Vol. 8, No. 3, June 2026

ISI Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

A Mixed Adversarial Awareness Technique for Improving Neural Network Defense

Moses Apambila Agebure[1], Sampson Kuma Konja[1], Stephen Akobre[2], Mohammed Ibrahim Daabo[1]

[1]Department of Computer Science, C.K. Tedam University of Technology and Applied Sciences, Ghana
[2]Department of Cyber Security and Computer Engineering Technology, C.K. Tedam University of Technology and Applied Sciences, Ghana

Received:
September 1, 2025
Revised:
March 11, 2026
Accepted:
June 9, 2026
Published:
June 26, 2026

Corresponding Author:
Author Name*:
Moses Apambila Agebure
Email*:
magebure@utas.edu.gh

DOI:
10.63158/journalisi.v8i3.1552

Abstract. Neural Network (NN) models, particularly Convolutional Neural Networks (CNNs), have achieved remarkable performance in computer vision tasks but remain highly vulnerable to adversarial attacks. Existing defense techniques mainly focus on detecting adversarial examples and often show limited effectiveness when adversarial perturbations coexist with significant noisy inputs. To address this limitation, this study proposes a Mixed Adversarial Awareness Technique (MAAT) based on kernel density estimation and a Bayesian uncertainty estimator. Kernel density estimation is used to model data manifolds in the input subspace, while the Bayesian uncertainty estimator, inspired by the Dirichlet process, quantifies predictive uncertainty in the input space. The proposed technique was evaluated on three benchmark datasets, CIFAR-10, CIFAR-100, and SVHN, using four adversarial attack schemes, namely FGSM, BIM, JSMA, and C&W, as well as Gaussian noise injection. The LeNet ConvNet model was employed as the test classifier. Experimental results show that MAAT effectively flags adversarial and noisy instances, improving detection performance with AUC values ranging from 0.84 to 0.96, compared with 0.61 to 0.94 achieved by selected state-of-the-art techniques. These findings demonstrate that combining density-based manifold modeling with uncertainty estimation provides a robust defense against mixed adversarial and noisy inputs.

Keywords: Adversarial Example Detection; Noise-Aware Defense; Kernel Density Estimation; Bayesian Uncertainty; Neural Network Defense

1. INTRODUCTION

Adversarial examples are legitimate inputs maliciously modified to force models to incorrectly classify them [1, 2]. Although neural networks have achieved successful implementation in several areas of application such as image processing, speech recognition, among others and are regarded as one of the state-of-the-art learning schemes, they are extremely vulnerable to adversarial attacks. This vulnerability militates against their successful application in security-critical domains [3]. An adversary in a neural network model comes in a form such that given a legitimate input x, and a target T, a new input that is very similar to x is crafted such that it impairs the network's ability to correctly predict the target, T [2, 4]. The presence and effects of adversarial examples were first noticed in the image classification domain, which suggested that it is possible to marginally perturb an image to cause a classifier to wrongly classify it with high confidence [5, 6].

There is an increase in the applications of machine learning, most especial neural networks in mission/security-critical domains [7]. In areas of application such as autonomous vehicles, little perturbations in the input image stream can cause them to make wrong decisions and as such take wrong actions, which could be injurious when operating in the real-world [8, 9, 10, 4]. Similarly, findings from recent studies indicate that, in speech recognition, it is possible to generate and manipulate hidden voice commands similar to that of a user using methods such as Gaussian Mixture and Hidden Markov Chain Models to control user devices without their knowledge [11, 12, 13]. To this end, several techniques have been introduced in literature in an attempt to improve models' defense against adversarial examples. Those that are confirmed to be inherently robust to adversarial detection include adversarial training, defensive distillation, gradient regularization, model compression, and activation pruning [14, 15, 12].

Among these, adversarial training has proven to be the most effective. Adversarial training, which is modelled on the min-max optimization problem, can be regarded as a data augmentation technique that enables neural network models to learn effectively on adversarial examples, therefore enabling them to recognise malicious examples when encountered [16, 7]. It consists of two loops; the inner loop and the outer loop. The inner loop iteratively generates the adversarial examples that try to maximise the model's loss,

while the outer loop updates the model's weights to minimise that loss. This technique is popular on image-classification and is mostly applied to autonomous vehicles, medical imaging, malware detection, and reinforcement learning. It is reported to highly effective against white-and black-box adversarial attacks but includes no explicit mechanism for the detection of noisy instances.

Defensive Distillation is another stable technique in adversarial detection. It trains a smaller "Student" network on the softened probability output of a larger "Teacher" network instead of hard labels. Based on the "Knowledge", the technique reduces the model's over-confidence and smoothens the decision boundary. The core mechanism uses a temperature-scaled softmax [17]. This technique was previously introduced as a post-training defence for image classifiers and later extended to larger CNNs and some natural-language tasks. Though these approaches reduce the success rate of gradient-based attacks by several factors, they offer limited defence against naturally occurring noise and has been extended by [18] to enable stronger, adaptive attacks detection.

In Gradient Regularisation, a penalty term is added to the loss function to discourage inputs with large gradients. This ensures that the model's decision boundary is less sensitive to small perturbations [19, 20]. Because of its nature of directly flattening the loss landscape, adversarial and moderate random noise detection has been improved. However, like other classical methods, it is not formulated for explicit detection of noise and their co-occurrence with adversarial instances.

Model compression is another relatively well-established technique for adversarial detection. In this approach, network size and complexity are reduced by techniques such as weight pruning, quantisation, and low-rank factorisation in an attempt to preserve accuracy and, in some cases, robustness [21]. This technique iteratively removes low magnitude weights and fine-tunes the remaining connections [22]. Compression has two side effects; it either leads to improved model robustness by eliminating redundant parameters through which an attacker can exploit, or increases the vulnerability of the model based on the compression ratio and method used [23]. While it can indirectly help in noisy input detection by simplifying the representations, it focuses on efficiency rather than joint adversarial-noise defence.

Stochastic Activation Pruning (SAP), on the other hand, works by dropping a subset of activation neurons with smaller magnitudes. As a game theory proposed strategy between the model and adversary with no retraining required, it can be applied directly to pretrained networks [24, 25]. Its main focus is on adversarial examples with no emphasis on noisy examples or both. It was formalised as a stochastic process that disrupts coordinated adversarial perturbations while maintaining calibration [26]. Its application in image-classification tasks has shown robustness against both white- and black-box attacks while providing negligible targeted benefits against noisy examples.

As alluded to in [27, 28], the co-existence of adversarial and noisy examples in datasets have significant negative effects on model performance. However, previous studies focused on the detection of adversarial examples with no specific mechanism for the detection of noise particularly, in situations where they co-exist with adversarial examples in a dataset. That is, existing adversarial detection techniques are not able to efficiently detect noisy instances whenever they co-exist with adversarial instances in a dataset because they are not primarily designed to detect noise. The objective of this paper is therefore, to propose, implement and evaluate a hybrid adversarial-noise detection technique with the ability to simultaneously detect the existence of adversarial and noisy instances. The contribution of this paper is anchored on the novel hybridisation of a nearest neighbour-based density estimation with Bayesian uncertainty estimation to effectively detect adversarial and noisy examples.

2. METHODS

This section presents the methods and the tools employed to accomplish the objective of the study. First the proposed technique is outlined, followed by the experimental setup in which a description of the datasets used for evaluation, architecture of the neural network models as well as the training and evaluation processes are presented.

2.1. Proposed Technique

In this paper, a Mixed Adversarial Awareness Technique (MAAT) that leverages the power of Kernel Density and Bayesian Uncertainty is proposed to facilitate adversarial examples detection. The proposed technique is incorporated into a neural network model between the last hidden layer and the output layer to flag feature vectors as adversarial, noise or

normal. For optimal performance, the proposed technique is modelled on the feature representation extracted by the neural network model in the course of its training. These feature vectors are then used to fit the proposed model using the processes outlined in the subsequent subsections and summarised in Algorithm 1. However, it is worth noting that the fitting of the proposed model is done after training of the neural network is completed and thus, is independent of the training processes of the neural network. It only uses the features extracted by the neural network. The core purpose of the proposed technique is to help flag potential adversarial examples before they are transitioned to the output neurons of the network. Being able to determine the true status of an example as to whether it is adversarial, noisy or normal before a model's prediction is envisaged to serve as a guide to the model in its decision making. Once an example is flagged as being adversarial or noisy, then actions that would have been taken can be ignored to avoid the potential consequences associated with it.

2.1.1. Kernel Density Estimation

The first phase of the proposed technique involves the estimation of the Kernel Density using a nearest neighbour-based estimation approach. The nearest neighbour-based estimation is a non-parametric method that is employed to estimate the Probability Density of a dataset using the distance of data points to their nearest neighbours in the dataset. The probability of a data point x_i, in a dataset $\mathrm{X} = \{x_1, x_2, \dots, x_n\} \subset R^d$ is estimated using Equation 1.

$$\hat{p}(\mathrm{x_i}) = \frac{k}{n.V_d.r_k(x_i)^d} \tag{1}$$

where k is the number of neighbours of x_i, n is the total number of data points, $\mathrm{V_d}$ the volume of a unit hypersphere in the R^d, $r_k(x_i)$ the Euclidean distance from point x_i to the $\mathrm{k^{th}}$ nearest neighbour, and d is the dimensional space of the dataset. In this study, $k = 5$ is adopted from Dudani [29], where it is demonstrated that over high-dimensional space, the nearest-neighbour count, k = 5 gives the best bias-variance trade-off to distance-weighted density estimation.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Algorithm 1 Mixed Adversarial Awareness Technique (MAAT)

Require: Training dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n}$,
Pre-trained LeNet ConvNet model $f(\cdot)$,
Number of nearest neighbors $k = 5$,
Weight parameters $w_1, w_2 > 0$,
Detection thresholds $\tau_{\text{adv}} = 0.05$, $\tau_{\text{noise}} = 0.10$,
New input sample x'

Ensure: Label $\ell \in \{\text{adversarial}, \text{noisy}, \text{normal}\}$ for x'

Extract feature vectors $\mathcal{F} = \{z_i\}_{i=1}^{n}$ from the last hidden layer of $f(\cdot)$ for all $x_i \in \mathcal{D}_{\text{train}}$

for each $z_i \in \mathcal{F}$ **do**

Compute $r_k(z_i)$: Euclidean distance to its k-th nearest neighbor in $\mathcal{F}$

Estimate density:

$$p(z_i) = \frac{k}{n \cdot V_d \cdot r_k(z_i)^d}$$

where V_d is the volume of the unit hypersphere in $\mathbb{R}^d$

end for

Store fitted density estimator $\hat{p}(\cdot)$

Approximate posterior $P(\theta \mid \mathcal{D})$ via Bayes' theorem using Monte Carlo dropout

Perform T stochastic forward passes (dropout enabled) to sample model weights $\theta^{(t)}$

Extract feature vector z' from the last hidden layer of $f(x')$

Compute KDE density estimate:

$$\hat{p}(z') = \frac{k}{n \cdot V_d \cdot r_k(z')^d}$$

Compute predictive distribution via Monte Carlo approximation:

$$P(y \mid x', \mathcal{D}) \approx \frac{1}{T} \sum_{t=1}^{T} p(y \mid x', \theta^{(t)})$$

Compute predictive entropy (uncertainty):

$$H(x') = -\sum_{c=1}^{C} p_c \log p_c, \quad p_c = \frac{1}{T} \sum_{t=1}^{T} p(c \mid x', \theta^{(t)})$$

Compute anomaly score:

$$S(x') = w_1 \cdot H(x') - w_2 \cdot \log \hat{p}(z')$$

if $S(x') < \tau_{\text{adv}}$ **then**

$\ell \leftarrow$ adversarial

else if $\tau_{\text{adv}} \leq S(x') < \tau_{\text{noise}}$ **then**

$\ell \leftarrow$ noisy

else

$\ell \leftarrow$ normal

end if

return ℓ

1

The assumption of using the nearest neighbour-based technique is that, regions with many nearby examples would have the highest densities. However, when perturbation is applied to an example, it will most likely shift away from its original place significantly by ϵ, putting it in a low-density region of the data space [2]. This implies that examples, which are adversarially modified will be shifted significantly from their original positions thereby forming less dense regions. Therefore, when these data points are passed through the technique, the technique using the density of the data around these examples, would be able to identify them as adversarially perturbed examples. In the case of noisy examples, it is assumed that, because they are often randomly generated by adding a Gaussian noise, $\epsilon \sim \mathcal{N}(0, \sigma^2)$ to examples, they are unlikely to shift significantly from normal examples in the data space.

2.1.2. Bayesian Uncertainty Estimation

The second phase of the technique seeks to confirm the confidence of the technique's decision by modelling the posterior distribution of the dataset. This approach incorporates uncertainty by treating model parameter, θ as a random variable with a prior distribution $p(\theta)$. In order to compute the posterior distribution using Baye's theorem as in Equation 2, the dataset is modelled as $\mathrm{D} = \{(x_i, y_i)\}_{i=1}^{n}$.

$$\mathrm{P}(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \tag{2}$$

Where $\mathrm{p}(D|\theta)$ is the likelihood, and $\mathrm{p}(\mathrm{D}) = \int \mathrm{p}(\mathrm{D}|\theta)\mathrm{p}(\theta)\mathrm{d}\theta$ is the computed marginal likelihood. The predictive distribution (Bayesian Uncertainty Estimate) of a new data point, x' is obtained using Equation 3, which is approximately the same as the Dirichlet Estimation Process:

$$\mathrm{P}(y|x', D) = \int p(y|x', \theta)p(\theta|D)d\theta \tag{3}$$

with a predictive entropy as a measure of uncertainty given as Equation 4.

$$H(x') = -\sum_{c=1}^{C} \bar{p}_c \log \bar{p}_c \tag{4}$$

Where $\bar{p}_c = \frac{1}{T}\sum_{t=1}^{T} p(c|x', \theta_t)$.

2.1.3. Anomaly Score

The use of density and entropy computation in MAAT is based on the hypothesis that, adversarial examples will typically have low density and deviate significantly from the data space with high predictive entropy for uncertainty. While noisy examples will have moderate entropy and density. Normal examples are assumed to always yield low entropy and high density. Based on these, the anomaly score, $S(x')$ of an example is determined using Equation 5, which is derived by hybridising the estimated density $\hat{p}(x')$ and predictive entropy $H(x')$ defined by Equation 1 and 4, respectively.

$$S(x') = w_1 . H(x') - w_2 . \log \hat{p}(x') \tag{5}$$

Where $w_1, w_2 > 0$ are weight balancing conditions for entropy and density. The negative log-density ensures that lower density increases the score, which aligns with higher entropy for adversarial inputs.

2.1.4. Anomaly Detection

The proposed Mixed Adversarial Awareness Technique (MAAT) serves as a plug-in layer between the last hidden layer and the output layer of a base neural network model. The Kernel-density and uncertainty estimation are applied directly on the extracted feature representation of input data transitioning from the last hidden layer to the output layer. The combined density and uncertainty estimates define the anomaly score, defined in Equation 5, which is compared to a calibrated thresholds to either classify an input as normal, adversarial, or noisy. The thresholds τ_{adv} and τ_{noise} for adversarial and noise are set to 0.05 and 0.10 respectively. These thresholds were arrived at after extensive experimentation with different configurations. If the anomaly score of an input, $S(x') < \tau_{adv}$ it is considered adversarial, however if $S(x')$ falls within the range, $\tau_{adv} < S(x') < \tau_{noise}$ (i.e. 0.05 – 0.10) it is considered as noise, else it is classified as normal (unperturbed input). The relatively lower threshold of 0.05 for adversarial examples and higher threshold above 0.10 for normal examples reflects the hypothesis that adversarial examples will always have lower densities compared to other examples.

2.2. Experimental Setup

2.2.1. Datasets

Three standard open-access imagery datasets (CIFAR-10, CIFAR-100, and SVHN) from the TensorFlow repository are used to evaluate the performance of the proposed technique. The choice of these datasets is mainly due to their diversity, complexity, and widespread usage for similar purposes. The CIFAR-10 dataset consists of 10 classes while CIFAR-100 comprise of 100 fine-grained categories of objects; offering varying levels of difficulty for model performance under adversarial perturbations. Also, the SVHN, a 10 classes dataset, introduces real-world visual challenges such as illumination changes, background noise, and digit variations, making it suitable for assessing generalization and robustness in practical scenarios. Furthermore, these datasets are standardised benchmarks with consistent image dimensions (32 × 32 pixels) and well-defined training and testing sets enabling easier reproducibility. Their use in this study is to facilitate fair comparison and validation of the proposed technique against existing defence mechanisms. A summary of these datasets is presented in Table 1.

Table 1. Summary of Datasets

Datasets	Training Samples	Testing Samples	No. of Classes	Image Size
CIFAR-10	50000	10000	10	32 x 32
CIFAR-100	50000	10000	100	32 x 32
SVHN	73257	26032	10	32 x 32

2.2.2. Adversarial and Noise Data Generation

Four adversarial attack schemes, the Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), Jacobian Saliency Map Attack (JSMA), and Carlini & Wagner (C&W) are used to generate adversarial examples for testing. The perturbation scale of $\epsilon = 0.1$ is used for generating adversaries in all attacking techniques unless otherwise stated. For noisy examples, a Gaussian noise, $\sigma = 0.1$ is added. The values for ϵ and σ are chosen to reflect what is generally used in literature. For each experiment, unless otherwise stated, adversarial and noisy examples were only generated from examples in test datasets that are correctly classified by the trained LeNet ConvNet model. This was done to ensure that the conditions for which an example is flag as adversarial or noisy is not from its

original nature but due to the perturbation. The adversarial attack techniques were implemented in TensorFlow. The Cleverhans library is used for FGSM and JSMA [30], while the implementation of C&W in the Torchattacks in PyTorch is adopted for C&W attacks.

2.2.3. Training and Evaluation

A LeNet ConvNet [31] with a dropout rate of 0.5 applied at the last pooling layer and after the inner-product layer is employed as the base model in this study. The dropout is a hyper-parameter that defines the probability of a neuron being used or ignored in an iteration during the training phase. The choice of 0.5 dropout rate is to ensure that 50% of the neurons are randomly ignored in a given training iteration. This helps to prevent overfitting of the model. The images in the various datasets were normalised to floating-point numbers on a range [0,1], which indicates an image scale representation change in pixel in a greyscale from full-on to full-off. This means that on an L_2 change of 1.0 and L_∞ of 1.0 and not 255 for the distance metrics. The training of the models was done using Adadelta optimizer with a batch size of 256. The performance of the models in the course of training and validation was monitored using categorical cross-entropy loss. A summary of the models used are shown in Tables 2 and 3.

Table 2. CIFAR-10 and CIFAR-100 Model Summary

Layer Type	Con2D	Con2D second layer	Activation	MaxPooling 2D	Flatten	Dropout	Dense first layer	Dense middle layer	Con2D middle Layer	Dense last layer
Properties (CIFAR-10)	Filter=32, kernel=(3,3)	Filter=64, kernel=(3,3)	ReLu	Poolsize=(2,2)	–	0.5	size=1024, l_2=0.01	size=512 l_2=0.01	filter=128 kernel=(3,3)	size=10
Properties (CIFAR-100)	Filter=32, kernel=(3,3)	Filter=64, kernel=(3,3)	ReLu	Poolsize=(2,2)	–	0.5	size=1024, l_2=0.01	size=512 l_2=0.01	filter=128 kernel=(3,3)	size=100

Table 3. SVHN Model Summary

Layer Type	Con2D	Activation	MaxPooling2D	Flatten	Dropout	Dense first layer	Dense middle layer	Dense last layer
Properties	Filter=64, kernel=(3,3)	ReLu	Poolsize=(2,2)	—	0.5	size=512	size=126	size=10

3. RESULTS AND DISCUSSION

3.1. Experimental Results

The experimental results of the Mixed Adversarial Awareness Technique (MAAT) and other state-of-the-art techniques on three benchmark datasets (CIFAR-10, CIFAR-100, and SVHN) are presented and discussed in this section. The performance of the base model without perturbed examples is presented, followed by the results when noise and adversarial examples are introduced into the test set, and then when adversarial detection techniques are incorporated into the model.

3.1.1. Effects of Perturbations on Model Performance

The AUC scores of the base model with and without noise and adversarial examples in the test dataset is presented in Table 4. On the clean dataset the model achieved relatively higher performance with AUC scores of 0.92 on the CIFAR-10 and SVHN datasets and 0.91 on the CIFAR-100 dataset. The performance of the model decreased to 0.79, 0.75 and 0.77 on the CIFAR-10, CIFAR-100 and SVHN datasets respectively, when noisy examples were introduced into the test dataset. This marks a reduction in model's performance up to 14.13%, 17.58% and 16.30% on the CIFAR-10, CIFAR-100 and SVHN datasets, respectively.

As illustrated in Figure 1, the model's performance degraded further when adversarial examples generated using four (4) attacking schemes (FGSM, BIM, JSMA and C&W) were introduced into the test dataset. On the CIFAR-10 dataset, the performance dropped to 0.65, 0.64, 0.77 and 0.62, respectively, when adversarial examples generated using FGSM, BIM, JSMA and C&W were introduced into test dataset. This reflects an average performance drop of about 27.71% with the minimum drop of 16.30% recorded when JSMA is used and the maximum of 32.61% when the C&W attacking scheme is employed. The JSMA scheme yielded the minimum reduction in model performance of about 21.98% and 20.65%, on the CIFAR-100 and SVHN datasets, respectively. The maximum reduction on these two datasets respectively, stood at 34.07% and 34.78%, when the BIM scheme was used and the average reduction was 29.67% on the CIFAR-100 dataset and 29.89% on the SVHN dataset. The results shown in Table 4 and the percentage drop in performance illustrated in Figure 1, confirm that in isolation, noisy and adversarial examples have severe negative impact on the performance of the base model considered in this study.

Table 4. Effects of Individual Attacks on Base Model

Dataset/Combinations	AUC Scores		
	CIFAR-10	CIFAR-100	SVHN
Clean Examples	0.92	0.91	0.92
Clean & Noise	0.79	0.75	0.77
Clean & FGSM	0.65	0.63	0.62
Clean & BIM	0.64	0.60	0.60
Clean & JSMA	0.77	0.71	0.73
Clean & C&W	0.62	0.62	0.63

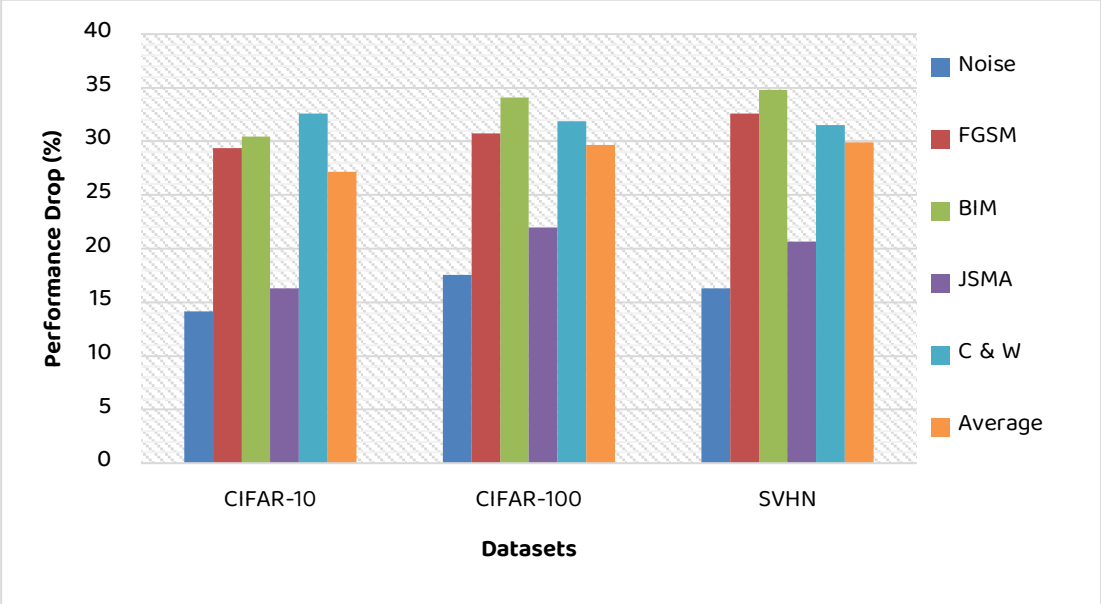

Figure 1. Percentage Drop in Performance Across Individual Attacks

To establish the co-effects of noisy and adversarial examples on the performance of the base model, it was tested using a mixture of these examples, which is presented in Table 5 with the percentage drop in performance illustrated in Figure 2. From Table 5, the best performance of the model across the datasets were recorded when the adversarial examples were generated using JSMA, which translated to 27.17%, 32.97%, and 26.09% reduction on the CIFAR-10, CIFAR-100 ad SVHN datasets, respectively, as illustrated in Figure 2. Conversely, the worse performance in terms of AUC stood at 0.59 on the CIFAR-10, 0.58 on CIFAR-100 and 0.61 on the SVHN dataset, representing 35.87%, 36.26% and 33.70% reduction, respectively, when the C&W attacking technique was used to generate the adversarial examples.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Table 5: Effects of Mixed Perturbations on Base Model

Dataset/Combinations	AUC Scores		
	CIFAR-10	CIFAR-100	SVHN
Noise & FGSM	0.63	0.59	0.67
Noise & BIM	0.60	0.60	0.63
Noise & JSMA	0.67	0.61	0.68
Noise & C&W	0.59	0.58	0.61

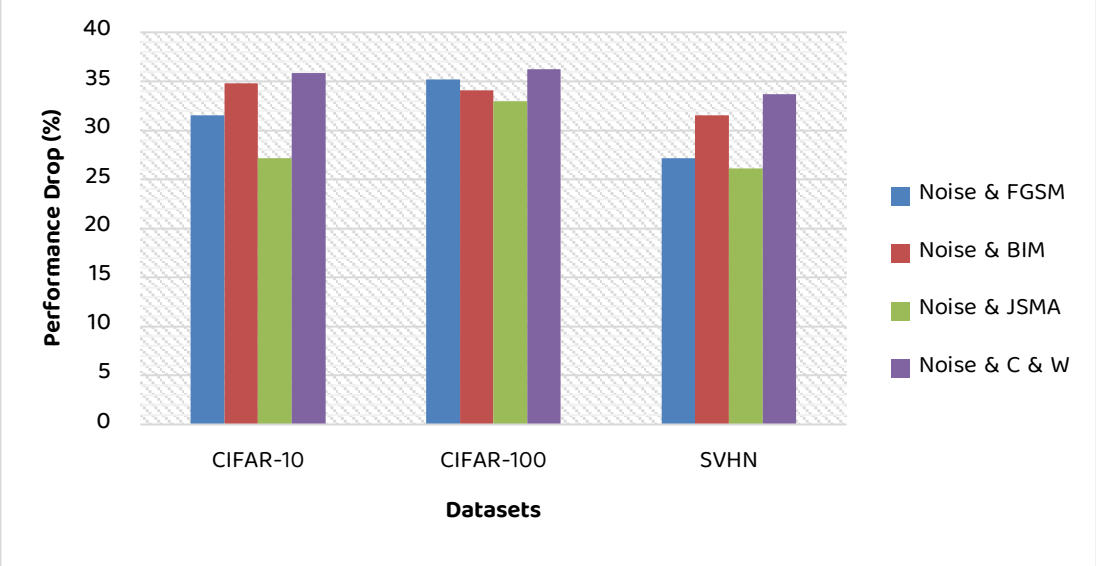

Figure 2. Percent Drop in Performance on Mixed Perturbations

The trend of the performance of the model across the test cases presented reveals a significantly negative impact of noise, adversarial examples and their combination on the performance of the LeNet CovNet model considered without the use of appropriate defense mechanisms. This confirms the assertions made in earlier studies on the effects of noise and adversarial examples on model performance and also, the hypothesis of this study that their co-occurrence presents a further strain on model performance.

3.1.2. Performance of Model with Defense Techniques

Having established the extent to which the presence of perturbations in test datasets affect the performance of the base neural network model. Experimental results on how the proposed Mixed Adversarial Awareness Technique (MAAT) improves the performance of the base model in the presents of perturbations is presented and compared to existing

techniques in this section. The performance of the base neural network model while using MAAT and three state-of-the-art defense techniques to detect mixed perturbations (a combination of noise and adversarial examples created using FGSM, BIM, JSMA and C&W) are presented in Table 6 and illustrated graphical in Figure 3.

From Table 6, it is evident that the performance of the base model improved in all cases when the various defense techniques are employed as compared to the results obtained without any defense mechanism shown in Table 5. On the CIFAR-10 dataset, the use of MAAT, AT and GM yielded averagely similar results across each perturbation scheme. MAAT, AT and GM each yielded 0.93 on Noise and FGSM attack scheme, 0.97 on Noise and BIM, 0.90 on Noise and JSMA, and 0.94 on Noise and C&W.

The results on the CIFAR-100 and SVHN datasets, however, showed a different trend. On the CIFAR-100 dataset, the use of the proposed technique (MAAT) yielded better model performance in three out of the four perturbation conditions. It yielded 0.89 AUC as against 0.80 for AT and GM and 0.70 for DD on the Noise & FGSM attack scheme, 0.90 against 0.76, 0.87 and 0.71 respectively, for AT, GM and DD on the Noise & JSMA attacks. It also, yielded 0.93 AUC against 0.80 for AT and DD and 0.89 for GM on the Noise & C&W schemes. MAAT, however, fell short in the Noise & BIM scheme where it came closely behind GM (0.88) with 0.84 but performed better than AT and DD as shown in Table 6. Results on the SVHN dataset, further confirms the superiority of MAAT against the existing defense techniques as it out performed all of them under all attack conditions with significantly higher AUC values except on the Noise & C&W attack scheme where GM produced the same AUC measure (0.91). The dynamics of the results as presented, suggests that, though all the defense techniques proof useful in detecting noise and adversarial examples in the test datasets, the proposed technique, MAAT is more resilient in most cases than the existing techniques. This is evident in Table 6 and Figure 3, where it out-performed almost all the existing techniques, particularly, the AT and DD with wider margins on the CIFAR-100 and SVHN datasets. The relatively higher increase in classifier performance on the CIFAR-100 and SVHN datasets when MAAT is employed can be attribute to the higher intra-class variability in these two datasets. Introducing perturbed instances (adversarial and noise) increases this variability and thereby making it difficult for the already straggling model to correctly classify them. Additionally, MAAT draws its strength from variations in the test data and as such perturbing instances that already

lie at the boarder lines of the data clusters further puts them away thereby making it easier for MAAT to detect them.

Table 6. Performance of Defense Techniques Against Mixed Perturbations

	Noise & FGSM			Noise & BIM			Noise & JSMA			Noise & C&W		
	C-10	C-100	SV	C-10	C-100	SV	C-10	C-100	SV	C-10	C-100	SV
AT	0.93	0.80	0.87	0.97	0.76	0.71	0.90	0.76	0.61	0.94	0.80	0.71
GM	0.93	0.80	0.94	0.97	0.88	0.87	0.90	0.87	0.91	0.94	0.89	0.91
DD	0.90	0.70	0.94	0.96	0.72	0.80	0.82	0.71	0.79	0.90	0.80	0.70
MAAT	0.93	0.89	0.96	0.96	0.84	0.93	0.90	0.90	0.96	0.93	0.93	0.91

** C-10 = CIFAR-10, C-100=CIFAR-100 & SV=SVHN*

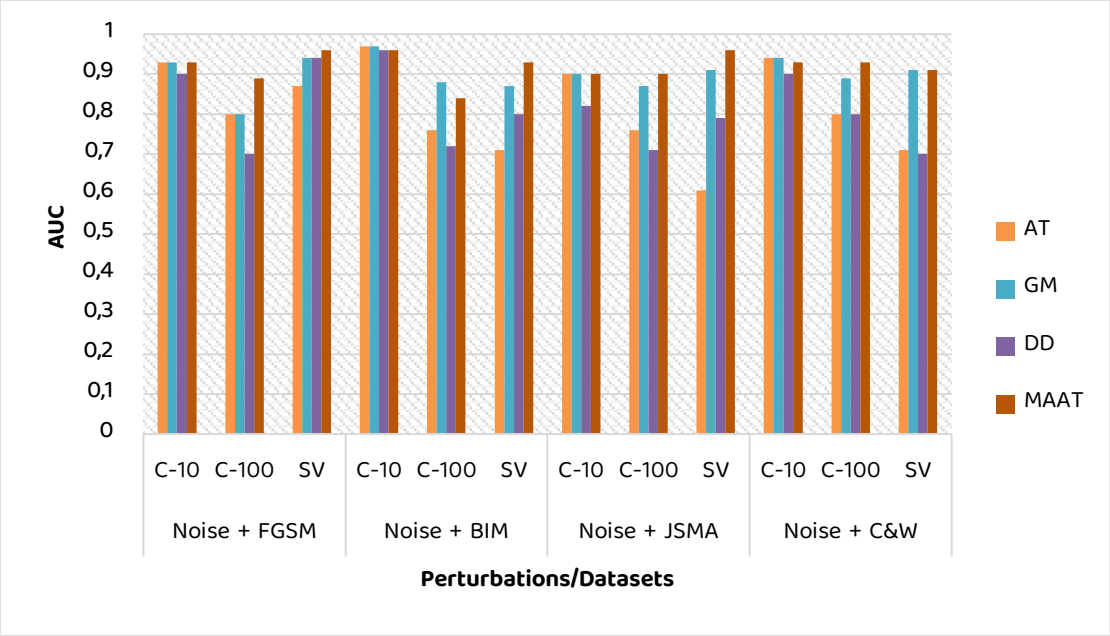

Figure 3: Comparison of Performance on Mixed Perturbations

3.2. Discussions

The experimental results presented above gives a complete overview of the behaviour of the base model in the presence of noise and adversarial examples and the robustness of MAAT viz-a-viz state-of-the-art techniques against different adversarial attacks. The results indicate that without appropriate defense mechanisms, the model suffers significant performance degradation in the presence of either Gaussian noise or adversarial examples or both. Having the performance of a model degrade averaging by

16% in the presence of noise and up to 33% in the presence of adversarial examples as shown Figure 1, and further in average terms up to 36% in the presence of both noise and adversarial examples in a test dataset as shown in Figure 2, confirms the dire impact of perturbed examples in model performance. It also, highlights the urgent need for improved detection techniques not only for adversarial examples, which has been the concentration in previous studies but techniques that have the ability to detect both noise and adversarial examples when they either exist in isolation or co-exist in a dataset. Another aspect of the results worth highlighting is the performance of the various adversarial detection techniques across the datasets and attacking schemes viz-a-viz the performance of MAAT. As illustrated in Figure 3, AT and GM almost drew parity with MAAT on the CIFAR-10 dataset across all noise and adversarial technique combinations with the exception of DD that recorded relatively lower AUC measures. These results suggest that no detection technique is inherently superior across all datasets, but the characteristics of a dataset has the tendency of influencing the impact of adversaries on base models and how well detection techniques can detect adversaries generated from the data using the schemes considered. However, based on the results, it can be firmly concluded that MAAT has the ability to scale well under varied data conditions and complexities than the existing techniques given the test cases considered. In practical terms, the AUC values indicate that, a base model that would have suffered performance degradation up to 36% when attacked by injecting noise and adversarial instances is able to, particularly, in the case of MAAT, flag these perturbed instances and maintained its performance almost at the same level when not under an attack. This points to model stability and resilience in real-world applications.

To confirm that the differences in AUC values across the various detection techniques as observed in Table 6 are indeed significant, an ANOVA analysis at a p-value of 0.05 was conducted. The outcome as shown in Table 7 confirms that there are significant differences in the measures since the recorded p-value is 0.001 (less than 0.05). Furthermore, a Turkey HSD test was carried out to identify the particular detection technique(s) that produced results that are significantly better than the others. From the HSD test results shown in Table 8, it is confirmed that the results obtained using MAAT are significantly better than those obtained when AT and DD are used. It is however, not significantly better than those obtained using GM though higher AUC measures were recorded across the test cases.

Table 7. ANOVA Analysis

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.117273	3	0.039091	6.438943	0.001037	2.816466
Within Groups	0.267125	44	0.006071			
Total	0.384398	47				

Table 8. Tukey HSD Test Results

Treatments Pair	Tukey HSD Q Statistic	Tukey HSD p-Value	Tukey HSD Inference
AT vs GM	3.8902	0.041291	* p<0.05
AT vs DD	0.0741	0.899995	Insignificant
AT vs MAAT	4.7423	0.008649	** p<0.01
GM vs DD	3.9643	0.036364	* p<0.05
GM vs MAAT	0.8521	0.899995	insignificant
DD vs MAAT	4.8164	0.007475	** p<0.01

4. CONCLUSION

In this paper, the task of detecting perturbed (crafted adversarial and noisy) examples from normal examples to improve adversarial attack defense in a LeNet CovNet model is addressed. This is achieved by proposing a Mixed Adversarial Awareness Technique (MAAT) that couples the kernel density and a variant of the Bayesian Uncertainty Estimator driven by the Dirichlet process and then incorporated between the last hidden layer and the output layer of the model to detect noisy and adversarial examples. The technique was tested using four adversarial attacks (FGSM, BIM, JSMA, and C&W) on three benchmarked datasets (CIFAR-10, CIFAR-100, and SVHN). The experimental results showed that co-existence of noise and adversarial examples generally pose a serious threat to a model's performance. The results confirms that the proposed technique presents a strong measure for the detection of adversarial and noisy examples within the test cases considered when compared to three (3) existing techniques (AT, GM and DD). Despite the improved performance recorded by the proposed technique, there is a need for further evaluation (ablation studies) focusing extensively on how different parameter settings affect the performance of MAAT when used with different neural network models with varying architectures and on datasets with different complexities. Key potential

limitations of the proposed technique that requires further exploration are need for an optimal technique for the determination of optimal thresholds and the computational

REFERENCES

[1] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: representing model uncertainty in deep learning," in *Proceedings of the 33rd International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 1050–1059.

[2] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2017, pp. 39–57.

[3] A. Mądry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *stat*, vol. 1050, no. 9, 2017.

[4] S. Kiani, S. Awan, C. Lan, F. Li, and B. Luo, "Two souls in an adversarial image: Towards universal adversarial example detection using multi-view inconsistency," in *Proceedings of the 37th Annual Computer Security Applications Conference*, ser. ACSAC '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 31–44, doi: 10.1145/3485832.3485904

[5] G. Liu, I. Khalil, and A. Khreishah, "Using single-step adversarial training to defend iterative adversarial examples," in *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '21. New York, NY, USA: Association for Computing Machinery, 2021, doi: 10.1145/3422337.3447841.

[6] G. Tao, W. Sun, T. Han, C. Fang, and X. Zhang, "Ruler: discriminative and iterative adversarial training for deep neural network fairness," in *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2022. New York, NY, USA: Association for Computing Machinery, 2022, p. 1173–1184, doi: 10.1145/3540250.3549169.

[7] R. Yang, Q. Sun, H. Cao, C. Shen, J. Cai, and D. Rong, "1+1>2: A dual-function defense framework for adversarial example mitigation," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 4121–4136, 2025.

[8] S. Y. Khamaiseh, D. Bagagem, A. S. Al-Alaj, M. Mancino, and H. W. Alomari, "Adversarial deep learning: A survey on adversarial attacks and defense mechanisms on image classification," *IEEE Access*, vol. 10, pp. 102 266–102 291, 2022.

[9] Y. Gao, Z. Lin, Y. Yang, J. Sang, X. Yang, and C. Xu, "Staying in the cat-and-mouse game: Towards black-box adversarial example detection," in *Proceedings of the 2nd International Workshop on Deep Multimodal Generation and Retrieval*, ser. MMGR '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 35–43, doi: 10.1145/3689091.3690090

[10] J. Zhao, S. Qiao, J. Wang, and G. Liu, "Generating image adversarial example by modifying jpeg stream," in *Proceedings of the International Conference on Computer Vision and Deep Learning*, ser. CVDL '24. New York, NY, USA: Association for Computing Machinery, 2024, doi: 10.1145/3653804.3654719

[11] J. Tian, C. Shen, B. Wang, X. Xia, M. Zhang, C. Lin, and Q. Li, "Lesson: Multi-label adversarial false data injection attack for deep learning locational detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, pp. 4418–4432, 2024.

[12] H. Kuang, H. Liu, X. Lin, and R. Ji, "Defense against adversarial attacks using topology aligning adversarial training," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 3659–3673, 2024.

[13] Y. L. Khaleel, M. A. Habeeb, and H. Alnabulsi, "Adversarial attacks in machine learning: Key insights and defense approaches," *Applied Data Science and Analysis*, 2024.

[14] S.-H. Choi, J.-M. Shin, P. Liu, and Y.-H. Choi, "Argan: Adversarially robust generative adversarial networks for deep neural networks against adversarial examples," *IEEE Access*, vol. 10, pp. 33 602–33 615, 2022.

[15] X. Yuan, Z. Zhang, X. Wang, and L. Wu, "Semantic-aware adversarial training for reliable deep hashing retrieval," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4681–4694, 2023.

[16] Y. Wang, T. Sun, S. Li, X. Yuan, W. Ni, E. Hossain, and H. Vincent Poor, "Adversarial attacks and defenses in machine learning-empowered communication systems and networks: A contemporary survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 2245–2298, 2023.

[17] X. Yue, M. Ningping, Q. Wang, and L. Zhao, "Revisiting adversarial robustness distillation from the perspective of robust fairness," *Advances in Neural Information Processing Systems*, vol. 36, pp. 30390–30401, 2023.

[18] L. Lu, S. Pang, X. Zheng, X. Gu, A. Du, Y. Liu, and Y. Zhou, "Ciard: Cyclic iterative adversarial robustness distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2025, pp. 350–359.

[19] C. Finlay and A. M. Oberman, "Scaleable input gradient regularization for adversarial robustness," *Machine Learning with Applications*, vol. 3, p. 100017, 2021.

[20] D. Sen, "Gradient Maximization Regularization for Signed Adversarial Attacks," *2023 14th International Conference on Electrical and Electronics Engineering (ELECO)*, Bursa, Turkiye, 2023, pp. 1-5, doi: 10.1109/ELECO60389.2023.10415930.

[21] C. Yu, T. Chen, Z. Gan, and J. Fan, "Spear: evaluate the adversarial robustness of compressed neural models," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI '24, 2024, doi: 10.24963/ijcai.2024/177

[22] I. Kraidia, A. Ghenai, and S. B. Belhaouari, "Defense against adversarial attacks: robust and efficient compressed optimized neural networks," *Scientific Reports*, vol. 14, 2024.

[23] Q. Liu and W. Wen, "Model compression hardens deep neural networks: A new perspective to prevent adversarial attacks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 3–14, 2023.

[24] S. Sharma, "Multi-sap adversarial defense for deep neural networks," *International Journal of Advanced Science Computing and Engineering*, vol. 4, no. 1, p. 32–47, Apr. 2022.

[25] A. Jordao and H. Pedrini, "On the effect of pruning on adversarial robustness," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. Los Alamitos, CA, USA: IEEE Computer Society, 2021, pp. 1–11.

[26] S. H. Zhong, Z. You, J. Zhang, S. Zhao, Z. LeClaire, Z. Liu, D. Zha, V. Chaudhary, S. Xu, and X. Hu, "One less reason for filter-pruning: gaining free adversarial robustness with structured grouped kernel pruning," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2023.

[27] Y. Xu, B. Du, and L. Zhang, "Assessing the threat of adversarial examples on deep neural networks for remote sensing scene classification: Attacks and defenses," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 2, pp. 1604–1617, 2021.

[28] Y. Zhu, L. T. Yang, J. Feng, and X. Xie, "Tensor-based gan to defense adversarial attacks for cyber-physical-social system," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021, doi: 10.1109/TNSE.2021.3077305.

[29] S. A. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 4, pp. 325-327, April 1976, doi: 10.1109/TSMC.1976.5408784.

[30] P. Harder, F.-J. Pfreundt, M. Keuper, and J. Keuper, "Spectraldefense: Detecting adversarial attacks on cnns in the fourier domain," *2021 International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9533442..

[31] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.