IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 November 2023, accepted 13 December 2023, date of publication 18 December 2023, date of current version 22 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3343754

RESEARCH ARTICLE

Associative Discussion Among Generating Adversarial Samples Using Evolutionary Algorithm and Samples Generated Using GAN

ARUNA PAVATE[1], (Member, IEEE), RAJESH BANSODE[2], PARVATHANENI NAGA SRINIVASU[3,4,5], JANA SHAFI[6], JAEYOUNG CHOI[7], (Member, IEEE), AND MUHAMMAD FAZAL IJAZ[8]

[1]School of CSIT, Symbiosis Skills and Professional University, Pune 412101, India
[2]Department of Information Technology, Thakur College of Engineering, Mumbai 400101, India
[3]Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza 60455-970, Brazil
[4]Department of Computer Science and Engineering, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada 520007, India
[5]INTI International University, Nilai 71800, Malaysia
[6]Department of Computer Science, College of Arts and Science, Prince Sattam bin Abdulaziz University, Wadi Ad-Dawasir 11991, Saudi Arabia
[7]School of Computing, Gachon University, Seongnam-si 13120, Republic of Korea
[8]School of IT and Engineering, Melbourne Institute of Technology, Melbourne, VIC 3000, Australia

Corresponding authors: Jaeyoung Choi (jychoi19@gachon.ac.kr) and Muhammad Fazal Ijaz (mfazal@mit.edu.au)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2022R1C1C1004590, and in part by Prince Sattam bin Abdulaziz University under Project PSAU/2023/R/1445.

ABSTRACT The remarkable accomplishments of deep neural networks (DNN) have led to their widespread adoption in various contexts, including safety-critical applications. Many strategies have been implemented to generate adversarial samples using DNN, raising the question of the security of the model. Adding slight magnitude noise to the input samples during training or testing can misguide DNN to produce different results than the actual one. DNNs are sensitive to indiscernible adversarial samples but readily identifiable by them. Currently, gradient-based approaches are used to generate adversarial samples. Gradient-based methods require internal details of the model, such as several parameters, model type, Etc. Usually, these details are practically unavailable, and calculating the gradient for non-differential models is impossible. In this work, we propose a novel DESapsDE framework based on evolutionary algorithms to generate adversarial samples from the probability of labels. We also incorporated the discussion with the various Generative Adversarial Networks (GANs) models, such as ACGAN, DCGAN, and SAGAN. It has been observed that GANs differ from adversarial sample generation methods and can be applied as defense mechanisms. The proposed method reduced model confidence to 13.09% for the ResNet50 model, 30.34% for the WideResNet model, and 23.1% for the DenseNet model, with an FID score of 16.45. The proposed model varies from the GAN model. It applies to attack-on-network models as a preventive major to make the model robust.

INDEX TERMS Adversarial examples, attacks, differential evolutionary algorithm, deep neural networks, generative adversary networks, optimization methods.

I. INTRODUCTION

Integrating deep machine learning with industrial automation solutions can significantly increase speed in all processes by avoiding human errors and reducing human interventions. The use of deep learning changed human life in many fields. Computer vision is the field of deep learning that is increasingly used in many applications, from disease prediction [1], [2], [3] to automated surveillance systems [4]. The advent of many deep learning technologies has given rise to protecting

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

computing systems from digital attacks [5]. Many of these applications exhibit better performance than humans. Despite having high performance, recent research demonstrates that though many of the network models are strong, they are not robust. The most popular term nowadays being used by an adversary is adversarial machine learning, which fools machine learning models with perturbed data. Adversarial machine learning is becoming one of the significant threats in machine learning. Adversarial machine learning considers both the generation and identification of samples. Adversarial samples are specially crafted to deceive the prediction model and are exposed in many areas, such as image classification, disease prediction, to face recognition.

An adversarial mechanism is an approach to producing adversarial samples. Adversarial samples are inputs to a classification model designed intentionally to make the model incorrect, despite resembling a valid input to the human eye. Researchers for the generation of adversarial samples propose many approaches. Adversarial mechanisms that assist in generating adversarial samples may be derivative-based (gradient-based) optimization techniques. Assuming no prior knowledge of the model, adversarial attacks are conceivable during testing and deployment without direct access to the model.

Szegedy et al. [6] first proposed the term adversarial examples using gradient-based evasion attacks. Recently, many researchers attempted adversarial example attacks on deep neural networks. Kurakin et al. [7] tried adversarial examples in the physical environment. In the same context, Carlini and Wagner [8] and Chen et al. [9] verified adversarial standards in speech recognition models (ASR) and Voice Controller systems (VCS). The recent work shows effective attacks in contrast to neural networks that resolve numerous problems. Initially, adversarial examples generated were not appropriately imperceptible. Most methods use distance metrics of L_p - norms (L_0, L_2, L_∞). Sharif et al. [10] showed that L_p norms are not essential for perceptual resembles. Secondly, several methods were proposed for constructing adversarial examples and making the network robust against adversarial examples. Currently, no single defense is available to accurately categorize the adversarial examples.

Many analysts use generative adversarial networks to generate different types of adversarial samples. The initial framework, called GAN, was suggested by Goodfellow et al. [11] for producing fresh instances from the entire dataset in deep learning. In recent years, GAN has progressed from making realistic human features to producing artistic artworks [12], [13]. The effectiveness of these models comes from the expense of computation and data. GAN models are data-hungry to produce high-accuracy images of many categories. GAN models require high-quality training samples with huge volumes. These massive datasets need time, significant human work, and expensive annotation costs to collect and process data. Generative modeling is applicable to produce real examples that result from a distribution of existing samples. For instance, producing new similar but distinct images from a collection of existing images. GAN works on image data and makes use of convolutional neural networks. Brock et al. [14] demonstrate how their BigGAN technique can produce synthetic photographs that are almost different from actual photographs. Applications such as Generate Realistic Photographs [11], Cartoon Characters, Text-to-Image Translation [13], Generate New Human Poses, Image-to-Image Translation, Photo Blending, Photo Inpainting, Clothing Translation, and Photograph Editing are designed using GAN and many more.

Adversarial attacks may be launched in several ways. These attacks are made primarily for image recognition issues and are made to be effective against Neural Network (NN) models. The training of Generative Adversarial Networks (GANs) is infamous. Research has been done from various perspectives to overcome the difficulty of training GAN. Discriminators or classifiers are vulnerable to hostile perturbations. The adversarial robustness of these models is increased when they are trained on data generated by GANs. Many defenses have been suggested to lessen the impact of adversarial attacks. Researchers use generative adversarial networks to defend against attacks [15], [16]. Many researchers concentrated on defensive mechanisms using GAN, such as Zhang et al. [17], who proposed a robust system to defend the gradient-based attack applied during the attacking and testing stages. The attacking phase works as a proactive mechanism to intercept the attacker from generating adversarial samples, and the testing stage allows them to discover the perturbed examples and avoid feeding into the classifier while preventing the attacker from developing malicious samples. The authors utilized a neural network to design the defense and allow the network to find the adversarial examples.

Defense mechanisms modify the samples to make the classifier more robust to the attack. Many defense mechanisms have limitations that apply to black-box or white-box attacks but not to both, and most of the defense mechanisms are specific to the attack and not applicable to the new attack.

This work addresses the associative discussion between the generation of adversarial examples using evolutionary algorithms (DESapsDE) [18] and the samples generated using generative adversarial networks (GAN). The proposed framework makes use to fool the different neural network architectures. It generates adversarial samples with a success rate while maintaining human perception and the speed of the generation of samples very rapidly. The previous work concentrates on generating adversarial samples using gradient optimization methods that need internal design aspects of the model, such as several parameters for training, training data, and neural network type [6], [19]. Several adversarial samples are created without understanding the model's essential details, like the internal organization of the model [9], [20]. Evolutionary algorithms work only on the probability of labels from the target model; no internal details are required.

The data is often fetched from physical devices, including mobile phones and cameras. In such scenarios, getting the gradient details in the real world is challenging. Deep neural network models are black-box and consist of multiple layers, and it is not easy to examine the model line by line, even if internal details are known. It is possible to provide cost-effective solutions using pretrained models; hence, the adversary can make the model generate the expected output. Evolutionary algorithms are the most robust, reliable, and stable solutions introduced by Su et al. to add small perturbations [21]. A differential evolutionary algorithm is a global optimizer that requires only three parameters, population, crossover, and scaling parameters, to search for a solution from a large space [22], [23]. Most of the existing evolutionary algorithms [21], [22], [23] concentrate on fixed population size that results in solutions getting stuck in local search space. Researchers have reported methods using evolutionary algorithms based on differential evolution and variants, but their success rate is low [21]. In a natural environment, the population size varies due to many parameters. The proposed solutions concentrate on changing the population size to provide more robust solutions. The proposed solution is more effective for low search space and focuses on only the probability of labels with flexibility regarding attack on any deep neural network model.

In 2017 google brain showed that any prediction system designed using machine algorithms could be fooled and allow the system to yield incorrect results with significantly less skill. Researchers can get them to provide any effect that they want. This vulnerability is a significant problem for the applicability of these safety-critical practices. Most existing machine-learning classifiers are vulnerable to adversarial examples [24], [25]. Machine learning algorithms, such as deep neural networks, have been weak to well-crafted input samples [6]. This weakness of adversarial mechanisms' deep neural networks becomes a significant threat to applying deep neural networks in safety-critical scenarios.

The creation of adversarial examples is an optimization issue with some conditions. The adversary aims to get the optimal solution by adding perturbation as a minimization or maximization function. Generating adversarial samples becomes a significant challenge when the gradient calculation is complex such that perturbation added can hide adversarial modification.

Deep neural networks have demonstrated unparalleled success in solving complex problems previously deemed challenging for traditional machine-learning approaches. Deep neural networks handle large amounts of data and model complex relationships, contributing to their success in diverse domains. The deep neural network generalizes its capacity to unseen data and adapts to various tasks, making it the go-to choice for many machine learning applications. Deep Neural Network automates the feature extraction process, eliminating the need for manual feature engineering saving time and resources for training the model. Deep neural networks offer remarkable capabilities but are not immune to vulnerabilities. Deep neural networks are susceptible to adversarial attacks, where small, carefully crafted input can lead to misclassification. Deep Neural networks raise a significant challenge to the security and reliability of DNN-based systems. Szegedy et al. [6] contributed to discovering and exploring vulnerabilities in neural networks. The critical vulnerability Szegedy highlights is the sensitivity of neural networks to small and imperceptible perturbations in input data. The reasons for the vulnerability of neural networks are as follows:

Non-Linearity: A deep Neural Network is a non-linearity in nature; small changes in input data can lead to disproportionately large differences in the activation of neurons and, consequently, in the final output.

High-Dimensional Input Space: Neural networks operate in high dimensional input space with millions of pixels. In high-dimensional areas, numerous directions exist, and small changes can cause significant alterations in the final output.

Lack of Robust Features: Deep neural networks often rely on features that might not be robust or stable across different inputs.

Limited Generalization: Deep neural networks demonstrate impressive generalization capabilities. They may need help to generalize effectively in the presence of adversarial examples.

The models may focus on learning patterns present in the training data but fail to capture the underlying structure of the data, making them vulnerable to manipulation.

Understanding and addressing these limitations are critical for developing and deploying deep neural networks. Most of the ongoing research focuses on mitigating these challenges and ensuring that deep neural networks are used ethically and effectively in various applications. There are many applications, such as style transfer, transferring one image's properties to another, 3-D object generations, generating faces, etc. Most applications using GAN generate similar to the original images but significantly differ nearby. This motivates us to work on how GAN is different from adversarial samples. Therefore, the attacks and defense strategies for generating adversarial mechanisms pulled great attention.

We introduce below a few basic terminologies to understand the concept of adversarial samples.

A. ADVERSARIAL SAMPLES

Examples are created by purposely adding minor worst-case perturbations to regular examples so humans can not recognize them easily. As shown in Figure 1, the original image x, after adding a small perturbation of ε (>0), makes the machine learning model change the output class with some confidence.

B. LOSS FUNCTION FOR ATTACK

The convolutional neural network (CNN) is a dominant deep learning model that trains network models to categorize pictures based on available patterns. It may then be

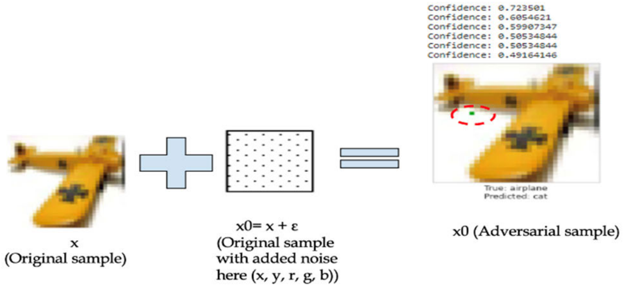


FIGURE 1. Adversarial sample generated using Novel DESapsDE differential evolutionary method.

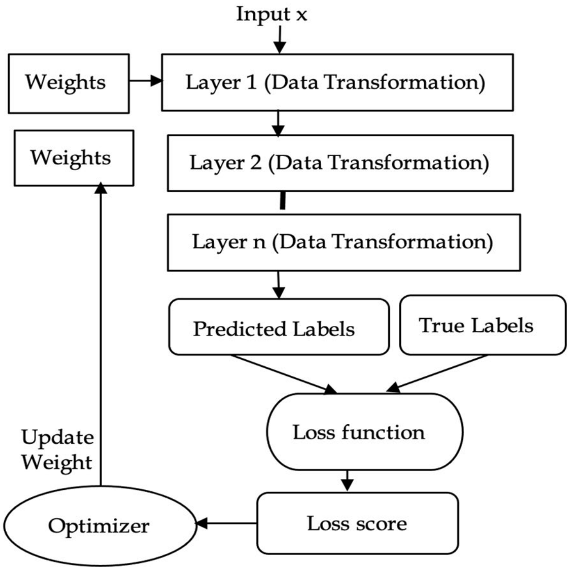


FIGURE 2. Deep learning as the optimization process.

taught to recognize things in photos. As shown in Figure 2, networks are developed by embedding an optimization procedure that involves a loss function to quantify the model's error.

The loss function evaluates the machine learning model's performance using various loss functions. As shown in Figure 3, the network model $f\theta$ is trained using optimization algorithms that calculate the error generated.

The loss function is used to upgrade the model by providing retraining. The purpose of retraining the model is to minimize loss, as minimum values represent an improved model than a larger value. Let us consider a network f parameterized by θ that transfers a sample x to a real label y^0. An adversary intends to use the function f to misclassify x0 to y^{false}. Here y^{false} is the output label other than the actual class label. Here y^0 is the original label, and y^{true} is the predicted label. The function's output after training is shown in (1). The input x never changes during training.

$$Training{:}LTrain\,(\theta) = c\left(y^0, y^{true}\right), \tag{1}$$

where $c(a, b)$ is a cost function between a and b.

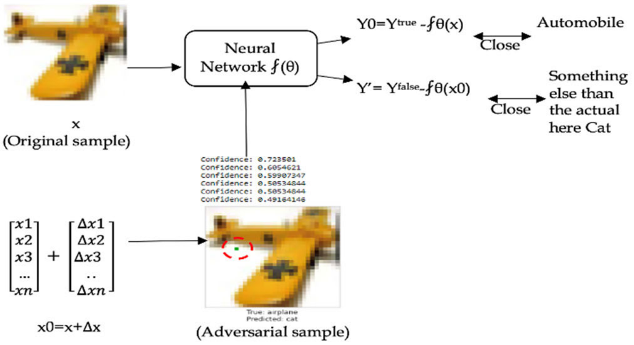


FIGURE 3. The loss function for adversarial sample attack.

C. NON-TARGETED ATTACK

This attack misguides the model to any one of the classes. Adversaries make the model give incorrect results. In a non-targeted attack, θ designates the number of parameters as constant, and the loss function is minimized optimal solution is as shown in (2).

$$L(x') = -c\left(y', y^{true}\right) \tag{2}$$

D. TARGETED ATTACK

Targeted attack misguides the deep neural network to a determined class. This attack is targeted to receive a specific class for the given input, making it more difficult to attack. The output may be any arbitrary class, but not the original one. In a targeted attack, the loss function is maximized optimal solution as shown in (3).

$$L(x') = -c(y', y^{true}) + c(y', y^{false}) \tag{3}$$

E. PERTURBATION MEASUREMENT METRICS

The correlation between the original image and the adversarial sample was assessed using L_p norms. The generally used p-norm metrics for assessing perturbation magnitude are L_0, L_2, and $d(x, x')$ is a distance constraint that should be less than some value ε as represented in (4). The similarity between the original and adversarial samples was assessed using L_p norms. The generally used p-norm metrics for quantifying perturbation magnitude are L_0, L_2, and L_∞. $d\left(x, x'\right)$ is a distance restriction that must be smaller than the value while adding perturbation to the sample, as shown in (4).

$$Constraint{:}d(x, x') \leq \varepsilon \tag{4}$$

F. L_0 -NORM

This norm gives the count of aggregated pixels altered in the perturbed samples. The maximum possible perturbation is one pixel, as represented in (5).

$$d\left(x, x'\right) = \left|\left|x - x'\right|\right|_0 \tag{5}$$

G. L_2 -NORM

For each pixel, calculating the variation between the actual input sample and the perturbed sample and summing it over all the pixels is called the L2 norm. Mathematically it is represented in (6).

$$d(x, x') = ||x - x'|| = (\Delta x1)^2 + (\Delta x2)^2 + (\Delta x3)^2 \quad (6)$$

H. L_∞ - NORM

The Euclidean distance measure finds the variation between the perturbed and actual samples. For each pixel, the variation between the actual sample and the perturbed image is computed, squared, and summed over all the pixels, as shown in (7).

$$d\left(x, x'\right) = \left|\left|x - x'\right|\right|^{\infty} = \max\{\Delta x1, \Delta x2, \Delta x3\} \quad (7)$$

This work is organized as follows: Section II discusses related work with adversarial machine learning; section III briefly introduces generative adversarial networks (GANs) and the proposed system. The experimental results and their comparison associated with generative adversarial examples are provided in section IV. Finally, section V presents the conclusion and future directions.

II. RELATED WORK

The related work concentrates on gradient-based attacks, evolutionary-based attacks, and work related to generative adversarial networks. The first adversarial attack (L-BFGS) [6] for deep neural networks was presented by Szegedy. By using a visual perturbation, the network can be utilized to classify an image incorrectly. The author demonstrated how various models and datasets might use the created adversarial attack. Iterative attack frequency and perturbation magnitude were utilized as the validation metrics. 2.1% error rate and 0.058 distortion rate. The L-BFGS method's reliance on an expensive linear search technique was time-consuming and challenging to execute.

Although linear behavior accelerates the training process, the authors [11] claim that the susceptibility of deep neural networks to adversarial perturbation arises from their collinear character. The validation metric was attack frequency multiplied by perturbation magnitude. A. Rozza [26] created the fast gradient value technique by altering the gradient's sign in the fast gradient sign technique using the raw gradient. The proposed method improved the system's dependability and accuracy. A practical saliency adversarial map, known as the Jacobian-based Saliency Map Attack, as stated by Papernot et al. [19]. A modest perturbation was created to track the neural network that could successfully produce massive output changes. The authors described two adversarial saliency maps to choose the feature to be created over each iteration. Only 4.02% of the input characteristics per sample were changed to attain their 97% adversarial success rate. Deepfool [27] is the author's approach for determining the shortest distance between the genuine input and adversarial samples' decision boundary. They used an iterative technique based on a linear approximation to deal with the high-dimension nonlinearity. Chen et al. [9] developed a strategy based on Zeroth Order Optimization (ZOO). Although this attack does not need gradients, it can be used immediately in a black box attack without delivering any data. The researchers also modified stochastic coordinate descent (SCD) techniques by converting the gradient function into a novel loss function called ZOO-ADAM, which resembles a hinge. The results demonstrated that the white box assaults used by ZOO and C&W functioned equally.

Lin et al. [28] presented the Black-box Momentum Iterative Fast Gradient Sign Method to create the adversarial samples. The major goal is to assure the DNN's resilience by considering model features such as input and output rather than internal details such as weight values, gradients, or model architectural information. On the ImageNet dataset, the suggested solution is tested for targeted and untargeted assaults. The author used differential evolution to enhance the model's inaccurate gradient direction and enabled double-step size and candidate reprocessing. The suggested system was tested against CIFAR10, MNIST, and ImageNet. In this study, the ResNet101 architecture is utilized as a basic model with 100 samples verified for both the targeted attack, with a success rate of 93.2%, and the non-targeted attack, with a success rate of 98.6%. The author claims this method takes less time and produces more transferrable samples than the Zoo approach. Shu et al. [29] developed a straightforward method for producing and identifying adversarial samples. Users may define the number of pixels affected, the chance of misclassification, and the targeted erroneous pixels. The disclosed method is a white box attack that can recognize vulnerable samples, i.e., pixels using a unique manifold-based F1 measure. According to the author, this attack is universal, rapid, and gradient-free over a sample size of 200, 500, and 1000 using particle swarm optimization methods. The ResNet32 model is used in this work to train and evaluate samples over the MNIST and CIFAR10 datasets.

In the study by Luo et al., [30] a random directed attack over the hill climbing method was to get the gradient direction for the generation of adversarial samples. The generated adversarial samples were applied for both the targeted and non-targeted labels without internal information available, and experiments were tested using MNIST, SVHN, CIFAR-10, and ImageNet-10 datasets. The model is trained for 100 epochs through the Adam optimizer and with different operations on samples like rotation, vertical shift, and horizontal flip. Experimental results examine the effect on the success rate of a different selected number of dimensions, the angle of rotation of samples, attack direction, and the number of iterations. The results given by the RDA method are aggressive in most of the analyses, which achieves the highest success rate of 100 % after multiple iterations.

In a novel attack known as compositional pattern-producing network-encoded EA (CPPN EA) [31], adversarial samples are classified with notable accuracy (99%) using a deep neural network. However, these objects are not identifiable to humans.

TABLE 1. Experiment conducted using GAN with adversarial samples.

Reference	Model	Knowledge Consideration	Performance evaluation	Attack Type	Dataset
[25]	Adversarial Networks - Adv-GAN	Semi-Whitebox and Black-box	Accuracy Black-box 92.76% Semi-Whitebox 88.93%	Targeted and Non-targeted Attack	MNIST CIFAR10
[42]	AdvGAN++:	Black-box	Attack success rate on Wide-Resnet 99.92%	Untargeted	MNIST CIFAR10
[43]	Defense-GAN	White-box and Black-box attacks	Accuracy White-Box 98.8 Black-Box 91.64	Targeted and Non-targeted	MNIST F-MNIST
[44]	XGAN	Black-box	82% success rate	Targeted	MNIST CIFAR10
[45]	Robust regularizations	Black-box	FID for robust feature matching RFM-	Targeted	CIFAR10
[46]	Rob-GAN	Black-box	Accuracy 81.45% on CIFAR10 Dataset	Adversarial training	CIFAR10, ImageNet
[47]	FastGAN (Free Adversarial Training)	Black-box	FID Score 12.97	Adversarial training	CIFAR10
[48]	ATGAN	Whitebox	Highest Accuracy 0.646	Targeted	MNIST SVHN and CIFAR-10
[49]	AI-GAN	Whitebox	Success rate 90%	Targeted	MNIST and CIFAR-10, CIFAR-100,
[50]	Novel GAN model	Blackbox	Clean data accuracy 85%	Targeted	CIFAR10

Pavate et al. [20] discussed the different adversarial generations using gradient-based methods and concluded that calculating the gradient is practically difficult. Evolutionary algorithms (EAs) have been used to generate hostile examples. It is challenging to avail the information about the model and calculate the gradient for the system designed using non-differential techniques. Evolutionary algorithms require only the probability of labels from the target model. For Evolutionary algorithms, it has been shown that 72.29%,72.32% & 61.28 % success rates for non-targeted attacks and 88.68%,83.63%, and 73.07% confidence with best parameter settings on three different types of networks [32]. As more effective methods are available, we can compare them with other categories of evolutionary algorithms [18], [21], [32], [33]. As varieties of evolutionary algorithms are available, implementing samples can be possible using more advanced algorithms such as Covariance Matrix Adaptation Evolution Strategy, Adaptive DE, SUNA, etc.

There are many GAN-based methods used for the attack [34], [35] and model protection [24], [37]. Radford et al. [38] proposed a DCGAN (deep convolutional GAN) system that is more secure and fast in most settings. Xiao et al. [25] proposed AdvGAN design perturbed instances from the original ones.

The generated adversarial samples were verified in Black Box and semi-White box settings. The generated model is a defense method against attack [11], [39]. The authors showed that the generated samples achieved a high success rate of 94.7 for the ResNet model and 99.3 for the WideResNet model in a semi-white box attack setting for the CIFAR-10 dataset.

The discriminator's loss function in the Least Squares Generative Adversarial Network (LSGAN) is designed to utilize the a-b coding technique in the least squares technique to solve the issue of gradients vanishing during the GAN training process [40]. The LSGAN helps to generate high-quality images. A representation learning technique with the potential to fully framework for the implementation of the disentangled design was introduced by Information Maximizing GAN (InfoGAN) [41]. InfoGAN, an unsupervised framework based on GAN, distinguishes continuous and discrete latent components, scales to huge datasets, and takes no further training time than GAN.

Xiao et al. [25] proposed AdvGAN for protecting the network model from adversarial attacks. The adversarial samples are generated by establishing perturbation into the real world.

For human perceptual testing, authors engaged humans to choose more realistic image pairs. The AdvGAN applies to high-resolution images. The advanced version of AdvGAN++ addressed the limitations of AdvGAN and improved the attack success rate concerning time [42].

Table 1 represents a variety of adversarial networks with performances. The metrics mentioned, such as accuracy, attack success rate, and FID score, provide insights into the robustness and effectiveness of these models under various attack scenarios and datasets. Many of the GAN models were used as defense mechanisms, whereas few of the models used adversarial examples to retrain the model.

According to the study, the primary source of attacks on machine learning models is that it remembers far too much. Because the model is nonlinear, parameters may be adjusted to match the training dataset. The opponent can use this advantage to reveal confidential information or alter the model's output. There is no guarantee that an adversarial picture will be labeled wrongly using these approaches; sometimes, the attacker wins, and sometimes the machine learning model prevails.

III. METHODOLOGY

A. DATASET & ARCHITECURES

The experiments were conducted on various deep neural network models such as LeNet, ResNet50, Network-In-Network, DenseNet, and WideResNet [18] as target image classifiers on the CIFAR-10 dataset [51]. The dataset contains 60000 images of sample size 32 × 32 in 10 classes. Each class has 1000 images. The simplified specification of all the models used for experimentation is shown in Table 2. The DenseNet architecture is flexible and can be adapted for different datasets. In this experiment has considered depth = 16, batch size =128 epochs =200, iterations=391 and weight decay = 0.0005 and other parametric setting is mentioned in Table 2 These models are used to attack adversarial samples generated using the DESapsDE Algorithm.

TABLE 2. **Specifications of models used for experimentation.**

	LeNet	ResNet_50	Net-in-Net	Dense_Net	WideResNet
Input layer	32x32x3	32x32x3	32x32x3	32x32x3	32x32x3
Filter size	5x5	3x3	3x3	3x3	3x3
No. of Filter	6	64	64	64	16
Activation function	ReLU	ReLU	ReLU	ReLU	ReLU
stride	1	1	1	1	1
Padding	Valid	3	Valid	Valid	Valid
Pooling type	Average pooling	Average pooling	Average pooling	Average pooling	Average pooling
Pool size	2x2	4x4	2x2	2x2	2x2
Stride	2	2	2	2	2
FC1	120	64	92		256
FC2	84	256	192		10
Output Layer	10, Softmax	10 , softmax	10 , softmax	10 , softmax	10 , softmax

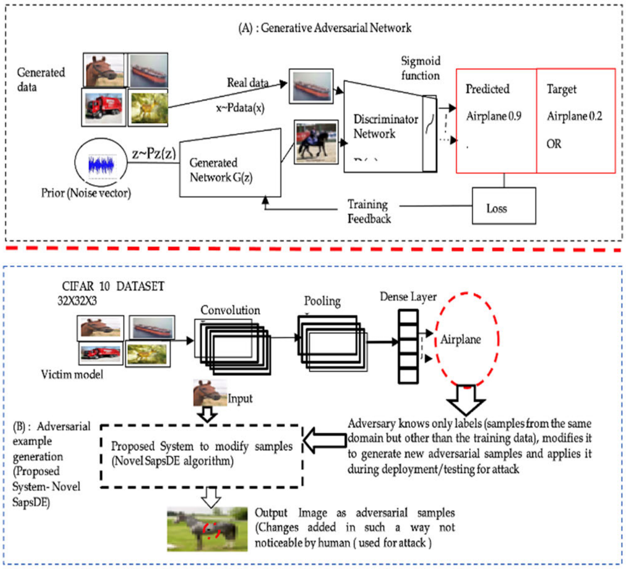

(A) : Generative Adversarial Network
Generated data
Real data
x~Pdata(x)
Sigmoid function
Discriminator Network
Predicted
Airplane 0.9
Target
Airplane 0.2
OR
z~Pz(z)
Prior (Noise vector)
Generated Network G(z)
Training Feedback
Loss
CIFAR 10 DATASET 32X32X3
Convolution
Pooling
Dense Layer
Airplane
Victim model
Input
(B) : Adversarial example generation (Proposed System- Novel SapsDE)
Proposed System to modify samples (Novel SapsDE algorithm)
Adversary knows only labels (samples from the same domain but other than the training data), modifies it to generate new adversarial samples and applies it during deployment/testing for attack
Output Image as adversarial samples (Changes added in such a way not noticeable by human (used for attack)

FIGURE 4. **Associated discussion between the (A): Adversarial samples generation using generative adversarial network and (B): Adversarial sample generation using novel DESaps-DE algorithm.**

B. METHOD

A summary of the systems methods is shown in Figure 4 and represents the associated discussion among two different models, GAN and novel DESapsDE.:

Figure 4A represents the overview of GAN's general architecture for generating samples. We use the proposed DESapsDE [18] system to generate adversarial samples, as Figure 4B highlighted with a blue dotted line. Group B generates the adversarial samples by training some other model. The first group, Figure 4A, consists of original images mixed with some noise images to generate the new samples using GAN. The working of each model is discussed below in sections.

C. SAMPLES GENERATIONS WITH GAN

This section concentrates on training the GAN model using CIFAR 10 dataset and comparing the functional performance of the GAN in synchronization with the proposed system. Figure 4 A shows the generation of adversarial samples using the general generative adversarial network. GAN comprises two models: generative(adversarial) and discriminator models. The model takes sample images with three color channels (R, G, B) and the 32 × 32 image from dataset CIFAR10 as input and outputs a binary class prediction of whether the sample is real (or fake). The image pixel values in the range (0,255) are scaled down to the range (−1,1).

The adversarial model generates the pixels using the tanh action function (1, −1). The adversary model creates new adversarial samples by adding random noise, and the discriminator model verifies whether the samples are fake or real. The discriminator model determines the samples taken from the dataset or adversarial samples. Mathematically the model is represented as shown in (8):

$$\begin{aligned} \min\left(G\right)\max\left(D\right)VDC\left(D,G\right) &= E_{X\sim pdata}\left(x\right)\left[\log D\left(x\right)\right] \\ &\quad + E_{Z\sim P_Z(Z)}[log(1{-}D(G(z))] \end{aligned} \tag{8}$$

Here G is the adversary, x is actual samples from the dataset, D is discriminator. z is generated samples, $D\left(x\right)$ is the discriminator network model, and $G(z)$ is a generator network model. The GAN is an unsupervised model based on the deep neural network architecture. The discriminative model acts as a supervised model. GAN models are trained like other network architecture models such as ResNet, DenseNet Etc. However, these models are complex to train. This model uses random noise with input samples to create new perturbed samples. Extending the number of output labels while training the model can improve the model's performance, but getting the number of output labels is practically challenging. The GAN models help to make the model more robust to attack [15]. In this work, GAN models incorporated for conversation are DCGAN, ACGAN, and SAGAN. The working of each model and experimental settings is discussed below.

1) SAMPLES GENERATION USING DCGAN

The generator model produces an image using up-sampling by adding random noise, as shown in Figure 5. The discriminator consists of stride, batch norms, and LeakyReLU activation function. The samples the generator produces are transferred to the discriminator along with images. The training model setting for Deep convolutional generative adversarial network (DCGAN) is as follows: Generator model settings include sride2, eliminated FC layer, and used inverse convolution for upscaling. Discriminator model setting: CNN, LeakyReLU, kernel size=5, b1 = 0:5, batch size = 64, epochs=100. Here it takes a 3 × 32 × 32 input image, and the output is 3 × 32 × 32.

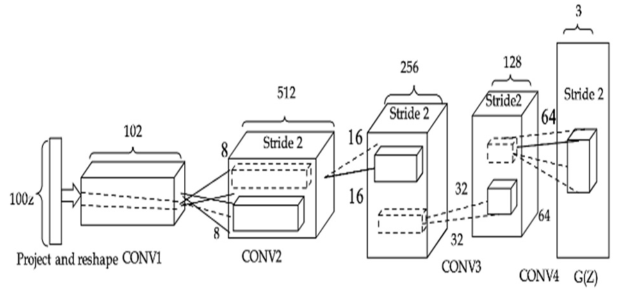


FIGURE 5. Architecture of deep convolutional generative adversarial network.

The samples produced by the generator are transferred to the discriminator and the actual images. The DCGAN causes the problem of mode collapse, where the generator over-optimizes, and the discriminator can never detect fake images; as a result, the generator generates many similar images [38]. The preprocessing images are scaled to a specific range of tanh activation functions.

2) SAMPLES GENERATION USING SAGAN

Self-Attention for Generative Adversarial Networks (SAGANs) [52] is a redraft of the original GANs, as shown in (Fig 6). Here, the idea is to generate global detailing samples. The discriminator and the generator layer contain convolution layer output followed by the attention layer. To deal with the problem of DCGAN, self-attention GAN introduces two time-scale updates in GAN training by providing different learning rates for the generator and discriminator [36]. This helps in solving the issue of slow learning and imbalanced updates. Self-Attention for GANs uses spectral normalization to avoid increased parameters and unwanted gradients. The f(x), g(x), and h(x) are the feature vectors. The feature vectors f(x) and g(x) have different dimensions than h(x), and both feature vectors are aggregated using matrix multiplication to calculate the attention. The aggregated results are passed to the SoftMax layer, which generates the attention map.

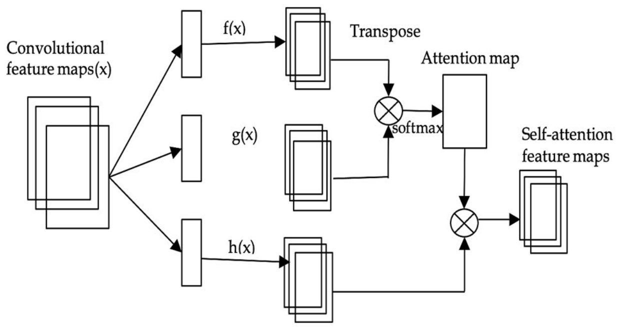


FIGURE 6. Architecture of self-attention for generative adversarial networks.

3) SAMPLES GENERATION USING ACGAN

Conventional GAN was designed for unsupervised learning with an output of the discriminator of dimension 1 with some real probability value. The auxiliary classifier GAN (ACGAN) [40] helps to create class-specific samples using the auxiliary classifier in the discriminator. The discriminator comprises two output layers, the first is used for determining whether the output is real or fake, and the second decides which input belongs to which class, as shown in Figure 7.

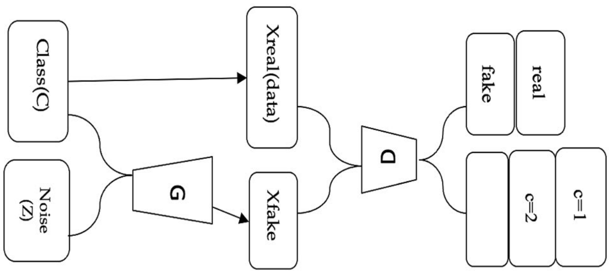


FIGURE 7. Architecture of auxiliary classifier GAN.

D. SAMPLES GENERATION USING NOVEL DESapsDE

The adversarial Mechanism for designing the model starts with collecting the input samples from a similar domain. In this work, the classifier attacked using images from the CIFAR-10 dataset [51]. As shown in figure 4B, the generation of adversarial mechanisms has two different models one is on the victim side, and the other is on the adversary side.

Adversarial mechanisms are the methods used to generate adversarial samples. An adversarial sample is an input to the neural network model designed by adding a small perturbation that causes a model to predict different class output than the actual one resembling an original input to a human. The adversary knows victim model labels. Adversary trains the model with domain samples and obtains similar results. Optimization is the math. Targeted attacks are maximization problems, whereas non-targeted attacks are minimization problems.

The algorithm shows the steps for generating adversarial samples, as discussed in [18]. This method is based on a differential evolutionary algorithm with changing population size. Previous work [21] concentrated on fixed-size populations, but naturally, this is not true as the population changes randomly.

This work concentrates on changing the population and increasing convergence speed. In this algorithm, input is the n-dimensional input as original image X=(x1,……xn). P is ((x1,y1,r1,g1,b1),(x2,y2,r2,g2,b2),…,(x100,y100,r100, g100,b100)) the population size, xm1,xm2,xm3 are the arbitrary indices of the range [1, P]. The differential evolutionary algorithm based on DE/Base/Num/Cross scheme. The base represents how the mutant vector is constructed, Num represents the number of differential vectors, and the cross represents the crossover scheme. θ decides on one of the mutation schemes $\theta \in [1, 0.1]$, e(p) is the additive perturbation w.r.t. natural image X, e(p)$*$ is the fitness function, for the targeted attack, it is considered a maximization function, and for non-targeted, it is a minimization function. The fitness value of each input sample is the probability value of the actual class for each input sample. L is the minimum constant value. Here, L is 1 for one-pixel perturbation, qi is the trial vector, x is the original, and g is the number of generations, initially set to g=0.

Algorithm - Adversarial Sample Generation (Novel DESapsDE)

```
Input: Images of size 32 × 32(CIFAR10 Dataset)
Set the initial population P= (X1,X2,……Xn) i.e., n is equivalent
to 100; Mutation set to 0.5F; Crossover set to 0.1;
    For all g = 1 to 75, do :
  Assess fitness e(p)* = maximize ftdav(p+e(p))
    e(p) * subject to ‖e(p)‖ ≤ L
  For i=1 to 100, do:
  Select any 3 vectors (xm1, xm2, xm3) randomly with different
  indices, where X1=xm1 = (x1, y1, r1, g1,b1) flat vector
  Assess n new_mutant using Xi=xm1 + F(xm2 - xm3)
  Generate trial vector qi through crossover_operation
    if f(qi)>= f(Xi)
        New_offspring = trial vector(qi)
    Else
        New_offspring = Xi
    P = (new_offspring, i=1,2,…,n) //Selection one of the scheme
    to speed up the process
    a. Remove 5% of individuals from the total population or
    b. Randomly selection of best individuals or
    c. Randomly select of best individuals and remove 5% from
    the total population
    Output: Perturbed samples
```

The algorithm starts with selecting the initial random population. At the start, it considers the whole search space. The second step obtains the mutation strategy. Crossover merges with individuals to make new offspring. Three population schemes are included to get the population according to the desired population distribution either 5% of the individuals from the whole population, randomly selects the best individual, or randomly selects the best and removes 5% of the individuals. The algorithm helps to include the perturbation in the input sample so that it is not easily detectable by human eyes. These samples are applied on different neural network models during the testing or deployment phase, and observed the results. The work concentrates on black-box attacks for both targeted and nontargeted attacks during the testing of the model.

IV. RESULTS AND DISCUSSION

The comparison and performance evaluation of images generated by evolutionary algorithms and GAN is challenging. The parameter setting for experimental purposes is mentioned in Table 3. The GAN uses different activation functions at the generator and discriminator as DCGAN uses the G>ReLu, Tanh and D->leaky ReLUs, SAGAN uses G>ReLu, Tanh D-> ReLUs, ACGAN uses at G>ReLu, Tanh and D-> Leaky ReLU, Sigmoid, Softmax whereas DESapsDE applies only one activation function. Generative adversarial Networks and Adversarial examples are distinct concepts with different purposes and applications. GAN is designed to generate new, realistic data samples. They consist of a generator and a discriminator, and both networks are engaged in a competitive process. The purpose of DESapsDE is to test the robustness and vulnerability of the model to small perturbations in the input data.

The parameters of DESapsDE involve the magnitude and direction of noise applied to the input data to cause misclassifications. DESapsDE calculates the fitness value for targeted labels and nontargeted labels. For targeted attacks, it is a maximization function; for nontargeted attacks, it is a minimization function to add minimal noise into the sample.

The novels DESapsDE and GAN have executed in Google Collab with GPU configuration.

In this work, we have used the FID score to check the model's performance and accuracy, as shown in Table 4. The previous works concentrate on different norms $L0$ to $L\infty$ to identify the amount of perturbation added into the samples, making the state-of-the-art complicated to perceive [6], [54]. FID provides a comprehensive evaluation that goes beyond single-image metrics. It considers the entire generated image distribution, offering a more holistic view of the model's performance. FID scores have been shown to correlate with human judgment of image quality.

Models that achieve lower FID scores tend to produce visually closer images to real images according to human perception. Frechet Inception Distance is an assessment metric that calculates the Wsserstein-2 distance between the actual and the constructed samples, where a lower FID score indicates optimal results for the models.

TABLE 3. Comparative parameter settings for experimentation.

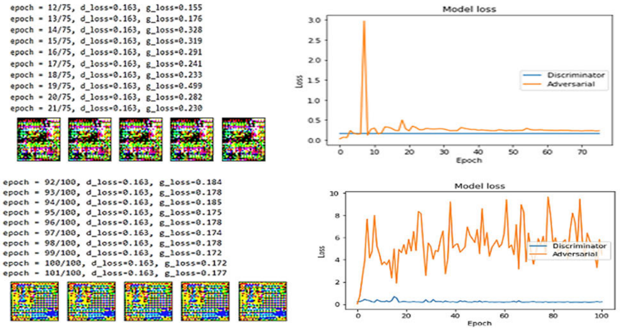
Model	DCGAN	SAGAN	ACGAN	DESapsDE
Purpose	Generate realistic synthetic images	Synthesize complex visual patterns	Realistic data belongs to specific classes	Testing Model Robustness
Component	Generator, Discriminator	Generator, Discriminator, Self-Attention Mechanism	Generator, Discriminator, Auxiliary Classifier	Input Data
Para -meter	Weights and Biases	Attention Mechanism (Weights and Biases learned during training)	Class Labels ,Noise Vector Dimension, Trade-off Parameters	Perturbation Magnitude
Input size	32X32X3	32X32X3	32X32X3	32X32X3
Kernel Size / Fitness value	5x5	3x3	4x4	3
Stride1/stride2	Stride2	stride2	stride2	stride2
Padding	Same	Same	Same	4
No of epoch	3120	1200	1000	200
Optimizer	Adam	RMSprop and Adam optimizer	Adam Optimizer	Adadelta
Seed	random	random seed	1337	100
Learning rate	0.0002	0.0002	0.0002	0.0001
Batch size	128	128	100	100
Loss function	Binary Cross entropy	Binary Cross entropy	SoftMax , Binary , Sigmoid .	Cross Entropy
Dataset	CIFAR10	CIFAR10	CIFAR10	CIFAR10
Model	DCGAN	SAGAN	ACGAN	LeNet, ResNet50, Network-In-Network, DenseNet, WideResNet
Concern	Likelihood of samples	Likelihood of samples	Likelihood of samples	Probability of labels
Applications	Image synthesis, data augmentation, artistic style	High resolution & realistic images, Semantic Segmentation	Conditional Image Synthesis	Security Systems Bypass Testing, Ensuring Ethical AI Practices

However, evaluating the model's performance is difficult based on the cost function and many other parameters. Many time cost functions address the vanishing gradient or gradient stuck in local optima. Becoming trapped in local optima is overcome using innovative DESapsDE evolutionary algorithms by considering dynamic population. However, it is dependent on the cost function. The performance of the GAN and DESapsDE is mentioned in Table 4. The training time required GAN to get the images is more. GANs can be notoriously difficult to train and may suffer from issues like mode collapse.

TABLE 4. Performance of the proposed system with GAN models (200 epoch for Accuracy) on the CIFAR 10 dataset.

MODEL	DCGAN	SAGAN	ACGAN	DESAPSDE
Training Time	40-45 hrs	40-45 hrs	40-45 hrs	20-28 hrs
Accuracy	67%, for real images	69% for real images,	71.89%, for real images	After attack- ResNet -79.22% DenseNet- 71.57 % WideResNet 65.00%
	93% for fake images	92% for fake images	91% for fake images	Before Attack- ResNet - 92.31% DenseNet 94.67% WideResNet 95.34%
FID	69.09%	84.85%	81.23%	16.45%
BEST FID on # epoch	3120	199	3100	200

The GAN model's loss for the discriminator and the generator is observed after every batch. After training, the model over many epochs displays images with some loss remains stable. The discriminator loss on the real and the generated samples is over 1.5. The loss for the adversary model trained using a discriminator over around 2.5 for much of the training process. The model's training starts at epoch 100, and the model starts getting the acceptable images at 3120 epochs as shown in figure 8 whereas for DESapsDE generates the acceptable images.

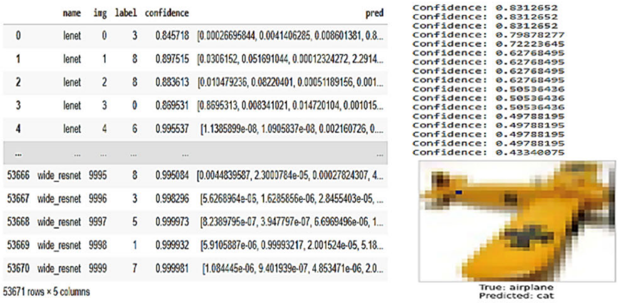


```
epoch = 12/75, d_loss=0.163, g_loss=0.155
epoch = 13/75, d_loss=0.163, g_loss=0.176
epoch = 14/75, d_loss=0.163, g_loss=0.328
epoch = 15/75, d_loss=0.163, g_loss=0.319
epoch = 16/75, d_loss=0.163, g_loss=0.291
epoch = 17/75, d_loss=0.163, g_loss=0.241
epoch = 18/75, d_loss=0.163, g_loss=0.233
epoch = 19/75, d_loss=0.163, g_loss=0.499
epoch = 20/75, d_loss=0.163, g_loss=0.282
epoch = 21/75, d_loss=0.163, g_loss=0.230
```

```
epoch = 92/100, d_loss=0.163, g_loss=0.184
epoch = 93/100, d_loss=0.163, g_loss=0.178
epoch = 94/100, d_loss=0.163, g_loss=0.185
epoch = 95/100, d_loss=0.163, g_loss=0.175
epoch = 96/100, d_loss=0.163, g_loss=0.178
epoch = 97/100, d_loss=0.163, g_loss=0.174
epoch = 98/100, d_loss=0.163, g_loss=0.178
epoch = 99/100, d_loss=0.163, g_loss=0.172
epoch = 100/100, d_loss=0.163, g_loss=0.172
epoch = 101/100, d_loss=0.163, g_loss=0.177
```

FIGURE 8. Evaluating model performance using generative adversarial network.

The model is adversarial, meaning the generator model changes after every batch until good-quality images can be produced. The quality of the images may vary, sometimes improve or even degrade with subsequent updates. The GAN models require more training time to get better-quality samples. The Figure 9 represents the model confidence (left) and sample generated after 100 epochs using the proposed system. As per observation, the GAN requires more



```
        name        img   label  confidence  pred
0       lenet       0     3      0.845718    [0.00026695844, 0.0041406285, 0.008601381, 0.8...
1       lenet       1     8      0.897515    [0.0306152, 0.051691044, 0.00012324272, 2.2914...
2       lenet       2     8      0.883613    [0.010479236, 0.08220401, 0.00051189156, 0.001...
3       lenet       3     0      0.869531    [0.8695313, 0.008341021, 0.014720104, 0.001015...
4       lenet       4     6      0.995537    [1.1385899e-08, 1.0905837e-08, 0.002160726, 0....
...     ...         ...   ...    ...         ...
53666   wide_resnet 9995  8      0.995084    [0.0044839587, 2.3000784e-05, 0.00027824307, 4...
53667   wide_resnet 9996  3      0.998296    [5.6268964e-06, 1.6285856e-06, 2.8455403e-05, ...
53668   wide_resnet 9997  5      0.999973    [8.2389795e-07, 3.947797e-07, 6.6969496e-06, 1...
53669   wide_resnet 9998  1      0.999932    [5.9105887e-06, 0.99993217, 2.001524e-05, 5.18...
53670   wide_resnet 9999  7      0.999981    [1.084445e-06, 9.401939e-07, 4.853471e-06, 2.0...
53671 rows × 5 columns
```

```
Confidence: 0.8312652
Confidence: 0.8312652
Confidence: 0.8312652
Confidence: 0.79878277
Confidence: 0.72223645
Confidence: 0.62768495
Confidence: 0.62768495
Confidence: 0.62768495
Confidence: 0.62768495
Confidence: 0.50536436
Confidence: 0.50536436
Confidence: 0.50536436
Confidence: 0.49788195
Confidence: 0.49788195
Confidence: 0.49788195
Confidence: 0.49788195
Confidence: 0.43340075
```

FIGURE 9. Evaluating model performance of Novel DESapsDE.

epochs to generate good-quality images as compare to Novel DESapsDE. The model predicts the image as a cat instead of an airplane with the effect of adding noise.

GAN models experimented using weights and biases on the MLOps platform with TESLA T4 configurations. Most of the previous work concentrated on gradient-based methods [6], [9], [18], [27] for generating adversarial samples, but practically getting gradient information is challenging, so many of the researchers concentrated on evolutionary algorithms. Much of the previous work was completed using GAN to visually evaluate images and it is difficult to assess the visual quality. The Frechet Inception distance (FID) [36] and Inception score (IS) [53] measures are most typically employed to assess image quality.

In DCGAN confidence, the label cannot infer the latent variable from input samples, and it requires low performance and produces many samples belonging to the same class. The images generated using SAGAN are more quality than DCGAN, but again it depends on the depth of the network. High-level feature maps gave better-quality images. The ACGAN produces the samples based on the class labels and does not require the probability to generate the images. The produced samples mostly show one of the classes. As shown in Figure 10, in the first row, most images are cars representing the latent space class conditional and partial.

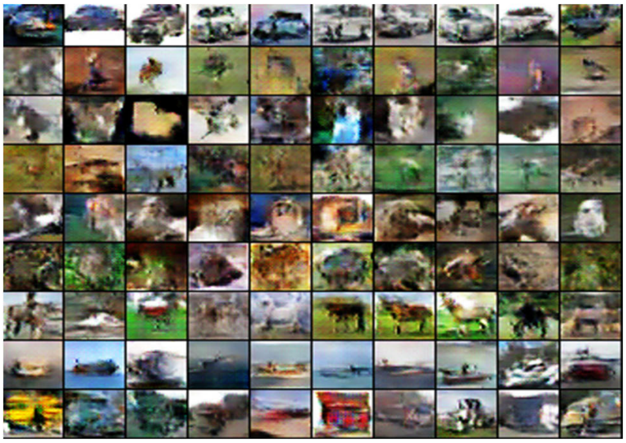


FIGURE 10. Adversarial images were generated using ACGAN.

Creating the complex structure is difficult because complex geometrical patterns require long-range information, which traditional convolution may not recognize. Specific categories of classes GAN can work well but often fail where non-local dependencies frequently appear in some classes of images.

Once the GAN model has been trained, the generative attack is quick and effective compared to the conventional optimization-based methods. The GAN black box attacks method does not work well and lacks transferability. In this experiment, DESapsDE is superior to adversarial attacks relating to accuracy. Compared to the results of GAN models, the quality of the images generated using novel DESapsDE is superior, as shown in Figure 11. The scenario is limited as samples contain only a few pixels of noise. Considering the attack rate, the WideResNet model has stronger resilience against noise attacks.

DESapsDE shows varying success rates across different models, with LeNet achieving the highest success rates in targeted and nontargeted attacks. Standard DE [21] and its variants demonstrate competitive performance, particularly in nontargeted attacks. JADE(Adaptive Differential Evolution) [54] also shows noteworthy success rates, with ResNet achieving high success rates in nontargeted attacks. Table 5 provides a clear comparison of the success rates of different models under targeted and non-targeted one-pixel attacks, offering insights into the robustness of these models against adversarial manipulations.

TABLE 5. Performance of the proposed system with other models on the CIFAR 10 dataset.

Methods	Models (1 pixel)	Success Rate (Targeted)	Success Rate (Non-Targeted)
DESapsDE	LeNet	60.07	85.13%
	ResNet-50	51	68.00%
	Net-in-Net	33.37	75.83%
	Dense_Net	54.11	93.67%
	WideResNet	24.40 %	42.77%
Std. DE[21]	All conv	23.46%	73.80 %
	Net-in-Net	26.32%	73.04 %
	VGG	19.78 %	66.08%
Std. DE[32] (Different Variant's)	All Conv	NA	71.86%
	Net-in-Net		77..64%
	VGG		56.47%
	BVLC		31.87%
JADE [54]	Lenet	53.5%	74.88%
	ResNet	32.5%	92.31%
	DenseNet	28.0%	94.67%

A deer or possibly a deer-horse-looking animal is the output of the classifier from the DCGAN model, and humans and other images can easily detect it, as shown in Figure 12. Most of the images generated do not belong to any of the classes. The images are familiar and similar to CIFAR-10 dataset images, but most images are not specified to one of the 10 classes. A human operator evaluates the quality of the images, knowing when to stop training the model is difficult in the GAN model. The training stops by observing the

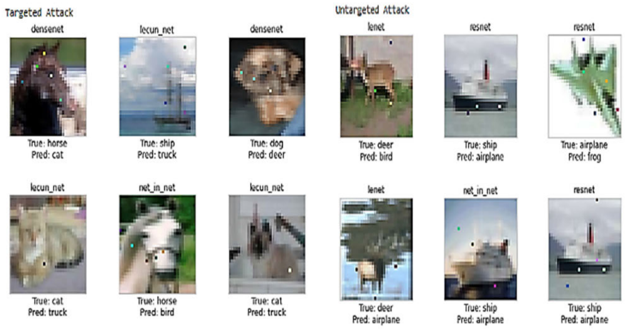


FIGURE 11. Adversarial samples were generated using DESapsDE (Differential evolution self-adaptive population resizing scheme) and verified using various CNN network models.

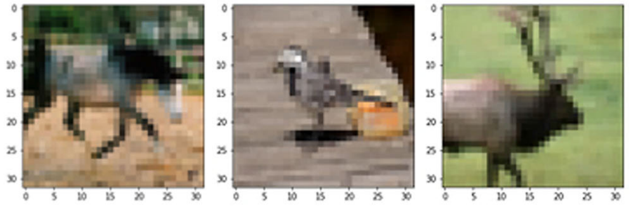


FIGURE 12. Images produced using the GAN model on CIFAR 10 dataset.

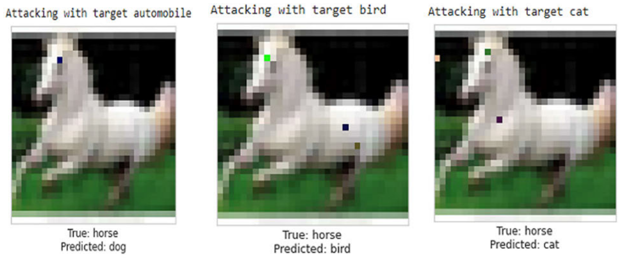

Attacking with target automobile
True: horse
Predicted: dog
Attacking with target bird
True: horse
Predicted: bird
Attacking with target cat
True: horse
Predicted: cat

FIGURE 13. Generated sample using DESapsDE applied over ResNet model and LeNet model.

generated images only. However, it is much more challenging to create geometrically complex structures.

Figure 13 shows the output generated using the Novel DESapsDE method. The real image corresponds to the horse class but is projected by the model as a dog, bird, or cat class. The goal is to undermine the confidence of models in the target class. These samples train the model and improve the system's resilience. The innovative DESapsDE method is effective for low-dimensional space.

V. CONCLUSION AND FUTURE WORK

Generative adversarial network models are more successful techniques and applicable in high dimensions. Many times, acquiring data may be costly. GAN works on both unsupervised and supervised learning data with handling multimodal capacity. The proposed work concentrates on low dimensional space and tries to solve the problem of the gradient being stuck in local space by including a population resizing scheme to increase convergence speed. GAN models frequently reject convergence due to switching between the generator and discriminator. This problem is solved by embedding the noise to the discriminator input or penalizing weights at the discriminator. Though researchers are working on convergence, the problem of stabilizing the network is still unresolved. GAN could be applicable to protect or defend against adversarial mechanisms. The discriminator model in GAN can be trained to resist the adversarial samples, and the system becomes more robust to such examples. The proposed model differs from the GAN and applies to attack-on-network models as a preventive major to make the model robust.

There are growing concerns about the security of deep neural networks (DNN) due to the susceptibility to adversarial samples.

The work introduces a novel DESapsDE framework based on evolutionary algorithms to generate adversarial samples, addressing the challenges associated with gradient-based methods. The approach is discussed in the context of various GAN models, emphasizing its potential as both an attack prevention measure and a way to enhance the robustness of deep neural networks against adversarial threats. The results demonstrate promising outcomes in reducing model confidence, providing valuable insights into improving the security of DNNs. The reported results show a reduction in model confidence for specific DNN models, such as ResNet50, WideResNet, and DenseNet, with an associated FID score of 16.45.

The future work concentrates on considering high dimensional space and more advanced differential evolutionary algorithms. The experiments can be conducted using changing population size, various strategies, constant of differentiation, number of steps included in the traversal phase, and constant of crossover.

ACKNOWLEDGMENT

This study is supported via funding from Prince sattam bin Abdulaziz University project number (PSAU/2023/R/1445).

REFERENCES

[1] A. Pavate and R. Bansode, "Design and analysis of adversarial samples in safety-critical environment: Disease prediction system," in *Artificial Intelligence on Medical Data* (Lecture Notes in Computational Vision and Biomechanics), vol. 37, M. Gupta, S. Ghatak, A. Gupta, and A. L. Mukherjee, Eds. Singapore: Springer, 2022, doi: 10.1007/978-981-19-0151-5_29.

[2] A. Vulli, P. N. Srinivasu, M. S. K. Sashank, J. Shafi, J. Choi, and M. F. Ijaz, "Fine-tuned DenseNet-169 for breast cancer metastasis prediction using FastAI and 1-cycle policy," *Sensors*, vol. 22, no. 8, p. 2988, Apr. 2022.

[3] S. P. Praveen, P. N. Srinivasu, J. Shafi, M. Wozniak, and M. F. Ijaz, "ResNet-32 and FastAI for diagnoses of ductal carcinoma from 2D tissue slides," *Sci. Rep.*, vol. 12, no. 1, p. 20804, Dec. 2022, doi: 10.1038/s41598-022-25089-2.

[4] S. Kamal, N. Dey, A. Ashour, S. Ripon, E. Balas, and M. Kaysar, "FbMapping: An automated system for monitoring Facebook data," *Neural Netw. World*, vol. 27, no. 1, pp. 27–58, Oct. 2017.

[5] A. A. Pavate and R. Bansode, "Analyzing probabilistic adversarial samples to attack cloud vision image classifier service," in *Proc. Int. Conf. Data Anal. Bus. Ind. (ICDABI)*, Oct. 2021, pp. 689–693.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014, *arXiv:1312.6199*.

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

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

[9] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," 2017, *arXiv:1708.03999*.

[10] M. Sharif, L. Bauer, and M. K. Reiter, "On the suitability of L_p-norms for creating and preventing adversarial examples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1–9.

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

[12] X. Gao, Y. Tian, and Z. Qi, "RPD-GAN: Learning to draw realistic paintings with generative adversarial network," *IEEE Trans. Image Process.*, vol. 29, pp. 8706–8720, 2020.

[13] C. Kotian, S. Lokhande, M. Jain, and A. Pavate, "D2F: Description to face synthesis using GAN," in *Proc. Int. Conf. Recent Adv. Comput. Techn. (IC-RACT)*, Jun. 2020, pp. 1–8.

[14] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*.

[15] F. Yu, L. Wang, X. Fang, and Y. Zhang, "The defense of adversarial example with conditional generative adversarial networks," *Secur. Commun. Netw.*, vol. 2020, Aug. 2020, Art. no. 3932584, doi: 10.1155/2020/3932584.

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

[17] J. Zhang, Y. Dong, M. Kuang, B. Liu, B. Ouyang, J. Zhu, H. Wang, and Y. Meng, "The art of defense: Letting networks fool the attacker," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 3267–3276, 2023, doi: 10.1109/TIFS.2023.3278458.

[18] M. Behera, A. Sarangi, D. Mishra, P. K. Mallick, J. Shafi, P. N. Srinivasu, and M. F. Ijaz, "Automatic data clustering by hybrid enhanced firefly and particle swarm optimization algorithms," *Mathematics*, vol. 10, no. 19, p. 3532, Sep. 2022, doi: 10.3390/math10193532.

[19] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Mar. 2016, pp. 372–387.

[20] A. A. Pavate and R. Bansode, "Generation of adversarial mechanisms in deep neural networks: A survey of the state of the art," *Int. J. Ambient Comput. Intell.*, vol. 13, no. 1, pp. 1–18, Mar. 2022, doi: 10.4018/ijaci.293111.

[21] J. Su, D. V. Vargas, and S. Kouichi, "One pixel attack for fooling deep neural networks," 2017, *arXiv:1710.08864*.

[22] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Black-box adversarial sample generation based on differential evolution," *J. Syst. Softw.*, vol. 170, Dec. 2020, Art. no. 110767.

[23] W. Luo, C. Wu, N. Zhou, and L. Ni, "Random directional attack for fooling deep neural networks," 2019, *arXiv:1908.02658v1*.

[24] J. Yoon, J. Jordon, and M. van der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in *Proc. Int. Conf. Learn. Represent.*, Los Alamitos, CA, USA, 2019, pp. 536–545.

[25] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, J. Lang, Ed., Stockholm, Sweden, Jul. 2018, pp. 3905–3911.

[26] A. Rozsa, E. M. Rudd, and T. E. Boult, "Adversarial diversity and hard positive generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 410–417.

[27] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.

[28] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Black-box adversarial sample generation based on differential evolution," 2020, *arXiv:2007.15310*.

[29] H. Shu, R. Shi, Q. Jia, H. Zhu, and Z. Chen, "MFI-PSO: A flexible and effective method in adversarial image generation for deep neural networks," 2020, *arXiv:2006.03243*.

[30] J. Renkhoff et al., "Exploring adversarial attacks on neural networks: An explainable approach," in *Proc. IEEE Int. Perform., Comput., Commun. Conf. (IPCCC)*, Austin, TX, USA, 2022, pp. 41–42, doi: 10.1109/IPCCC55026.2022.9894322.

[31] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 427–436.

[32] J. Su, D. V. Vargas, and K. Sakurai, "Attacking convolutional neural network using differential evolution," *IPSJ Trans. Comput. Vis. Appl.*, vol. 11, no. 1, p. 1, Dec. 2019.

[33] C. Veal, M. Lindsay, S. D. Kovaleski, D. T. Anderson, and S. R. Price, "Evolutionary algorithm driven explainable adversarial artificial intelligence," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Canberra, ACT, Australia, Dec. 2020, pp. 913–920.

[34] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "LOGAN: Membership inference attacks against generative models," *Proc. Privacy Enhancing Technol.*, vol. 2019, no. 1, pp. 133–152, Jan. 2019.

[35] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 250–258.

[36] 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 *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, 2017, pp. 1–12.

[37] G. Acs, L. Melis, C. Castelluccia, and E. De Cristofaro, "Differentially private mixture of generative neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 6, pp. 1109–1121, Jun. 2019.

[38] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.

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

[40] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 2642–2651.

[41] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2813–2821.

[42] S. Jandial, P. Mangla, S. Varshney, and V. Balasubramanian, "AdvGAN++: Harnessing latent layers for adversary generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2045–2048.

[43] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," presented at the 6th Int. Conf. Learn. Represent. (ICLR), Vancouver, BC, Canada, 2018, pp. 1–17.

[44] X. Fang, G. Cao, H. Song, and Z. Ouyang, "XGAN: Adversarial attacks with GAN," *Proc. SPIE*, vol. 11321, Nov. 2019, Art. no. 113211G, doi: 10.1117/12.2543218.

[45] B. Zhou and P. Krahenbuhl, "Don't let your discriminator be fooled," in Proc. 7th Int. Conf. Learn. Represent. (ICLR), New Orleans, LA, USA, May 2019, pp. 1–17.

[46] X. Liu and C.-J. Hsieh, "Rob-GAN: Generator, discriminator, and adversarial attacker," 2018, *arXiv:1807.10454*.

[47] J. Zhong, X. Liu, and C.-J. Hsieh, "Improving the speed and quality of GAN by adversarial training," 2020, *arXiv:2008.03364*.

[48] D. Wang, W. Jin, Y. Wu, and A. Khan, "Improving global adversarial robustness generalization with adversarially trained GAN," 2021, *arXiv:2103.04513*.

[49] T. Bai, J. Zhao, J. Zhu, S. Han, J. Chen, B. Li, and A. Kot, "AI-GAN: Attack-inspired generation of adversarial examples," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Anchorage, AK, USA, Sep. 2021, pp. 2543–2547, doi: 10.1109/ICIP42928.2021.9506278.

[50] W. Zhao, Q. H. Mahmoud, and S. Alwidian, "Evaluation of GAN-based model for adversarial training," *Sensors*, vol. 23, no. 5, p. 2697, Mar. 2023, doi: 10.3390/s23052697.

[51] *Dataset*. Accessed: May 3, 2023. [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html

[52] G. Liu, S. Lan, T. Zhang, W. Huang, and W. Wang, "SAGAN: Skip-attention GAN for anomaly detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 2468–2472.

[53] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 2234–2242.

[54] J.-I. Kushida, A. Hara, and T. Takahama, "Generation of adversarial examples using adaptive differential evolution," *Int. J. Innov. Comput., Inf. Control*, vol. 16, no. 1, pp. 405–414, Feb. 2020.

ARUNA PAVATE (Member, IEEE) is currently an Assistant Professor with the School of CSIT, Symbiosis Skills and Professional University, Pune, India, where she is involved in new advances in the field of medical and engineering to improve the healthcare domain. Her research interests include machine learning and security, data mining, and data science. She is a member of ISTE, IAENG, AICTSD, and Insc. She has been a Reviewer of many conferences and journals, such as *Journal of Electrical Engineering and Technology*, *Journal of Experimental and Theoretical Artificial Intelligence*, *Expert Systems with Applications*, and *Applied Artificial Intelligence*, and an Ad Hoc Reviewer of *International Journal of Ambient Computing and Intelligence*.

RAJESH BANSODE received the B.Tech. degree in electronics and communication engineering from Jawaharlal Nehru Technological University, Hyderabad, in 1999, the M.Tech. degree from Devi Ahilya Vishwavidyalaya, Indore, in 2001, and the Ph.D. degree in information technology (engineering and technology) from Sant Gadge Baba Amravati University, Amravati, in 2016. He is currently a full-time Professor with the Department of Information Technology, MR, and I.Q.A.C. Coordinator with the Thakur College of Engineering and Technology, Mumbai. He guided 30 bachelor's and 36 master's projects to date. He has published a total number of research publications in national and international conferences is 39 and international journals is 46. He is guiding six Ph.D. I.T. students and three completed with the Thakur College of Engineering and Technology. His research interests include network security, wireless communication in MIMO OFDM, and light-weight cryptography. He is a member of the Research Progress Monitoring Committee in other affiliated institutes of Mumbai University.

PARVATHANENI NAGA SRINIVASU received the bachelor's degree in computer science engineering from SSIET, JNTU Kakinada, in 2011, and the master's degree in computer science technology from the Gandhi Institute of Technology and Management, Visakhapatnam, in 2013. He is an Associate Professor with the Department of Computer Science and Engineering, Prasad V. Potluri Siddhartha Institute of Technology, India. He holding a Post-Doctoral Fellowship at the Department of Teleinformatics Engineering, Federal University of Ceará, Brazil, he also serves as a Research Fellow at INTI International University, Malaysia. His doctoral research at GITAM University focused on automatic segmentation methods for volumetric estimation of damaged areas in astrocytoma instances identified from 2D brain MR imaging. He expertise spans biomedical imaging, soft computing, explainable AI, and healthcare informatics, with a significant impact on academic literature through numerous publications in esteemed peer-reviewed journals and edited book volumes with renowned publishers, including Springer, Elsevier, IGI Global, and Bentham Science. Actively contributing to the scholarly community, he is a diligent reviewer for over 75 journals indexed in the Web of Science. He is also a Guest Editor and Technical Advisory Board Member for various internationally recognized conferences. His diverse contributions reflect a steadfast dedication to advancing research and knowledge in the fields of healthcare informatics and biomedical engineering.

JANA SHAFI is currently with the Department of Computer Science, Prince Sattam bin Abdulaziz University, Saudi Arabia. She has more than eight years of teaching and research experience. She has published in numerous journals, such as *Sensors*, IEEE ACCESS, *Diagnostics*, *Symmetry*, *Mathematics*, and *Wireless Communications and Mobile Computing*. Her research interests include online social networks, wearable technology, artificial intelligence, machine learning, deep learning, smart health, and the IoMT.

JAEYOUNG CHOI (Member, IEEE) received the B.S. and M.S. degrees from the Department of Mathematics, Korea University, South Korea, in 2008 and 2013, respectively, and the Ph.D. degree from the Department of Electrical Engineering, KAIST, in 2018. From 2018 to 2020, he was an Assistant Professor with the Department of Automotive Engineering, Honam University, South Korea. Since 2020, he has been an Assistant Professor with the School of Computing, Gachon University, South Korea, where he has been an Associate Professor, since 2023. His research interests include the intersection of applied mathematics and statistical inference, including social networks, wireless vehicular networks, and probabilistic graphical models.

MUHAMMAD FAZAL IJAZ received the Dr.Eng. degree in industrial and systems engineering from Dongguk University, Seoul, South Korea. He was a Research Assistant, a Visiting Guest Professor, and an Assistant Professor with tertiary institutes, including the Dongguk University; Technology De Monterrey, Campus Mexico City and Guadalajara, Mexico; Sejong University, Seoul, and the University of Melbourne, Australia. He has published numerous research articles in several international peer-reviewed journals, including IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE INTERNET OF THINGS JOURNAL, *Scientific Reports*, *Cancers*, and *Human-centric Computing and Information Science*. His research interests include machine learning, blockchain, healthcare engineering, the Internet of Things, big data, and data mining.