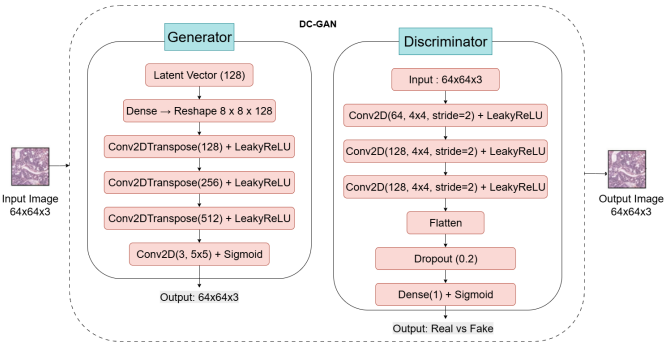

Fig. 1. Architecture describing the generator and discriminator of our DCGAN

First, the DCGAN is trained on the dataset and generated 64×64 synthetic histopathological images. As shown in Fig. 1, the generator progressively upsampled a latent representation through transposed convolutional layers to produce three channel RGB outputs. The discriminator classified the images using a series of convolutional layers with downsampling. Training was conducted for 50 epochs using Adam optimization and binary cross entropy loss.

For the ResNet50 model, the final fully-connected layer was replaced and a dropout layer (p=0.5) was added before classification to prevent overfitting. Training was performed for 50 epochs using the Adam optimizer, cross entropy loss, and a learning-rate scheduler. The model achieved a high accuracy of 98.9% on the dataset. Table I provides the hyperparameters used for the DCGAN and ResNet50 classifier.

TABLE I
Training Hyperparameters for DCGAN and ResNet Classifier

Parameter	DCGAN	ResNet-50
Learning Rate	1×10^{-4}	1×10^{-4}
Optimizer	Adam	Adam
Weight Decay	–	1×10^{-4}
Loss Function	Binary Cross-Entropy	Cross-Entropy
Epochs	50	50 (with early stopping)
Dropout Rate	0.2 (Discriminator)	0.5 (Classifier)
Learning Rate Scheduler	–	ReduceLROnPlateau

The unattacked DCGAN's image quality is quantified using a metric called Fréchet ResNet Distance (FRD), which is a custom modification of the original Fréchet Inception Distance (FID) using features extracted from our ResNet50 model, also trained on the histopathological images. The rate of misclassification, i.e., the proportion of images misclassified by the ResNet50 classifier, is quantified as the Attack Success Rate (ASR) metric.

For the adversarial attacks, this study adapted three standard white-box adversarial attacks, Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), and Jacobian-based Saliency Map Attack (JSMA), to target the DCGAN's latent space during training.

To investigate how impactful sequential combinations of at-

tacks are on the DCGAN and downstream classifier, the search for harmful attack chains is formulated as a combinatorial optimization problem. An attack sequence S of length L is defined as an ordered list of attacks: $S = [A_1, A_2, \ldots, A_L]$, where $A_i \in \{\text{FGSM}, \text{PGD}, \text{JSMA}\}$. In this experimental setup that uses 3 attacks, the total size of search space grows exponentially as 3^L, making exhaustive search infeasible as the sequence length L increases. An EA is thus employed to search for optimal attack sequences (e.g., FGSM $\rightarrow$ JSMA $\rightarrow$ PGD). A fitness function F evaluates each attack sequence as a weighted combination of ASR and FRD.

A sensitivity analysis was then conducted to explore the behavior of the evolutionary algorithm across different hyperparameter configurations. EA hyperparameters such as population size ($N \in [5, 10]$) and mutation rate ($p_m \in [0.46, 0.5, 0.6]$) were studied for a fixed sequence length ($L = 3$). This analysis identifies parameter configurations that efficiently discover potent attack sequences while maintaining computational feasibility.

B. Adapted Attacks

A major contribution of this study is the modification of the three chosen standard white-box adversarial attacks (FGSM, PGD, and JSMA) to attack the DCGAN's latent space during training. The idea behind this modification is that, in a GAN, the latent space is a lower-dimensional vector space that serves as the input to the Generator of the GAN. Modifying this input latent vector z effectively can alter the synthesized output of the GAN, thus generating poisoned images.

Therefore, given a latent vector $\mathbf{z} \sim \mathcal{N}(0, I)$, the generator produces an image $G(\mathbf{z})$, which is evaluated by the discriminator D. The adversarial objective is defined using the standard generator loss function given in equation 1.

$$\mathcal{L}_{\text{adv}}(\mathbf{z}) = -\log D(G(\mathbf{z})), \tag{1}$$

This implies that the attacker aims to craft a perturbation $\boldsymbol{\delta}$ such that the adversarial latent vector $\mathbf{z}_{adv} = \mathbf{z} + \boldsymbol{\delta}$ maximizes $\mathcal{L}_{adv}$.

In this work, the three white-box attacks are modified to perturb $\mathbf{z}$ during GAN training. The attacks FGSM, PGD, and JSMA are modified as follows.

1) *Modified FGSM:* The original FGSM [17] perturbs an input image x in the direction that maximally increases the classifier loss. The modified FGSM applies this attack directly into the latent space, thus perturbing the latent vector $\mathbf{z}$ by a perturbation budget ϵ in the direction of the gradient (equation 2), where, $\nabla_z \mathcal{L}_{adv}(\mathbf{z})$ is the gradient of the adversarial loss $\mathcal{L}_{adv}$ w.r.t. the latent vector z.

$$\mathbf{z}_{\text{adv}} = \mathbf{z} + \epsilon \cdot \text{sign}\left(\nabla_{\mathbf{z}} \mathcal{L}_{\text{adv}}(\mathbf{z})\right), \tag{2}$$

2) *Modified PGD*: PGD iteratively refines FGSM. The original method [18] applies FGSM repeatedly and projects onto an ϵ-ball. The modified PGD begins with a randomly perturbed initialization $z_{\text{adv}}^{(0)} = z + \mathcal{U}(-\epsilon, \epsilon)$, where $\mathcal{U}(-\epsilon, \epsilon)$ samples uniformly from an ℓ_∞-bounded hypercube.

Then, for each iteration $t = 0, \ldots, T-1$, the gradient is computed using equation 3 and the latent vector is updated using equation 4, where α is the PGD step size and the $\text{Clip}_{z,\epsilon}(\cdot)$ enforces the constraint $\|z_{\text{adv}}^{(t+1)} - z\|_\infty \leq \epsilon$, producing a strong, optimized adversarial latent vector.

$$g^{(t)} = \nabla_{z_{\text{adv}}^{(t)}} \mathcal{L}_{\text{adv}}(z_{\text{adv}}^{(t)}) \tag{3}$$

$$z_{\text{adv}}^{(t+1)} = \text{Clip}_{z,\epsilon}\big(z_{\text{adv}}^{(t)} + \alpha \,\text{sign}(g^{(t)})\big) \tag{4}$$

3) *Modified JSMA:* The original JSMA [20] perturbs salient input pixels for a targeted attack. The modified attack on the latent-space works by modifying only the K most influential latent dimensions. First, the saliency values are computed using equation 5, the top K latent dimensions are selected using equation 6, and the sparse perturbation is finally applied using equation 7.

$$S = |\nabla_z \mathcal{L}_{\text{adv}}(z)| \tag{5}$$

$$M = \text{OneHot}(\text{TopKIndices}(S, K)) \tag{6}$$

$$z_{\text{adv}} = z + \epsilon \cdot \text{sign}(\nabla_z \mathcal{L}_{\text{adv}}(z)) \odot M \tag{7}$$

Here, $\odot$ denotes element-wise multiplication, ensuring that only the top-K salient latent components are perturbed.

TABLE II
EA PARAMETERS

Parameter	Value
Gene Pool	{1,2,3}
Chromosome Length	3
Population Sizes	5, 10
Number of Generations	10
Mutation Rates	46%, 50%, 60%
Number of Runs	10
Termination Criteria	Max generations, Stagnation

C. Evolutionary Algorithm

The proposed EA formulates the search for harmful adversarial attack chains as a combinatorial optimization problem. The gene pool (alleles) consists of the three selected white-box attacks (FGSM, PGD, and JSMA). Each chromosome of a population represents an ordered sequence of the three attacks, encoded as integers. Thus, a chromosome such as $[3, 1, 2]$ corresponds to the chaining of JSMA followed by FGSM followed by PGD, attacked during DCGAN training. Initial populations of size 5 and 10 are created by randomly sampling such sequences from the three-attack gene pool.

To evaluate each sequence, the EA uses a fitness function F designed to measure the degree to which a given attack chain degrades both the GAN and a downstream classifier.

F uses two complementary metrics, ASR and FRD. The ASR measures the proportion of generated images that are misclassified by a ResNet50 classifier and therefore reflects how effectively the attacked generator induces classifier errors, while the FRD quantifies how far the distribution of generated images drifts away from that of real histopathology data.

$$\text{ASR} = \frac{\text{Number of misclassified images generated}}{\text{Total number of images generated}} \times 100\%$$

FRD is computed by comparing deep feature statistics extracted from real and generated images. Since FRD varies in scale compared to ASR, it is first normalized (FRD_{norm}) so that all FRD values lie between 0 and 1, like ASR. The final fitness value (F) of a chromosome (attack sequence) is then computed as the weighted combination of its ASR and FRD_{norm}:$F = (0.5 \times \text{ASR}) + (0.5 \times \text{FRD}_{\text{norm}})$, ensuring that the EA favors attack sequences that simultaneously maximize classification errors and disrupt the quality of generated samples.

Chromosomes with the best fitness values, indicating more powerful attack sequences, are selected and passed on to future generations. New individuals are created solely through mutation. No crossover operator is used in this study. Mutation is applied independently to each gene, replacing it with a randomly selected attack from the gene pool. Experiments were conducted with mutation probabilities of 46%, 50%, and 60% and for population sizes of 5 and 10 for every generation. Each configuration, defined by population size and mutation probability, was executed for 10 independent runs starting from different random populations, with 10 generations per run. All the EA parameters used are listed in Table II.

It is worth mentioning that the experiments were executed on Google Colab NVIDIA L4 Tensor core GPU with 24GB RAM with each chromosome fitness evaluation taking over 200 minutes to execute. Consequently, due to computational limitations, the experimental search space comprises of only $3^3 = 27$ possible attack sequences for a gene pool of $\{1, 2, 3\}$ and chromosome length 3.

IV. Results and Analysis

To evaluate the sensitivity of the evolutionary algorithm, six configurations were tested by taking various combinations of population size $\in \{5, 10\}$ and mutation probability $\in \{0.46, 0.50, 0.60\}$. Each configuration was implemented for ten independent runs, with ten generations per run. Table III lists the top ten attack sequences according to their fitness values that have been identified by the evolutionary algorithm guided search.

From Table III and Figure 2, we see that employing EA with different hyperparameter configurations has revealed the most harmful attack sequence to be `[2,1,2]`, corresponding to `PGD` → `FGSM` → `PGD`, achieving a fitness value of 0.99. This indicates it was the most disruptive sequence for both the DCGAN and the ResNet50 classifier. The repeated dominance of this sequence across EA hyperparameter configurations

TABLE III
Top 10 Chromosomes identified by EA along with Fitness Values

Chromosome	**Fitness** F
$[2, 1, 2]$	0.9903
$[3, 2, 1]$	0.9523
$[1, 2, 2]$	0.7977
$[3, 2, 3]$	0.6588
$[2, 1, 1]$	0.5667
$[1, 1, 1]$	0.5515
$[3, 1, 1]$	0.5376
$[3, 3, 3]$	0.5089
$[1, 2, 3]$	0.5000
$[2, 2, 3]$	0.5000

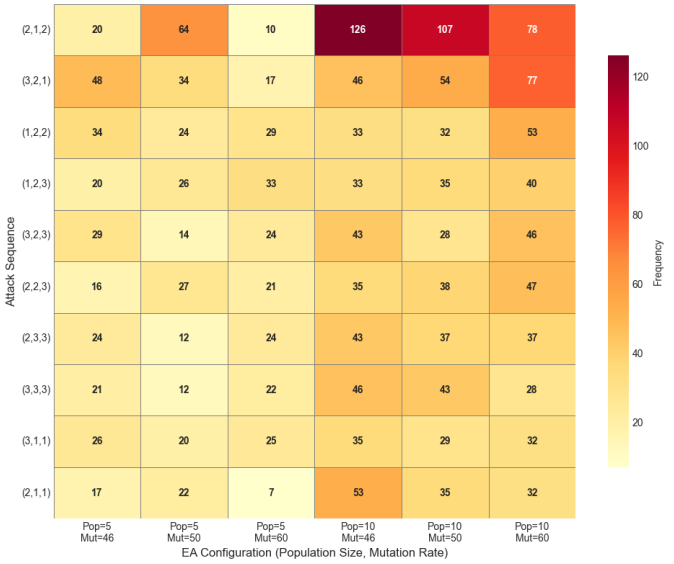

Fig. 2. Frequency of Top 10 Attack Sequences by EA Configuration

suggests that beginning with an iterative attack like PGD effectively weakens the model, while concluding with either PGD or JSMA provides a strong finishing impact, underscoring the importance of both attack ordering and terminal strength in sequence effectiveness. Figure 2 shows the frequency with which each of the top 10 attack sequences (rows) appeared under each EA hyperparameter configuration (columns). The concentration of high values (darker cells) for sequence `[2,1,2]` in configurations with larger population and lower mutation rates, particularly Pop=10 with Mut=0.46, where it appears 126 times, shows that it is an optimal attack. On the other hand, configurations with higher mutation and smaller population (e.g., Pop=5, Mut=0.60) show more dispersed coloring across multiple sequences, indicating greater exploration by the EA.

Figure 3 shows the average of the best fitness value per generation across the 10 runs (y-axis) against the generation index (x-axis), for all six EA configurations. It is observed that all six EA configurations show improvement in fitness values, from lower values to the highest fitness value of 0.99, corresponding to the attack sequence [2,1,2], from generation 1 to 10. However, the resulting curves differ significantly,

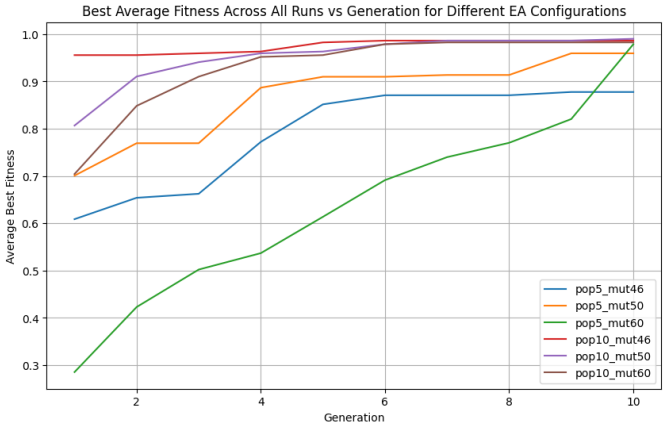

Fig. 3. Best average-fitness across all Runs v.s. Generation for Different EA hyperparameter configurations

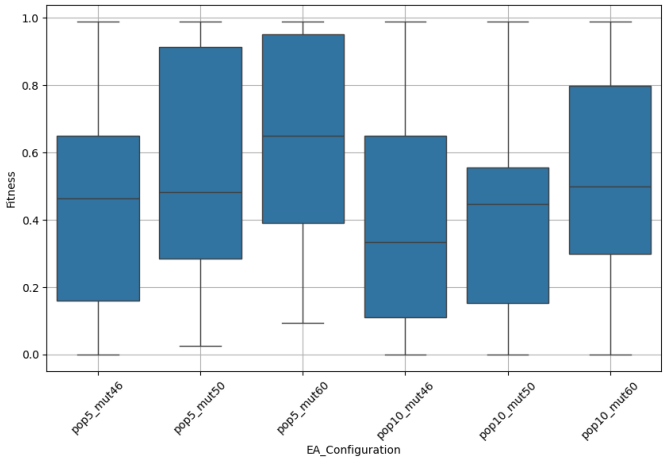

Fig. 4. Fitness values of final-generation across the ten independent runs for each EA hyperparameter configurations

Fig. 5. Convergence Heatmap: Generation number where Fitness ≥ 0.99 was reached

likely due to the small experimental search space of this study. The population-10 configurations attain the best final fitness of 0.99 rapidly, indicating early convergence. For example, `pop10_mut0.46` reaches high fitness within the first few generations and then plateaus, with minimal variation across runs. This behavior does not indicate strong evolutionary search, possibly due to the large population size sampling most of the available search space immediately, causing early convergence. Mutation rates of 0.46 and 0.50 for the population size of 10 also do not sufficiently exhibit exploration. On the other hand, the population-5 configurations show more meaningful evolutionary algorithm behavior. After starting at lower fitness values, the candidates improve gradually, and show more variation across runs, indicating better exploration of the search space. Consequently, it can be observed that the greatest diversity among all six EA configurations is obtained when population size is 5 with a mutation rate of 0.60 (figure 3). It is also worth highlighting that though lower mutation rates (0.46 and 0.50) converge faster, there is less exploration and the behavior is similar to the early convergence patterns observed for population size 10. Figure 4 further highlights this observation by showcasing the distribution of the final-generation fitness values across the ten independent runs for each hyperparameter configuration.

Figure 5 shows the generation number at which each EA configuration first reached a fitness value ≥ 0.99. The number within each cell indicates the generation at which, for a particular run and a particular EA configuration, the final best fitness value of 0.99 was achieved. If the number is less, it indicates that the best fitness value was achieved at a very early generation (early convergence). Lighter cells indicate faster convergence (lower generation numbers), while darker cells represent slower convergence or late discovery of high-fitness solutions. Blank cells correspond to runs that never achieved the 0.99 (best) fitness. Configuration `population-5` with `mutation = 0.60` shows the most consistent and reliable convergence behavior. Mostly all runs for this configuration reach the 0.99 fitness, most of them converging between generations 9–10. This shows that exploration and exploitation are balanced.

On the other hand, other `population-5` configurations (`mutation = 0.46`, `mutation = 0.50`) show inconsistent convergence patterns: some runs reach high fitness rapidly, while others fail to converge entirely. This suggests insufficient mutation rates for adequate exploration of the search space. Convergence patterns are irregular for `population-10` configurations as well. Although certain runs show rapid convergence (generation numbers 1–3), many either fail to reach the 0.99 or converge inconsistently on repeating across runs. This shows early convergence and diversity collapse within larger populations.

In short, it can be observed that population size plays a critical role in this experimental study. Larger populations (Pop=10) converge to top fitness sequences like `[2,1,2]` and `[3,2,1]` rapidly, while increasing the risk of early convergence in such a constrained space. Conversely, smaller populations (Pop=5) produce more variation in the best sequence found across runs, reflecting increased exploration. Mutation rate further modulates this trade-off. Lower mutation (0.46) drives strong exploitation, especially with Pop=10, where `[2,1,2]` appeared 126 times. Higher mutation (0.60)

encourages exploration, reducing the frequency of `[2,1,2]` to as low as 10 in Pop=5 and 78 in Pop=10, while allowing other sequences to surface more prominently.

V. Conclusion

This study adapted three white-box attacks originally designed to attack classifiers, to perform poisoning attacks on Generative Adversarial Networks (GANs). It also explored the potential of evolutionary algorithms to find highly disruptive attack chains that can degrade the performance of medical imaging DCGANs, along with a subsequent sensitivity analysis using six different EA hyperparameter configurations. Results show that chaining attack sequences significantly degrade the performance of GANs to generate quality medical images. It is seen that specific sequences, particularly the attack chain where first, PGD (Projected Gradient Descent) is executed, followed by FGSM (Fast Gradient Sign Method) and PGD (again), have great attacking power, as exhibited by their high fitness values. The experiments reveal that EA hyperparameters critically influence search behavior. Within the small search space of this study, the EA shows a trade-off between exploring new sequences and exploiting chromosomes with high fitness values. We see that if an EA configuration is exploitation-oriented (larger populations with lower mutation rates), it will lead to finding the optimal solution very soon, leading to early convergence, while exploration-oriented EA configurations (smaller populations with higher mutation rates) promote greater diversity in the top sequences identified across runs.

Future work will explore scaling to larger search spaces by using longer attack sequences and the development of defense mechanisms to safeguard both synthetic data quality and classifier reliability.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[2] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv*, vol. abs/1411.1784, 2014.

[3] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2016.

[4] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[5] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[6] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[7] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[8] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[9] S. Kazeminia, C. Baur, A. Kuijper, B. Menze, A. Lotter, D. Rueckert, S. Albarqouni, and N. Navab, "GANs for medical image analysis," *Artificial Intelligence in Medicine*, vol. 109, p. 101938, 2020.

[10] H. Yang, Q. Zhang, X. Liu, and Y. Xu, "Generative adversarial networks in medical imaging: A comprehensive survey," *arXiv:1809.07294*, 2019.

[11] A. Makhlouf, M. Njeh, and H. Hamam, "Medical image augmentation using GANs: A review," *Neural Computing and Applications*, vol. 35, no. 23, pp. 16731–16757, 2023.

[12] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "GAN-based synthetic medical image augmentation for improved liver lesion classification," *IEEE Transactions on Medical Imaging*, vol. 38, no. 3, pp. 628–639, Mar. 2018.

[13] V. Sandfort, J. Yan, P. Tian, and R. M. Summers, "Data augmentation using CycleGAN for CT images: Application to colorectal liver metastases," *Scientific Reports*, vol. 9, p. 16852, 2019.

[14] S. K. Venu and S. Ravula, "GAN-based chest X-ray image augmentation for classification," *Future Internet*, vol. 13, no. 1, p. 14, 2021.

[15] A. A. Borkowski, M. L. Borkowski, E. W. Thomas, and S. Doyle, "Liver and colon histopathological image dataset (LC25000)," *arXiv:1912.12142*, 2019.

[16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.

[17] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of International Conference on Learning Representations. (ICLR)*, 2015.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[19] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proceedings of IEEE Symposium on Security and Privacy (SP)*, 2017.

[20] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proceedings of IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016.

[21] J. Kos, I. Fischer, and D. Ha, "Adversarial examples for generative models," *arXiv:1702.06832*, 2018.

[22] V. Webster, Z. Lipton, and A. G. Wilson, "Defending and attacking generative models," *arXiv:2006.05084*, 2019.

[23] A. Rawat, P.-Y. Chen, S. Chang, C.-M. Lee, and K.-H. Wang, "The devil is in the GAN: Backdoor attacks and defenses in deep generative models," *arXiv:2103.08606*, 2021.

[24] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer, 2003.

[25] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[26] K. A. De Jong, *Evolutionary Computation: A Unified Approach*. MIT Press, 2006.

[27] S. Shyju and R. Murali, "ATLAS – A co-evolutionary framework for automatic tuning of adversarial neural networks," in *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, 2023.

[28] V. A. Ashwin, G. M. Harish, and C. S. Velayutham, "A study on morphological and behavioral adaptations from multi-task morphoevolution by voxel-based soft robots," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2025.

[29] V. K. Nair and C. S. Velayutham, "EVGAN: Optimization of generative adversarial networks using Wasserstein distance and neuroevolution," in *Evolutionary Computing and Mobile Sustainable Networks*, Lecture Notes on Data Engineering and Communications Technologies, vol. 116, Springer, Singapore, 2022.

[30] R. Murali, P. Thangavel, and C. S. Velayutham, "Evolving malware variants as antigens for antivirus systems," *Expert Systems With Applications*, 2023.

[31] M. Alzantot, Y. Sharma, S. Chakraborty, B. Karlas, and M. Srivastava, "GenAttack: Practical black-box attacks with gradient-free optimization," in *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, 2019.

[32] F. Brau, M. Pintor, A. E. Cinà, R. Mura, L. Scionis, L. Oneto, F. Roli, and B. Biggio, "TransferBench: Benchmarking ensemble-based black-box transfer attacks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.