IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 20, 2019, accepted October 31, 2019, date of publication November 4, 2019, date of current version November 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2951526

DDSA: A Defense Against Adversarial Attacks Using Deep Denoising Sparse Autoencoder

YASSINE BAKHTI[1,2], SID AHMED FEZZA[1], WASSIM HAMIDOUCHE[2,3], (Member, IEEE), AND OLIVIER DÉFORGES[2]

[1]National Institute of Telecommunications and ICT, Oran 31000, Algeria
[2]INSA Rennes, CNRS, IETR - UMR CNRS 6164, Université de Rennes, 35700 Rennes, France
[3]IRT b<>com, 35510 Cesson-Sévigné, France

Corresponding author: Wassim Hamidouche (wassim.hamidouche@insa-rennes.fr)

ABSTRACT Given their outstanding performance, the Deep Neural Networks (DNNs) models have been deployed in many real-world applications. However, recent studies have demonstrated that they are vulnerable to small carefully crafted perturbations, *i.e.*, adversarial examples, which considerably decrease their performance and can lead to devastating consequences, especially for safety-critical applications, such as autonomous vehicles, healthcare and face recognition. Therefore, it is of paramount importance to offer defense solutions that increase the robustness of DNNs against adversarial attacks. In this paper, we propose a novel defense solution based on a Deep Denoising Sparse Autoencoder (DDSA). The proposed method is performed as a pre-processing step, where the adversarial noise of the input samples is removed before feeding the classifier. The pre-processing defense block can be associated with any classifier, without any change to their architecture or training procedure. In addition, the proposed method is a universal defense, since it does not require any knowledge about the attack, making it usable against any type of attack. The experimental results on MNIST and CIFAR-10 datasets have shown that the proposed DDSA defense provides a high robustness against a set of prominent attacks under white-, gray- and black-box settings, and outperforms state-of-the-art defense methods.

INDEX TERMS Deep neural network, security, adversarial attacks, defense, sparse autoencoder, denoising.

I. INTRODUCTION

Due to the increase use of deep neural networks (DNNs) models in many practical applications, especially security-sensitive applications, this raises an important issue as to their robustness against adversarial attacks. Recently, it has been shown that these models are vulnerable to small quasi-imperceptible perturbations, *i.e.*, adversarial examples, that can cause erroneous outputs [1], [2]. Different works have demonstrated successful generation of adversarial examples for different machine learning applications, such as speech recognition [3], robot vision [4] and image classification [5]. For instance, in the image classification domain, an adversarial example can be defined as an input image carefully crafted by an adversary that is correctly classified by humans, while is misclassified by the target DNN.

This is possible as all DNN models are based on the Independent and Identically Distributed (IID) assumption, which means that the training and the test sets roughly falls on a similar data distribution or the same manifold. Thus, the aim of an attacker is to maximize the loss of the targeted DNN on a given input, subject to the constraint that the perturbations remain imperceptible [6] and at the mean time cause a misclassification. To reach this objective, an attacker has to carefully craft inputs that are not drawn independently from each other and are not drawn from an identical distribution to what the model is trained on. In other words, the adversarial attacks are basically trying to find the shortest direction to push an input out of its decision boundary to either a targeted class or any other class (see Figure 1). The former is known as a *targeted* attack, and the latter as *untargeted* attack.

Different approaches for generating adversarial examples have been proposed, depending on the adversary's knowledge, these attacks can be divided into three main categories. The taxonomy of these categories is shown in Figure 2 and are defined as follows:

- *White-box* attacks: in this setting, the adversary has full access to both the defense strategy and the target model's architecture and parameters.

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

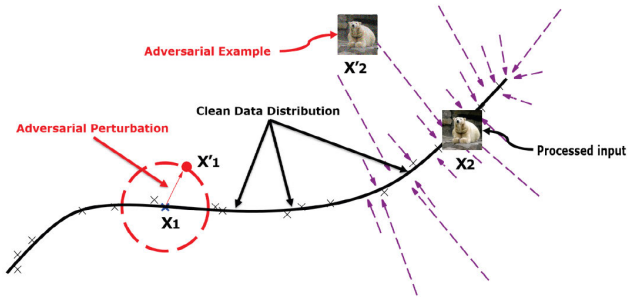


FIGURE 1. Data distribution over the manifold.

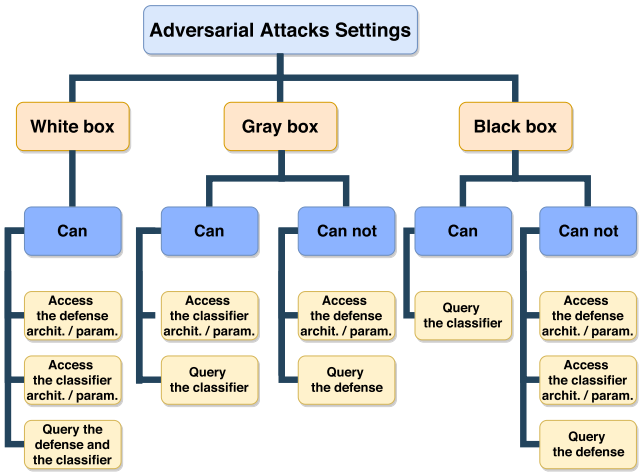


FIGURE 2. White, Gray and Black box attacks settings.

- *Black-box* attacks: in this setting, the adversary has no access to the model's architecture and parameters. The attacker only knows the output of the model (label or confidence score) for a given input. In this case, in order to generate an adversarial example, the attacker may train another model or a substitute one based on the output of the target model and exploits the transferability property of adversarial examples [5], [7].
- *Gray-box* attacks: in this setting, the adversary has full access to the model's architecture and parameters, but does not have any knowledge about the defense technique.

A defense against adversarial example x' aims to make the predicted class label of x' equals to that of clean sample x. The majority of defenses proposed in the literature are either attack-specific or model-specific defenses with strong limitations on the attacker, such as the allowed norm of the perturbation ϵ, the number of iterations or the black-box settings. Therefore, such defenses do not fulfill the *Kerckhoffs* principle [8], [9] and are not effective against new attacks.

In this paper, we propose a novel defense method based on a Deep Denoising Sparse Autoencoder (DDSA). Our approach aims to remove the adversarial noise from the input sample, using image denoising as a preprocessing. Then, the output of the proposed DDSA block is fed to the classifier, as illustrated in Figure 3. Knowing that the adversarial perturbations gradually increase as the image propagates through the network during the forward pass [10], [11], this leads to more noisy feature maps and inappropriate activations. These latter route the model's prediction to an incorrect label. To address this, we propose a shallower architecture with a sparsity constraint that ensures that a neuron fires only for meaningful patterns, which limits the hallucinated activations produced by the adversarial noise. Thus, the inclusion of sparsity constraint to the denoising autoencoder with training data including both clean and attacked samples, improves the robustness of DNN against highly challenging state-of-the-art adversarial attacks. In addition, the proposed method is thought to be a universal defense, which can defend against any attack without requiring any a priori knowledge about it.

The rest of this paper is organized as follows. Section II reviews several attack techniques and defense mechanisms that have been proposed in the literature. The proposed defense solution is detailed in Section III. The architectures and the performance of the proposed defense on both MNIST and CIFAR-10 datasets are provided and analysed in Section IV. Finally, Section V concludes this paper.

It is important to note that the most notations and symbols used in this paper are provided in Table 1.

II. RELATED WORK

In this section, first, the problem formulation of adversarial examples is introduced, then we present different attack models used to generate adversarial examples. Finally, some defense mechanisms against these attacks are described.

A. PROBLEM FORMULATION

Given an image space $\xi = [0, 1]^{H\times W\times C}$, a target classification model $f(\cdot)$ and a legitimate input image $x \in \xi$. An adversarial example is a perturbed image $x' \in \xi$ such that $f(x') \neq f(x)$ and $d(x, x') \leq \epsilon$, where $\epsilon \geq 0$. d is a distance metric to quantify the similarity between the perturbed and clean unperturbed inputs [6]. In the literature, three metrics are commonly used for generating adversarial examples, and all three are L_p norms, including L_0 distance, the Euclidean distance (L_2) and the *Chebyshev* distance (L_∞ norm) [12].

B. ADVERSARIAL EXAMPLES

In order to understand how DNN models make their classification decisions, Szegedy *et al.* [13] have studied their performance on the non-IID setting, which led to unexpected results. Starting with a clean image, they modified it by following the gradient of the probability of another class until the class changed, whereas the texture and shape are not modified. This pioneer work demonstrated that very small perturbations can completely change the output prediction of the classifier.

In the following, we describe some prominent attacks that we considered in the evaluation of our defense. For a complete description of the state-of-the-art attacks, the reader is refereed to the following review papers [14]–[16].

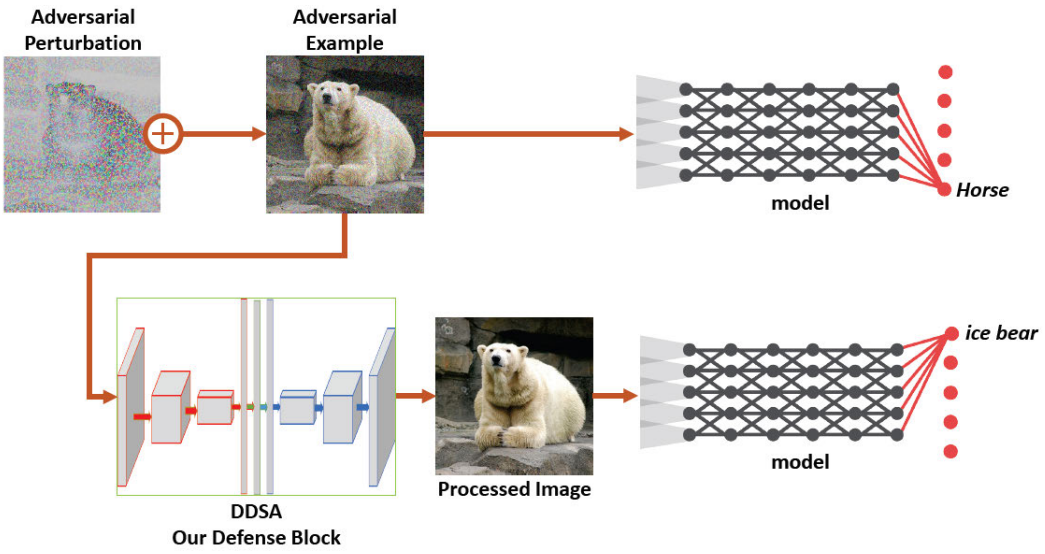


FIGURE 3. DDSA defense block against adversarial examples.

TABLE 1. The used notations and symbols.

Notations and Symbols	Definition
x	Clean image
x'	Adversarial image
c	Class label
$f(\cdot)$	Deep learning model (for classification task, $f \in F : \mathbb{R}^n \rightarrow c$)
ϵ	Perturbation norm
ξ	Image space
H, W, C	Height, Width, Channels of an image
θ	Parameters of the deep learning model f
α	Learning rate parameter
J	Loss function
∇	Gradient
$\|\| \cdot \|\|_p$	L_p norm
$a_i^{(j)}$	Activation of the i^{th} hidden unit at the j^{th} hidden layer
δ	Perturbation magnitude
μ	Sparsity parameter or Target activation
$\hat{\mu_i}$	Running estimate
b_i^{j-1}	Bias term of the i^{th} hidden unit in layer j
β	Learning rate to satisfy the sparsity

1) FAST GRADIENT SIGN METHOD

Goodfellow *et al.* [1] introduced a fast attack method called Fast Gradient Sign Method (FGSM). The FGSM performs only one step gradient update along the direction of the sign of gradient at each pixel as follows

$$x' = x + \epsilon \, sign(\nabla_x J_\theta(x, y)), \tag{1}$$

where θ is the set of model's parameters and $\nabla J(\cdot)$ computes the gradient of the loss function J around the current value of θ w.r.t. x. The sign$(\cdot)$ denotes the sign function and ϵ is a small scalar value that controls the perturbation magnitude.

Therefore, we can conclude that FGSM applies a first-order approximation of the loss function to construct the adversarial examples.

2) BOOSTING ADVERSARIAL ATTACKS WITH MOMENTUM

This attack is also refereed as Momentum Iterative Method (MIM). Dong *et al.* [17] increased the effectiveness of the FGSM attack by introducing the momentum term into its iterative process, which improves the transferability of adversarial examples. The gradients are calculated by

$$g_{t+1} = \omega \, g_t + \frac{\nabla_x J_\theta(x'_t, y)}{||\nabla_x J_\theta(x'_t, y)||_1}, \tag{2}$$

where g_t is the gradient at iteration t, ω is the decay factor and $||\cdot||_1$ is the L_1 distance.

Then, the adversarial example is calculated as follows

$$x'_{t+1} = x'_t + \epsilon \, sign\,(g_{t+1})\,. \tag{3}$$

3) PROJECTED GRADIENT DESCENT

The Projected Gradient Descent (PGD) has been introduced by Madry *et al.* in [18]. The authors formulated the generation of an adversarial example as a constrained optimisation problem. Specifically, they introduced the following saddle point optimization problem

$$\begin{aligned} &\min_\theta \rho(\theta), \\ &\text{with} \;\; \rho(\theta) = \mathbf{E}_{(x,y)\sim D}[\max_{\delta \in S} J_\theta(x+\delta, y)], \end{aligned} \tag{4}$$

where $\mathbf{E}$ is a risk function and δ is the magnitude of the perturbation.

This classic saddle point problem is a composition of an inner maximization problem and an outer minimization problem. The inner maximization is the same as attacking a neural network by finding an adversarial example. On the other hand, the outer minimization aims to minimize the adversarial loss.

4) CARLINI & WAGNER

Carlini and Wagner [12] introduced three attacks under three different distance metrics: L_0, L_2 and L_∞. The C&W attack aims to minimize a trade-off between the perturbation intensity $||\delta||_p$ and the objective function $g(x')$, with $x' = x + \delta$ and $g(x') \leq 0$ if and only if $f(x') = c$ and $f(x) \neq c$

$$\min_{\delta} ||\delta||_p + \lambda\, g(x'), \quad \text{such that } x' \in [0,1]^n, \tag{5}$$

where c is the target class and $\lambda > 0$ is a constant calculated empirically through binary search.

5) RANDOMIZED FAST GRADIENT SIGN METHOD

Rand+FGSM attack is an enhanced version of FGSM, aiming to increase its power against adversarial training defense [19]. Before using FGSM, the authors propose to add a small random noise to the clean input

$$\hat{x} = x + \phi\, sign(X), \tag{6}$$

where ϕ is the noise factor and X is a random vector defined in $\mathbb{R}^n$ by the multivariate Gaussian distribution $\mathcal{N}(0^n, I^n)$ of mean vector 0^n and identity co-variance matrix I^n, and $n = W \times H \times C$. Then, the FGSM attack is applied on the noisy version $\hat{x}$ as follows

$$x' = \hat{x} + (\epsilon - \phi)\, \nabla_{\hat{x}} J(\hat{x}, y) \quad \text{with } \phi < \epsilon. \tag{7}$$

C. DEFENSES

Different defense strategies have been used to deal with adversarial attacks [14]. In the following, we outline some of them.

1) ADVERSARIAL TRAINING

An obvious defense approach is to augment the training dataset with adversarial examples, which regularizes the network and reduce over-fitting, and therefore make the network more robust against adversarial attacks [1], [13], [20].

This defense can be useful if it has already been trained with the same kind of attack as that exploited by an adversary. However, adversarial training defense not tend to generalize across different attack strategies, thus leaving the classifier vulnerable to new/unknown attack models. Moreover, an adversarial example can again be computed on an already brute force trained networks [20].

2) DEFENSIVE DISTILLATION

Papernot *et al.* [21] exploited the notion of distillation introduced by Hinton *et al.* [22] as a defense mechanism against adversarial examples. First, the network is normally trained to the exception of raising the temperature of the softmax function to a large value (40-50), so that it produces smooth probability vectors. Then, these probability vectors are used to label the training data and the same architecture is retrained using these new labeled datasets to obtain the new distilled network. Finally, the prediction is performed, and subsequently, the temperature has to be set back to 1 at test time.

Therefore, instead of constraining the network to provide only the correct class, defensive distillation allows it to produce some scores for the other classes.

Defensive distillation tends to make the network more robust to white-box attacks. However, it does not perform well against black-box attacks [12].

3) MAGNET

Meng and Chen [23] introduced an effective defense framework that consists of two components:

1) *Detector*: rejects examples that are far from the manifold boundary.
2) *Reformer*: given an example x, reformer aims to find an example $\hat{x}$ within or close to the manifold. Thus, $\hat{x}$ is a close approximation to x.

For the detector part, the authors used an autoencoder to reconstruct any input x and map it with $\hat{x}$. Then, the two images are classified and, based on their probability divergence, it is decided whether it is adversarial or not. If it is not, the probability of the reconstructed image is considered as classification result.

III. PROPOSED METHOD

In this section, we first enumerate our supposed threat model, and subsequently, we will describe in details the proposed defense method.

A. THREAT MODEL

A threat model is an adversary with a specific set of assumptions. The proposed defense is build according to the threat model outlined below:

1) We assume that the attacker has full access to the classification model, *i.e.*, gray-box attack.
2) We consider x' as an adversarial example if and only if $d(x, x') \leq \epsilon$ and $f(x') \neq f(x)$.
3) We do not consider a specific classification model, we defend all types of classification models against any attack.

B. DEEP DENOISING SPARSE AUTOENCODER AS A DEFENSE AGAINST ADVERSARIAL ATTACKS

In order to deal with the adversarial perturbations, we propose to add a preprocessing block before any classification model (see Figure 3). As a preprocessing block, we propose a deep denoising sparse autoencoder or DDSA for short, as shown in Figure 4.

The DDSA is a variant of autoencoder. An autoencoder is a feed-forward unsupervised neural network algorithm that is trained to learn a compressed representation of an input, *i.e.,* identity function $f_\theta(x) \approx x$. Typically, the feed-forward autoencoder consists of two parts, the encoder and the decoder, where the encoder compresses the input into a lower-dimension, while the decoder reconstructs the

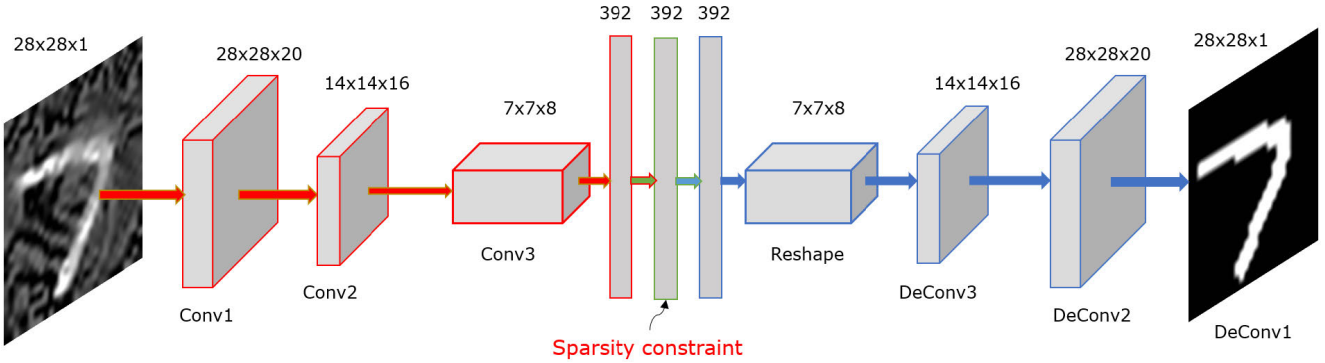


FIGURE 4. Deep denoising sparse autoencoder (DSSA).

output from this compressed representation. Furthermore, the autoencoders exploit the fact that the data distribution concentrates around a low-dimensional manifold and aim to learn the structure of that manifold [24]. The adversarial perturbations offset the data away from the manifold, and the function of the proposed DDSA is to push back this corrupted data to the learned manifold.

Thus, the idea behind the use of a denoising autoencoder is to learn a representation extracted from the autoencoder that is robust to adversarial perturbations. In other words, the denoising autoencoder performs a dimensionality reduction, thus allowing to remove or reduce the adversarial noise. This is done by minimizing its cost function J expressed as the mean squared error loss function

$$J = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - f_\theta(x_i')\right)^2, \tag{8}$$

where n is the number of samples and x' is the perturbed version of x. It is important to note that we only used the PGD attack to generate the perturbed samples. The latter have been exploited in the training stage, because it has been shown that adversarial training with PGD attack tends to generalize well across a wide range of attacks [18]. This assumption has been also verified by Carlini *et al.* in [25].

In addition, in order to maintain a good accuracy on clean samples and to regularize the proposed DDSA block, we used a training dataset including a mixture of clean and perturbed samples. The training formulation is designed as follows.

$$\theta^* = \arg\min_\theta \Big[\mathbb{E}_{(x,f_\theta(x'))\in\hat{p}_{data}}\big(\max_{\delta\in S} J\left(x, f_\theta(x')\right)\big) + \mathbb{E}_{(x,f_\theta(x))\in\hat{p}_{data}}\big(J(x, f_\theta(x))\big)\Big], \tag{9}$$

where $\hat{p}_{data}$ is the training data distribution.

Moreover, by adding a sparsity constraint to the Fully Connected (FC) layers of the proposed DDSA block, we constraint the neurons to be inactive most of the time in order to extract only meaningful and relevant features. Specifically, the sparsity constraint allows to force the activations of hidden units to be equal to some target activation μ. As we use a ReLU activation function, the value of μ is set to 0.1 (*i.e.*, a value close to 0)

$$\mathbb{E}_{x\sim D}[a_i^{(j)}] = \mu, \tag{10}$$

where x is the input image sampled from a distribution D and $a_i^{(j)}$ is the activation of the i^{th} hidden unit at the j^{th} hidden layer, which is the 2^{nd} one in our case.

Then, in each iteration of gradient descent, the activation of the hidden units is calculated for each i and a running-estimate $\hat{\mu}_i$ of the expectation is updated by the following formula

$$\hat{\mu}_i := 0.999\,\hat{\mu}_i + 0.001\,a_i^{(j)}, \tag{11}$$

where $\hat{\mu}_i$ is initialized to 0 ($\hat{\mu_0} = 0$), while the particular choice of 0.999 and 0.001 allows $\hat{\mu}_i$ to be an exponentially-decayed weighted average of about the last 1000 observed values of $a_i^{(j)}$.

Finally, the following learning rule is used

$$b_i^{j-1} := b_i^{j-1} - \alpha\,\beta\,(\hat{\mu}_i - \mu), \tag{12}$$

where b_i^{j-1} is the bias term that is used as a regulator for our constraint and α is the auto-encoder learning rate. Thus, when the condition $\hat{\mu}_i > \mu$ is valid, then we decrease the activation by decreasing b_i^{j-1} and vice versa, with β is the learning rate trying to satisfy the sparsity constraint.

Overall, at each iteration, the following steps are performed:

1) Forward pass to compute FC layers activation.
2) Backpropagation to update the weights using (9) and, consecutively, $\hat{\mu}_i$ and b_i^{j-1} are updated using (11) and (12), respectively.

IV. EXPERIMENTAL RESULTS

In order to evaluate the efficiency of the proposed defense method, we tested it against different set of attacks, including FGSM [1], Rand+FGSM [19], MIM [17], PGD [18] and C&W [12]. These latter represent a broad range of attack models proposed in the literature.

TABLE 2. Classifiers architectures for MNIST and CIFAR-10.

MNIST Architecture	CIFAR-10 Architecture
Conv2D(32, (3, 3))	Conv2D(96, (3, 3))
ReLU()	ReLU()
MaxPooling2D((2, 2))	Conv2D(96, (3, 3))
Dropout(0.25)	ReLU()
Flatten()	MaxPooling2D((2, 2))
Dense(1024)	Dropout(0.25)
ReLU()	Conv2D(64, (3, 3))
Dropout(0.25)	ReLU()
Dense(10)	Conv2D(64, (3, 3))
Softmax()	ReLU()
	Flatten()
	Dense(1024)
	ReLU()
	Dense(10)
	Softmax()

Additionally, the performance of the proposed DDSA defense have been compared to two state-of-the-art defenses, namely MagNet [23] and adversarial training [1], in addition to the trained Deep Denoising Autoencoder (DDA) corresponding to the proposed DDSA method without the sparsity constraint. The experiments were performed on the MNIST [26] and CIFAR-10 [27] datasets:

- **MNIST** dataset [26]: consists of handwritten digits. MNIST was split into training, validation and test sets with 50,000, 10,000 and 10,000 samples, respectively.
- **CIFAR-10** dataset [27]: consists of 60,000 images, where each image has the following dimension $32 \times 32 \times 3$. CIFAR-10 contains ten different classes, and each class is divided into 5,000 training images and 1,000 test images.

Our implementation is build on open-source software *CleverHans* library [28] based on TensorFlow [29], with NVIDIA GEFORCE GTX 1080 ROG.

In the following, first, we describe the different classifiers and defenses architectures that we used for MNIST and CIFAR-10 datasets. Then, the performance of the DDSA defense under the three attack levels, namely black-box, gray-box and white-box, are presented.

A. CLASSIFIERS AND DEFENSES ARCHITECTURES

1) CLASSIFIERS ARCHITECTURES

Table 2 describes classifiers architectures of MNIST and CIFAR-10 datasets. For these considered classifiers, we obtained accuracies of 98.96% and 81.42% on the MNIST and CIFAR-10 datasets, respectively, which is not far from state-of-the-art performance on these datasets.

The two architectures are based on convolutional neural networks with the format Conv2D and fully connected layers with the format dense (number of hidden units). We used the rectified linear unit activation function and a dropout layer with a rate of 0.25 to prevent overfitting.

Table 3 presents the training parameters used to train our classifiers. We used the Adam optimization algorithm for the MNIST classifier and RMSprop for the CIFAR-10 classifier,

TABLE 3. Training parameters of classification models.

Parameters	MNIST	CIFAR-10
Optimization Method	Adam	RMSprop
Learning Rate	0.01	0.001
Batch Size	256	32
Epochs	64	64

TABLE 4. Deep denoising sparse autoencoder architectures.

DDSA MNIST Architecture	DDSA CIFAR-10 Architecture
Conv2D(20, (3, 3))	Conv2D(128, (3, 3))
ReLU()	ReLU()
Conv2D(16, (3, 3))	BatchNormalization()
ReLU()	Conv2D(64, (3, 3))
MaxPooling2D((2, 2))	ReLU()
Conv2D(8, (3, 3))	Dropout(0.25)
ReLU()	MaxPooling2D((2, 2))
MaxPooling2D((2, 2))	Conv2D(32, (3, 3))
Flatten()	ReLU()
Dense(392)	MaxPooling2D((2, 2))
Dense(392)	Flatten()
Dense(392)	Dense(2048)
Reshape((7, 7, 8))	Dense(2048)
Conv2DTranspose(8, (3, 3))	Dense(2048)
ReLU()	Reshape((8, 8, 32))
Conv2DTranspose(16, (3, 3))	Conv2DTranspose(32, (3, 3))
ReLU()	ReLU()
UpSampling2D((2, 2))	UpSampling2D((2, 2))
Conv2DTranspose(20, (3, 3))	Conv2DTranspose(64, (3, 3))
ReLU()	ReLU()
UpSampling2D((2, 2))	UpSampling2D((2, 2))
Conv2DTranspose(1, (3, 3))	Conv2DTranspose(128, (3, 3))
Sigmoid()	ReLU()
	Conv2DTranspose(3, (3, 3))
	Sigmoid()

which are the most commonly used learning algorithms. The learning rate was set to 0.01 for the MNIST classifier and to 0.001 for the CIFAR-10 classifier, and we have not used the learning rate decay as it slowed down the convergence significantly. The two classifiers have been trained for 64 epochs with a batch size of 256 and 32 for MNIST and CIFAR-10 datasets, respectively.

2) DEFENSES ARCHITECTURES

Table 4 summarizes the architectures of our DDSA defense for both MNIST and CIFAR-10 datasets. The two architectures are based on convolutional neural networks with Conv2D layer, dense layers and a dropout using the ReLU activation function. In addition, we used a Conv2DTranspose for the deconvolution process and BatchNormalization to normalize the hidden units activations and therefore speed up the learning.

The network architecture of the reformer (encoder) used for the MagNet defense, as specified in [30], is provided in Table 5.

Finally, for the adversarial training defense, in addition to the clean samples, also the training was performed using the FGSM adversarial samples. These latter have been generated

TABLE 5. Architecture of MagNet neural network encoder.

Encoder
Conv(64, 5x5, 2)
LeakyReLU(0.2)
Conv(128, 5x5, 2)
LeakyReLU(0.2)
Conv(256, 5x5, 2)
LeakyReLU(0.2)
FC(128)
Tanh()

using a perturbation magnitude of $\epsilon = 0.3$. These choices are the most adopted in the literature.

B. DEFENSE EVALUATION

By following the recommendations of [31], our defense was evaluated under the three different attack settings: 1) Black-box attack, 2) Gray-box attack and 3) White-box attack, using a plethora of different attacks, as is described in the next sections.

1) RESULTS ON BLACK-BOX ATTACKS

In this section, we present defense results against FGSM, Rand+FGSM, MIM, PGD and C&W as black-box attackers. As previously described, in black-box setting, the attacker has no access to the classifier and defense parameters. Thus, as performed in [19], [32], to simulate black-box attack, we train a substitute network with 150 samples taken randomly from the test set augmented with synthetic images labeled according to the output of the target classifier. Then, the adversarial examples generated by this substitute network are used to attack the classifier.

Table 6 reports the performance of our DDSA defense against the five considered black-box attacks and is compared to three other defense strategies. Theses results have been obtained with $\epsilon = 0.3$ as perturbation magnitude for FGSM, Rand+FGSM, MIM and PGD attacks. In addition, for Rand+FGSM, we fixed the first noise of magnitude to $\alpha = 0.05$, while we used the L_2 norm for C&W attack with a maximum of iterations of 1000, a confidence of 10 and a learning rate of 1.0.

According to the obtained results, it is clear that all the attacks have succeeded in greatly reducing the classifier accuracy of up to 89% (*i.e.*, without any defense solution). On the other hand, all the defenses have globally succeeded in decreasing the effect of the attacks, with relatively large differences in performance. For instance, the adversarial training defense achieved 76.6% and 68.5% of classification accuracy for FGSM and Rand+FGSM attacks, respectively. Thus, performed better than MagNet defense which respectively obtained 60.7% and 52.2% accuracy for the same attacks. These results were expected since the same FGSM attack has been used to train the adversarial training method. This is confirmed by the low classification accuracy obtained by the adversarial training for the remaining attacks. Whereas, Magnet defense achieved somewhat stable results across the different attacks. However, the performance of these defenses are clearly lower than the proposed DDSA defense that achieved significant increase in the classification accuracy. Moreover, although our method was trained using PGD attack, the improvements achieved are not attack-dependent and concerns the whole considered attacks, making our method a general defense applicable to any attacks. In addition, compared the results of our DDSA method to those of DDA show clearly the add value of the sparsity constraint, thus allowing a consistent performance improvement.

Consequently, thanks to the use of denoising autoencoder with sparcity constraint as preprocessing block, the proposed method is efficient in removing the adversarial noise from the input images, making the classifier safer.

2) RESULTS ON GRAY-BOX ATTACKS

In gray-box attack level, the attacker has knowledge of the classifier's parameters without the defense strategy. Table 7 presents our defense results for the gray-box settings as well as for the DDA defense on the MNIST dataset. From this table, we can see that the gray-box attacks are more efficient than the black-box attacks in reducing the classification accuracy, making them more challenging to defend.

As with the previous black-box attacks, the proposed defense performs well against the gray-box attacks. Despite the classification accuracy is slightly smaller than the results of black-box attacks, however, compared to no defense case the DDSA increases the accuracy by up of 87%, which is more significant than the black-box results.

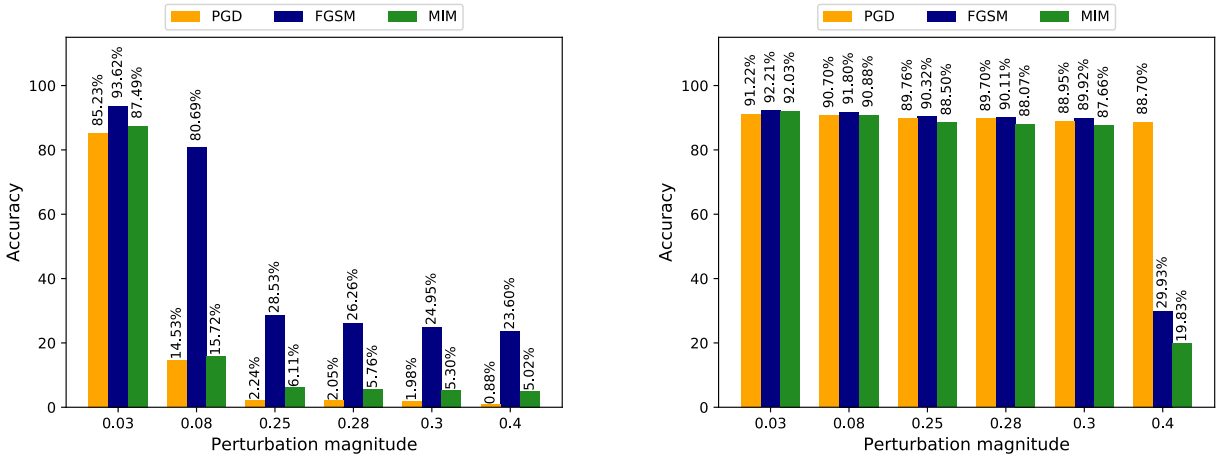
As illustrated in table 7, the obtained results confirm the importance of using a denoising sparse autoencoder instead of a normal DDA method with a mean squared error loss function. In contrast to black-box results, where DDA defense provides an acceptable result, against gray-box attacks, the DDA is far from sufficient. However, the proposed DDSA method shows that it is also robust against gray-box attacks. Consequently, it is clear that our defense is performing way better than the DDA method under all considered gray-box attacks.

In addition, a defense is considered effective, if it resists to a wide variety of attacks as well as different attack parameters. Thus, in order to assess the effect of different attack parameters on DDSA's performance, in Figure 5, we show the accuracy of classifier on MNIST dataset with and without any defense against PGD, FGSM and MIM gray-box attacks, with different perturbation magnitudes. In Figure 5(a), it can be seen that the accuracy of classifier gradually decreases while augmenting the magnitude of the perturbation ϵ. Therefore, the attacks are more successful with higher ϵ as the accuracy dropped with nearly 60%, however, at the expense of being more detectable and visually perceptible.

Figure 5(b) shows the accuracy of MNIST classifier using our DDSA defense under the same gray-box attacks with the same perturbations magnitude. Thanks to the inclusion

TABLE 6. Classification accuracy using different defense strategies under various black-box attacks on the MNIST dataset. Note that DDA refers to denoising autoencoder without sparsity constraint.

Attack	No Attack	No Defense	**DDSA**	MagNet	Adv. Training	DDA
FGSM $\epsilon = 0.3$	0.989	0.412	**0.902**	0.607	0.766	0.885
Rand+FGSM $\epsilon = 0.3, \alpha = 0.05$	0.989	0.245	**0.889**	0.522	0.685	0.849
C&W L_2 norm	0.989	0.091	**0.841**	0.588	0.290	0.721
MIM $\epsilon = 0.3$	0.989	0.180	**0.889**	0.674	0.608	0.799
PGD $\epsilon = 0.3$	0.989	0.254	**0.911**	0.671	0.672	0.896



(a) without any defense

(b) with our DDSA defense

FIGURE 5. Accuracy of the classifier against FGSM, PGD and MIM gray-box attacks on the MNIST dataset: (a) without any defense, and (b) after adding our DDSA defense block.

TABLE 7. Classification accuracy using different defense methods under various gray-box attacks on the MNIST dataset.

Attack	No Attack	No defense	**DDSA**	DDA
FGSM $\epsilon = 0.3$	0.989	0.249	**0.899**	0.715
Rand+FGSM $\epsilon = 0.3, \alpha = 0.05$	0.899	0.165	**0.849**	0.382
C&W L_2 norm	0.989	0.011	**0.811**	0.140
MIM $\epsilon = 0.3$	0.989	0.053	**0.876**	0.238
PGD $\epsilon = 0.3$	0.989	0.019	**0.889**	0.741

of DDSA defense, the classifier is now highly confident and robust against the attacks even for large values of ϵ. The enhancement is significant for all ϵ values, except for the $\epsilon = 0.4$ that generates very noisy image, making them difficult to classify, even by a human. However, this is not the case for PGD attack, which may be explained by the fact that our DDSA was trained including this attack. Some adversarial images defended by DDSA from MNIST dataset are shown in Figure 7.

Furthermore, we investigated the effect of varying the number of PGD attack iterations, ranging from 10 to 1000,

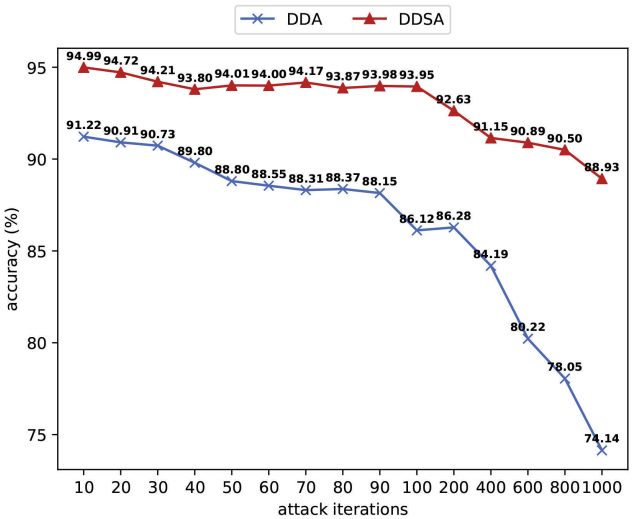


FIGURE 6. Defense against gray-box PGD attack with 10 to 1000 attack iterations with a perturbation of $\epsilon = 0.3$.

on the defense performance. Figure 6 shows that our defense is more robust than DDA defense, where the latter decreases rapidly with the increase of attack iterations, this is more

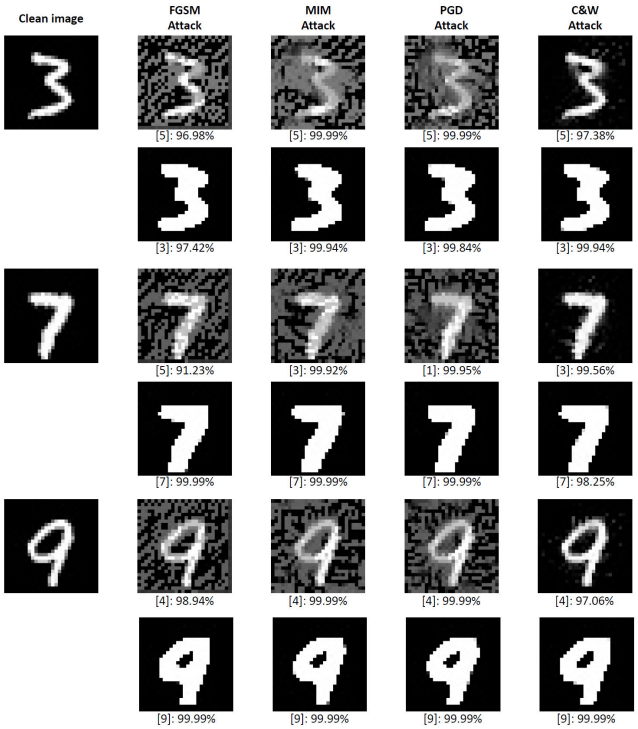


FIGURE 7. Examples of defended images by DDSA from MNIST dataset. For each digit, the top row shows the adversarial images with $\epsilon = 0.3$, while the bottom row represents the defended (denoised) images. The predicted class label and its corresponding probability are provided for each image.

TABLE 8. Classification accuracy under various gray-box attacks on the CIFAR-10 dataset.

Attack	No Attack	No Defense	**DDSA**	DDA
FGSM $\epsilon = 0.2$	0.814	0.131	**0.610**	0.489
Rand+FGSM $\epsilon = 0.2, \alpha = 0.05$	0.814	0.092	**0.543**	0.297
C&W L_2 norm	0.814	0.000	**0.367**	0.095
MIM $\epsilon = 0.2$	0.814	0.036	**0.584**	0.428
PGD $\epsilon = 0.2$	0.814	0.021	**0.611**	0.583

noticeable between 90 and 1000 iterations. However, our defense approach provides somewhat stable results even under the utmost 1000-iteration PGD attack for which our DDSA has obtained 88.93% accuracy compared to only 74.14% accuracy achieved by the DDA.

Table 8 reports the performance of our defense against different gray-box attacks on the CIFAR-10 dataset. FGSM, Rand+FGSM, MIM and PGD attacks have been performed with a perturbation magnitude of $\epsilon = 0.2$ that represents, according to [31], a high distortion that makes most images completely unrecognizable and hardly to classify.

The obtained results on CIFAR-10 dataset are lower than those achieved on MNIST dataset, it is clear that the CIFAR-10 dataset is much more challenging to defend than the MNIST one. However, here again our DDSA provides

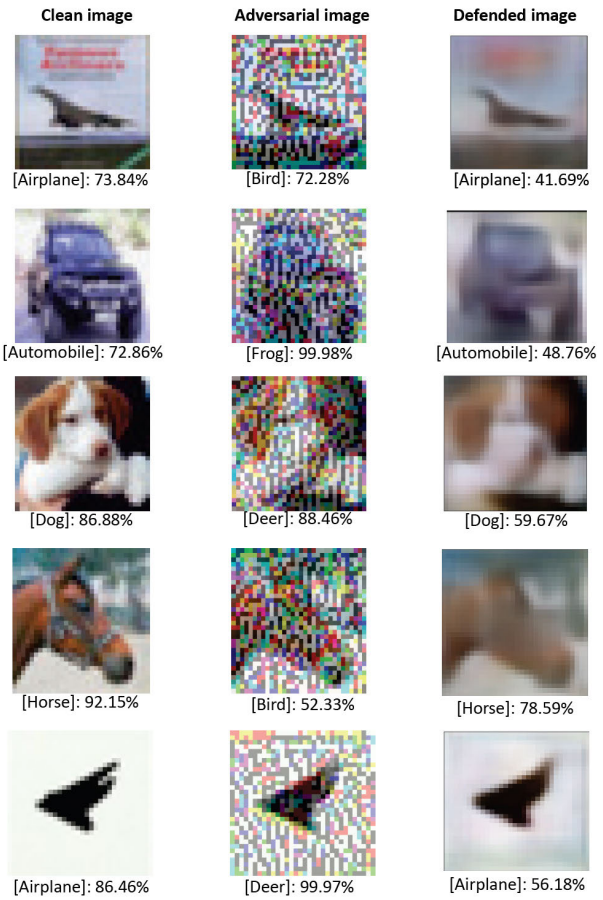


FIGURE 8. Images from CIFAR-10 dataset are ordered form left to right as the clean image, adversarial image perturbed with FGSM ($\epsilon = 0.2$) and the defended (denoised) image. The predicted class label and its corresponding probability are provided for each image.

an increase in the classification accuracy and surpassed the DDA method for all the considered attacks. The proposed DDSA achieves the highest accuracy against the PGD attack with 61.14% accuracy, while obtained the lowest accuracy against C&W attack. Because it is important to specify that the C&W attack is one of the most powerful attacks that generates adversarial samples that are visually very close to the clean image, *i.e,* adversarial images with very low noise. As evidence of its effectiveness, it reduces the classification accuracy to 0% and, by using the DDSA defense, the accuracy of classifier goes to 36.7%.

Furthermore, on the CIFAR-10 dataset, except for the C&W attack, the remaining attacks provide adversarial images that are highly perturbed and, therefore, are hardly classified even for the human. Some visual examples from CIFAR-10 dataset are illustrated in Figure 8. The achievements of our DSSA method are comparable to the state-of-the-art accuracy performance on the CIFAR-10 dataset [12], [33], [34].

3) RESULTS ON WHITE-BOX ATTACKS

In white-box attacks, the attacker has access to all of the information about the classifier and defense mechanism.

TABLE 9. Classification accuracy using different defense strategies under various white-box attacks on the MNIST dataset.

Attack	No Attack	No Defense	**DDSA**	MagNet	Adv. Training	DDA
FGSM $\epsilon = 0.3$	0.989	0.152	0.614	0.094	**0.651**	0.295
Rand+FGSM $\epsilon = 0.3, \alpha = 0.05$	0.989	0.104	0.427	0.115	**0.539**	0.300
C&W L_2 norm	0.989	0.000	**0.038**	0.021	0.010	0.022
MIM $\epsilon = 0.3$	0.989	0.010	**0.345**	0.264	0.204	0.200
PGD $\epsilon = 0.3$	0.989	0.000	**0.838**	0.177	0.162	0.197

In Table 9, we report the classification accuracy under FGSM, Rand+FGSM, C&W, MIM and PGD white-box attacks. As can be seen, our DDSA method outperforms MagNet, adversarial training and DDA defenses for C&W, MIM and PGD attacks, but does not perform as well as adversarial training on FGSM and Rand+FGSM attacks. This can be explained by the fact that adversarial training defense has been trained on an FGSM images perturbed set. However, the DDSA results for both FGSM attacks may be acceptable compared to non-use of a defense.

Moreover, although, adversarial training method obtained a good result on both FGSM and Rand+FGSM attacks, it achieved the lowest accuracy on the C&W, MIM and PGD attacks. This is due to the fact that this defense does not generalize as well as DDSA on other attacks.

V. CONCLUSION

In this paper, we proposed DDSA method, a defense against adversarial attacks. The proposed defense, which consists of applying sparsity constraint on a denoising autoencoder, is used as a preprocessing block applied to the input samples for removing the effect of adversarial noise. The proposed method allows to increase the adversarial robustness of DNNs and has been designed in such a way that can be deployed with any classifier without any change to its architecture or training stage. In addition, since it does not require any knowledge about the attack, the DDSA defense can be used against any type of attack.

The proposed defense has been evaluated against FGSM, Rand+FGSM, C&W, MIM and PGD attacks under black-box, gray-box and withe-box settings on two standard datasets. The experimental results demonstrated that the proposed defense can provide a significant improvement in the robustness of DNNs against adversarial attacks, and outperforms two state-of-the-art defenses.

Even though the DDSA defense is providing satisfactory results, we seek to improve our method against the highly challenging C&W attack and white-box attack settings. In addition, we believe that enhancing our defense by making it hardly differentiable and randomized can increase the robustness against gradient-based and non gradient-based white-box attacks.

REFERENCES

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*. [Online]. Available: https://arxiv.org/abs/1412.6572

[2] 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.

[3] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured prediction models," 2017, *arXiv:1707.05373*. [Online]. Available: https://arxiv.org/abs/1707.05373

[4] M. Melis, A. Demontis, B. Biggio, G. Brown, G. Fumera, and F. Roli, "Is deep learning safe for robot vision? Adversarial examples against the iCub humanoid," in *Proc. IEEE Int. Conf. Comput. Vis. (CVPR)*, Oct. 2017, pp. 751–759.

[5] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," 2016, *arXiv:1611.02770*. [Online]. Available: https://arxiv.org/abs/1611.02770

[6] S. A. Fezza, Y. Bakhti, W. Hamidouche, and O. Déforges, "Perceptual evaluation of adversarial attacks for CNN-based image classification," in *Proc. 11th IEEE Int. Conf. Qual. Multimedia Exper. (QoMEX)*, Jun. 2019, pp. 1–6.

[7] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: From phenomena to black-box attacks using adversarial samples," 2016, *arXiv:1605.07277*. [Online]. Available: https://arxiv.org/abs/1605.07277

[8] A. Kerckhoffs, "La cryptographie militaire," *J. des Sci. Militaires*, vol. 9, pp. 5–38, Jan. 1883.

[9] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[10] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," 2017, *arXiv:1711.00117*. [Online]. Available: https://arxiv.org/abs/1711.00117

[11] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1778–1787.

[12] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*. [Online]. Available: https://arxiv.org/abs/1312.6199

[14] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[15] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[16] S. Qiu, Q. Liu, S. Zhou, and C. Wu, "Review of artificial intelligence adversarial attack and defense technologies," *Appl. Sci.*, vol. 9, no. 5, p. 909, 2019.

[17] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 9185–9193.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*. [Online]. Available: https://arxiv.org/abs/1706.06083
[19] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," 2017, *arXiv:1705.07204*. [Online]. Available: https://arxiv.org/abs/1705.07204
[20] 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.
[21] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597.
[22] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: https://arxiv.org/abs/1503.02531
[23] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 135–147.
[24] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 899–907.
[25] N. Carlini, G. Katz, C. Barrett, and D. L. Dill, "Provably minimally-distorted adversarial examples," 2018, *arXiv:1709.10207v2*. [Online]. Available: https://arxiv.org/pdf/1709.10207v2.pdf
[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[27] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009. [Online]. Available: https://openreview.net/pdf?id=Hki-ZlbA-
[28] N. Papernot *et al.*, "Technical report on the CleverHans v2.1.0 adversarial examples library," 2016, *arXiv:1610.00768*. [Online]. Available: https://arxiv.org/abs/1610.00768
[29] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," White Paper, Nov. 2015. [Online]. Available: https://arxiv.org/pdf/1603.04467.pdf
[30] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," 2018, *arXiv:1805.06605*. [Online]. Available: https://arxiv.org/abs/1805.06605
[31] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," 2019, *arXiv:1902.06705*. [Online]. Available: https://arxiv.org/abs/1902.06705
[32] P. Nicolas, M. Patrick, G. Ian, J. Somesh, C. Z. Berkay, and S. Ananthram, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519.
[33] V. Srinivasan, A. Marban, K.-R. Müller, W. Samek, and S. Nakajima, "Robustifying models against adversarial attacks by Langevin dynamics," 2018, *arXiv:1805.12017*. [Online]. Available: https://arxiv.org/abs/1805.12017
[34] S. Song, Y. Chen, N.-M. Cheung, and C.-C. J. Kuo, "Defense against adversarial attacks with saak transform," 2018, *arXiv:1808.01785*. [Online]. Available: https://arxiv.org/abs/1808.01785

YASSINE BAKHTI is currently pursuing the degree in engineering with the National Institute of Telecommunications and ICT, Oran, Algeria. He is an Intern with INSA Rennes, CNRS, IETR, Rennes, France, working on adversarial attacks and defenses against deep neural networks. His research interests include machine learning and data science.

SID AHMED FEZZA received the degree in engineer from the University of Dr. Tahar Moulay, Saïda, Algeria, in 2007, and the Ph.D. degree from the Djillali Liabes University of Sidi-Belabbes, Algeria, in 2015, all in computer science. He is currently an Associate Professor with the National Institute of Telecommunications and ICT (INTTIC), Oran, Algeria. His research interests include image/video processing, image/video coding, visual quality assessment, immersive multimedia communication, multimedia security, and multimedia content protection. He was a recipient of two top 10% Best Paper Awards in ICIP 2014, the 2015 Algerian Paper of the Year Awards from the Algerian Network for Academics, Scientists and Researchers and has authored several publications in top journals and conferences on image and video processing.

WASSIM HAMIDOUCHE (M'15) received the Ph.D. degree in signal and image processing from the University of Poitiers, France, in 2010. From 2011 to 2012, he was a Research Engineer with the Canon Research Centre, Rennes, France. Since 2015, he has been an Associate Professor with INSA Rennes. He is currently a member of the Institute of Electronics and Telecommunications of Rennes (IETR), UMR CNRS 6164. His research interests include video coding, efficient real time and parallel architectures for the new generation video coding standards, multimedia transmission over heterogeneous networks, and multimedia content security.

OLIVIER DÉFORGES received the Ph.D. degree in image processing, in 1995. In 1996, he joined the Department of Electronic Engineering, National Institute of Applied Sciences of Rennes (INSA), Scientific and Technical University. He is currently a Professor with INSA. He is a member of the Institute of Electronics and Telecommunications of Rennes (IETR), UMR CNRS 6164. He has authored more than 180 technical articles. His principal research interests include image and video lossy and lossless compression, image understanding, fast prototyping, and parallel architectures.