EPiC Series in Computing

Volume 111, 2026, Pages 95–110

Proceedings of The 21st Asia Joint Conference on Information Security

Adversarial Defense Using Latent Anomaly Detection and Image Purification

Kyo Matsumura[1], Yuya Tarutani[1], and Atsuko Miyaji[1]

The University of Osaka, Osaka, Japan
kyo.matsumura@cy2sec.comm.eng.osaka-u.ac.jp, tarutani@comm.eng.osaka-u.ac.jp, miyaji@comm.eng.osaka-u.ac.jp

Abstract

Recently, Convolutional Neural Networks (CNNs) have demonstrated high performance in image recognition tasks, but they remain vulnerable to adversarial examples, which intentionally cause misclassification through imperceptible perturbations. Conventional defense methods using Autoencoders (AEs) rely solely on anomaly detection based on reconstruction error or image purification. These approaches are often insufficient against white-box attacks, as attackers can bypass either of the defenses, making robust protection difficult. In this study, we propose "DSVDD-AE," a two-stage defense method that integrates anomaly detection based on feature distance in the latent space using Deep SVDD (DSVDD) with image purification via AE. This two-stage approach enables the detection of attacks that cannot be effectively purified, providing robust defense even under white-box environments. In our evaluation, we constructed two types of models using Contractive AE (CAE) and Variational AE (VAE) and verified their defense performance against PGD attacks on the CIFAR-10 dataset. Under a white-box attack environment, the proposed method (DSVDD-VAE) achieved a defense success rate of 30.28%, marking an improvement of approximately 30 percentage points compared to the existing Defense-VAE.

1 Introduction

The advancement of deep learning in recent years has significantly impacted our society. In particular, the progress in image recognition, natural language processing, and speech processing has been remarkable. Image recognition technologies driven by Convolutional Neural Networks (CNNs) are increasingly being deployed in tasks requiring high reliability, such as facial recognition systems, medical image diagnosis, and autonomous driving.

However, alongside the widespread adoption of deep learning models like CNNs, critical security vulnerabilities have been identified. Notably, adversarial examples involve adding imperceptible perturbations to input images, imperceptible to humans, to intentionally cause the model to misclassify. For instance, an attack can deceive a model into classifying a stop sign—which appears normal to a human—as a speed limit sign. Consequently, adversarial examples pose a severe risk of causing catastrophic incidents, such as accidents involving autonomous vehicles.

T.-T. Tsai, H.K. Kim, Y. Wang and A. Yamada (eds.), AsiaJCIS 2026 (EPiC Series in Computing, vol. 111), pp. 95–110

Representative methods for generating adversarial examples include the Fast Gradient Sign Method (FGSM) [3] proposed by Goodfellow et al., and Projected Gradient Descent (PGD) [9] proposed by Madry et al. These attacks have become increasingly powerful and stealthy over the years.

Defense mechanisms against adversarial examples are broadly categorized into "model robustification" and "attack detection and purification." A representative approach of the former is adversarial training, which includes adversarial examples in the training data. While robust against known attacks, this method is vulnerable to unknown attacks and incurs high computational costs due to the training data increasing proportionally with the variety of targeted attacks. Conversely, the latter approach defends against attacks by preprocessing input data. Specific techniques include methods utilizing Autoencoders (AEs) [5] and diffusion models [6]. Diffusion-based methods can eliminate adversarial perturbations during the process of converting all input data to noise, demonstrating high purification capabilities. However, the iterative calculations required to restore the original image from noise result in prohibitively high computational costs. AE-based methods, while less powerful in purification capability than diffusion models, offer significantly lower computational costs. Since low computational costs allow for deployment across various resource-constrained devices, this study focuses on researching an AE-based defense method.

Existing defense methods based on image purification via AE include Defense-VAE [8], proposed by Li et al., which performs adversarial training on a Variational Autoencoder (VAE), and Adversarial Memory VAE (AdMVAE) [13], proposed by Yin et al., which randomly masks parts of the input image and reconstructs the entirety, including the missing parts, based on normal features for purification. Furthermore, as an anomaly detection method using AE, approaches relying on reconstruction error—the difference between input and output—are widely known. Because an AE trained exclusively on normal data struggles to accurately reconstruct attack data, resulting in a large reconstruction error, these methods identify an input as anomalous when the reconstruction error exceeds a predefined threshold.

However, because these defense mechanisms rely on the reconstruction process, they face the challenge of being unable to completely defend against attacks calculated with consideration of the reconstruction process, or attacks that cause misclassification while keeping the reconstruction error small, particularly in white-box environments where the attacker has full knowledge of the classifier and the defense mechanism. MagNet [10], which proposed anomaly detection based on reconstruction error, also pointed out its vulnerability to white-box attacks.

In this study, we position the white-box environment as the strongest attack model. Since existing AE-based defense methods depend on the reconstruction process, they risk being bypassed when an attacker incorporates this process into their optimization. Therefore, to achieve detection that does not rely solely on reconstruction error, we introduce anomaly detection based on the feature distribution in the latent space. While adversarial examples can successfully maintain a small reconstruction error, they tend to significantly deviate from the original data distribution within the latent space. As a latent space anomaly detection method, Deep Support Vector Data Description (DSVDD) [12], proposed by Ruff et al., learns a minimum hypersphere enclosing normal data within the feature space, and has proven effective for high-dimensional data with complex distributions.

In this study, we propose "DSVDD-AE." The proposed method is a two-stage defense mechanism that conducts anomaly detection based on feature distance in the latent space without relying on reconstruction error, followed by purification. This allows it to defend against attacks that cannot be purified by relying on anomaly detection. This approach applies "DSVDD-CAE," proposed by Aktar et al. [1] in the field of IoT security, to image recognition tasks. In

our experiments, we implemented two models using different AEs: the Contractive AE (CAE) [11], a deterministic model, and the VAE [7], a probabilistic generative model. We evaluated their defense performance against ResNet-18 using the CIFAR-10 dataset. The evaluation utilized three metrics: the detection rate of DSVDD, the classification accuracy of ResNet-18, and the defense success rate of DSVDD-AE. The defense success rate is defined as the proportion of adversarial examples that were either "successfully detected as anomalous" or "correctly classified by the classifier after AE purification despite passing the detection."

The core novelty of this research lies in extending the DSVDD-CAE framework from the IoT domain to multi-class image recognition tasks, constructing an integrated two-stage defense structure combining "detection via feature distance in the latent space" and "purification via AE reconstruction." Assuming attacks in a white-box environment, this method forces conflicting tradeoffs between evading anomaly detection and inducing misclassification, thereby realizing robust defense against attacks that could not be defended by image purification alone.

Table 1: Comparison of Defense Performance Across Baseline Models

Method	Normal Data	Gray-box	White-box
Defense-VAE	67.47%	65.92%	0.71%
AdMVAE	86.33%	76.25%	37.66%
DSVDD-VAE	**79.47%**	**79.34%**	**30.28%**

The primary contributions of this research are as follows:

1. **Demonstration of the effectiveness of DSVDD-AE in image recognition**: Against gray-box attacks, DSVDD-VAE achieved a defense success rate of 79.34%, demonstrating performance approximately 14 percentage points higher than the existing Defense-VAE, and comparable to AdMVAE. Against White-box attacks, DSVDD-VAE achieved a 30.28% defense success rate, an improvement of about 30 percentage points compared to Defense-VAE's 0.71%, though it was approximately 7 percentage points lower than AdMVAE.
2. **Confirmation of tradeoffs in adaptive attacks**: Against "adaptive white-box attacks," where the attacker also aims to evade anomaly detection, we confirmed a tradeoff where the attempt to evade detection weakens the perturbation, resulting in improved classification accuracy.

1.1 Structure of this Paper

The remainder of this paper is organized as follows. Section 2 provides preliminaries on adversarial attack methods and AEs. Section 3 describes the foundational methods of our proposal and existing defense methods. Section 4 details the proposed method and its training process. Section 5 presents the experimental setup and results. Section 6 discusses the performance of the proposed method based on the experimental results. Finally, Section 7 concludes the study.

2 Preliminaries

2.1 Adversarial Examples and Attack Methods

Adversarial examples are inputs to which microscopic, human-imperceptible perturbations have been added to intentionally cause misclassification by the model. The primary attack methods relevant to this study are detailed below.

Fast Gradient Sign Method (FGSM) Goodfellow et al. [3] proposed FGSM as a method for generating adversarial examples. FGSM adds perturbation only once in the direction of the sign of the gradient. The algorithm is as follows:

$$x_{adv} = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(\theta, x, y))$$

where sign$(\cdot)$ is the sign function, and ϵ represents the magnitude of the perturbation.

Projected Gradient Descent (PGD) Madry et al. [9] proposed PGD. PGD adds random perturbation within the vicinity of the original sample as an initial input and iterates multiple times within a constrained range. The algorithm is as follows:

$$x_{t+1} = \mathcal{P}\left(x_t + \alpha_{pgd} \cdot \text{sign}(\nabla_x \mathcal{L}(\theta, x_t, y))\right)$$

where $\mathcal{P}(\cdot)$ is a projection function into the constrained range, and α_{pgd} is the step size.

2.2 Autoencoder (AE)

An Autoencoder [5] is a neural network that learns to compress input data into a low-dimensional latent space and reconstruct the original input data from that latent representation. It consists of an encoder, a latent space, and a decoder. The encoder f transforms the input $x \in \mathbb{R}^d$ into a latent variable $z \in \mathbb{R}^k$, and the decoder g reconstructs $\hat{x}$ from z, where d is the dimensionality of the input data and k is the dimensionality of the latent space. The loss function is defined using Mean Squared Error (MSE) as follows:

$$\mathcal{L}_{AE} = \frac{1}{n} \sum_{i=1}^{n} ||x_i - g(f(x_i))||^2$$

Contractive Autoencoder (CAE) Contractive Autoencoder (CAE) [11] adds a regularization term to the standard AE to penalize the rate of change of the encoder's output with respect to its input (the Frobenius norm of the Jacobian matrix). This stabilizes the latent space against minor variations in the input.

Jacobian Matrix $J_f(\mathbf{x})$. This is a $k \times d$ matrix containing the first-order partial derivatives of the encoder's output $f(\mathbf{x}) \in \mathbb{R}^k$ with respect to the input $\mathbf{x} \in \mathbb{R}^d$.

$$J_f(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_k}{\partial x_1} & \cdots & \frac{\partial f_k}{\partial x_d} \end{pmatrix}$$

Each element of this matrix represents the degree of influence that a minute change in the input has on each dimension of the latent space.

Frobenius Norm $\|\cdot\|_F$. This is the square root of the sum of the squares of each element in the matrix, used as a scalar metric to evaluate the overall magnitude of the matrix.

$$\|J_f(\mathbf{x})\|_F^2 = \sum_{i=1}^{k} \sum_{j=1}^{d} \left(\frac{\partial f_i(\mathbf{x})}{\partial x_j}\right)^2$$

The loss function of the CAE is defined by adding the Frobenius norm of the Jacobian matrix to the AE loss function:

$$\mathcal{L}_{CAE} = \mathcal{L}_{AE} + \lambda \|J_f(\mathbf{x})\|_F^2 \tag{1}$$

Variational Autoencoder (VAE) Variational Autoencoder (VAE) [7] is a generative model that treats the latent variables as probability distributions. Training involves minimizing the following loss function, which is equivalent to maximizing the Evidence Lower Bound (ELBO) of the marginal log-likelihood:

$$\mathcal{L}_{VAE} = -\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] + D_{KL}(q_\phi(z|x)||p(z))$$

The first term is the reconstruction error, and the second term is the Kullback-Leibler (KL) divergence, which acts as a regularization term measuring the proximity between the posterior distribution $q_\phi(z|x)$ output by the encoder and the prior distribution (standard normal distribution $\mathcal{N}(0, I)$). During training, the reparameterization trick ($z = \mu + \sigma \odot \epsilon$) enables error backpropagation.

2.3 Deep Support Vector Data Description (DSVDD)

Support Vector Data Description [12] is an anomaly detection method that uses only normal data to find the smallest hypersphere (center c, radius R) that encloses the majority of the data in the feature space. In Deep SVDD (DSVDD), a neural network is trained to map the data into the latent space instead of using a traditional kernel function.

$$\mathcal{L}_{DSVDD}(\theta) = R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \max(0, ||f(x_i) - c||^2 - R^2) \tag{2}$$

Here, c is the center of the hypersphere, R is its radius, and ν is a parameter determining the tolerance for outliers.

3 Related Work

3.1 Anomaly Detection Methods

Aktar et al. [1] proposed "DSVDD-CAE," an anomaly detection model integrating DSVDD and CAE, to address cyberattacks such as zero-day attacks in IoT networks. By combining the robust feature extraction capability of CAE with the anomaly detection capability of DSVDD, this method achieved high-precision anomaly detection through semi-supervised learning using only normal data. The loss function is expressed by combining Equation (1) and Equation (2):

$$\mathcal{L}_{DSVDD-CAE}(\theta) = \mathcal{L}_{CAE}(\theta) + \alpha \mathcal{L}_{DSVDD}(\theta). \tag{3}$$

Here, α is a hyperparameter that adjusts the tradeoff between the two terms.

Anomaly judgment is performed using an anomaly score $s(x_i)$ for the input data. The anomaly score is calculated using the squared distance between the latent representation and the center of the hypersphere, along with the reconstruction error:

$$s(x_i) = ||z_i - c||^2 + \beta \times MSE(x_i, \hat{x}_i).$$

Here, β is a hyperparameter adjusting the balance between the latent representation distance and the MSE. The optimal threshold for the anomaly score was determined by calculating the F1-score on validation data (90% normal, 10% anomalous) across multiple candidate thresholds and selecting the one with the highest F1-score. In their experiments, Aktar et al. demonstrated that anomaly detection using the optimal threshold achieved an F1-score of 99.25% on the ToN-IoT dataset, significantly outperforming K-Means and One Class SVM. However, its application to adversarial examples in image data was not verified.

3.2 Image Purification Methods for Adversarial Examples

Li et al. [8] proposed Defense-VAE, a defense method that purifies adversarial examples using a VAE. Unlike a standard VAE, which learns to reconstruct the input data itself, Defense-VAE is trained to receive an adversarial example x_{adv} as input and output the original clean image x, thereby purifying the attacked image. The loss function during training is defined as follows:

$$\begin{aligned}\mathcal{L}_{Defense-VAE} = &-\mathbb{E}_{q(z|x_{adv})}[\log p(x|z)] \\ &+ D_{KL}(q(z|x_{adv})||p(z)).\end{aligned}$$

Through experiments on the CIFAR-10 dataset, Li et al. showed that Defense-VAE outperformed Defense-GAN. However, a significant vulnerability remains: its defense performance drops substantially against unknown attacks not included in the training data (e.g., CW attacks against a model trained only on FGSM).

Yin et al. [13] proposed Adversarial Memory Variational AutoEncoder (AdMVAE), a defense method integrating a memory module into a VAE. In this method, parts of the input image are randomly masked before being fed to the encoder. Instead of passing the obtained latent variables directly to the decoder, they are aggregated with clean features of normal data stored in a memory module before reconstruction. This enables image purification based on normal features, even in the presence of adversarial noise.

AdMVAE is trained by combining a VAE and a GAN [2]. GAN is a generative model that learns by pitting two neural networks—a Generator (G) and a Discriminator (D)—against each other. By using a VAE as the generator, the impact of reconstruction loss is mitigated. The loss function for training is defined as:

$$\begin{aligned}\mathcal{L}_{AdMVAE} = &\min_G \max_D L_{GAN} + \alpha_{mem}\mathcal{L}_{Memory} \\ &+ \beta_{rec}\mathcal{L}_{rec} + D_{KL}[Q(z|X)||P(z)].\end{aligned}$$

Here, $\mathcal{L}_{rec}$ is the reconstruction error, $\mathcal{L}_{Memory}$ minimizes the feature distance between the latent representation and memory items, $\mathcal{L}_{GAN}$ is the adversarial error improving image sharpness, and $\mathcal{L}_{kl}$ is the KL divergence regularizing the latent space. α_{mem} and β_{rec} adjust the weights of $\mathcal{L}_{Memory}$ and $\mathcal{L}_{rec}$, respectively.

Furthermore, during inference, to prevent adversarial examples, "spherical sampling" is introduced, which scatters noise at a certain distance from the center in the latent space. This realizes a probabilistic defense making accurate gradient computation by the attacker difficult. Additionally, by using a tempered softmax function during memory readout, only the single most similar memory item is extracted, blocking noise contamination. Yin et al.'s experiments demonstrated that AdMVAE maintains reconstruction accuracy for normal data while showing high defense performance against Gray-box attacks on datasets like CIFAR-10.

4 Proposed Method

4.1 Objective of this Study

The objective of this research is to verify the effectiveness of anomaly detection and image purification using the DSVDD-AE model as a defense mechanism against adversarial examples on image datasets.

Existing AE-based defense methods rely on anomaly detection based on reconstruction error or image purification via reconstruction. However, these methods fail to completely defend against attacks in white-box environments, where attackers are aware of the defense mechanisms. Therefore, this study applies the DSVDD-CAE model, previously used in the IoT domain, to image recognition tasks. By considering feature distance in the latent space in addition to reconstruction error, we aim to improve detection accuracy. Furthermore, we propose a two-stage defense method that combines this with AE-based image purification.

In our experiments, we constructed two specific implementations of DSVDD-AE: "DSVDD-CAE" using a deterministic CAE, and "DSVDD-VAE" using a probabilistic generative VAE.

4.2 DSVDD-CAE and DSVDD-VAE Models

The DSVDD-CAE model implemented in this study is based on CAE and is trained to map data into class-specific hyperspheres in the latent space. Unlike the IoT dataset used by Aktar et al., we use a multi-class dataset; thus, we establish k centers $c_1, \ldots, c_k$ in the latent space corresponding to each class.

To form hyperspheres that enclose all normal data, we formulate the loss function without considering outlier tolerance from Equation (2), aiming solely to minimize the feature distance between the data and the centers:

$$\mathcal{L}_{DSVDD}(\theta) = \frac{1}{n}\sum_{i=1}^{n} ||z - c_k||^2.$$

The total loss function is defined as the sum with the CAE loss function:

$$\mathcal{L}_{DSVDD-CAE}(\theta) = \mathcal{L}_{CAE}(\theta) + \alpha\mathcal{L}_{DSVDD}(\theta). \tag{4}$$

Here, α is a hyperparameter representing the weight coefficient of the DSVDD loss function.

The DSVDD-VAE model is based on VAE and learns to map data into class-specific hyperspheres in the latent space similarly to CAE. While the basic structure is identical, the encoder outputs parameters of the probability distribution of the latent variables, generating latent vectors via the Reparameterization Trick.

The loss function $\mathcal{L}_{DSVDD-VAE}$ is defined by adding the DSVDD loss function to the VAE loss function, which consists of the reconstruction error $\mathcal{L}_{MSE}$ and KL divergence $\mathcal{L}_{KLD}$. As with DSVDD-CAE, centers corresponding to each class are established in the latent space.

$$\mathcal{L}_{DSVDD-VAE}(\theta) = \mathcal{L}_{MSE} + \gamma\mathcal{L}_{KLD} + \alpha\mathcal{L}_{DSVDD}. \tag{5}$$

Here, γ is a hyperparameter representing the weight coefficient of the KL divergence term.

4.3 Anomaly Score and Optimal Threshold Determination

Using the trained DSVDD-AE model, an anomaly score $s(x)$ is calculated for the input data. Images subjected to adversarial attacks deviate from the normal feature distribution; thus, the

feature distance from the center in the latent space is expected to increase. The anomaly score is defined as:

$$s(x) = \min_k ||z - c_k||^2 + \beta||x - \hat{x}||^2.$$

Here, $\min_k ||z - c_k||^2$ represents the feature distance to the nearest class center, and $||x - \hat{x}||^2$ represents the reconstruction error. To evaluate the defense performance in terms of the classification accuracy of reconstructed images, data judged as anomalous are still reconstructed by the decoder and fed into the subsequent classifier.

We describe the method for determining the optimal threshold for anomaly detection using the anomaly score. In this study, the threshold is defined as the p-th percentile of the anomaly scores across the entire training dataset. In other words, the threshold is set such that $p\%$ of all training data is judged as normal. To find the optimal percentile, p was varied in the range of $[90, 100]$.

When determining the optimal threshold, evaluations were based on the True Negative Rate (TNR) of normal data and the True Positive Rate (TPR) of Gray-box, White-box, and adaptive White-box attack data. Generally, setting a smaller threshold p improves the TPR for attack data but lowers the TNR for normal data, presenting a tradeoff. For practical anomaly detection, both high TNR for normal data and high TPR for attack data are required. Therefore, we find the p that maximizes the sum of TNR and TPR and adopt it as the optimal threshold.

4.4 Training and Testing of the DSVDD-AE Model

Model training proceeds as follows:

1. **Initialization:** The centers c_k for each class are randomly placed and fixed in the latent space, maintaining a certain feature distance from each other. Centers are fixed to prevent hypersphere collapse during updates.
2. **DSVDD-AE Training:** The DSVDD-AE model is trained to minimize Equations (5) and (4) using a training dataset consisting exclusively of normal data.
3. **Threshold Determination:** Anomaly scores $S(x)$ are calculated for the entire training dataset, and the proportion judged as normal is varied from 90% to 99% to find a balanced optimal threshold.
4. **Classifier Training:** An image classifier is trained using images reconstructed by the trained DSVDD-AE.

The training pipeline for the DSVDD-AE model is illustrated in Figure (1). $\mathrm{D}(z, c_k)$ represents the feature distance between z and each class center $c_k (k = 1, ..., 10)$. P represents the penalty terms of CAE or VAE. In the testing process (Figure 2), input images are passed through DSVDD-AE, and if the anomaly score exceeds the threshold, they are flagged as anomalous. Following this, images are reconstructed by the decoder and classified by ResNet-18.

5 Experiments

5.1 Experimental Setup, Environment, and Model Architecture

The experiments were conducted under the hardware and software environment detailed in Table 2. We utilized the CIFAR-10 dataset (50,000 training images, 10,000 test images). The

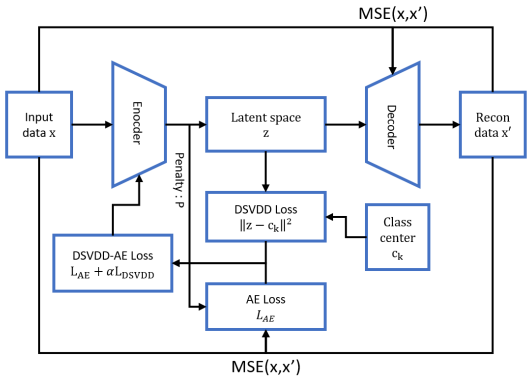

Figure 1: Training Pipeline

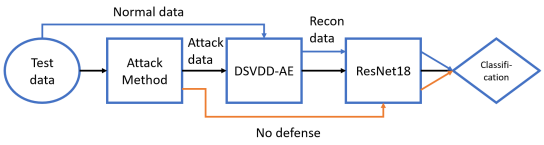

Figure 2: Testing Pipeline

training dataset was augmented with RandomCrop and RandomHorizontalFlip, and pixel values were normalized to the range $[0, 1]$.

Table 2: Experimental Environment

Item	Specification / Version
OS	Windows 11
CPU	Intel Core i7-13700F
Memory	32 GB
GPU	NVIDIA GeForce RTX 4060 (8GB)
Python	3.12.10
PyTorch	2.6.0+cu124

An 18-layer ResNet-18 [4] was employed as the subsequent classifier (Table 3). It was trained from scratch using reconstructed images without pre-training.

The architecture of the proposed DSVDD-AE model is shown in Table 4. The encoder consists of 3 convolutional layers and 1 fully connected layer, compressing the image into a 128-dimensional latent vector. Batch Norm and LeakyReLU are applied after each convolutional layer. The decoder reconstructs the image using a fully connected layer and transposed convolutional layers.

Adam was adopted as the optimization method, with a learning rate of 0.001 and a batch size of 128. Both models were trained for 100 epochs. Hyperparameters were set to $\alpha = 0.1$, $\beta = 0.1$, $\lambda = 10^{-4}$, and $\gamma = 10^{-5}$.

Table 3: Model Architecture of ResNet-18

Layer	Output Size	ResNet-18
conv1	32×32	$3 \times 3, 64$, stride 1
conv2_x	32×32	$[3 \times 3, 64; 3 \times 3, 64] \times 2$
conv3_x	16×16	$[3 \times 3, 128; 3 \times 3, 128] \times 2$
conv4_x	8×8	$[3 \times 3, 256; 3 \times 3, 256] \times 2$
conv5_x	4×4	$[3 \times 3, 512; 3 \times 3, 512] \times 2$
average pool, 10-d fc, softmax		

Table 4: Architecture of DSVDD-CAE / VAE Models

Encoder	**Output Size**
Input	$3 \times 32 \times 32$
Conv2d (32, k=4, s=2), BN, LeakyReLU	$32 \times 16 \times 16$
Conv2d (64, k=4, s=2), BN, LeakyReLU	$64 \times 8 \times 8$
Conv2d (128, k=4, s=2), BN, LeakyReLU	$128 \times 4 \times 4$
Flatten, Linear(2048, 128)	128 (Latent z)
Decoder	**Output Size**
Linear(128, 2048)	2048
Reshape	$128 \times 4 \times 4$
ConvTrans2d (64, k=4, s=2), BN, ReLU	$64 \times 8 \times 8$
ConvTrans2d (32, k=4, s=2), BN, ReLU	$32 \times 16 \times 16$
ConvTrans2d (3, k=4, s=2), Sigmoid	$3 \times 32 \times 32$

5.2 Attack Settings and Evaluation Metrics

We employed PGD attacks. Parameters were set to $\epsilon = 8/255$, iters $= 20$, and $\alpha_{pgd} = 2/255$. Three attack scenarios were assumed:

- **Gray-box:** The attacker knows only the parameters of the classifier and is unaware of the defense model. The objective function is:

$$L_{adv} = Sign(L_{classifier}) \tag{6}$$

- **Standard White-box:** The attacker knows both the classifier and the defense mechanism. The attack optimizes solely to maximize the classification error after passing through the DSVDD-AE model:

$$L_{adv} = Sign(L_{AE+classifier}) \tag{7}$$

- **Adaptive White-box:** The attacker aims to simultaneously degrade classification accuracy and evade anomaly detection by adding an anomaly score minimization term to the objective function:

$$L_{adv} = Sign(L_{AE+classifier}) + h \cdot s(x) \tag{8}$$

  Here, h balances classification error maximization and anomaly score minimization.

5.3 Evaluation Metrics

To quantitatively evaluate the performance of the proposed method, we use the following metrics based on True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

- **Classification Accuracy**: The proportion of all test samples correctly predicted by the downstream classifier:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

- **False Positive Rate (FPR)**: The proportion of normal samples mistakenly detected as anomalous by the model. In our experimental results, this corresponds to the detection rate of normal data:

$$FPR = \frac{FP}{TN + FP}.$$

- **True Negative Rate (TNR)**: The proportion of normal samples correctly identified as normal:

$$TNR = \frac{TN}{FP + TN}.$$

- **True Positive Rate (TPR)**: The proportion of attack samples correctly detected as anomalous. In our experimental results, this corresponds to the detection rate of attack data:

$$TPR = \frac{TP}{TP + FN}.$$

- **Defense Success Rate (Total Success Rate)**: The proportion of attacked samples where either the classifier predicted correctly or the anomaly detector successfully flagged the attack. Let N be the total number of samples, $\hat{y}_i$ be the predicted label, y_i be the ground truth label, and $d_i \in \{0, 1\}$ be the detection result (1 for detected, 0 for undetected). It is defined as:

$$\text{Success Rate} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[(\hat{y}_i = y_i) \vee (d_i = 1)].$$

  Here, $\mathbb{I}[\cdot]$ is the indicator function that returns 1 if the condition is true and 0 otherwise, and $\vee$ denotes the logical OR operation.

5.4 Experimental Results

5.4.1 Optimal Threshold

The trends of TNR, TPR, and their sum against each percentile p are shown in Figures 3 and 4. In Figure 3, the orange line denotes TNR and Gray-box TPR, the green line denotes TNR and White-box TPR, and the red line denotes TNR and adaptive TPR. In Figure 4, the red line indicates TNR alongside Gray-box and White-box TPRs, while the purple line evaluates TNR integrated with all attack TPRs. As p increases, TNR inherently rises, but TPR declines. At $p = 100$, TPR drops drastically since the threshold becomes excessively wide. Evaluating the sum of TNR and TPR, the maximum was reached at $p = 90$. However, because the threshold at $p = 90$ results in a high FPR of 10%, we also verify defense performance under a stricter threshold of $p = 99$ prioritizing FPR.

Table 5 shows the results for ResNet-18 without defense. It achieved 94.40% classification accuracy on normal data but dropped to 0.03% under PGD attacks, confirming extreme vulnerability.

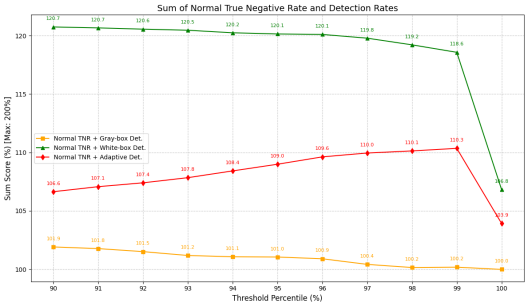

Figure 3: Threshold evaluation 1

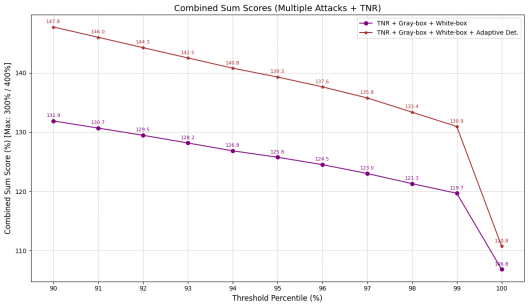

Figure 4: Threshold evaluation 2

Table 5: Experimental Results: No Defense (ResNet-18 Only)

Input Data	Classification Accuracy
Normal Data	94.40%
White-box PGD	0.03%

5.4.2 Evaluation of DSVDD-CAE

Tables 6 and 7 present the results for DSVDD-CAE at $p = 90$ and $p = 99$. Classification accuracy is calculated over data judged normal by anomaly detection. At $p = 90$, classification accuracy for normal data was 71.57%. Against Gray-box attacks, CAE image purification yielded 59.79% classification accuracy, combining with anomaly detection (6.08%) for a 65.87% defense success rate. In standard White-box attacks, classification accuracy fell to 0.25%, but the TPR rose to 11.61%, achieving an 11.86% defense success rate. Under adaptive White-box attacks, classification accuracy recovered to 22.49%, improving the defense success rate to 24.86%.

Conversely, at $p = 99$, the FPR decreased to 0.46%, improving normal data classification accuracy to 73.71%. However, attack detection sensitivity declined simultaneously, limiting defense success rates to 62.45% for Gray-box and 5.16% for standard White-box attacks. The adaptive White-box defense success rate stood at 23.25%, confirming an overall defense degradation tradeoff compared to $p = 90$.

Table 6: DSVDD-CAE ($p = 90$)

Input Data	Acc	TPR	Success
Normal	71.57%	5.38%	—
Gray PGD	59.79%	6.08%	65.87%
White (Std)	0.25%	11.61%	11.86%
White (Adp)	22.49%	2.37%	24.86%

Table 7: DSVDD-CAE ($p = 99$)

Input Data	Acc	TPR	Success
Normal	73.71%	0.46%	—
Gray PGD	62.24%	0.21%	62.45%
White (Std)	0.24%	4.92%	5.16%
White (Adp)	22.44%	0.81%	23.25%

5.4.3 Evaluation of DSVDD-VAE

Tables 8 and 9 show the results for DSVDD-VAE at $p = 90$ and $p = 99$. At $p = 90$, normal classification accuracy was 79.47%. Against Gray-box attacks, VAE image purification yielded 68.26% classification accuracy, combining with anomaly detection (11.08%) to achieve a 79.34% defense success rate, outperforming DSVDD-CAE. For standard White-box attacks,

while classification accuracy dropped to 0.18%, the TPR climbed to 30.10%, resulting in a 30.28% defense success rate. Under adaptive White-box attacks, classification accuracy and TPR reached 38.35% and 16.02%, yielding a 54.37% defense success rate.

At $p = 99$, normal FPR dropped to 0.97%, but overall attack TPR declined. Defense success rates resulted in 73.97% for Gray-box, 20.13% for standard White-box, and 50.12% for adaptive White-box attacks. Similar to CAE, stricter thresholds prevent normal data false positives but degrade attack detection capability. Nevertheless, across all conditions, the probabilistic VAE consistently demonstrated superior and more stable defense performance than the deterministic CAE.

Table 8: DSVDD-VAE ($p = 90$)

Input Data	Acc	TPR	Success
Normal	79.47%	9.19%	—
Gray PGD	68.26%	11.08%	79.34%
White (Std)	0.18%	30.10%	30.28%
White (Adp)	38.35%	16.02%	54.37%

Table 9: DSVDD-VAE ($p = 99$)

Input Data	Acc	TPR	Success
Normal	79.47%	0.97%	—
Gray PGD	72.83%	1.14%	73.97%
White (Std)	0.20%	19.93%	20.13%
White (Adp)	38.86%	11.26%	50.12%

5.4.4 Optimal h for Adaptive White-box Attacks

The effectiveness of adaptive White-box attacks fluctuates depending on h. We varied h within [0, 10] and examined TPR, classification accuracy, and defense success rates, as illustrated in Figure 5. The blue line denotes TPR, the green line denotes classification accuracy, and the red line denotes the defense success rate. As h increases, the TPR drops, but classification accuracy significantly improves, elevating the overall defense success rate. Attack optimization was most threatening to the system when $h = 0.0$.

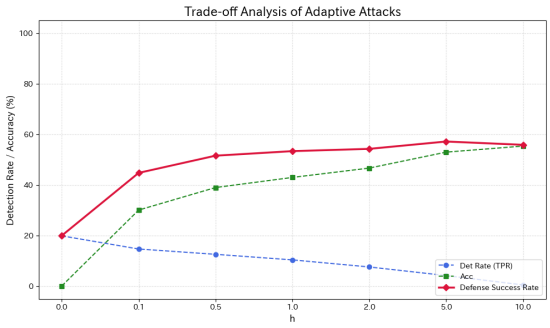

Figure 5: Optimal h evaluation

6 Performance and Discussion

6.1 Comparison between CAE and VAE

Our findings confirm that the proposed method possesses a complementary two-stage structure combining image purification and anomaly detection. Image purification effectively mitigates

relatively weak attacks like Gray-box, whereas latent space anomaly detection heavily contributes to defending against powerful White-box attacks. This demonstrates that the method utilizes distinct defensive mechanisms depending on attack strength.

For both CAE and VAE models, classification accuracy against Gray-box attacks remained high (60% to 70%), while the TPR was low (2% to 5%). Conversely, against White-box attacks, classification accuracy dropped below 1%, but the TPR notably increased (7% to 25%). This demonstrates that anomaly detection captures unpurifiable White-box samples, proving the complementary nature of image purification and anomaly detection.

Comparing deterministic DSVDD-CAE with probabilistic DSVDD-VAE, VAE achieved a 25.59% defense success rate under standard White-box attacks. This suggests that the inferential diversity introduced by VAE's probabilistic latent representation successfully disrupts accurate gradient computation by attackers relying on fixed perturbations. In adaptive White-box attacks, defense success rates reached 26.33% for CAE and 52.57% for VAE, demonstrating that VAE consistently provides more robust defense.

The drop in DSVDD-CAE's TPR under White-box attacks likely stems from CAE's specific training constraints. While CAE is regularized with a penalty term to stabilize the latent space for robust feature extraction, White-box attacks highly likely exploit this exact penalty term to shift latent representations toward another class's center, improperly minimizing the anomaly score. Furthermore, computing the Frobenius norm during CAE training imposes heavy computational costs. Therefore, from an implementation perspective, the probabilistic DSVDD-VAE model is more suitable as an adversarial defense mechanism.

6.2 Comparison of Defense Performance Across Methods

Table 10 compares the defense performance of DSVDD-VAE, Defense-VAE, and AdMVAE. Against Gray-box attacks, our proposed method achieved an approximate 10 percentage point improvement in defense success rate over Defense-VAE, performing on par with AdMVAE. Under White-box attacks, it improved by about 25 percentage points compared to Defense-VAE, but was roughly 12 percentage points lower than AdMVAE. These results highlight that each method operates on distinct defensive mechanisms, validating that our integrated architecture enables flexible protection suited to varying attack strategies.

Table 10: Comprehensive Comparison of Defense Performance Across Methods

Method	Normal Data	Gray-box	White-box
Defense-VAE	67.47%	65.92%	0.71%
AdMVAE	86.33%	76.25%	37.66%
DSVDD-VAE	**79.47%**	**79.34%**	**30.28%**

6.3 Tradeoffs in Adaptive White-box Attacks

From the optimal h search experiments, setting a larger h in adaptive White-box attacks—incorporating anomaly score minimization into the objective function—lowers the TPR compared to standard White-box attacks but significantly improves the classifier's accuracy rate, ultimately increasing the system's overall defense success rate. This implies that a standard White-box attack ignoring anomaly detection ($h = 0.0$) remains the optimal attack strategy.

This behavior can be attributed to insufficient inter-class separation within the latent space resulting from extending DSVDD to multiple classes. Originally, DSVDD is a one-class clas-

sification method designed to learn a minimal hypersphere enclosing a single normal class; extending it to 10 classes highly likely caused individual feature distributions to cluster in close proximity or overlap near boundaries. Under such narrow inter-class margins, an attacker can easily guide data points into adjacent class territories using minimal perturbations without triggering significant anomaly score increases. This architectural limitation restricted detection performance under White-box conditions, rendering attacks that ignore anomaly scores ($h = 0.0$) the optimal solution.

However, when attackers explicitly incorporate anomaly score evasion into their objective function ($h > 0$), the target model's classification accuracy is largely improved. This proves that an operational tradeoff is forced upon the attacker. To induce misclassification, input data must be moved outside the decision boundary; simultaneously, to prevent anomaly score escalation, data must remain strongly anchored near the class centers. These conflicting constraints restrict the attacker from adding sufficient misclassifying perturbations, consequently increasing cases where the classifier maintains correct predictions.

This result implies that introducing anomaly detection, even if the detection itself is bypassed, inherently restricts the attacker's degrees of freedom and effectively enhances the overall robustness of the system.

7 Conclusion

In this study, we proposed "DSVDD-AE," a two-stage defense mechanism integrating AE and DSVDD, which combines image purification and anomaly detection to counter adversarial examples against deep learning models. Specifically, we implemented a deterministic model, DSVDD-CAE, and a probabilistic model, DSVDD-VAE, and verified their effectiveness through evaluation experiments on the CIFAR-10 dataset.

The results showed that against Gray-box attacks, both CAE and VAE implementations realized effective noise removal and anomaly detection, achieving higher defense success rates compared to the existing Defense-VAE method. Furthermore, we demonstrated that the proposed method acts as a strong constraint on adversarial perturbations. We confirmed a tradeoff where the constraint of attempting to evade anomaly detection strips the attacker of degrees of freedom, which paradoxically suppresses the degradation of classification accuracy.

Future work will involve further enhancing the system's robustness by improving the AE for better reconstruction accuracy and by extensively refining the feature distance calculations of anomaly detection to enforce wider inter-class margins.

8 Acknowlegement

This work is partially supported by JSPS KAKENHI Grant Number 26H02496 and SECOM Science and Technology Foundation.

References

[1] Sharmin Aktar and Abdullah Yasin Nur. Robust anomaly detection in iot networks using deep svdd and contractive autoencoder. In *2024 IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2024.

[2] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[3] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[8] Xiang Li and Shihao Ji. Defense-vae: A fast and accurate defense against adversarial attacks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 191–207. Springer, 2019.

[9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[10] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017.

[11] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th international conference on international conference on machine learning*, pages 833–840, 2011.

[12] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

[13] Sheng-lin Yin, Xing-lan Zhang, and Li-yu Zuo. Defending against adversarial attacks using spherical sampling-based variational auto-encoder. *Neurocomputing*, 478:1–10, 2022.