Biologically Inspired Variational Auto-Encoders for Adversarial Robustness

Sameerah Talafha[1(✉)], Banafsheh Rekabdar[2], Christos Mousas[3], and Chinwe Ekenna[4]

[1] Southern Illinois University Carbondale, Carbondale, IL, USA
sameerah.talafha@siu.edu
[2] Portland State University, Oregon, USA
rekabdar@pdx.edu
[3] Purdue University, West Lafayette, Indiana, USA
cmousas@purdue.edu
[4] University at Albany, Albany, NY, USA
cekenna@albany.edu

Abstract. Deep Neural Networks (DNNs) have recently become the standard tools for solving problems that can be prohibitive for human or statistical criteria, such as classification problems. Nevertheless, DNNs have been vulnerable to small adversarial perturbations that cause misclassification of legitimate images. Adversarial attacks show a security risk to deployed DNNs and indicate a divergence between how DNNs and humans perform classification. It has been illustrated that sleep improves knowledge generalization and improves robustness against noise in animals and humans. This paper proposes a defense algorithm that uses a biologically inspired sleep phase in a Variational Auto-Encoder (Defense–VAE–Sleep) to purge adversarial perturbations from contaminated images. We are demonstrating the benefit of sleep in improving the generalization performance of the traditional VAE when the testing data differ in specific ways even by a small amount from the training data. We conduct extensive experiments, including comparisons with the state–of–the–art on three datasets: CelebA, MNIST, and Fashion-MNIST. Overall, our results demonstrate the robustness of our proposed model for defending against adversarial attacks and increasing the classification robustness solutions compared with other models: Defense–VAE and Defense–GAN.

Keywords: Defense mechanism · VAE · Sleep algorithm · Adversarial robustness

1 Introduction

Although DNNs have demonstrated success on various classification tasks, adversarial attacks [1] formulated as minute perturbations to inputs drive DNNs to classify incorrectly. For images, these perturbations can drastically impact the classifiers based on DNNs, even though these perturbed inputs are imperceptible to humans. Two general categories: (a.) white-box attack and (b.) black-box attack have been tested to attack DNNs, posing a severe threat to different safety-critical applications such as

I. Awan et al. (Eds.): DBB 2022, LNNS 541, pp. 79–93, 2023.
https://doi.org/10.1007/978-3-031-16035-6_7

autonomous driving, healthcare, and education [2]. Third generation Neural Networks (NNs), Spiking Neural Networks (SNNs) – being recent members of the neural network's family – usage for combating adversarial attacks is still new and limited compared to what was achieved with DNNs. However, the latest researches show that SNNs are more robust than DNNs under some types of attacks [1]. Sleep mechanism is essential to animals' and humans' brain functions, including how their neurons contact each other. During sleep, neurons in the previously learned activity get reactivated, likely simulating similar observations of the spatiotemporal patterns as training in the awake phase. Sleep–inspired algorithm built using SNN has proved its ability to decrease catastrophic forgetting by reactivating the previous tasks, increasing the network's efficiency to generalize on noisy or alternative variants of the training dataset letting the network perform forward transfer learning. In this paper, we introduce Defense–VAE–Sleep model, which combines Defense-VAE [3] with the sleep mechanism [1] by utilizing the sleep phase in Defense–VAE to increase generalization performance by decreasing the impact that imperceptible input changes might have on the task output. During the sleep phase, Mirrored Spike-Timing Dependent Plasticity (mSTDP or mirrored STDP) used as sleep functions' learning rules [4] leads to an increased ability of neurons' to form logical communication in-between memories, and hence could reduce VAE loss function and subsequently resulting in less dispersed latent codes and increased output interpretability. The downstream CNN target classifiers in our proposed model are fed a clean version of the input contaminated images by removing the adversarial perturbations. First, adversarial images are generated based on an attack. Then, these adversarial images are fed into Defense–VAE–Sleep for reconstruction. Defense VAE–Sleep can generate images reconstructed from the underlying clean image by removing most adversarial perturbations. This paper introduces a framework as well as reports our initial findings on mimicking the biological sleep phase for defense against adversarial attacks on deep generative models like VAE. Contributions to our work include:

1. We report that our proposed model works well for defense against white–box and black–box attacks by reporting our model's performance on three different datasets (MNIST, Fashion–MNIST, and CelebA). For most results, after the sleep phase was applied in Defense–VAE, the reconstructed images resemble the underlying clean images more than the ones generated by Defense–VAE [3] (without sleep).
2. We demonstrate that using the Defense–VAE–Sleep algorithm leads to information-rich latent codes relative to the ones generated by Defense–VAE [3] and Defense–GAN [5].
3. We show that Defense–VAE–Sleep's architecture is more robust when compared to Defense–VAE [3] and Defense–GAN [5] which creates decision boundaries that more closely resemble the actual classes.
4. We demonstrate that Defense–VAE–Sleep algorithm is better suited for multi-label classification problems (by experiments on the CelebA dataset) compared to Defense–VAE [3] and Defense–GAN [5].

The rest of the paper is organized as follows: a review of literature providing background on adversarial attacks and defense mechanisms is provided in Sect. 2, followed

by our model's description in Sect. 3. Section 4 delineates our experimental protocol and results. Finally, the paper is concluded with remarks on the results in Sect. 5.

2 Related Work

Defense algorithms and adversarial attacks are widely researched machine learning fields, with many defense models developed last decade. The research focuses on the creation/design of robust models that perform well on training sets with perturbed adversarial augmentations.

The defense models such as [6] have improved the accuracy classification scores and even enhanced the accuracy of the clean image in some image datasets. A two-step defense model is proposed in [7], where the adversarial input is detected and then reformatted using divergence amongst clean and adversarial examples' manifolds. The use of various filters(Gaussian low–pass, median and averaging, etc., among others) is suggested in [8] as a mechanism to counter the effects of adversarial noise. [9] prepossessed images with JPEG compression, a standard, widely-used image encoding and compression method, where the impact of adversarial noise is reduced using the JPEG compression as a defense mechanism. [10] used a saturating network whose performance is robust in the presence of adversarial noise by modifying the loss function such that activations are in their saturating regime.

Saman–gouei et al. proposed a Defense–GAN model that used generating model "GAN" [5] for adversarial defense. Initially, the authors trained a GAN model with clean images, and therefore, the model learned clean images' distribution. Then, back-propagation was used to identify the optimal latent space of the clean image when provided with an adversarial image. Eventually, the GAN reconstructed an image through optimal latent space, which is expected to resemble the clean image. Xiang Li and Shihao Ji proposed Defense–VAE [3] removes the adversarial perturbations, which leads to reconstructed images that closely resemble the underlying clean image. Compared to Defense–GAN [5], Defense–VAE can directly identify an optimal latent code by using the forward–propagating algorithm an adversarial image through its encoder network; rendering the whole process to be much faster when compared with Defense–GAN. Subsequently, Defense–VAE's decoder network reconstructs the clean image using that latent code.

Recently, biologically inspired learning with sleep algorithms in DNNs has been mimicked in [1] to increase adversarial robustness and generalizations of DNNs by using a model which augments the backpropagation training phase with an STDP based unsupervised learning phase in the Spiking domain modeled after how the biological brain utilizes sleep to improve learning. The sleep algorithm improves the generalization accuracy for noise and blurs in the testing dataset, especially when training using clean images, and exploits using such attacks can be a severe threat; as mentioned before, security-sensitive applications of DNNs are vulnerable to adversarial attacks. To the best of our knowledge, Defense–VAE [3] is the state–of–the art for defense against adversarial attacks outperforming general DNNs' defense mechanisms [1] and Defense–GAN [5]. In this work, we suggest increasing Defense–VAEs' robustness by utilizing a biologically inspired sleep phase in VAE to both adversarial attacks and general image distortions, with high generalization performance.

3 Proposed Model

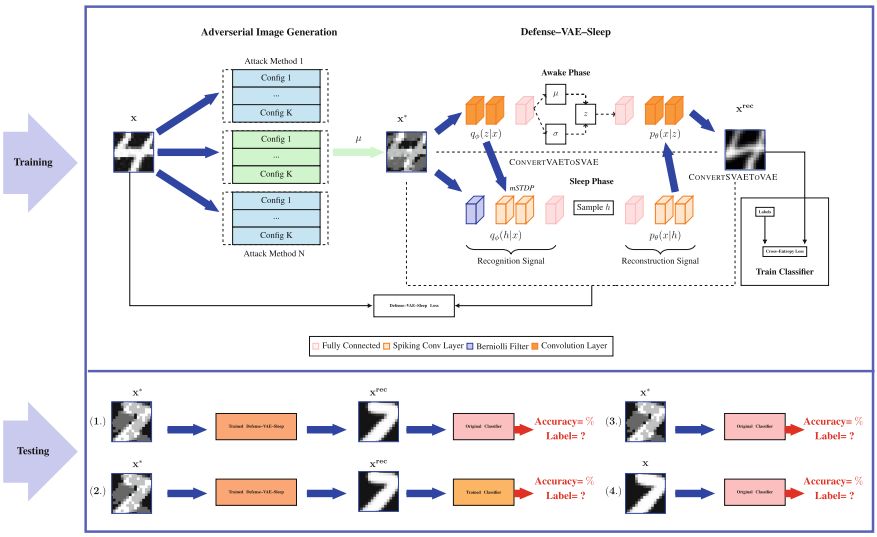

Fig. 1. Up left shows the training pipeline used for defense–VAE–sleep, and the classifier is shown on up right. The testing pipeline of defense–VAE–sleep is shown downwards.

3.1 Varitional Auto-Encoder (VAE)

VAE is a probabilistic generative architecture [11] consisting two independent cooperating networks: (a.) an encoder $q_\phi(z\|x)$ and (b.) a decoder $p_\theta(x\|z)$. These two networks are cooperating agents that perform space transformation were given an input (image in our case) x, encoder transforms it from feature space to what is referred to as a latent space to a latent variable z. The decoder's role in the network is the opposite. During training, VAE regularizes its encoding distribution such that the transformation of input from feature space to latent space captures necessary properties to facilitate the reverse transformation. Assuming parameters θ and ϕ represent the weight and biases of VAE, its loss function (lower bound on log-likelihood $\log p_\theta(x)$), also referred to as Evidence Lower BOund (ELBO) [11] can be calculated using Jensen's inequality as follows:

$$\mathcal{L}_{\theta,\phi;x} = \mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x|z)\big] - \mathcal{D}_{KL}(q_\phi(z|x)\|p_\theta(z)), \tag{1}$$

where,

$$\log p_\theta(x) = \int p_\theta(x|z)p(z)\frac{q_\phi(z|x)}{q_\phi(z|x)}dz \geqslant \mathcal{L}_{\theta,\phi;x}. \tag{2}$$

ELBO consists of two terms; (1.) the reconstruction term, which to maximizing the expected log-likelihood under the q_ϕ distribution, and (2.) Kullback-Leibler (KL) term,

which compares the learned distribution q_ϕ with prior distribution p_θ. When the approximate and true posterior probabilities are equal i.e. $q_\phi(z\|x) = p_\theta(z\|x)$, ELBO is minimized, where the bounds are determined by the KL divergence $q_\phi(z\|x)$ and $p_\theta(z\|x)$. When used with images, CNN architectures can be used to build VAE's encoder and decoder models for enhanced performance on the transformations, such that encoders can better capture distinct perceptual features (example, spatial correlation) of the input image [12]. However, the model's reconstruction process regarding fidelity and naturalness can be improved as the output of VAEs are generally blurry and unrealistic.

3.2 Spiking Varitional Auto-Encoder (SVAE)

SVAE [12] maps probabilistic mapping of encoder and decoder as Leaky Integrate–And–Fire (LIF) neurons based SNNs. Hidden spiking signals $X_{\mathcal{H}}$ (or latent spike trains) distributed according to a parameterized distribution are created by the encoder $q_\phi(X_{\mathcal{H}}\|X_{\mathcal{V}})$, given the visible spiking signal $X_{\mathcal{V}}$, while the decoder $p_\theta(X_{\mathcal{V}}\|X_{\mathcal{H}})$ will reconstruct the spiking signal $X_{\mathcal{V}}$ given the hidden spike signal $X_{\mathcal{H}}$. Usually Bernoulli or Poisson filters are used to encode input images to the spike trains. Learning SVAE depends on changing the synaptic strengths between neurons. During training SVAE, the encoder should try to learn the simpler distribution $q_\phi(X_{\mathcal{H}}\|X_{\mathcal{V}})$ such that it is as close as possible to the target distribution $p_\theta(X_{\mathcal{H}}\|X_{\mathcal{V}})$ based on KL divergence [5], which is defined as:

$$KL_{\left[q_\phi\left(X_{\mathcal{H}}|X_{\mathcal{V}}\right)\|p_\theta\left(X_{\mathcal{H}}|X_{\mathcal{V}}\right)\right]} = \int \mathcal{D}_{X_{\mathcal{H}}}\, q_\phi(X_{\mathcal{H}}|X_{\mathcal{V}}) \log \frac{q_\phi(X_{\mathcal{H}}|X_{\mathcal{V}})}{p_\theta(X_{\mathcal{H}}|X_{\mathcal{V}})}, \tag{3}$$

where $\mathcal{D}_{X_{\mathcal{H}}}$ is a measure of integration over latent spike trains. The distribution considered as "best fit" for the data is obtained by minimizing the KL divergence. Maximizing the marginal–log value leads to optimization of KL divergence, which can be formulated in terms of KL as:

$$\begin{aligned} &\log p_\theta(X_{\mathcal{V}}) = \log \int \mathcal{D}_{X_{\mathcal{H}}}\, p_\theta(X_{\mathcal{V}}, X_{\mathcal{H}}) = \int \mathcal{D}_{X_{\mathcal{H}}}\, q_\phi(X_{\mathcal{H}}\|X_{\mathcal{V}}) \log \frac{q_\phi(X_{\mathcal{H}}\|X_{\mathcal{V}})}{p_\theta(X_{\mathcal{H}}\|X_{\mathcal{V}})} + \\ &\int \mathcal{D}_{X_{\mathcal{H}}}\, q_\phi(X_{\mathcal{H}}\|X_{\mathcal{V}}) \log \frac{p_\theta(X_{\mathcal{V}}, X_{\mathcal{H}})}{q_\phi(X_{\mathcal{H}}\|X_{\mathcal{V}})} = KL_{\left[q_\phi(X_{\mathcal{H}}\|X_{\mathcal{V}})\|p_\theta(X_{\mathcal{H}}\|X_{\mathcal{V}})\right]} + \mathcal{L}_{(\theta,\phi;X_{\mathcal{V}})}, \end{aligned} \tag{4}$$

where $\mathcal{L}_{(\theta,\phi;X_{\mathcal{V}})}$ is the loss function (or ELBO) for the SVAE . The gradient of Eq. 4 $\nabla_{w_{ij}} \log p(X_{\mathcal{V}})$ of both the spike observed (or the visible spiking signal) $X_{\mathcal{V}}$ and the latent spike neurons $X_{\mathcal{H}}$ between two neurons i and j w.r.t to the synaptic efficacy w_{ij}, is given by;

$$\langle \nabla_{w_{ij}} \log p_\theta(X_{\mathcal{V}}) \rangle_{p_\theta(X_{\mathcal{H}}|X_{\mathcal{V}})} = \sum_{k\in(\mathcal{V}\cup\mathcal{H})} \int_0^T d\tau \frac{\partial \log \rho_k(\tau)}{\partial w_{ij}} [X_k(\tau) - \rho_k(\tau)], \tag{5}$$

where $\rho_k(\tau)$ is firing rate function of LIF neuron guided by mSTDP. The gradient $\nabla_{w_{ij}} \log p(X_{\mathcal{V}})$ can be calculated by updating the weight according to gradient descent $\Delta w_{ij} \propto \nabla_{w_{ij}} \log p(X_{\mathcal{V}})$ yields mSTDP learning rules.

3.3 VAE–Sleep

It has been hypothesized that the sleep mechanism is crucial to humans and animals' brain functions, including but not limited to how neurons communicate with each other by connecting the recently learned memories and the old learned memories; leading to improvement in memories, learning, increased attention and robustness against noise [13]. Limitations of traditional VAEs include the creation of non–interpretable latent codes when the inputs are noisy/disturbed and not observed during training, which is problematic and makes them ineffective for classifications [14]. This lack of generalization presents issues when VAEs are utilized in the real world. Compared to DNNs, SNNs simulate the behavior of natural neural networks more closely [15], and they have been proven to be more robust vis-á-vis deterioration to noisy inputs than their DNNs counterparts. Moreover, computing elements used in SNNs leverage the spike–domain processing that operates on spikes, making SNNs energy–efficient when compared to DNNs that consume energy sporadically [11]. The operation of VAE–Sleep [12], which consists of two distinct (sleep and awake) phases, can be summarized in the following points:

1. In the awake phase, train a regular VAE using the re–parameterization trick.
2. After training, convert the VAE network into a SVAE network (ConvertVAEToSVAE) by mapping the weights as in [16]. For the conversion, it is assumed that ReLU is used as an activation function and there are no biases.
3. This conversion facilitates the sleep phase where the input (pixel intensities) are converted to Bernoulli spike–trains $X_{\mathcal{V}}$ which simulate the mSTDP sleep from the network's visible to hidden layers
4. Reconvert the SVAE back to VAE (ConvertSVAEToVAE).

We refer the readers to [12] for more details about VAE–Sleep.

3.4 Mirrored STDP (mSTDP)

STDP [17] and anti–Hebbian STDP (aSTDP) [18] combined together guide the feed-0forward and feedback connections for mSTDP. Under mSTDP, for both visible and hidden pins, generated synaptic plasticity is identical for both feedback and feedforward connections (guided by a scaling factor), hence improving the performance of SVAE for input reconstruction. STDP considerably enhances synchrony in the feedforward network. Precise spike timing between pre and postsynaptic neurons induces plasticity in the standard STDP paradigm, where the strengthening(pre before post) and weakening(post before pre) of a synapse is determined by the spike timing. On the other hand, aSTDP is the opposite of Hebbian STDP, where (pre before post) leads to the weakened synapse. Maintaining this symmetry throughout the learning process requires the changes in feedforward feedback synaptic strengths to be constrained by the following rule: any weakening of the feedforward synaptic strength should result in an equivalent weakening of the synaptic feedback strength and vice versa. Feedforward and feedback synapses in the real neurons are two separate physical structures; a model mimicking them should explain aforementioned neurons will experience almost identical plasticity. In mSTDP, visible neuron spikes are substituted in place of presynaptic

neurons if STDP, and hidden neuron spikes are substituted in place of presynaptic neurons of aSTDP, and the postsynaptic neurons are substituted accordingly. The mSTDP rules are governed by the Eqs. 6 and 7.

$$\delta_{i\in\mathfrak{S}_{vis},j\in\mathfrak{S}_{hid}} = \begin{cases} \alpha\phi_r a_r^+ + \beta\phi_p a_p^- & t_j - t_i \leq 0, \\ \alpha\phi_r a_r^- + \beta\phi_p a_p^+ & t_j - t_i > 0, \end{cases} \tag{6}$$

$$\Delta w_{ij} = \delta_{ij} w_{ij}(1 - w_{ij}), \tag{7}$$

where $\mathfrak{S}_{vis}$, and $\mathfrak{S}_{hid}$ are the set of visible neuron spikes, and the set of hidden neuron spikes respectively; t_i and t_j are the time of the neuron spike i and j respectively; a_r^+, a_r^-, a_p^+, and a_p^- are the scale magnitude of weight change and signify the direction of weight change; α, β, ϕ_r, and ϕ_p are controlling factors; δ_{ij} denotes one of the mSTDP functions (also called learning window), which ensures the weights remain between the range [0,1] ensuring stability of the weight changes until convergence; and Δw_{ij} is the synaptic weight modification. For more information regarding mSTDP rules and input prepossessing strategy in mSTDP, we refer readers to [4]. During the learning process, k–Winner Take All (k–WTA) ensures that earlier fired neurons perform mSTDP and thwart other neurons' firing. k neurons with the quickest spike times are chosen, and those with the highest internal potentials are selected. Of the selected ones, a $r \times r$ inhibition window is used to indicate the winner where the winner neuron will be at the center of the window, which will be imposed on the feature maps to avert the selection.

3.5 Defense–VAE–Sleep

Algorithm 1. Defense–VAE–Sleep

procedure MAIN

$x^* =$ **adv-attack**(x, y, θ, ϵ)

Initialize (vae)

Train $vae(x^*, x)$

Minimize $\mathcal{L}_{\theta,\phi;x,x^*}$ in VAE

$\mathbb{E}_{q_\phi(z\|x^*)}\left[\log p_\theta(x\|z)\right] - \mathcal{D}_{KL}(q_\phi(z\|x^*)\|p_\theta(z))$

$svae, scales =$ **ConvertVAEtoSVAE**(vae)

$svae =$ **Sleep**$(svae, x^*, scales)$

Minimize $\mathcal{L}_{\theta,\phi;X_\mathcal{Y},X_\mathcal{X}^*}$ in SVAE

$\log p_\theta(X_\mathcal{Y}) - KL(\left[q_\phi(X_\mathcal{H}\|X_\mathcal{Y}^*)\|p_\theta(X_\mathcal{H}\|X_\mathcal{X})\right])$

$vae =$ **ConvertSVAEtoVAE**$(svae)$

Reconstruct x^{rec}

Minimize Binary Cross-Entropy

$C = -\frac{1}{n}\sum_{x^{rec}}(y \ln Q(x^{rec}) + (1-y)\ln(1 - Q(x^{rec})))$

end procedure

We propose a new defense model that uses our proposed Defense–VAE–Sleep algorithm and a target deep learning classifier trained in an End–to–End (E2E) fashion within random weights initialization by taking into consideration the loss function of the Defense–VAE–Sleep algorithm as well as the classifer. Defense–VAE–Sleep can create the proper latent codes to correctly reconstruct adversarial examples (denoise examples) classified to their classes. The pseudo–code of our Defense–VAE–Sleep algorithm is shown in Algorithm 1. Given a clean image x_m, we use different adverserial attack methods to generate multiple adverserial images x^*_{mk}, resulting in a many–to–one mapping between adverserail samples and their clean counterpart rendering Defense–VAE–Sleep to serve as a robust yet generic defense algorithm for different types of attacks. After that, we apply Defense–VAE–Sleep that consists of two phases; the awake and sleep phase (see Fig. 1 (left)). In the awake phase, we initialize VAE's parameters, and then train VAE to update the weights based on VAE's loss function using reparameterization trick [19]. Then, we apply CONVERTVAETOSVAE to transfer the weights from VAE to SVAE. In the sleep phase, we train the SVAE using mSTDP wherein the loss function is optimized concerning parameters of encoder (ϕ) and decoder (θ) . The mSTDP lets us backpropagate based on ELBO. After that, CONVERTSVAETOVAE is applied to transfer the weights from SVAE to VAE. The encoder and decoder in Defense–VAE–Sleep, are defined as follows:

$$z \sim Enc(x^*) = q_\phi(z|x^*), x \sim Dec(z) = p_\theta(x|z), \quad \text{(awake-phase)}, \tag{8}$$

$$X_{\mathcal{H}} \sim Enc(X^*_{\mathcal{V}}) = q_\phi(X_{\mathcal{H}}|X^*_{\mathcal{V}}), X_{\mathcal{V}} \sim Dec(X_{\mathcal{H}}) = p_\theta(X_{\mathcal{V}}|X_{\mathcal{H}}), \quad \text{(sleep-phase)}, \tag{9}$$

Finally, the reconstructed image x^{rec} from Defense–VAE–Sleep is used to train a downstream target classifier Q (see Fig. 1 (right)) by minimizing the cross–entropy loss calculated between target label and prediction labels: $y\ Q(x^{rec})$ respectively [20]. The loss function for an E2E training pipeline(Defense–VAE–Sleep and the target classifier) is given as follows:

$$\mathcal{L}_{E2E} = \mathcal{L}_{VAE} + \mathcal{L}_{SVAE} + \mathcal{L}_{Cross-Entropy} \tag{10}$$

In the test stage (see Fig. 1 (down)), we investigate the possibilities of threats/remedies beyond classification tasks by (1.) Testing the classification accuracy of an adversarial perturbation elimination framework to eliminate the adversarial examples' perturbation before feeding it into the original target classifier trained with clean images before the attack (a defense model), (2.) Testing the classification accuracy of a defense model E2E learning (a defense model E2E), (3.) Testing the classification accuracy of an adversarial image on the original target classifier (No Defense), (4.) Testing the classification accuracy of a clean image on the original target classifier (No Attack). In our model, we consider combining ensemble methods with our defense mechanism [21]; the following ensemble methods are used;

1. We train each target classifier (C1, C2, C3, and C4) (see Table 1) multiple times, initialized with different random initial weights, which leads to quite diverse classifiers with different final weights. Once an ensemble of classifiers is trained, it predicts by allowing each classifier in the ensemble to vote for a label, and then the predicted value is selected to be the label with the maximum or average of the softmax probability distribution of the output from each classifier.
2. We train multiple adversarial attacks (white and black box attacks) with Defense–VAE–Sleep to obtain a set of adversarial training samples to increase the model's capability and, therefore, formulate a defense algorithm that is generic and has improved robustness for a spectrum of perturbations.
3. We train our model using Gaussian distortions samples besides multiple attack algorithms to make each target classifier more robust against noise/perturbation.

4 Experiments

MNIST and Fashion–MNIST were used to evaluate the performance of our defense model, and additionally, CelebA was also used to classify celebrity gender (male, female) based on face image.

4.1 Network Architectures and Training Parameters

Table 1. Description of the substitute models and classifiers for white–box and black-box attacks.

C1	C2	C3	C4
Dropout.0.2.	3×3 conv. 128 ReLU. stride 1. padding 1.	FC1.200.	FC1.200
8×8 conv. 64 ReLU. stride 2. padding 5.	5×5 conv. 128 ReLU. stride 2. padding 0	ReLU	ReLU
6×6 conv. 128 ReLU. stride 2. padding 0.	Dropout.0.25.	Dropout.0.5.	FC2.200
5×5 conv. 128 ReLU. stride 1. padding 0.	FC1.128.	FC2.200.	ReLU
Dropout.0.5	ReLU	ReLU	FC3.10 + Softmax
FC1.10. + Softmax	Dropout.0.5.	Dropout. 0.25.	
	FC2.10. + Softmax	FC3.10. + Softmax	

The details of VAE and SVAE architectures used to build our Defense–VAE–Sleep are given in Table 2, and Table 3 respectively. In the awake phase, the Convolutional VAE model is used. The encoder's outputs (the mean and log standard deviation) are parameterized by a factorized Gaussian, while the decoder's outputs are parameterized by Bernoulli distributions over the pixels. For the sleep phase, our SVAE model is constructed by mapping the convolution layers of the VAE model in awake phase into spiking convolution layers with LIF neurons. To apply mSTDP to a SVAE layer, the controlling factors are $\phi_r = 1$, $\phi_p = 0$, $\alpha = 1$, and $\beta = 0$. The CNN target classifiers (C1, C2, and C3) and the substitute models [5] of black-box CNN models (C1, C4) referred in this section are depicted in Table 1.

Table 2. Details of the encoder and decoder architectures of defense–VAE–sleep in awake phase used in the experiments.

Encoder	Decoder
Input $64 \times 64 \times 1$ image	FC. 128. ReLU
5×5 conv. 64 ReLU. stride 1. padding 2. +BN	4×4 deconv. 256 ReLU. stride 2. padding 1. +BN
4×4 conv. 64 ReLU. stride 2. padding 3. +BN	4×4 deconv. 128 ReLU. stride 2. padding 1. +BN
4×4 conv. 128 ReLU. stride 2. padding 1. +BN	4×4 deconv. 64 ReLU. stride 2. padding 3. +BN
4×4 conv. 256 ReLU. stride 2. padding 1. +BN	5×5 deconv. 64 ReLU. stride 1. padding 2. +BN
FC1. 4096., FC2. 4096.	

Table 3. Values of the spiking-layer of defense–VAE–sleep in sleep phase parameters are used in the experiments. Threshold: neuronal firing threshold for each layer of neurons, α_r^+: upper bounds range of weights, α_r^-: lower bounds range of weights, α_p^+: upper bounds range of weights, α_p^-: lower bounds range of weights, k: The number of winners used in k–WTA mechanism [22], and r: The radius of lateral inhibition used in k–WTA mechanism [22].

Layer	Number of feature maps	Input window	Stride	Padding	Threshold	α_r^+	α_r^-	α_p^+	α_p^-	k	r
S1	64	5×5	1	2	36	0.004	-0.003	0	0	5	3
S2	64	4×4	2	3	23	0.004	-0.003	0	0	5	2
S3	128	4×4	2	1	23	0.004	-0.003	0	0	8	1
S4	256	4×4	2	1	36	0.004	-0.003	0	0	1	0

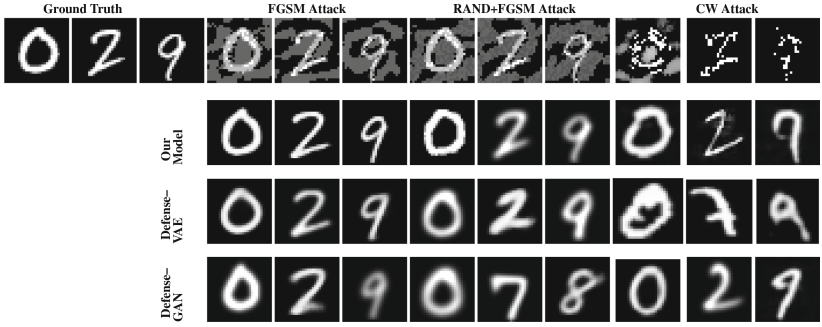

Fig. 2. Randomly chosen reconstructions of the latent variables using defense–VAE, defense–GAN, and defense–VAE–sleep (our model) on input perturbed with different white-box attacks on the MNIST dataset. The white–box attacks used are FGSM, RAND+FGSM, and CW.

4.2 Results of White-box Attacks

Complete knowledge of the network architecture, training data, and saved weights are assumed to be known for white-box attacks where the attacker can leverage any of these to perform an adversarial attack.

They can range from having full information, such as a gradient-based attack, to score-based attacks that only utilize predicted scores of the model [5]. Experiments for different white-box attacks: FGSM [23], RAND+FGSM [21], and CW [24] were

performed. Table 4 depicts the performance of different classifier models (C1, C2, and C3) across the three datasets' different attack and defense strategies. Nine adversarial samples: 3 different configurations × 3 attack scenarios were generated. For a performance comparison between our model vs. Defense–VAE [3] and Defense–GAN [5], we included the results of the models (under the same configurations used to evaluate our model) as well. As shown, our model achieves superior performance over the other two models in most attacks and recovers almost all the losses inaccuracy as a result of the adversarial attacks. We show examples of adversaries created by white-box attacks in Fig. 2.

Table 4. Classification accuracies of various defense mechanisms under white-box attacks for different datasets: MNIST, Fashion–MNIST, and CelebA.

Dataset		MNIST						Fashion–MNIST						CelebA					
Attack	Classifier model	No attack	No defense	Our model	Our model (E2E)	Defense–VAE	Defense–GAN	No attack	No defense	Our model	Our model (E2E)	Defense–VAE	Defense–GAN	No attack	No defense	Our model	Our model (E2E)	Defense–VAE	Defense–GAN
FGSM	C1	96.20	2.20	96.12	**97.13**	95.92	95.60	74.70	10.20	71.09	**79.03**	70.88	62.90	94.68	9.95	91.13	**93.60**	90.05	90.10
$\epsilon = 0.3$	C2	99.60	33.10	99.01	**99.33**	98.41	98.90	93.30	13.90	90.61	**92.02**	85.80	89.60	94.59	4.60	91.83	**93.01**	92.47	91.45
	C3	99.20	3.80	98.00	**98.60**	97.56	98.00	89.20	8.20	87.81	**89.03**	85.36	87.50	94.76	6.05	92.13	**93.13**	90.05	92.05
RAND+FGSM	C1	96.20	1.70	96.00	**97.24**	95.83	94.40	74.70	13.10	72.55	**79.02**	71.12	66.10	94.68	17.85	91.22	**93.41**	90.55	89.20
$\epsilon = 0.3$	C2	99.60	10.30	98.70	**99.40**	98.33	98.50	93.30	10.50	90.28	**91.07**	86.42	89.30	94.59	4.70	92.08	**94.40**	91.70	91.80
$\alpha = 0.05$	C3	99.20	5.00	97.92	**98.45**	97.81	98.00	89.20	9.10	88.57	**90.01**	85.77	86.20	94.76	6.65	92.27	**94.01**	91.42	90.33
CW	C1	96.20	3.20	92.44	**95.50**	87.66	91.60	74.70	17.20	70.11	**74.80**	67.43	65.60	94.68	5.75	92.11	**93.60**	90.65	73.16
$l_2$norm	C2	99.60	12.6	95.67	97.42	94.46	**98.90**	93.30	6.30	80.22	88.89	78.64	**89.60**	94.59	4.35	93.53	**94.53**	93.28	79.15
	C3	99.20	3.20	95.74	97.68	83.42	**98.30**	89.20	9.00	82.61	87.40	64.38	**87.50**	94.76	6.60	92.02	**94.20**	91.15	76.11
Average		98.33	8.34	96.62	**97.86**	94.38	96.91	85.73	10.83	81.54	**85.70**	77.31	80.48	94.68	7.39	92.04	**93.77**	91.26	85.93

4.3 Results of Black-box Attacks

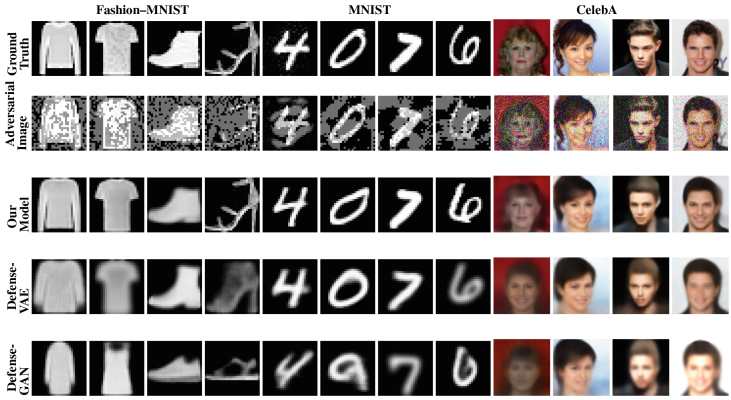

Fig. 3. Randomly chosen reconstructions of the latent variables using defense–VAE for different dataset: MNIST, fashion–MNIST and CelebA when tested under FSGM black–box attacks.

Black–box attacks are usually carried out under the scenarios where the target model is treated as a complete black box, and the attacker does not have access to the dataset the model was trained under where the model can be queried for collecting information on certain input/output pairs [25].

A small dataset is created by augmenting the samples labeled by the original target model. It is used to train a substitute model with the hopes that the substitute can be used as a surrogate for the actual target model. An adversarial example can be created by applying an attack on the generated substitute model. We used our Defense–VAE–Sleep model trained under white-box attacks to defend against black-box attacks. Only FGSM attack computed based on a substitute model [25] is considered, with the results of MNIST, and 5 reports the results of the datasets used for our experiments. The performance of defenses on the Fashion–MNIST dataset is noticeably lower than on MNIST and CelebA. A qualititive analysis of example reconstruction (see Fig. 3) demonstrates that our model (Defense–VAE–Sleep) is more robust when compared with Defense–VAE for black–box attacks. Assiging initial random weights performed an E2E training for Defense–VAE–Sleep and a target classifier to improve the target model's accuracy. For some cases depicted in 5, it even improves over the original target classifier. In particular, according to the results in Table 5, Defense–VAE–Sleep trained in an E2E configuration outperforms Defense–VAE–Sleep with each network trained independently (without take classifier's loss in consideration during the training).

Table 5. Classification accuracies of various defense mechanisms under black-box attacks for different datasets: MNIST, Fashion–MNIST, and CelebA (Only FGSM black–box attack was used).

Dataset	MNIST						Fahion-MNIST						CelebA					
Classifier/Substitute	No attack	No defense	Our model	Our model (E2E)	Defense–VAE	Defense–GAN	No attack	No defense	Our model	Our model (E2E)	Defense–VAE	Defense–GAN	No attack	No defense	Our model	Our model (E2E)	Defense–VAE	Defense–GAN
C1/C1	96.18	28.16	96.00	**96.90**	95.89	91.05	74.70	40.17	74.13	**84.71**	73.66	55.30	93.69	20.02	89.25	**90.17**	88.02	70.02
C1/C4	96.18	21.28	96.10	**97.40**	96.16	88.92	74.70	31.23	69.01	**80.30**	69.29	41.87	93.69	18.20	86.13	**88.67**	85.67	74.90
C2/C1	99.59	66.48	98.09	**99.30**	97.91	93.22	93.34	26.35	86.01	**90.40**	83.64	60.79	95.62	22.01	92.24	**94.12**	91.01	60.01
C2/C4	99.59	80.50	98.45	**99.40**	98.30	91.82	93.34	20.66	80.19	**87.01**	76.27	46.25	95.62	10.50	88.80	**90.34**	87.03	68.54
C3/C1	99.20	46.41	97.73	**97.92**	97.68	93.23	89.23	45.41	81.33	**85.02**	80.31	58.53	94.89	19.32	90.13	**92.37**	89.32	74.58
C3/C4	99.20	39.31	97.41	**97.81**	97.72	91.55	89.23	25.43	81.44	**85.80**	70.66	47.30	94.89	7.05	87.14	**89.05**	86.14	60.20
Average	98.32	47.02	97.30	**98.12**	97.23	91.63	79.54	31.54	78.69	**85.54**	75.64	51.67	94.73	16.18	88.95	**90.79**	87.87	68.04

4.4 How Come is Defense–VAE–Sleep Mighty and Efficient?

Why does the sleep phase [1] have such an enormous leap in increasing the robustness of Defense-VAE [3] to different attacks? It has been shown that the sleep phase based on SVAE [12] tends to promote an increase in more robust weights while pruning weaker weights, thus increasing the width of the weight's distribution. This results in consolidating strong relations at the cost of diminishing weak connections between neurons. Strengthening the strong relations between neurons also makes our defense model robust and noise invariant, producing better generalization, and maintaining a high baseline accuracy. Moreover, VAE–Sleep algorithm [12] uses "weight normalization" [16] as a technique of adjusting the synaptic weights to acquire lossless transformation from VAE to SVAE. It considers each LIF neuron as the activation neuron function due to its functional resemblance to ReLU, without any leak or refractory period. The accuracy reported for training in this way is high, compared to traditional VAE, even for large-scale networks. The results demonstrate that our model is superior compared to Defense–VAE and Defense–GAN for the benchmark datasets(MNIST, Fashion–MNIST, and CelebA). CelebA dataset [26] is a face attributes dataset, each one with 40 attribute annotations. To prove the efficacy of Defense–VAE–Sleep over

the other two models tested, a multi-label classification was applied to distinguish more than one attributes such as “Attractive”, “Gray hair”, “bald”, “wearing lipstick”, etc. Table 6 shows that the images reconstructed by Defense–VAE–Sleep have a better classification accuracy (on the aforementioned multi-label classification), proving our hypothesis that the use of the sleep phase for Defense–VAE results in improved classification accuracy and increased robustness concerning the number of latent dimensions.

Table 6. Classification accuracies for different number of attributes on the CelebA dataset for our model, defense–VAE and defense–GAN under FGSM based black–box attacks

# Attributes	2 Attributes						4 Attributes						6 Attributes					
Classifier/Substitute	No attack	No defense	Our model	Our model (E2E)	Defense–VAE	Defense–GAN	No attack	No defense	Our mode	Our model (E2E)	Defense–VAE	Defense–GAN	No attack	No defense	Our model	Our model (E2E)	Defense–VAE	Defense–GAN
C1/C1	92.25	18.14	90.9	**91.01**	90.01	65.03	89.33	15.34	84.67	**89.40**	76.24	63.23	85.14	13.77	80.31	**84.91**	70.23	58.40
C1/C4	92.25	16.11	85.04	**89.80**	78.91	55.02	89.33	14.27	82.15	**87.77**	74.44	52.11	85.14	12.55	82.11	**85.04**	69.25	48.56
C2/C1	94.61	20.31	91.08	**94.10**	83.11	62.11	91.13	18.23	86.01	**91.05**	78.11	55.41	88.45	15.44	82.70	**88.00**	70.06	50.04
C2/C4	94.61	8.01	89.22	**92.11**	75.33	69.34	91.13	6.04	85.17	**91.03**	69.90	55.22	88.45	5.77	83.12	**88.09**	65.05	48.19
C3/C1	93.22	17.10	90.00	**93.33**	82.01	69.41	90.15	15.23	86.45	**90.04**	77.16	58.44	85.35	14.22	81.45	**85.01**	69.01	52.41
C3/C4	93.22	5.71	87.10	**90.00**	78.02	55.54	90.15	4.91	83.65	**88.24**	74.66	55.34	85.35	3.88	80.15	**84.90**	68.03	50.43
Average	93.36	14.23	88.89	**91.725**	81.23	62.74	90.20	12.34	84.68	**89.59**	75.09	56.625	86.31	10.99	81.64	**86.00**	68.61	51.34

5 Conclusion and Future Research Directions

Defense strategy inspired by the biological processes (sleep algorithm) presented in the paper yield increased robustness of classification models against adversarial attacks and distortion. Our experiments on standard computer vision datasets demonstrate that Defense–VAE–Sleep provides a better defense for adversarial attacks when compared with other models (Defense–GAN and Defense–VAE). We hypothesized that more realistic feature representations are created because of the sleep phase in Defense–VAE–Sleep, hence leading to more natural decision boundaries that closely resemble right classes, thus increasing the robustness of the network. Additionally, a comprehensive analysis of the performance of Defense–VAE–Sleep was presented using qualitative comparisons for different adversarial attacks. Although being a robust mechanism of defense against adversarial attacks, for some types of attacks, classification accuracy deteriorates with increased robustness. Future work includes addressing the deficiencies in our model to be robust for different types of attacks by enhancing spike encoding of the spiking convolution layer.

Acknowledgement. This work is supported by Google Cloud credits for academic research. We thank the Google cloud platform for giving us access to computing power that will make the next big thing possible.

References

1. Tadros, T., Krishnan, G., Ramyaa, R., Bazhenov, M.: Biologically inspired sleep algorithm for increased generalization and adversarial robustness in deep neural networks. In: International Conference on Learning Representations (2019)
2. Han, X., et al.: Adversarial attacks and defenses in images, graphs and text: a review. Int. J. Autom. Comput. **17**, 151–178 (2020)

3. Li, X., Ji, S.: Defense-VAE: a fast and accurate defense against adversarial attacks. In: Cellier, P., Driessens, K. (eds.) ECML PKDD 2019. CCIS, vol. 1168, pp. 191–207. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43887-6_15
4. Kendra S Burbank. Mirrored stdp implements autoencoder learning in a network of spiking neurons. PLoS Comput. Biol. **11**(12), e1004566 (2015)
5. Samangouei, P., Kabkab, M., Chellappa, R.: Defense-GAN: protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605 (2018)
6. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
7. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 135–147 (2017)
8. Osadchy, M., Hernandez-Castro, J., Gibson, S.J., Dunkelman, O., Pérez-Cabo, D.: No bot expects the deepcaptcha! introducing immutable adversarial examples with applications to captcha. IACR Cryptol. ePrint Arch., vol. 2016, p. 336 (2016)
9. Gu, S., Rigazio, L.: Towards deep neural network architectures robust to adversarial examples. arXiv preprint arXiv:1412.5068 (2014)
10. Nayebi, A., Ganguli, S.: Biologically inspired protection of deep networks from adversarial attacks. arXiv preprint arXiv:1703.09202 (2017)
11. Bagheri, A.: Probabilistic spiking neural networks: Supervised, unsupervised and adversarial trainings (2019)
12. Talafha, S., Rekabdar, B., Mousas, C., Ekenna, C.: Biologically inspired sleep algorithm for variational auto-encoders. In: Bebis, G., et al. (eds.) ISVC 2020. LNCS, vol. 12509, pp. 54–67. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64556-4_5
13. Walker, M.P., Stickgold, R.: Sleep-dependent learning and memory consolidation. Neuron **44**(1), 121–133 (2004)
14. Roy, S.S., Ahmed, M., Akhand, M.A.H.: Noisy image classification using hybrid deep learning methods. J. Inf. Commun. Technol. **17**(2), 233–269 (2018)
15. Ankit, A., Sengupta, A., Panda, P., Roy, K.: Resparc: a reconfigurable and energy-efficient architecture with memristive crossbars for deep spiking neural networks. In: Proceedings of the 54th Annual Design Automation Conference 2017, pp. 1–6 (2017)
16. Rueckauer, B., Liu, S.C.: Conversion of analog to spiking neural networks using sparse temporal coding. In: 2018 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5. IEEE (2018)
17. Caporale, N., Dan, Y.: Spike timing-dependent plasticity: a hebbian learning rule. Annu. Rev. Neurosci. **31**, 25–46 (2008)
18. Koch, G., Ponzo, V., Di Lorenzo, F., Caltagirone, C., Veniero, D.: Hebbian and anti-hebbian spike-timing-dependent plasticity of human cortico-cortical connections. J. Neurosci. **33**(23), 9725–9733 (2013)
19. Kim, H., Mnih, A.: Disentangling by factorising. arXiv preprint arXiv:1802.05983 (2018)
20. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. Adv. Neural Inf. Process. Syst. **31**, 8778–8788 (2018)
21. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: attacks and defenses. arXiv preprint arXiv:1705.07204 (2017)
22. Maass, W.: Neural computation with winner-take-all as the only nonlinear operation. Adv. Neural Inf. Process. Syst. **12**, 293–299 (2000)
23. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
24. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy (SP), pp. 39–57. IEEE (2017)

25. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506–519 (2017)
26. Liu, Z., Luo, P., Wang, X., Tang, X.: Large-scale celebfaces attributes (celeba) dataset. Retrieved August **15**(2018), 11 (2018)