Fortifying Brain Signals for Robust Interpretation

Kanan Wahengbam, Kshetrimayum Linthoinganbi Devi, and Aheibam Dinamani Singh

***Abstract*—Brain-Media, is the discipline of decoding sophisticated human brain activity such as imagination, memories, colours, textures, patterns, etc. Existing efforts either classify brain signals or map them to an image of the same class. The second technique has only been investigated in a few papers using Electro Encephalography (EEG) datasets based on ImageNet and MNIST images. The role of Deep Neural Networks (DNN) must be researched to make them robust against malicious noise. Existing frameworks ignore the existence of such disruptive noises. The current research provides a multimodality time-series and spatial-domain hybrid framework and a unique ResilientNet Generator to classify signals robustly. In the first stage, two teachers are trained using a time series and an image dataset that shares the class. The second phase trains a ResilientNet-Generator with a new penalized-reconstruction loss-function apart from the adversarial loss. Finally, the trained ResilientNet-Generator is used as a pre-processing module for training a DNN-classifier in the third phase. It is noted that representing time-series data by the spatial domain can significantly improve accuracy compared to existing approaches. An ablation study on the resilience of trained classifiers against attacked test samples shows that DNNs can be fortified using the proposed framework.**

***Index Terms*—Brain Media, Deep Learning, EEG, Multimodality learning, Signal disruption.**

I. Introduction

THE ability to read the human mind has been a hot topic in recent decades. Brain Computer Interface (BCI) has piqued the interest of scientists in a variety of domains, including neurology, neuroimaging, therapeutic healthcare, and so on [1], [2]. BCI is a technology that establishes a direct contact between the human brain and a computer [1], [2]. Electro Encephalography (EEG) evoked potentials from carefully produced stimuli have been used in investigations to understand brainwave activity [3], [4]. Multivariate pattern categorization has used in EEG investigations to evaluate category selectivity. This method allows the analysis to be done at once rather than necessitating the pre-selection of spatial or temporal components of the brain [5], [6]. It can also provide a data-driven method of identifying spatial, temporal, and spectral components underlying category discrimination Multivariate EEG may interpret image categories such as faces, objects, and surroundings [7], [8]. [15], [16], [17] presented pattern recognition approaches to address the challenge of multimedia-evoked brain cognition.

They are all based on real-world observations. As such, they are all external sources like text, image, music. The authors of [9] proposed a new path known as "Brain Media," in which unseen media such as dreams, ambitions, and emotional experiences are represented by reproducing the natural RGB image. Working with EEG data is fraught with challenges. The signal-to-noise ratio of EEG recordings is very low and noise arise from various sources. Excessive activity, eye movements, and blinking are further causes of unwanted electrical noise. Typically, only specific forms of brain activity are relevant. The signal should be separated from background activities. EEG on the scalp has weak spatial resolution, with extra spatial blurring caused by skull. Nonetheless, it offers a great temporal resolution for capturing slowly changing and fast-evolving brainwave activity patterns. Deep learning (DL) techniques might be useful in this situation. EEG data is high-dimensional and complicated. However, massive datasets are required to train deep networks. Convolutional Neural network (CNN) may address this issue of data scarcity by using a 2D spatial form of data-augmentation as discussed further in the current research.

Multi-modal DL techniques outperform unimodal systems. The current level of multi-modal machine learning assumes that all modalities are available, synchronized, and noise-free, during training and validation. Nevertheless, in real-world operations, more than one modality is typically missing, distorted, insufficiently annotated data, possess inaccurate classes, sparse in training or testing. Multi-modal co-learning is a helpful resource that can tackles this issue. MentorNet [10] oversees the student training process by offering a prototype weighting scheme to pick clean instances to guide training. Decoupling [11] trains two networks and updates them only when their outputs diverge. The discrepancy decreases as the classifier improves, and the proportion of noise changes. With high to low noisy labels, co-teaching [12] exhibited improved performance. Here, the two networks are trained at the same time, and in each mini-batch, each network considers its modest loss cases to be helpful. On the other hand, the approach permits each network to teach others in each mini-batch, allowing for error to flow between them.

Another growing issue that actively undermines Deep Learning models is the introduction of intentional noise into a signal/image high-dimensional image's manifold. It was first proposed and effectively proven in an image classification task [13]. It drastically degraded the predictive ability of numerous cutting-edge deep learning models. Since then, adversarial perturbation has

Manuscript received 26 June 2022; revised 2 October 2022; accepted 11 November 2022. Date of publication 15 November 2022; date of current version 23 February 2023. Recommended for acceptance by Prof. Lanjing Zhang. *(Corresponding author: Kanan Wahengbam.)*

Kanan Wahengbam is with the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati 781039, India (e-mail: wahengbam.kanankumar@gmail.com).

Kshetrimayum Linthoinganbi Devi and Aheibam Dinamani Singh are with the Department of Electronics and Communication Engineering, National Institute of Technology Manipur, Imphal 795004, India (e-mail: babythoi22@gmail.com; ads@nitmanipur.ac.in).

Digital Object Identifier 10.1109/TNSE.2022.3222362

mostly been applied to image-based approaches such as image classification, semantic segmentation, object identification, and so on. However, defense measures have also been developed, including *"adversarial training"* [14]. Following the realisation of this vulnerability, its implementation in sectors such as remote sensing, autonomous vehicles, biomedical imaging, facial recognition, and so on has just begun. However, the discussion of adversarial disruption in time-series domains such as "Brain-Media" are still in their infancy. It is expected to be an important area of research, especially as DL is of prime importance in the recognition of time-series signals.

A. Related Works

Minimally invasive methods like Functional magnetic resonance imaging (fMRI), EEG, and Magnetoencephalography (MEG) have been used to decipher human thought processes linked to vision in cognitive science and imaging research [15], [16], [17]. Among the neuroimaging methods available, EEG has various attributes, making them especially suited for this investigation. EEG is a low-cost technology with superior high resolution than MRI and fMRI. DL approaches have been utilized to interpret EEG data. Using the Bonn University EEG datasets, investigators of [18] provided a 13-layered DNN training on unprocessed signals enabling seizure detection, achieving an efficiency of 88.67%. [19] developed an attention technique for automatically identifying features of every channel and investigated a complex DNN for learning an unadulterated seizure-sensitive interpretation using signals via adversarial learning. [20] used a discriminative feature weighting integration methodology to generate tightly packed CNNs methodology for seizure diagnosis identification and epileptic parietal state classification. According to a recent study, EEG data obtained from human participants watching pictures from ImageNet [21] can be used to build a computer-vision model using the learned classifier. ImageNet images are utilized as stimuli for individual subjects undergoing EEG testing. To predict the stimuli category using the recorded signal, a long-short term memory (LSTM) comprising a fully linked layer and a ReLU layer is trained. [22] worked on EEG data reconstruction and categorization. [23] explored the impact of CNN on decoding and visualising EEGs. The authors evaluated many CNNs on EEG deciphering activities and discovered that DL features like exponential linear units and BN seem to be critical for obtaining high deciphering precision. Both CNN and Recurrent Neural Network (RNN) can effectively address EEG supervised classification. This is previously studied for visualizing a person's neural functioning while performing visual tasks [16], [24], [25]. The author used fMRI pictures in [16], [24] to highlight the visual stimuli formed in brain signals while a person observes a movie clip. Furthermore, the researchers of [25], [26] proposed utilizing EEG recordings to demonstrate the effects of visual stimulation on activity in the brain. [27] discussed combining RNN and CNN to collect EEG features for categorization. [28] constructed a classifier that could distinguish EEG brain waves caused by 12 distinct object categories having an accuracy of 29%, which is significantly lower than SOTA performance. [29] used an RNN to concurrently input video attributes and EEG signals to accomplish video classification depending on client liking. [30] introduced ThoughtViz, an EEG-based deep learning approach for picturing humans using a GAN-based network. [25] provided an autonomous vision recognition framework that increased the efficiency of the 40-class prediction to 21.8%. They used a siamese network to investigate the internal association between EEG and pictures, raising the efficiency of the 40-category classification job from 21.8% to 48.1%. [31] introduced a multiclass EEG signal identification approach based on enhanced Common Spatial Pattern (CSP) features with DNN with up to 69.27% accuracy. [32] developed three modality-dependent encoders to collect low-level features. HyperdenseNet [33] proposed a double DNN featuring links for various MRI modes. [34] combines final characteristics from modality-dependent routes to get inferences. MMFNet [35] has a more complicated framework for fusing multivariate MRI images. Researchers have also presented various cross-modality data analysis strategies [36], [37], [38]. [37] advocated utilizing a competitive multiple feature autoencoding model to accomplish a reciprocal predictive model of many modalities.

Some limitations observed in the existing works are discussed below.

i. Few researchers used very deep CNNs in Brain-media EEG signal categorization. Training a highly deep CNN from scratch for EEG signals necessitates a substantial amount of classful EEG data. Signals are also difficult to identify manually. Furthermore, a minimal volume of data is likely to result in underfitting. As a result of the poor SNR of the EEG and individual variances, EEG data are scant.

ii. These approaches are inflexible in controlling the semantic content of the inferred modalities. For example, the accuracies of the SOTA inception score (IS) produced by [26] and [39] are only 5.07% and 82.9%, correspondingly offering substantial room for additional study and advancement.

iii. These approaches are insufficiently versatile to accommodate numerous modalities. For reciprocal mapping across two modalities, most available approaches require learning two translators travelling in opposing directions. Therefore, for ramping up to $n(n-1)$ modalities, $n(n-1)$ interpreters are required. It becomes difficult as the number of modalities rises. Adjusting the magnitude of the bottleneck allows one to regulate the features' complexity and contraction rate deliberately. On the other hand, a tiny bottleneck adds to the sophistication of the decoder's operation and risks a greater distortion rate. This barter encourages the model to maintain just the needed changes in the source data to rebuild the input while avoiding redundancy and distortion across the input.

iv. None of the existing methods discussed the emerging issue of adversarial attacks on a clean signal which causes a fully trained DNN model to completely collapse. It is a critical issue because such noise are crafted according to the gradients target classifier.

B. Contribution.

The contribution of the current work are as follows.

- A multimodality framework that predicts a multimedia-based Brain time-series signal precisely by harnessing the

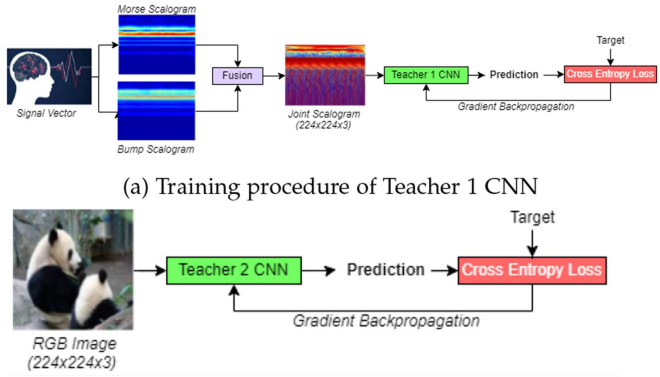
potential of a Joint-Scalogram based Spatial domain with a robust CNN framework is proposed. After studying its robustness to adversarial attacks on brain signals, it is named 'as ResilientNet.'

- Existing methods related for EEG data-augmentation and spatial transformation used either Analytical Morlet or Bump or Morse wavelets based scalogram. For instance, researchers of [40] demonstrated good success rate by training multiple DNNs with Amor, Bump and Morse scalograms and fusing the softmax predictions. However, use of multiple such datasets took a large amount of training time. To this end, we propose a method to utilize the diverse information capturing capability of the Morse and Bump time-frequency scalogram into a scalogram known as Joint-Scalogram (JS). It is a data-augmentation process that maps diverse features of different scalograms into a single spatial high-dimensional manifold which saves computing power that would have otherwise taken to train multiple times to get the same amount of features.
- A ResilientNet-based Generator that avoids redundant features in the bottleneck latent feature is proposed. A new loss function that uses perceptual loss and correlation loss is also proposed to provide a good reconstruction. This loss function encourages the Generator learn pair-wise correlations between the features.
- A fully-trained ResilientNet-Generator is proposed to be used as a plug-and-play module during both the training and testing phases. Training allows a classifier to learn the reconstructed version of an input signal. During testing, it acts as a pre-processing unit that filters an incoming signal to prohibit it from disrupting the prediction capability of the trained CNN classifier. This means that the CNN learns the true structure of a reconstructed JS during training, and it is protected from malicious features, thereby enhancing protection over two stages.
- Quantitative evaluation is carried out in standard conditions where test images are the original signals and under adversarial conditions where test signals are associated with malicious features. The Ablation study provides an in-depth study into the robustness that the framework can provide even in the event of distorted signals. The extensive evaluation shows that the proposed method provides significant improvement and robustness.

II. Proposed Methodology.

This paper proposes a multimodality time-series, spatial-domain hybrid framework for determining the right label of a brain-media input. One of the important attribute in this work is the use of 2D spatial version (i.e. JS) of the input as a data-augmentation process without actually increasing the number of samples.

The main objective is to provide a plug-and-play ResilientNet-Generator that can be used to train and test CNN classifiers. Three distinct training periods are undertaken to achieve this goal. A) Two identical CNNs are trained independently by using JS and RGB image datasets. They are named as Teacher-1 and Teacher-2. Following training, these two networks will serve as teachers. They will jointly oversee the Generator's training via backpropagation in the second stage. B) A Generative Adversarial Network (GAN) is trained using the JS dataset, Jensen-Shannon loss, and adversarial loss. The Generator receives a JS as input, resulting in a fake JS. By comparing the generated JS to the original JS input, the Discriminator determines if it is authentic or fake. The design of the Generator, as well as the non-intrusive supplemental gradient backpropagation with Jensen-Shannon Divergence loss, are the main characteristics offered in the second stage. The Generator has been specifically engineered to tolerate modest and precisely constructed attacks. This is accomplished by employing a very deep layered network as the encoder's base and a unique loss function aiming at improving feature connection while lowering reconstruction loss and inter-feature loss to avoid redundancy. These characteristics will help the Generator to train with dark features for a longer period. C) The final stage involves training a CNN classifier with reconstructed JS generated by the trained ResilientNet-Generator with cross-entropy loss in the standard way. The following are the main advantages of this process. (i) Signal denoising to avoid duplicate characteristics from interfering with categorization. (ii) The GAN module uses a spatial domain version of the brain signal to augment W the number of features by using the capabilities of CNN. (iii) Gradient backpropagation to the Generator enriches the dark features and allows them to train for a longer period of time. The two sets of soft labels are joined with a Jensen-Shannon divergence loss rather than integrating the features created by applying transfer learning to the two teachers. (iv) And, strengthening the CNN classifier to be resistant to malevolent noise artifacts.



(a) Training procedure of Teacher 1 CNN

(b) Training procedure of Teacher 2 CNN

Fig. 1. Phase 1 training: Isolated training of Teacher Networks.

A. Teacher Training Phase

Fig. 1(a), (b) shows the overview of the teacher training phase. The EfficientNetB7 [41] CNN is used as the teachers. EfficientNet is a state-of-the-art (SOTA) model that recently provided the highest classification rate on the ImageNet dataset classification problem. It has 8 different models named as EfficientNet B0 to B7, where B7 has the highest number of models with 813 number of layers in total. Teacher 1 is trained by using a JS image dataset while Teacher 2 is trained with the

same classes as Teacher 1 but with RGB images obtained from the ImageNet dataset. First, we describe the mathematical construct of a JS as follows.

Joint Scalogram It is a wavelet. Its magnitude is squared due to the fusion of Morse and Bump (M-B) time-frequency scalogram of an input speech signal. It captures the diverse patterns of two different scalograms of a brain signal into a single spatial domain high-dimensional manifold. It takes away the need and time taken to train the classifier multiple times with varying datasets of scalogram. It distinctly brings out regions with common power levels and unravels the phase correlation between them. A phase delay that is uniform and slowly varying will be observed in regions when the M-B scalogram has common attributes. It can be visualized more as a localized correlation between two different continuous wavelet Transfers (CWT). This portrays a local phase interlocking pattern, resulting in a new fused scalogram. If $f(t)$ and ψ are the input signal and mother wavelet respectively. Then the CWT can be expressed as:-

$$CWT(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t) * \psi\left(\frac{t-b}{a}\right) dt, \tag{1}$$

where, $a \in R^+ - 0$, $b \in R$. The cumulative addition of time-scaled ψ and $f(t)$ produces the CWT.

$$CWT(scale, position) = \int_{-\infty}^{+\infty} f(t) * \psi(scale, position, t) dt. \tag{2}$$

Morse Wavelet's Fourier Transform is given by -

$$\psi_{P,\gamma}(\omega) = \coprod(\omega) a_{P,\gamma} \omega^{\frac{P^2}{\gamma}} e^{-\omega^{\gamma}}, \tag{3}$$

where, $\coprod(\omega)$ denotes the unit-step function, P^2 is the time-bandwidth product, $a_{P,\gamma}$ is the normalization constant and γ is the symmetry parameter. Morse wavelets are created using various combinations of P^2 and γ. It is used with the CWT's coefficients from (2) to generate the Morse Scalogram. This coefficient is also combined with the bump wavelet to create the bump scalogram. A bump wavelet's Fourier transformation is given by

$$\psi(s\omega) = e^{\left(1 - \frac{1}{1-(s\omega-\mu)^2/\sigma^2}\right)} 1_{\left[\frac{(\mu-\sigma)}{s}, \frac{(\mu+\sigma)}{s}\right]}, \tag{4}$$

where, μ and σ is responsible for localizing the time and frequency of the transformed signal. The transformation into the time-frequency domain allows signal to propagate efficiently and more rapidly with frequency and phase. The scalograms formed by using (3) and (4) is now fused by using the following relation.

$$Y = \frac{\left|S(C^*_{\psi_{P,\gamma}(\omega)}(a,b) C_{\psi(s\omega)}(a,b))\right|^2}{S(|C_{\psi_{P,\gamma}(\omega)}(a,b)|^2) S(|C_{\psi(s\omega)}(a,b)|^2)}, \tag{5}$$

where, $C_{\psi_{P,\gamma}(\omega)}(a,b)$ and $C_{\psi(s\omega)}(a,b)$ are the Continous Wavelet Transformation of $\psi_{P,\gamma}(\omega)$ and $\psi(s\omega)$ respectively with a as scale and b as position. S is the time-scaling and smoothing operation. $*$ is a complex conjugate. JS may either be real of complex in nature as per the type of time series used. Since JS is constructed purely by using the correlated signal components of the two signals, it has a very high probability of cancelling out noise artifacts.

All of the signals are converted to JS spatial format and scaled to 224×224×3. The image samples with identical class signal samples are acquired for training the teacher no. 2 network. The MindEEG-ImageNet[1] and ImageNet [21] image classification datasets are used in this research. A parallel dataset is created with the same number of classes and labels. One dataset has JS samples that are a direct descendent of brain signals. The second dataset comprises RGB images that correlate to a graphical representation of brain activity. For example, if there are three classes of EEG signal samples, Aeroplane, Bird, and Panda, the RGB image samples for these classes are obtained from the ImageNet image dataset. More details about the dataset are discussed in Section III. In each iteration, the gradient for L_{CE} is backpropagated to update the teacher networks.

$$L_{CE} = -log(\frac{exp(z[y])}{\sum_i exp(z[i])}), \tag{6}$$

where z is the classifier's output logits, and y is the ground truth target. The i^{th} element of z is represented as $z[i]$. The gradient for L_{CE} in each iteration is allow to back-propagate to update the parameter for each classifier.

B. Adversarial Training Phase

The teachers can produce softlabels or scalar prediction score vector when tested against training samples. The idea for co-teaching with two teachers is to have them provide dissimilar probability distributions. The Kullback-Leibler (KL) divergence, in general, quantifies the overall distinction between different probabilistic weightings. Therefore, in contrast to the KL divergence or CE loss, the Jansen-Shannon divergence is chosen because it is symmetrical and an inherently unbiased over such a co-teaching system. A JS sample is obtained from the JS dataset, and an RGB images is obtained from the image dataset. We execute joint training in the GAN network to detect whether the input signal corresponds to a genuine label, and hence one image from the image dataset is picked at random for each JS sample. The only condition is that the two input data modalities must belong to the same class. The GAN is biased during training to learn to reconstruct a fake JS image. As illustrated in Fig. 2(a), the Siamese networks of teachers 1 and 2 yield two probability distributions. The JS divergence is calculated using two distributions. The gradients are backpropagated to the generator. As a result, the loss function calculated from the normalized logits after the softmax operation is as follows.

[1] http://www.mindbigdata.com/

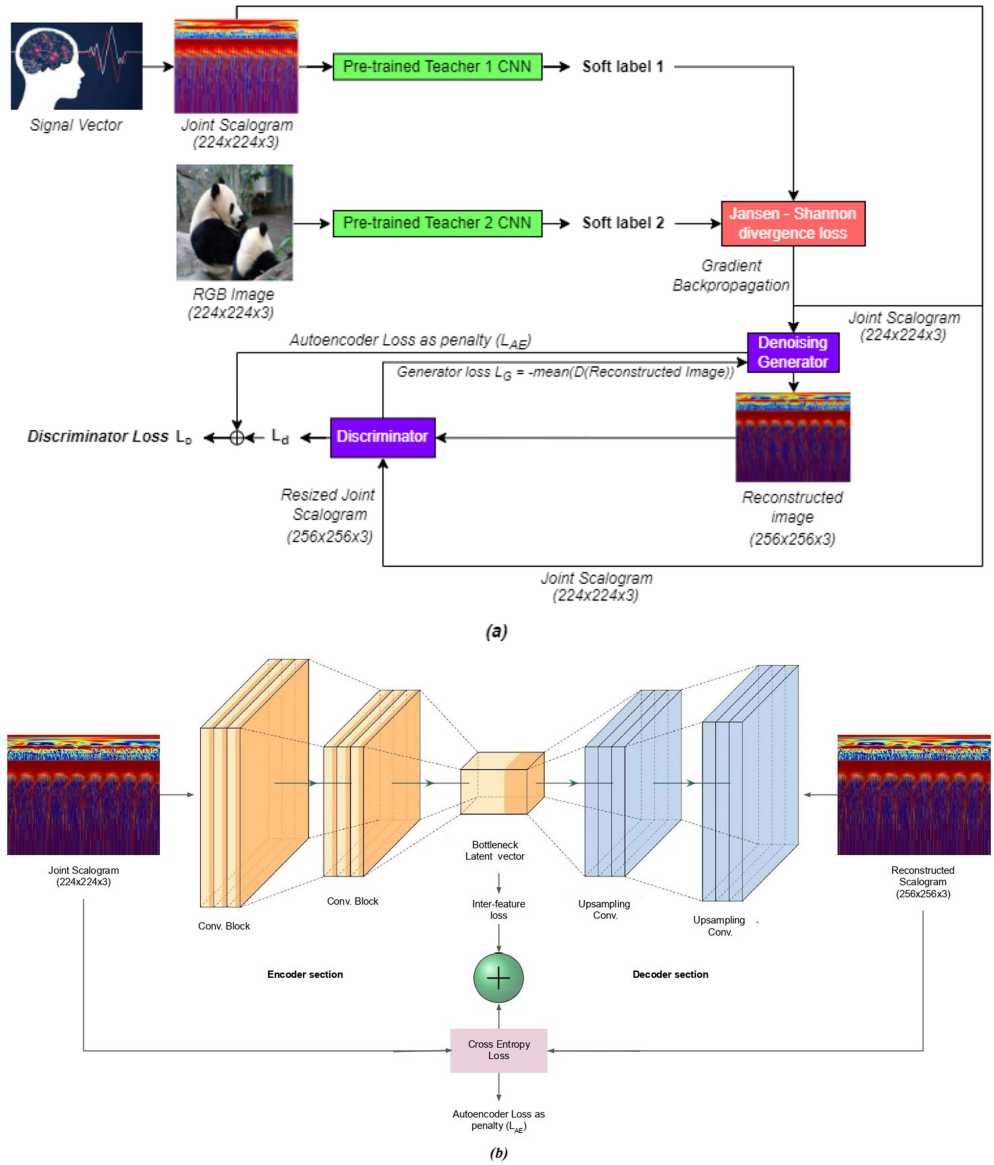


Fig. 2. Phase 2 training: (a) Adversarial training pipeline. (b) The internal structure of the Denoising Generator.

$$
\begin{aligned}
L_1 &= JS(C_0(x)||C_1(x)) \\
&= \frac{1}{2} KL\left(C_0(x)||\frac{C_0(x)+C_1(x)}{2}\right) \\
&\quad + \frac{1}{2} KL\left(C_1(x)||\frac{C_0(x)+C_1(x)}{2}\right),
\end{aligned} \tag{7}
$$

where, x denoes the normalized scores generated by applying softmax normalization. And $C_0(x)$ and $C_1(x)$ are the classifiers that provide the soft labels.

Then, using the JS dataset, the Generator, and Discriminator networks are trained, and the loss (L_1) is obtained from the teacher. The Generator attempts to transform the input JS into a filtered version. Next, the discriminator evaluates if the generator has created a genuine or fake example. The Generator is designed to represent the general distribution of training data and create realistic-looking samples. So, the discriminator will frequently be unclear about the genuiness of the obtained JS. Therefore, the Generator will strive to enhance its output by reducing ambiguity in brain signals and optimizing signal potentials by accounting for the teacher's logits.

The Vanilla GAN [42] is extremely unstable, and Wasserstein GANs [43] are also susceptible to model collapse. BigGAN [44] demands a substantial amount of computing power. To guarantee that the fake instances are of high quality and robustness, the original image's high-dimensional manifold must be preserved while enhancing class-specific characteristics and decreasing noise. One option is to use real images as input to a generator equipped with a denoising Autoencoder structure. Such generator is capable of extracting numerous features while minimizing reconstruction loss and inter-feature loss in the bottleneck. Fig. 2 shows the proposed ResilientNet-Generator's architecture. It has an hourglass design and three components: encoder, bottleneck, and decoder. The baseline for the encoder is EfficientNetb0 [41]. EfficientNetb0 contains a total of 290 layers. The JS input has a dimension of

TABLE I
LAYERS OF THE DECODER. WHERE, K=KERNEL SIZE, S=STRIDE, P=PADDING, F=FILTERS

Layer No.	Layer	Parameter	Actual size
1	TC-1	K=4, S=1,P=0,F=256	4×4×256
2	TC-2	K=4,S=2,P=1,F=128	8×8×128
3	TC-3	K=4, S=2,P=1,F=64	16×16×64
4	TC-4	K=4, S=2,P=1,F=32	32×32×32
5	TC-5	K=4, S=2,P=1,F=16	64×64×16
6	TC-6	K=4, S=2,P=1,F=8	128×128×8
7	TC-7	K=4, S=2,P=1,F=3	256×256×3

Where, K=Kernel size, S=Stride, P=Padding, F=Filters.

224×224×3. A fully connected layer after the EfficientNetb0 provides a 1280×1×1 bottleneck feature vector. The bottleneck is highly compressed and contains all of the rich data from the high-dimensional JS input. A decoder network with seven transposed convolutional layers (TC) is used for upsampling the bottleneck. The real structure of the TC is shown in Table I. It creates a reconstructed image with a dimension of 256×256×3.

Image denoising refers to the process of recovering a clean image from its noisy predecessor. CNN-based autoencoders have frequently been used to accomplish this task due to their plug-and-play network designs [45], [46]. Using noisy specimens as input and selecting the clean copy as the output objective, the model learns to retain just the critical information from the image while eliminating the noise. Autoencoders do this task by reducing the reconstruction loss across the training samples. It leads to a compact 'on-redundant' representation of the data. The reconstruction loss is further reduced in the current study by minimising the inter-feature loss in the bottleneck. For a given training data, $\{x_i\}_{i=1}^{N}$ and an encoder $g(1) \in R^D$, the correlation between the i^{th} and j^{th} features, g_i and g_j can be expressed as

$$c(g_i, g_j) = \frac{1}{N}\sum_n (g_i(x_n) - \mu_i)(g_j(x_n) - \mu j), \tag{8}$$

where, $\mu_i = \frac{1}{N}\sum_n (g_i(x_n))$ is the average output of the i^{th} neuron. This minimizes the pairwise covariance between different features. Therefore, the standard loss $L(\{x_i\}_{i=1}^{N})$ is now updated as

$$\begin{aligned} L(\{x_i\}_{i=1}^{N})_{AE} &\triangleq L(\{x_i\}_{i=1}^{N}) + \alpha \sum_{i\neq j} c(g_i, g_j) \\ &= \frac{1}{N}\sum_{i=1}^{N} D(x_i, f \circ g(x_i)) \\ &\quad + \alpha \sum_{i\neq j} \left(\frac{1}{N}\sum_n (g_i(x_n) - \mu_i)(g_i(x_n) - \mu_j) \right), \end{aligned} \tag{9}$$

where, α is a hyperparameter used to control the contribution of the additional term to the total loss of the model. The first term in (9) is the conventional autoencoder loss. It depends on both the encoder and decoder sections. The second term is solely dependent on the encoder. It aims to increase the variety of the learnt feature while ensuring that the encoder learns less associated non-redundant features.

The proposed loss component acts as an unsupervised regularization term on top of the encoder. It gives additional input throughout stochastic gradient descent to lessen the correlation of the encoder's output. It may be integrated as a plug-in into any autoencoder-based model and optimized in batch mode. The discriminator is updated according to L_d is given as

$$L_d = -E_{x\sim P_{data}}[\log D(x) + \log(1 - D(G(x))], \tag{10}$$

where, x is the original image, D and G are the generator and discriminator. The gradient with respect to adversarial loss L_{adv} is backpropagated to the generator after updating the discriminator as

$$L_{adv} = E_{x\sim P_{data}}[\log(1 - D(G(x))]. \tag{11}$$

It is seen from the above discussion and also from Fig. 2(a) that there are four losses. L_1 is a loss that is backpropagated to the generator. L_{AE} is a combination of Reconstruction loss and with inter-feature loss. L_d and L_{adv} are the backpropagated discriminator and generator losses. The L_d and L_{AE} is now added to form a new loss as

$$L_D = L_d + L_{AE}. \tag{12}$$

The loss L_{AE} decreases as the Generator's performance improves. However, it will lengthen the training/convergence period.

C. Classifier Training Stage

The training model of a classifier is shown in Fig. 3. Initially, the pretrained Generator crafts a new JS image. As in the teacher training phase, the classifier trains by minimizing the cross-entropy loss between prediction and ground truth labels. The no. of classes for the MindBigData-ImageNet and MindBigData-MNIST experiments are 40 and 11 respectively. More details about the classes, training procedure and hyperparameters are discussed in Section III.

III. IMPLEMENTATION RESULTS AND DISCUSSION

The performance of the framework is evaluated and compared with state-of-the-art methods on two public datasets: (1) MindEEG-ImageNet,[2] and (2) MindEEG-MNIST.[3]

A. Experimentation With MindBigData-Imagenet and ImageNet Dataset

The

a) Description of the datasets: Work given in created publicly accessible EEG dataset for brain activity mapping and categorization. It is collected using a 128-channel cap with active, low-impedance electrodes. The signals were obtained by inviting individuals to view visual triggers and

[2] http://www.mindbigdata.com/opendb/imagenet.html
[3] http://www.mindbigdata.com/opendb/index.html

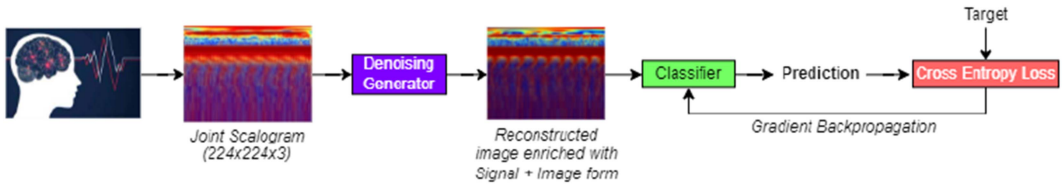


Fig. 3. Phase 3 Training: Isolated training of classifier with pre-trained Denoising generator.

images from the ImageNet database. It is a 40-class subset of the ImageNet dataset. Fifty visuals from ImageNet were displayed to six individuals for 0.5 seconds each for every 40 classes [39]. As a result, 2000 images were used in the experiment, and 12000 visually-evoked EEG data were collected. Consequently, only 11466 128-channel acceptable EEG sequences were retained. All the 11466 EEG samples are initially converted to the JS format. To generate the JS map, a signal is converted to the intermediate M-B scalogram by using a time-bandwidth product value equal to 60 and filter-bank symmetry parameter as 3, 4 signals. Here, 10 wavelet bandpass filters are used in each octave. The filters are normalized by setting a passband magnitude of 2. The value '3' provides zero skew and an ideal symmetric wavelet in the frequency domain for the two wavelets. The two scalograms have a matrix dimension of 31×16212. It shows that matrix multiplication and smoothing operations of (5) can be performed. Each JS has a dimension of 193×137160. As shown in Fig. 4(b), the arrows describe the phase relationship. The one without the arrow is the version used for training. The reddish color regions are the frequency range over which the two scalograms show similar attributes. It is observed that the two time-frequency images exhibit a similar oscillatory pattern near 1 kHz. Some regions exhibit a common pattern. Most arrows are pointed towards the left. It means 0 radians delay in the Morse Scalogram. In certain lower frequencies, phase delays of $\pi/4$ are also observed. This process results in the formation of 11466 JS images.

b) Implementation procedure: The implementation is divided into three stages: (a) teacher training, (b) adversarial training, and (c) classifier training.

• *Teacher training phase* - The teacher training stage is a multimodal stage that uses JS and RGB image datasets. The EfficientNet-b7 CNN structure is used for both teacher networks, namely - Teacher-1 and Teacher-2. For training Teacher-1, we acquire the 11466 JS images and split them up in the ratio of 80:20 or 9172:2294 training and validation sets respectively. And for training Teacher-2, we acquire a total of 2000 images from the ImageNet dataset in such a way that for the same 40 classes as the JS dataset, we gather 50 images for each classes. This exercise results in the production of 50 image $\times$ 40 classes = 2000 images. Using the same 80:20 split, we have 1600 and 400 training and validation sets respectively. Prior to training, all the JS and RGB images are downsampled to a dimension of $224\times224\times3$, which is a standard size of most CNNs. After having acquired the corresponding datasets, the two teacher networks are trained separately by using the hyperparameters, namely - learning rate 0.001, mini-batch size 24, maximum number of epochs

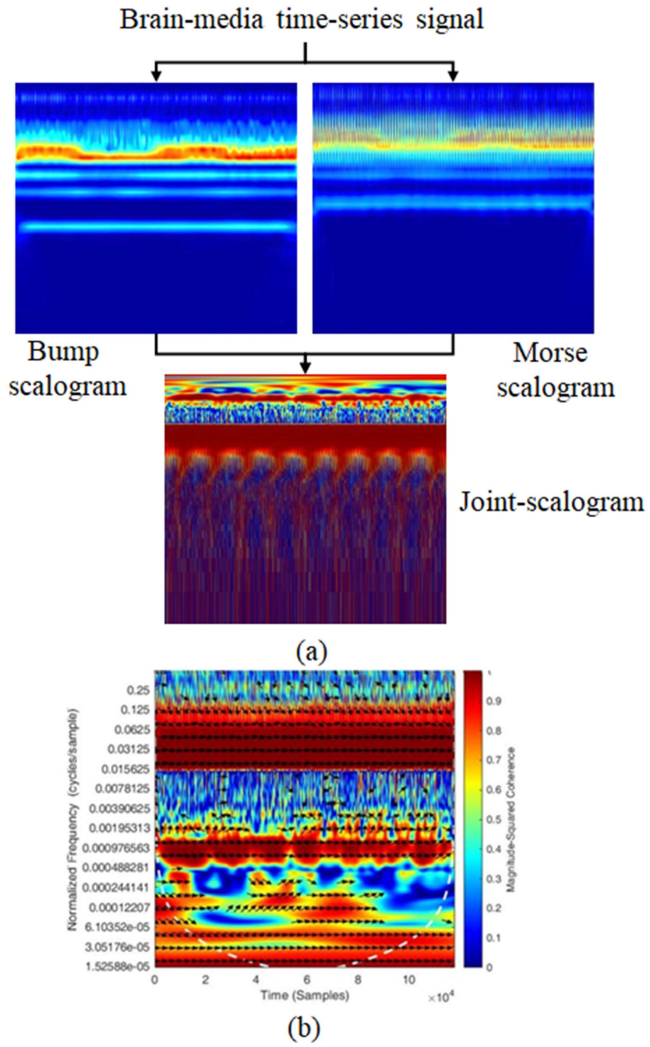


Fig. 4. Graphical depiction of a 2D JS spatial data. (a) formation of a Joint-scalogram, (b) phase relationship between the two scalograms.

40. During training, for both the teacher networks, we evaluate the cross-entropy loss between the target labels and the predicted labels. The Adam optimized is selected for both the training process. All implementations are done on our system bearing the following configurations: Intel(R) Xeon(R) CPU @ 2.30 GHz, Tesla P100-PCIe-16 GB GPU, 24 GB RAM, and Pytorch library. It was noted that Teacher-1 took a considerable amount of training time than its Teacher-2 counterpart. Teacher-1 and 2 converged at the 38^{th} and 24^{th} epoch respectively. Since both the training uses only 40 classes, the training accuracy stopped flattened at around 98%. However, our main motto is to utilize the two teachers for transfer learning in the adversarial training phase. Therefore, the teacher 1 and

teacher 2 CNNs are now fully trained with the same number of classes but with images having varied modalities. This multimodal training aims to impart intrinsic human-like perception in which images, when visualized, evoke an EEG sensation inside the brain. In the current work, a GAN performs this process by using the stimuli of both signals for training the generator. This aspect is observed in the adversarial training phase.

• *Adversarial training phase*- The next phase is the adversarial training stage, which involves the two pre-trained teachers, the Denoising Generator (G) and the Discriminator. The primary goal of this step is to fine-tune the Denoising Generator so that it can serve as an effective image reconstruction module. Importantly, it protects any target classifier from any intentional or unintentional manipulation of the input test data. Its main goal is to give resistance to white-box attacks. A white-box attack occurs when the enemy has complete knowledge of the CNN. In practise, an adversary may create a perturbation using white-box technique and inject it into the input test data, but the generator will denoise the image first. As a result, the target classifier will only get a purified test data.

The steps taken in one iteration of the Adversarial training are as follows. Initially, a JS and an RGB images are applied to the pre-trained Teacher-1 and Teacher-2 networks. Then, the softlabels or scalar probability vector are applied to the Jansen-Shannon divergence loss function to compute loss. This loss is used to update the teacher networks by decreasing their stochastic gradients. For teacher 1 and teacher 2, the stochastic gradients are $\nabla_{\theta_{t_1}} \phi(T_1(X_{JSreal}), y)$ and $\nabla_{\theta_{t_2}} \phi(T_2(X_{JSreal}), y)$ respectively. In the next step, the original JS image is passed on to the denoising Generator to yield a fake image $x_{fake} = G(x_{real})$, where, x_{JSreal} = original Joint scalogram image, G is the generator and x_{fake} is the generated fake image. Now from Fig. 2(a), we can also observe both the inputs to the Discriminator (D) are JS images. The only difference is that one of D's input is the original JS image that, and the other JS image is the reconstructed image provided by the Generator that has characteristics of RGB image as well. Now, D is updated by ascending its stochastic gradient $\nabla_D[\log D(x_{JSreal}) + log(1 - D(x_{JSfake})]$ by using the loss of (10). Next, G is updated twice in a row: a) firstly by descending its stochastic gradient $\nabla_G log(1 - D(x_{JSfake})$, b) and secondly with the Jensen-Shannon loss (L_1) of (7). The use of the loss (L_1) increase the divergence capability of the reconstructed image. The adversarial minimax training is performed by using all the 11466 JS and 2000 RGB images that is explained in section 3.1.0.1. The hyperparameters of this training are described as follows. The learning of the G and D are set as 10^{-5} and 10^{-3}. Here, the learning rate of D is kept smaller than that of G to make it learn well in advance. This provides a good adversarial training because D is able to anticipate G's output data as real/fake well. The minibatch size is set to 12, maximum number of training epochs is set to 40. The parameter α of (9) is set as 10^{-4}. During the minimax training, it was observed that G's loss started from near 0 value and flattened at around 0.4322. And D's loss decreased from approximately 1.3 to 0.5412. These final value of losses are however a slight deviation from the ideal 0.5 mark. It is of vital importance that G and D's losses attains near optimal mark to a) avoid model collapse, b) provide a good reconstructed image that resembles JS image to a large extent, and c) to be able to provide enough resilence against malicious test data. The training is performed in our PC whose specifications are mentioned in the 'Teacher training phase' subsection. Finally, the denoising generator is ready to be deployed in the third stage training.

• *Classifier training phase* -The prime motivation of our proposed methodology is the use of three disjoint training phases instead of an end-to-end form of training that most method employ. It allows any of the top CNNs to be trained with the denoising generator without placing much computational burden. Therefore, in the final phases we apply one of the training samples to the trained ResilientNet-Generator to produce a purified JS. The resulting image is then fed into the target CNN for learning. The CNN learns by using the cross entropy loss between its predicted and target labels. The training is performed with the same training, validation and test samples as the Teacher-1 training session. However, the 2294 validation sample described earlier now subdivided into a validation and test samples in the ratio 50:50. Therefore, given a batch os 11466 samples, we split them into 80:10:10 training, validation and test samples. This standard practice has been reportted in the pioneering works in this domain and in existing methods used in our comparison. The training hyperparameters are as follows. The learning rate is fixed at 10^{-4}, minibatch size is 16, maximum number of epochs is 100. The Adam optimizer is used here. Here, the learning rate and minibatch size is intentionally kept low to allow the CNN to learn steadily. Due to this factor, the training accuracy converged slowly and thus it became necessary to increase the number of epochs to 100. This set of hyperparameters is used for training ten CNN model for a comprehensive evaluation. They are - (1) GoogleNet, (2) ResNet-50, (3) VGG-19, (4) MobileNet, (5) ShuffleNet, (6) NasNet-A, (7) EfficientNet-B7, (8) EfficientNet-B5, (9) EfficientNet-B0, and (10) ResNet-101.

c) Quantitative results and comparison: Two sets of tests are performed. a) Conventional testing: It is carried out by prior methods. It is extremely prone to inaccurate prediction due to the high probability of noise being included in a real-world scenario. b) Robust testing: It shows that deep learning model can be fortified against numerous undesired threats. The results in this subsection's are based on conventional testing. The second type of investigation is described in Section 3.3 under 'Ablation study'. The effectiveness of the proposed ResilientNet is demonstrated by deploying it with the top ten CNN. A sample is initially run through the trained ResilientNet-Generator during testing to get a cleaned JS image. The generated image is then sent to the trained classifier. The test is run using 1147 signals for this dataset. Table II compares existing results in the same dataset with the proposed findings. Among the previous works, encoder [9] has the highest success percentage (99.1%). Other approaches, such as the Siamese network, Multimodal network, CNN-ResNet50, CNN-ResNet101, SVM, and 1D CNN showed more than 90% accuracy. Their findings are noteworthy since they used 1D signal vectors, which provide fewer features when combined with machine learning and 1D DNN models. The results of several

TABLE II
QUANTITATIVE COMPARISON WITH MINDEEG-IMAGENET

Sr.	Model Name	Accuracy (%)
1	Encoder [9]	99.1
2	RNN-based model [9]	82.1
3	Siamese Network [9]	93.7
4	Multimodal Network [9]	94.1
5	CogniNet [9]	89.6
6	RS-LDA [9]	13
7	LDA [47]	80.27
8	LSTM [47]	83.09
9	CNN [47]	85.6
10	CNN-Alex [47]	88.8
11	CNN-VGG16 [47]	90.97
12	CNN-ResNet50 [47]	92.65
13	CNN-ResNet101 [47]	92.99
14	kNN [48]	42.9
15	SVM [48]	94
16	MLP [48]	49
17	1D CNN [48]	97.4
18	Disco GAN [49]	58.75
19	Cycle GAN [49]	66.02
20	SimVAE [49]	69.02
21	DSML-Lxy [49]	76.68
22	DSML [49]	81.24
23	Resilient-Net+GoogleNet	**99.21**
24	Resilient-Net+ResNet50	94.07
25	Resilient-Net+VGG19	95.29
26	Resilient-Net+MobileNet	95.81
27	Resilient-Net+ShuffleNet	92.68
28	Resilient-Net+NasNetA	98.17
29	Resilient-Net+EfficientNetB7	98.52
30	Resilient-Net+EfficientNetB5	96.77
31	Resilient-Net+EfficientNetB0	95.73
32	Resilient-Net+ResNet101	96.25

CNN models in Serial numbers 10-13 demonstrate that CNN can enhance prediction efficiency when compared to machine learning rivals such as RSLDA, LDA, kNN, and SVM. The investigation of Resilient-Net with a variety of CNNs, ranging from low parameter models like ResNet50 and VGG19 to high parameter models like EfficientNetB7 and NasNetA shows that the amount of parameters does affect accuracy, since low parameter models performed badly when applied with ResilientNet. However, GoogLeNet, which has far fewer parameters than NasNetA and EfficientNetB7, performed better. This demonstrates that, among the top CNNs, GoogLeNet still has enormous potential. The Encoder [9] architecture shows a robust performance of 99.1%, which is somewhat higher than our ResilientNet-GoogLeNet approach. In this regard, we may compare on two fronts: the number of parameters and the training duration. In terms of parameters, GoogLeNet has less because we are not using an end-to-end training. In our training, we take a JS image and denoise it using a pre-trained Generator before passing it to the GoogLeNet model. We backpropagate the weights to the GoogLeNet classifier in the traditional manner during this training. In the case of the encoder [9], however, there are two structures: a feature encoder that employs a Recurrent Neural Network (RNN) and an Encoder-Decoder GAN unit. The number of parameters increases with end-to-end training, resulting in an n-fold increase in training time, where n is the number of target classifiers. In our situation, however, the adoption of a two-staged training approach assured that if we wanted to utilise another target classifier, we just have to incorporate the pre-trained generator into the training pipeline as illustrated in Fig. 3 with less complexity. Furthermore, we know that GoogLeNet has a small number of network parameters (7 million approximately). As a result, although having somewhat greater performance than Encoder [9], the simplicity in (re)usability and training time is more optimised. It is also observed that GoogLeNet's 9 inception modules, with their diverse kernel sizes (1x1 to 5x5), ensures that features of various scales be retrieved without producing a parameter explosion, which is the primary cause of overfitting. This is also demonstrated by its superiority over NasNetA and EfficientNetB7, which have a significantly higher number of parameters.

TABLE III
COMPARISON WITH THE MINDEEG-MNIST DATASET

Sl. No.	Model	Sample Modality	Accuracy (%)
1	EEGNet [50]	EEG Vector	32.98
2	Autoencoder kNN [51]		27.6
3	Autoencoder RF [51]		27.5
4	4 CNN Network [51]		35.2
5	GRU Network [51]		33.8
6	LSTM [52]		10.77
7	AdaBoosted DEvo MLP [52]		31.35
8	DEvo MLP [52]		27.09
9	Correlation-CNN [53]		70.1
10	Resilient-Net+GoogleNet	2D aug.	**98.02**
11	Resilient-Net+ResNet50		92.30
12	Resilient-Net+VGG19		94.27
13	Resilient-Net+MobileNet		94.01
14	Resilient-Net+ShuffleNet		90.03
15	Resilient-Net+NasNetA		97.21
16	Resilient-Net+EfficientNetB7		96.88
17	Resilient-Net+EfficientNetB5		97.37
18	Resilient-Net+EfficientNetB0		94.05
19	Resilient-Net+ResNet101		96.47

The proposed enhancement to existing CNN models, which use a 2D signal format and a Denoising Encoder trained in an Adversarial Fashion, has the potential to increase performance. The implementation of ResilientNet with GoogLeNet and NasNetA produced an accuracy of 99.21% and 98.17%, respectively, among the recommended ten results.

B. Experimentation With MindEEG-MNIST Dataset

The experimentation follows a similar fashion as the MindEEG-ImageNet dataset.

A. Description of the datasets. It is a dataset for visual brain decoding that is freely available to the public. Brain signals are recorded when a subject observes and thinks about a visual stimulus for roughly 2 seconds. The dataset was created by recording EEG signals using four commercial EEG equipment. They are the following: NeuroSkyMind Wave, Emotiv EPOC, Interaxon Muse, and Emotiv Insight. The dataset comprises 11 classes: 0-9 digits and '-1'. The '-1' class contains samples taken using random actions. When all 11 classes are included, there are 1,207,293 samples. The total number of samples collected by the four devices listed above is 67635,

TABLE IV
ROBUSTNESS OF THE PROPOSED MODEL OVER A WIDE OF RANGE OF ITERATION FOR THREE POPULAR ATTACK METHODS. THE RESULTS FOR THE TWO DATASETS DISPLAYED IN ONE CELL IN THE SYNTAX A/B, WHERE A AND B ARE THE RESULTS OF MINDEEGIMAGENET AND MINDEEGMNIST RESPECTIVELY. BOLD NUMERICS INDICATES BEST PERFORMANCE IN EACH SET

Sr. No.	Model	Attack Name	Robust accuracy with increase iteration of attack						Attack without defense
			10	30	50	70	90	100	
1	ResilientNet+GoogleNet	PGD	**93.9**/92.8	**94.3**/**93.1**	**95.1**/**94.0**	**93**/**91.9**	**92.4**/**91.3**	**94.3**/**93.2**	14.4/14.2
2	ResilientNet+ResNet50		89.1/87.4	89.4/87.7	90.2/88.5	88.2/86.5	87.6/85.9	89.4/87.7	9.9/10.3
3	ResilientNet+VGG19		90.2/89.3	90.6/89.6	91.4/90.4	89.3/88.4	88.7/87.8	90.6/89.6	8.7/9.0
4	ResilientNet+MobileNet		90.7/89.0	91/89.3	91.9/90.2	89.8/88.1	89.2/87.5	91.1/89.4	12/12.3
5	ResilientNet+ShuffleNet		87.8/85.2	88.1/85.6	88.9/86.3	86.9/84.4	86.3/83.8	88.1/85.6	11.6/12.3
6	ResilientNet+NasNetA		92.9/92.0	93.3/92.4	94.1/93.2	92/91.1	91.4/90.5	93.3/92.4	14.3/14.2
7	ResilientNet+EfficientNetB7		93.3/91.7	93.6/92.1	94.5/92.9	92.3/90.8	91.7/90.2	93.6/92.1	19.2/19.1
8	ResilientNet+EfficientNetB5		91.6/**92.2**	92/92.5	92.8/93.4	90.7/91.3	90.1/90.6	92/92.5	16/16.2
9	ResilientNet+EfficientNetB0		90.6/89.0	91/89.4	91.8/90.2	89.7/88.2	89.1/87.6	91/89.4	10.4/10.6
10	ResilientNet+ResNet101		91.1/91.3	91.5/91.7	92.3/92.5	90.2/90.4	89.6/89.8	91.5/91.7	13/13.3
11	ResilientNet+GoogleNet	CW	**97**/**96.5**	**95.9**/**96.1**	**96.7**/**96.5**	**94.9**/**95.7**	**96**/**96.4**	**96.1**/**96.3**	20.2/19.9
12	ResilientNet+ResNet50		92/91.0	91/90.5	91.7/90.8	90/90.2	91.1/90.8	91.1/90.8	8.6/8.9
13	ResilientNet+VGG19		93.2/92.9	92.1/92.4	92.9/92.8	91.2/92.0	92.2/92.7	92.3/92.6	10.5/10.8
14	ResilientNet+MobileNet		93.7/92.6	92.6/92.1	93.4/92.5	91.7/91.8	92.7/92.4	92.8/92.4	17.5/17.9
15	ResilientNet+ShuffleNet		90.6/88.7	89.6/88.2	90.4/88.6	88.7/87.9	89.7/88.5	89.8/88.4	13.9/14.7
16	ResilientNet+NasNetA		96/95.7	94.9/95.3	95.7/95.7	93.9/94.9	95/95.6	95.1/95.5	17.1/17.1
17	ResilientNet+EfficientNetB7		96.3/95.4	95.3/94.9	96.1/95.3	94.3/94.6	**95.4**/95.2	95.4/95.2	16/15.9
18	ResilientNet+EfficientNetB5		94.6/95.9	93.6/95.4	94.4/95.8	92.6/95.0	93.7/95.7	93.7/95.7	19.2/19.4
19	ResilientNet+EfficientNetB0		93.6/92.6	92.6/92.2	93.3/92.5	91.6/91.8	92.7/92.5	92.7/92.4	12.4/12.7
20	ResilientNet+ResNet101		94.1/95.3	93.1/94.8	93.8/94.9	92.1/94.4	93.2/95.1	93.2/95.1	11.6/11.8
21	ResilientNet+GoogleNet	MIFGSM	**97.7**/**95.9**	**97.2**/**95.0**	**97.6**/**94.6**	**96.8**/**94.2**	**97.5**/**96.0**	**97.5**/**95.1**	20.9/17.9
22	ResilientNet+ResNet50		92.8/90.3	92.3/89.5	92.6/89.0	91.9/88.7	92.6/90.4	92.5/89.6	14.4/14.8
23	ResilientNet+VGG19		93.9/92.2	93.4/91.4	93.8/90.9	93/90.6	93.7/92.3	93.6/91.5	12.6/11.8
24	ResilientNet+MobileNet		94.4/91.9	93.9/91.1	94.3/90.7	93.5/90.3	94.2/92.0	94.1/91.2	17.4/7.0
25	ResilientNet+ShuffleNet		91.3/88.1	90.8/87.3	91.2/86.9	90.5/86.5	91.1/88.1	91/87.4	16.8/12.7
26	ResilientNet+NasNetA		96.7/95.1	96.2/94.2	96.6/93.8	95.8/93.4	96.5/95.2	96.4/94.3	20.7/20.7
27	ResilientNet+EfficientNetB7		97/94.8	96.5/93.9	96.9/93.5	96.2/93.1	96.8/94.8	96.8/94.0	27.9/25.7
28	ResilientNet+EfficientNetB5		95.3/95.2	94.8/94.4	95.2/93.9	94.5/93.6	95.1/95.3	95.1/94.5	23.2/20.2
29	ResilientNet+EfficientNetB0		94.3/92.0	93.8/91.2	94.2/90.7	93.4/90.4	94.1/92.1	94/91.3	15/15.4
30	ResilientNet+ResNet101		95.1/94.3	94.6/93.5	94.7/93.1	94.2/92.7	94.9/94.4	94.8/93.6	18.9/19.2

The results for the two datasets displayed in one cell in the syntax A/B, where A and B are the results of MindEEGImageNet and MindEEGMNIST respectively. Bold numerics indicates best performance in each set.

980476, 163932, and 65250, in that order. Only 100,000 samples are kept for the devised work. We use the same set of parameters as discussed in the Section 3.1.0.1 to convert these signals to JS images. For the RGB image dataset, we acquire 200 images from the MNIST dataset for each of the 10 digit classes, i.e. 0-9. Addionally, to represent the '-1' class, we acquire 10 images from each of the 0-9 classes. This exercise results in 2200 MNISt images.

B. Implementation procedure. The implementation process is done over three phases as described below.

In the teacher training phase, both the JS and RGB datasets are divided into the 80:20 training:validation standard ratio. For instance, this creates 80000 training and 20000 validation samples for the JS dataset. The training hyperparameters of the two teacher are - (a) learning rate is fixed at 10^{-3}, (b) minibatch size is set to 10, (c) maximum epochs is set as 10. While, the value of remaining parameters and training specifications were kept similar to the MindBigData-ImageNet. The EfficientNetb7 architecture is also used here as the teacher networks, but trained with different spatial modalities. Next, in the adversarial training stage, the Teacher-1, Teacher-2, denoising Generator and the Discriminator CNNs undergo training. The same training procedure, parameters and hyperparameters elaborated in Section 3.1.0.2 under the 'Adversarial training phase' is also employed here with the only difference that minibatch size is set to 4. After the adversarial training stage, we train the target classifiers by using the same procedure, parameters and hyperparameters explained in case of MindBigData-ImageNet.

C. Quantitative results and comparison. The MindEEGMNIST dataset was proposed only recently. As a result, there are few implementations of them. Researchers have entered the field of multimodal signal identification recently. There are two types of classification methods. The first is binary classification. The second is a multiclass forecast using all 11 classes. Present work is based on the 11 classes. The results in Table III are based on 11 classes. All prior approaches are based on extracting cognition from EEG signal vectors. So, they suffered from a lack of datasets to accurately predict. The MindEEG-MNIST dataset has around 1.2 million signals for use. The machine learning models and 1D DNN models are not as accurate in extracting features as CNN. Thus, the best success rate among all prior outcomes is 70.1%. It was achieved with Correlation-CNN. The findings of GoogLeNet, NasNetA, and EfficientNetB7 offered more than 97 % accuracy among the 10 CNN models that is employed with the suggested ResilientNet. As a result, the suggested technique achieves a prediction accuracy of 98.02%. However, in light of the most current issue of adversarial perturbation on DNN

models, the outcomes of our experiments needs to be validated using perturbed test signals.

C. Ablation Study

DL has demonstrated exceptional proficiency in categorizing brain signals with high accuracy. The experimental results in Tables II, III are vulnerable to threats produced by the insertion of a minute amount of noise in the signal. Researchers in [13] described a mechanism to push DNN to produce misclassification for the first time. The method of [13] is a single step in which a noise sample is made in only one iteration. It was followed by stronger assaults that used many phases to generate noise capable of crippling a fully trained DNN. Projected Gradient Descent (PGD) [54], Carlini Wagner (C&W) [55], and Momentum Iterative FGSM (MIFGSM) [56] are some notable and powerful assaults. Table IV highlights an extensive study on the impact of such attacks on brain signals for various iterations ranging from 10 to 100. The goal is to determine if the trained Resilient-Net Denoising Generator can filter noise, such as attacked test data while allowing the DNN classifier to make accurate predictions.

The plan for this experiment is given below. A test signal is collected and converted to JS format. Depending on the nature of attack method used, the noisy version of the clean image is then created. Another critical parameter is ϵ, the amount of image distortion. Table IV displays all **60 tests** under consideration for $\epsilon = 0.05$. The noisy image is processed by the ResilientNet-Generator. The trained classifier is fed with the resultant JS. Finally, the ablation study analyses all 10 CNNs for these two brain signal datasets. As a result, 60 sets of tests were carried out. The results are documented in Table IV. The last column in Table IV displays the performance of the specific CNN that did not undergo training using the Resilient-Net DAE Generator. Furthermore, the noisy JS is fed to it during testing without the denoiser unit. When such noisy JS is used, the categorization rate is seen to have reduced dramatically, with the maximum accuracy being 27.89%. The results in the columns represent the robust performance even when the input JS is treated to the deliberately created noise. Compared to C&W and MIFGSM, the noise produced with PGD produces the highest disruption in both brain signal datasets, MindEEG-ImageNet and MindEEG-MNIST. The results for both the datasets reveal that the Resilient-Net+GoogLeNet platform can deliver the best accuracy against all three forms of noise. Despite the large iteration values, the proposed workflow retains its resistance to third party noise that is added to the input signal. As a result, even in noise, the proposed technique could deliver a strong accuracy of 97% to 96% on average.

IV. Conclusion

In this paper, a method to enable robust predictions of brain signals by combining a novel JS method with the ResilientNet DAE Generator is proposed. The time-series signal vectors are translated into the 2D Spatial domain, and two distinct forms of scalograms are brought under one roof to add to the richness. The current study fills a critical research gap between the prediction stage and translating the predicted class to an image for showing the object thought by a person. The problem has been solved by fortifying the classifier against well-crafted noise. Existing research attempts to map a brain signal to an image or predict the class of an incoming signal. However, no technique has investigated the significance of shielding the classifier's gradients against assaults in both circumstances. During testing, it was discovered that the ResilientNet-Generator could create semantically coherent reconstructed spatial data (or JS) with the ground truth class. Furthermore, using an additional regularizer in the Generator penalizes the pair-wise correlation of the neurons at the encoder's output, forcing it to acquire more varied and compact representations for the input samples. The proposed framework's utility over two datasets is demonstrated by comparing it to existing approaches. In-depth study on the framework's resilience in the presence of adversarially altered test samples also demonstrates that the ResilientNet-data Generator's reconstruction is capable of protecting against formidable attacks.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clin. Neuriophysiol.*, vol. 113, no. 6, pp. 767–791, 2002.

[2] N. Lu, T. Li, X. Ren, and H. Miao, "A deep learning scheme for motor imagery classification based on restricted Boltzmann machines," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 6, pp. 566–576, Jun. 2017.

[3] A. M. Green and J. F. Kalaska, "Learning to move machines with the mind," *Trends Neurosci.*, vol. 34, no. 2, pp. 61–75, 2011.

[4] D. Zhang et al., "Cascade and parallel convolutional recurrent neural networks on EEG-based intention recognition for brain computer interface," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32.

[5] C. Wang, S. Xiong, X. Hu, L. Yao, and J. Zhang, "Combining features from ERP components in single-trial EEG for discriminating four-category visual objects," *J. Neural Eng.*, vol. 9, no. 5, 2012, Art. no. 056013.

[6] I. Simanova, M. V. Gerven, R. Oostenveld, and P. Hagoort, "Identifying object categories from event-related EEG: Toward decoding of conceptual representations," *PLoS One*, vol. 5, no. 12, 2010, Art. no. e14465.

[7] T. Carlson, D. A. Tovar, A. Alink, and N. Kriegeskorte, "Representational dynamics of object vision: The first 1000 ms," *J. Vis.*, vol. 13, no. 10, pp. 1–1, 2013.

[8] M. Cauchoix, G. Barragan-Jason, T. Serre, and E. J. Barbeau, "The neural dynamics of face detection in the wild revealed by MVPA," *J. Neurosci.*, vol. 34, no. 3, pp. 846–854, 2014.

[9] J. Jiang, A. Fares, and S.-H. Zhong, "A brain-media deep framework towards seeing imaginations inside brains," *IEEE Trans. Multimedia*, vol. 23, pp. 1454–1465, 2021.

[10] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "MentorNet: Learning data-driven curriculum for neural networks on corrupted labels," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2304–2313.

[11] E. Malach and S. Shalev-Shwartz, "Decoupling" when to update" from" how to update," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30.

[12] B. Han et al., "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018,vol. 31.

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

[14] F. Tramér, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," 2017, *arXiv:1705.07204*.

[15] A. G. Huth, W. A. D. Heer, T. L. Griffiths, F. E. Theunissen, and J. L. Gallant, "Natural speech reveals the semantic maps that tile human cerebral cortex," *Nature*, vol. 532, no. 7600, pp. 453–458, 2016.

[16] A. G. Huth, T. Lee, S. Nishimoto, N. Y. Bilenko, A. T. Vu, and J. L. Gallant, "Decoding the semantic content of natural movies from human brain activity," *Front. Syst. Neurosci.*, vol. 10, 2016, Art. no. 81.

[17] D. Wang et al., "Epileptic seizure detection in long-term EEG recordings by using wavelet-based directed transfer function," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 11, pp. 2591–2599, Nov. 2018.

[18] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals," *Comput. Biol. Med.*, vol. 100, pp. 270–278, 2018.
[19] X. Zhang, L. Yao, M. Dong, Z. Liu, Y. Zhang, and Y. Li, "Adversarial representation learning for robust patient-independent epileptic seizure detection," *IEEE J. Biomed. health Informat.*, vol. 24, no. 10, pp. 2852–2859, Oct. 2020.
[20] J. Cao, J. Zhu, W. Hu, and A. Kummert, "Epileptic signal classification with deep EEG features by stacked CNNs," *IEEE Trans. Cogn. Devel. Syst.*, vol. 12, no. 4, pp. 709–722, Dec. 2020.
[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2009, pp. 248–255.
[22] A. Gogna, A. Majumdar, and R. Ward, "Semi-supervised stacked label consistent autoencoder for reconstruction and analysis of biomedical signals," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 9, pp. 2196–2205, Sep. 2017.
[23] R. T. Schirrmeister et al., "Deep learning with convolutional neural networks for EEG decoding and visualization," *Hum. Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.
[24] S. Nishimoto, A. T. Vu, T. Naselaris, Y. Benjamini, B. Yu, and J. L. Gallant , "Reconstructing visual experiences from brain activity evoked by natural movies," *Curr. Biol.*, vol. 21, no. 19, pp. 1641–1646, 2011.
[25] S. Palazzo, C. Spampinato, I. Kavasidis, D. Giordano, and M. Shah, "Generative adversarial networks conditioned by brain signals," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3410–3418.
[26] I. Kavasidis, S. Palazzo, C. Spampinato, D. Giordano, and M. Shah, "Brain2image: Converting brain signals into images," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1809–1817.
[27] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Learning representations from EEG with deep recurrent-convolutional neural networks," 2015, *arXiv:1511.06448*.
[28] B. Kaneshiro, M. P. Guimaraes, H.-S. Kim, A. M. Norcia, and P. Suppes , "A representational similarity analysis of the dynamics of object processing using single-trial EEG classification," *Plos One*, vol. 10, no. 8, 2015, Art. no. e0135697.
[29] T. Ogawa, Y. Sasaka, K. Maeda, and M. Haseyama, "Favorite video classification based on multimodal bidirectional LSTM," *IEEE Access*, vol. 6, pp. 61401–61409, 2018.
[30] P. Tirupattur, Y. S. Rawat, C. Spampinato, and M. Shah, "ThoughtViz: Visualizing human thoughts using generative adversarial network," in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 950–958.
[31] H. Yang, S. Sakhavi, K. K. Ang, and C. Guan, "On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification," in *Proc. IEEE 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2015, pp. 2620–2623.
[32] K.-L. Tseng, Y.-L. Lin, W. Hsu, and C.-Y. Huang, "Joint sequence learning and cross-modality convolution for 3D biomedical segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6393–6400.
[33] J. Dolz, K. Gopinath, J. Yuan, H. Lombaert, C. Desrosiers, and I. B. Ayed, "HyperDense-Net: A hyper-densely connected CNN for multi-modal image segmentation," *IEEE Trans. Med. Imag.*, vol. 38, no. 5, pp. 1116–1126, May 2019.
[34] D. Nie, L. Wang, Y. Gao, and D. Shen, "Fully convolutional networks for multi-modality isointense infant brain image segmentation," in *Proc. IEEE 13Th Int. Symp. Biomed. Imag.*, 2016, pp. 1342–1345.
[35] H. Chen et al., "MMFNet: A multi-modality MRI fusion network for segmentation of nasopharyngeal carcinoma," *Neurocomputing*, vol. 394, pp. 27–40, 2020.
[36] L. Tran, X. Liu, J. Zhou, and R. Jin, "Missing modalities imputation via cascaded residual autoencoder," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1405–1414.
[37] C. Du, C. Du, X. Xie, C. Zhang, and H. Wang, "Multi-view adversarially learned inference for cross-domain joint distribution matching," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1348–1357.
[38] L. Cai, Z. Wang, H. Gao, D. Shen, and S. Ji, "Deep adversarial learning for multi-modality missing data completion," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1158–1166.
[39] C. Spampinato, S. Palazzo, I. Kavasidis, D. Giordano, N. Souly, and M. Shah, "Deep learning human mind for automated visual classification," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2017, pp. 6809–6817.
[40] K. Wahengbam, M. P. Singh, K. Nongmeikapam, and A. D. Singh, "A group decision optimization analogy-based deep learning architecture for multiclass pathology classification in a voice signal," *IEEE Sensors J.*, vol. 21, no. 6, pp. 8100–8116, Mar. 2021.
[41] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[42] I. J. Goodfellow et al., "Generative adversarial networks," 2014, *arXiv:1406.2661*.
[43] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.
[44] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*.
[45] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep learning on image denoising: An overview," *Neural Netw.*, vol. 131, pp. 251–275, 2020.
[46] B. Goyal, A. Dogra, S. Agrawal, B. S. Sohi, and A. Sharma, "Image denoising review: From classical to state-of-the-art approaches," *Inf. Fusion*, vol. 55, pp. 220–244, 2020.
[47] Z. Jiao, H. You, F. Yang, X. Li, H. Zhang, and D. Shen, "Decoding EEG by visual-guided deep neural networks," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 1387–1393.
[48] R. Li, "Training on the test set? an analysis of spampinato et al.[31]," 2018, *arXiv:1812.07697*.
[49] C. Du, C. Du, and H. He, "Multimodal deep generative adversarial models for scalable doubly semi-supervised learning," *Inf. Fusion*, vol. 68, pp. 118–130, 2021.
[50] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: A compact convolutional neural network for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, 2018, Art. no. 056013.
[51] B. L. K. Jolly, P. Aggrawal, S. S. Nath, V. Gupta, M. S. Grover, and R. R. Shah, "Universal EEG encoder for learning diverse intelligent tasks," in *Proc. IEEE 5th Int. Conf. Multimedia Big Data*, 2019, pp. 213–218.
[52] J. J. Bird, D. R. Faria, L. J. Manso, A. Ekárt, and C. D. Buckingham, "A deep evolutionary approach to bioinspired classifier optimisation for brain-machine interaction," *Complexity*, vol. 2019, 2019.
[53] R. Mishra, K. Sharma, and A. Bhavsar, "Visual brain decoding for short duration EEG signals," in *Proc. IEEE 29th Eur. Signal Process. Conf.*, 2021, pp. 1226–1230.
[54] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016, *arXiv:1611.01236*.
[55] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.
[56] Y. Dong et al., "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9185–9193.