IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 January 2023, accepted 25 January 2023, date of publication 31 January 2023, date of current version 3 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3241096

RESEARCH ARTICLE

StAIn: Stealthy Avenues of Attacks on Horizontally Collaborated Convolutional Neural Network Inference and Their Mitigation

ADEWALE A. ADEYEMO[1], (Graduate Student Member, IEEE), JONATHAN J. SANDERSON[1], TOLULOPE A. ODETOLA[1], FAIQ KHALID[2], (Member, IEEE), AND SYED RAFAY HASAN[1], (Senior Member, IEEE)
[1]Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, TN 38505, USA
[2]Department of Computer Engineering, Vienna University of Technology, 1040 Wien, Austria

Corresponding author: Syed Rafay Hasan (shasan@tntech.edu)

This work was supported in part by the Faculty Research Grant from Tennessee Tech University's Office of Research and Center of Manufacturing Research, and in part by NSF under Grant CNS-1852126.

ABSTRACT With significant potential improvement in device-to-device (D2D) communication due to improved wireless link capacity (e.g., 5G and NextG systems), a collaboration of multiple edge devices (called horizontal collaboration (HC)) is becoming a reality for real-time Edge Intelligence (EI). The distributed nature of HC offers an advantage against traditional adversarial attacks because the adversary does not have access to the entire deep learning architecture (DLA). Due to the involvement of multiple untrusted edge devices in HC environment, the possibility of malicious devices cannot be eliminated. In this paper, we unearth some attacks that are very effective and stealthy even when the attacker has minimal knowledge of the DLA as is the case in HC-based DLA. We are also providing novel filtering methods to mitigate such attacks. Our novel attacks leverage local information available on output feature maps (FMs) of a targeted edge device to modify the regular adversarial attacks (e.g. Fast Gradient Signed Method (FGSM) and Jacobian-based Saliency Map Attack (JSMA)). Similarly, a customized convolutional neural network (CNN) based filter is empirically designed, developed, and tested. Four different CNN models (LeNet, CapsuleNet, MiniVGGNet, and VGG16) are used to validate the proposed attacks and defense methodologies. Our three attacks on four different CNN models (with two variations of each attack) show a substantial accuracy drop of 62% on average. The proposed filtering approach is able to mitigate the attack by recovering the actual accuracy back to 75.1% on average. To the best of our knowledge, this is the first work that investigates the security vulnerability of DLA in the HC environment, and all three of our attacks are scalable and agnostic to the partition location within the DLA.

INDEX TERMS Horizontal collaboration, convolutional neural network, machine learning security, adversarial machine learning, deep learning, edge intelligence.

I. INTRODUCTION

Deep learning (DL) models, particularly Convolution Neural Networks (CNNs), have achieved success in many fields of computer vision, resulting in increased usage in mission-critical applications [1]. DL inference models that are capable of outperforming humans in computer vision applications require intensive computation and have a large memory footprint [2]. Recently, due to the discovery of the threats posed by adversarial images to Deep Learning Architectures (DLA), robustness against such attacks has become one of the critical factors in its deployment [3]. In this paper, we called such attacks "adversarial data attacks" or ADA (we purposely used the word data in ADA instead of images for reasons that will be clear later on in this paper). An example of ADA was demonstrated by "Tencent's Keen Security Lab".

The associate editor coordinating the review of this manuscript and approving it for publication was Paul D. Yoo.

They are able to trick a Tesla Model S into switching lanes so that it drives directly into oncoming traffic [4]. Similarly, in [5] Morgulis et al. also fooled a commercial car's perception system using adversarial traffic signs.

Traditional adversarial machine learning research focuses on how to generate capable adversarial samples that can deceive a model with as little perturbations as possible and subsequently help in developing a model that is robust against these adversarial samples. Common methods of defense against ADAs include adversarial training [1] and defensive distillation [6]. Defensive distillation is able to mitigate the effects of Fast Gradient Signed Method (FGSM) attacks and Iterative Fast Gradient Signed Method (IFGSM) attacks because of the small gradient size, as small distortions would not be able to generate a significant loss function [7]. However, these techniques are useful during training a model, and cannot be directly used in inference attacks.

By leveraging the characteristics of distributed inference in edge computing, it may be possible to reduce ADAs, especially on edge devices [8], [9]. Modern intelligent computer vision applications such as video analytics, augmented & virtual reality (AR/VR), etc., need intensive computation to be done near data generating devices (edge devices) to achieve real-time performance [10], [11], [12], [13]. However, due to the limited memory available in these edge devices, they are unable to accelerate inference with larger CNN models [14]. Edge computing allows edge devices to use edge servers and instead of sharing the information with the cloud network, a server (edge server) near the edge device does the computation, which in turn reduces the latency and improves the privacy and security of the systems with edge intelligence (EI).[1] [10].

With significant potential improvement in device-to-device (D2D) communication in near future due to improved wireless link capacity (e.g., 5G and NextG systems), collaboration of multiple edge computing devices is becoming a reality for real-time EI [15], [16]. In the literature, such collaboration among edge devices only is called Horizontal collaboration (HC) [17], [18], [19], whereas collaboration between edge devices and edge servers (or cloud framework) is called as vertical collaboration (VC) [19], [20], [21]. In HC, DLAs, e.g. Convolutional Neural Networks (CNNs), need to be partitioned either between edge device(s) and edge server [10], [11], [22] or among multiple edge devices as shown in Fig. 1 [23], [24].

The deployment of the inference phase of DLA into multiple edge computing devices is enabled by HC-based EI. Hence, if the security of a particular edge device is compromised, a full-fledged white-box attack cannot be launched nor can the model be reverse-engineered because the adversary does not have access to the entire DLA. Therefore, in this research, we investigate the vulnerability of DLAs deployed in HC environments in the context of a gray-box attack, in which the attacker has no access to the trained model's parameters but only the output feature maps (FMs) of a target edge device. In contrast to a white-box attack, a gray-box attacker must devise strategies for generating adversarial FMs that can fool the target network. Traditionally adversarial noise (e.g. FGSM noise) is designed and injected under the premise that the attacker has complete knowledge of DLA [25], however, for HC settings, these traditional techniques will fail since the data is only perturbed within the sub-module available to the attacker (i.e. with only partial DLA). As a defense, we propose using an adversarially trained convolution filter as pre-processing at the input to all edge devices in HC-based DLA. Hence, we propose a novel method of adversarial training via output FMs of the target edge devices. Our results show the effectiveness of the novel attacks and our proposed defense framework. It is also worth mentioning that our defense methodology is general enough to cater to different proposed attacks and their variants.

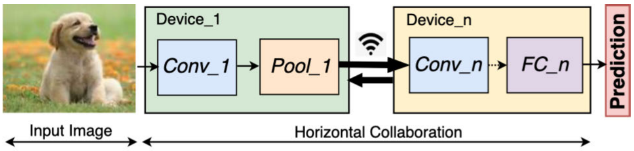

FIGURE 1. Horizontal Collaboration - A trained CNN model is partitioned and offloaded to different edge device(s) for inference.

A. RESEARCH CHALLENGES

1) **How to effectively introduce ADAs on targeted edge devices given the limited information available?** Though partitioned DLA provides privacy since only limited data (FMs) needs to be transferred over an insecure wireless communication network, but distributed nature of DLA may contain one or more untrusted nodes (edge devices) which can give rise to a new type of security issue. For example, if one of the nodes introduces adversarial noises as an output then it may lead to misclassification or decrease the accuracy. Existing adversarial attacks on inference require knowledge of the entire CNN model [26] to generate adversarial images, or at the very least the availability of the entire DLA as a black-box [27], [28], [29]. However, an adversary in partitioned DLA is not privy to the parameters of the deployed trained DLA, and may not have the access to complete architecture even as a black-box, hence launching effective adversarial attacks in HC environment set forth extra challenges. Some researchers have demonstrated breakthrough attacks in HC-based DLA inference through hardware intrinsic attacks [30], [31], [32]. But these attacks require either complete access to hardware models or memory of the targeted layer. So, how an adversarial attack like FGSM can be effectively deployed in HC settings are yet to be investigated. Since an adversary in HC settings can only access feature map data, not

[1] EI is a system where edge computing devices perform artificial intelligence-based tasks.

images, we continue to refer to such attacks as adversarial data attacks (ADA) rather than adversarial image attacks.

2) **How to safeguard against potential ADAs due to untrusted nodes in partitioned DLA:** Mitigating breakthrough attacks in partitioned DLA is another challenge. Traditionally, the impacts of adversarial attacks on the CNN are neutralized by deploying pre-processing noise (convolution) filters trained on adversarial images and integrated with ML-inference module [33]. However, in HC-based DLA, the compromised layer can be anywhere in the distributed network, rendering the traditional convolution filter useless because it cannot process FMs. Hence, the second research challenge is to investigate how the convolution filters need to be modified to accommodate the requirements of HC-based DLA. We propose using convolution filters as preprocessing at the input of each participating edge device in the HC-based DLA. This approach is a low-cost technique to mitigate adversarial attacks. The effectiveness of the proposed convolution filter has been validated by observing the top-1 accuracy of the attack model on DLA with our proposed mitigation techniques vs DLA with traditional convolution filters.

B. NOVEL CONTRIBUTIONS

The following are the major contributions of this paper:

- A novel set of ADAs (adversarial data attacks) are proposed to demonstrate the vulnerability of HC-based DLA. These attacks are designed on the assumption that only one of the participating edge devices is compromised.
- A comprehensive investigation into how to reduce ADAs on HC-based DLA using novel techniques of deploying adversarially trained convolution filters. Our method involves training convolutional filters with FMs rather than input images and then deploying the filters on edge nodes. The technique can protect edge devices within the distributed DLA.

Key Results: To examine our proposed attack methodologies, we perturbed four DLAs deployed in HC environment using three ADAs. The following are the key outcomes of these experiments:

1) First, we demonstrate how to modify traditional white-box adversarial attacks to perturb feature maps of DLA in HC environments. With three ADAs, we perturb the FMs of the first layer (L1) of the DLAs. A modified Gaussian Noise Attack (mGNA) attack on an MNIST dataset trained on CapsNet [34] DLA reduced classification accuracy from 99.6% to 52.4%. We also discovered that the modified Jacobian-based Saliency Map attack (mJSMA) on the Cifar10 dataset trained on the MiniVGGNet [35] model has the highest robustness, with classification accuracy dropping from 81.2% to 57.8%.
2) Second, it is shown that a pre-processing filter can reduce the impact of adversarial attacks. However, in the HC environment, the traditional method of deploying pre-processing filters (i.e. before the DLAs) is found to be ineffective. For example, when a traditional pre-processing filter is used to mitigate mJSMA ADA on MiniVGGNet (trained with Cifar-10 dataset), the classification accuracy remained very low (approx. 13%). Similar results are observed with other DLA.
3) Thirdly, we use pre-processing filters between layers of DLAs to protect against potential ADAs in untrusted nodes. With our technique, substantial improvement in classification accuracy is observed. For example, an average classification accuracy of 97% is observed when our filtering technique is applied to LeNet DLA subject to ADAs.
4) Lastly, for demonstrating the scalability of our approach, a trained VGG16 model (trained on the ImageNet dataset) is deployed in an HC environment. We generate adversarial FMs for the model's first layer (L1) using mGNA, mJSMA, and mFGSM (modified Fast Gradient Signed Method Attack) respectively. The perturbed output FMs of L1 served as input to the next CNN layer, in another edge device. Results show that the top-1 accuracy dropped from 91.71% to 49.8% (γ=0.5), 28.6% (ϵ=0.5), and 20.2% (θ=0.5) respectively. Deploying the pre-processing filters after layer one (L1), the top-1 classification accuracy increased to 80.5%, 67.6%, and 79.1% respectively for the given γ, ϵ, and θ values. Similar trends are observed on other layers of the VGG16 model.

II. THREAT MODEL

Deep learning architecture (DLA) can be attacked based on several factors which include but not limited to: attacker's knowledge of the deep learning system, attacker's goal, access to dataset, and the frequency of attack [36]. For this work, we consider the following two threat models :

- **Traditional Threat Model:** In this threat model, the traditional white box attack scenario is assumed where an attacker has complete access to the trained model like weights, training, and testing dataset. This threat model is basically used as a benchmark to compare with our work. Three different state-of-the-art perturbations, Gaussian Noise Attack (GNA), FGSM, and JSMA (Jacobian-based Saliency Map Attack) are introduced during the inference phase to demonstrate the viability of the attack (see Fig. 2(a)).
- **Threat Model for HC:** In this threat model, we explore ADAs in HC under the assumption that the attacker has access to only a portion of the CNN model (target layer) in a gray-box attack scenario. The attacker is privy to only the input image/FMs and output FMs of the target layer. Similar to the traditional threat model, the three state-of-the-art ADAs are modified for the purpose of

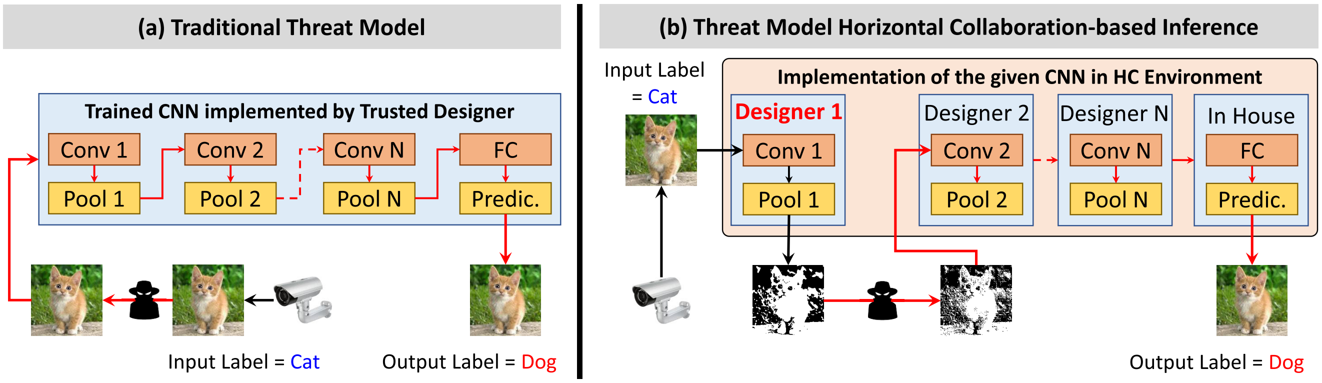

FIGURE 2. A pictorial comparison between traditional threat model for adversarial attacks with threat model for HC-based inference. (a) In the traditional threat model, the attacker has access to the input and output of the CNN, but the attacker may or may not have access to the CNN parameters. (b) In HC-based inference, the attacker has access to the input and the output of a particular segment of CNN, e.g., in this figure, the attacker (designer 1) has access to the input and output of the segment, which consists of Conv1 and Pool1 layers. Like the traditional threat model, the attacker may or may not have access to the parameters of the segment of CNN that is accessible to them. Note, in HC-based inference, the attacker does not have access to the input or output of the entire CNN.

perturbing the output FMs of the target layer during inference phase to demonstrate the potency of the modified attacks (see Fig. 2(b)).

III. BACKGROUND

This section presents the related work and background knowledge to improve the readability of paper.

A. RELATED WORK

Since this paper focuses on adversarial data attacks (ADA) and mitigation strategies, we discussed the related work for both. As noted previously, a traditional adversarial attack perturbs the DLA's input images and attempts to change the pixel intensity distribution by injecting specifically designed imperceptible perturbations to confuse a DLA [37], [38] for targeted or untargeted misclassifications. These attacks can be classified as white-box,[2] gray-box,[3] or black-box attacks.[4] In addition to adversarial attacks, several security attacks have been proposed that can steal models or perform hardware-intrinsic-attacks without complete knowledge of DLA, e.g., model stealing via reverse engineering [27], [29] and hardware intrinsic attack by shuffling of weights and FMs [39]. In our work, we are proposing modified adversarial attacks. Since the attacks are applied on FMs, therefore, we are calling it ADA instead of an adversarial image attack.

HC is a form of edge computing that involves partitioning the DL model horizontally, by intelligently selecting partition points based on: i) resource needs of different CNN layers, ii) total cost per specific layer configuration and iii) delay and energy requirements [40]. In addition to the advantages of HC, such as reduced latency [41] and energy efficiency [18], [42], HC limits an adversary's access to complete model parameters, which reduces the risk of ADAs. Despite all these advantages, HC-based CNN inference is also vulnerable to several threats. For example, Odetola et al. developed a stealthy hardware Trojan attack that performs a statistical analysis on the layer-by-layer output of CNN models in a horizontal collaborative environment [30]. Adeyemo et al. [31] investigated CapsNet's [34] robustness when exposed to noise-based inference attacks in an HC environment. Mohammed et al. [32] proposed Trojan attacks on CNN implemented across multiple nodes in a distributed edge network. Note, all the existing attacks in HC-based inference require modification of the model, hardware access, and a trigger design [30], [32], [39]. Although these attacks successfully perform misclassification, they are not applicable under the assumed threat model, where the adversary has access to *only* the input and output data of one layer (target layer) of the DLA. *Hence, we propose a novel methodology to modify the traditional white-box adversarial attack for HC settings using partial information of the DLA.*

Several techniques have been proposed to mitigate adversarial attacks [36], [43], for example, pre-processing-based defenses [33], [44], [45], [46], [47], [48], gradient masking [49], adversarial training [50], and dataset encryption [51], but these techniques are model-specific or require access to the complete model parameters [52]. Hence, these techniques cannot be applied in the more stringent (and arguably realistic) threat model for HC-based inference. On the other hand defense techniques for edge devices like secure channel [41], and multiple-input multiple-output (MIMO) beamforming [53], do not consider HC environment in which one of the participating nodes itself can be an adversary. Hence, the existing mitigation techniques against the proposed ADA cannot be directly applied to the stringent threat model for HC-based inference. *Therefore, we propose to modify the existing pre-processing filtering techniques for the HC environment because of its generalization ability and ease of design in terms of effectiveness.*

[2] Adversaries have complete knowledge of the targeted DLA.
[3] The adversaries' knowledge is limited to DLA.
[4] Adversaries can only access the input and limited output of DLA.

B. ADVERSARIAL ATTACKS

In this study, we utilized three state-of-the-art adversarial attacks which include: Gaussian Noise Attacks (GNA), Fast Gradient Signed Method (FGSM) and Jacobian-based Saliency Map Attack (JSMA).

1) GAUSSIAN NOISE-BASED ADVERSARIAL ATTACK (GNA)

This attack adds stochastically generated Gaussian noise (magnitude of the noise is between 0 and 1, and the standard deviation of 0.5) to each pixel of the input image for misclassifying the CNN, as described below:

$$x' = x + y \tag{1}$$

where x and x' represent input image and adversarial image, respectively, and y represents stochastically generated perturbations such that $y = [y_1 \ldots .y_n] \in [0, 1]$.

2) FGSM ATTACK

FGSM is a white box attack that uses the gradients of the CNN to generate an adversarial example [54] that maximizes the loss for an input image, as described below:

$$x' = x + \epsilon * sign(\nabla_x J(\theta, x, l)) \tag{2}$$

where, x', J, x, θ, l and ϵ represent adversarial image, loss, input images, model parameters, output labels, and multiplier to ensure imperceptibility, respectively.

3) JSMA ATTACK

The JSMA used the Jacobian matrix[5] to define an adversarial saliency map for choosing the features/pixels that should be perturbed to produce the necessary changes in model outputs. Then few of these selected features/pixels are saturated to their maximum or minimum values [54], [55], as described below:

$$S^+(x_{(i)}, c) = \begin{cases} 0 \; if \; \frac{\partial f(x)_{(c)}}{\partial x_{(i)}} < 0 \; or \; \sum_{c' \neq c} \frac{\partial f(x)_{(c')}}{\partial x_{(i)}} > 0 \\ -\frac{\partial f(x)_{(c)}}{\partial x(i)} \cdot \sum_{c' \neq c} \frac{\partial f(x)_{(c')}}{\partial x_{(i)}} \; otherwise \end{cases} \tag{3}$$

where $x_{(i)}$, i, c', c and $f(x)$ represent input images, number of input images, classes of dataset trained on the model, prediction ($argmax_{c'} f(x)_{(c')}$) and the softmax probability vector, respectively. $S^+()$ measures how much $x_{(i)}$ positively correlates with c, while also negatively correlating with all other classes. $\sum_{c' \neq c} \frac{\partial f(x)_{(c')}}{\partial x_{(i)}}$ is the summed gradient contribution across all non-targeted classes.

IV. MAHCI: Modification OF ADAs FOR HC-BASED Inference

This section provides a detailed, step-by-step procedure for adapting existing adversarial attacks for use with HC-based CNN inference using the proposed methodology, MAHCI. The implementation of StAIn was done using TensorFlow (version 2.2) and scikit-learn (version 1.2.0) on an Ubuntu operating system. The CNN models were trained on NVIDIA GPUs equipped with 128 GB of RAM and 28 Intel Xeon cores. To generate adversarial examples for HC-based inference, the proposed MAHCI requires the following steps (as shown in Fig. 3).

1) **Selection of the target layer:** In HC-based inference, the untrusted designer has access to a limited number of layer(s) in the DLA. The input layer in HC-based inference is not accessible to the adversary, which makes our threat model fundamentally different from the traditional threat model. Consequently, traditional attacks are inapplicable to HC environments. Therefore, the first step of the proposed methodology is to select the target layer(s) of the DLA that are accessible to the attacker. Empirically, it is known to the machine learning community that noise added farther from the output layer can be filtered out more easily due to the redundancies in the DLA. Hence, we demonstrate in our work that the proposed attack leads to a substantial accuracy drop even when the perturbations are introduced to the layers farthest from the output (i.e. the first layer).
2) **Feature Map Extraction and Label Generation:** In HC-based inference, the attacker does not have access to the input data and their respective labels (i.e., the output of the targeted CNN). Therefore, the second step of MAHCI is to generate the FMs for each available input using accessible layers. In some adversarial attacks, the output labels are required to estimate the gradient sign to optimize the adversarial noise. Hence, each generated FM is assigned to a unique pseudo-label.
3) **Modifying ADA for HC:** The third step of MAHCI is to modify the selected ADA from the ADA library. The required modification in ADA for HC-based implementation depends on the type of ADA. The detailed explanation of this step for the selected ADA is presented in Sections IV-A, IV-B and IV-C.

Finally, the generated FMs are used by the modified ADAs to generate the adversarial noise. Then this noise is inserted into the original FMs to generate modified FMs (mFM). These modified FMs are fed to the subsequent layer, leading to misclassification. The detailed implementation of the selected ADAs. i.e., GNA, FGSM, and JSMA, in an HC environment, along with their evaluation, are presented in the subsequent subsections.

A. MODIFIED GAUSSIAN NOISE ATTACK (mGNA)

In the proposed MAHCI, the modification of GNA for HC-based environment requires three steps after extracting FMs of the targeted layer(s) of CNN, illustrated in Fig. 3-A:

1) First, computation of the size of the FM of the targeted layer (S_{FM_t}) is made such that $FM_t = f(z)$, where z is the input of the targeted layer t.
2) Second, a noise vector y is generated following the Gaussian distribution such that $0 \leq y \leq 1$ and size(y) = S_{FM} and the standard deviation of 0.5.
3) Finally, generate the modified FM (mFM) by adding the noise vector y into the FM_t ($mFM = FM_t + y$).

[5] It shows how the input pixels affect the logit model outputs for various classes.

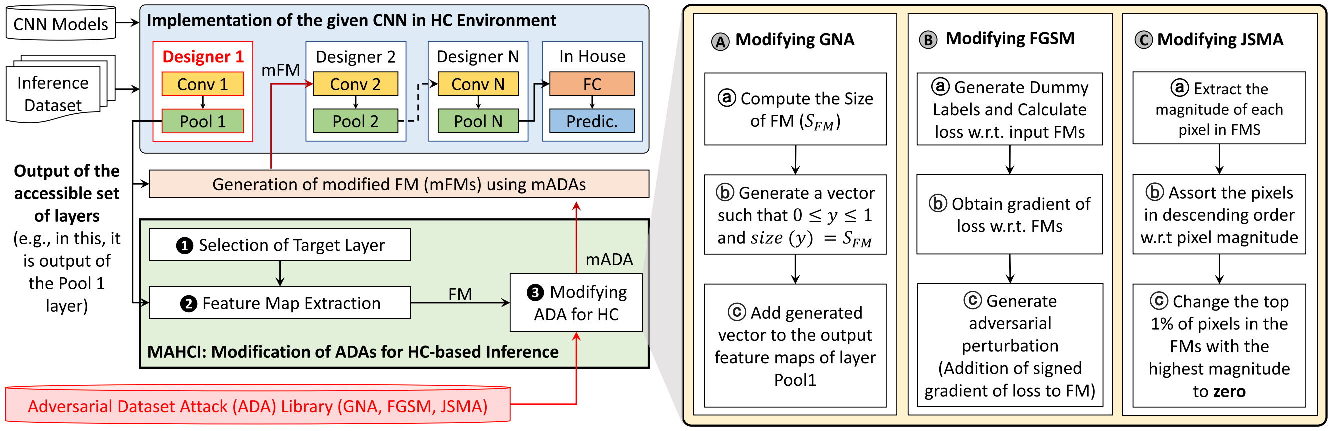
CNN Models
Inference Dataset
Implementation of the given CNN in HC Environment
Designer 1
Conv 1
Pool 1
mFM
Designer 2
Conv 2
Pool 2
Designer N
Conv N
Pool N
In House
FC
Predic.
Output of the accessible set of layers
(e.g., in this, it is output of the Pool 1 layer)
Generation of modified FM (mFMs) using mADAs
❶ Selection of Target Layer
❷ Feature Map Extraction
FM
❸ Modifying ADA for HC
mADA
MAHCI: Modification of ADAs for HC-based Inference
Adversarial Dataset Attack (ADA) Library (GNA, FGSM, JSMA)
Ⓐ Modifying GNA
ⓐ Compute the Size of FM (S_{FM})
ⓑ Generate a vector such that $0 \leq y \leq 1$ and $size\ (y) = S_{FM}$
ⓒ Add generated vector to the output feature maps of layer Pool1
Ⓑ Modifying FGSM
ⓐ Generate Dummy Labels and Calculate loss w.r.t. input FMs
ⓑ Obtain gradient of loss w.r.t. FMs
ⓒ Generate adversarial perturbation (Addition of signed gradient of loss to FM)
Ⓒ Modifying JSMA
ⓐ Extract the magnitude of each pixel in FMS
ⓑ Assort the pixels in descending order w.r.t pixel magnitude
ⓒ Change the top 1% of pixels in the FMs with the highest magnitude to zero

FIGURE 3. The proposed attack methodology, MAHCI, which modifies existing ADAs for HC-based inference, is illustrated step by step. Note, the sub-steps required to modify the GNA, FGSM, and JSMA are labeled A, B, and C, respectively, on the right of the figure.

1) PERFORMANCE EVALUATION

a: EXPERIMENTAL SETUP

To illustrate the effectiveness of the MAHCI for mGNA, we evaluated it on some of the common CNNs in the literature. For our experiments, we trained MiniVGGNet, LeNet, and CapsNet trained on MNIST dataset, and we also trained MiniVGGNet on Cifar-10 dataset. Furthermore, to ensure comprehensiveness, we performed the analysis for two different values of γ,[6] i.e., 0.25 (25% of the FMs are perturbed) and 0.5 (50% of the FMs are perturbed). Finally, we also compared mGNA with traditional GNA for the same experimental setup.

b: RESULTS

Fig. 5(a) shows some examples of FMs that are obtained at the output of the first convolution layer before (top-row) and after (bottom-row) performing the mGNA on LeNet trained for the MNIST dataset. Fig. 4 shows the results obtained from the evaluation of the traditional GNA and mGNA (GNA for HC-based inference) by applying different amounts of Gaussian noise to different DLA at the output of the first convolution layer. By analyzing these results, we made the following key observations:

- The mGNA is more powerful than the traditional GNA with lower perceptibility. For example, the accuracy drop in the case of mGNA on MiniVGGNet is 42.66% for $\gamma = 0.25$. The reason behind this observation is that mGNA introduces more localized noise in FMs rather than the input of the CNNs. Note, for higher perceptible noise, the performance of mGNA is comparable to GNA. For example, in the case of mGNA and GNA on MiniVGGNet for $\gamma = 0.5$, the accuracy drops are 53.26% and 57.39%, respectively, which are comparable to each other.
- Amongst all the evaluated DLA, CapsNet trained on the MNIST dataset performs worst in the case of mGNA, this indicates that the FM generated at the first convolution layer in CapsNet are very sensitive to the noise. For example, accuracy drop in the evaluated cases are 47.2% and 63.1% for for $\gamma = 0.25$ and $\gamma = 0.5$, respectively.
- GNA and mGNA are almost equally catastrophic for MiniVGGNet model trained on Cifar-10 dataset because it is very sensitive to noise. For example, the accuracy drop in the cases of GNA and mGNA for $\gamma = 0.25$ is 67.87% and 73.6%, and for $\gamma = 0.5$ is 70.58% and 67.79%, respectively.

B. MODIFIED FGSM ATTACK (mFGSM)

In HC-based inference, the attacker only has access to the input and output FMs of a target layer Li, which makes FGSM not applicable as it requires complete knowledge of the DLA. Therefore, in our proposed MAHCI technique, the modification of FGSM for the HC environment requires three steps after extracting FMs of the targeted layer(s) of CNN, as shown in Fig. 3-B (Algorithm 1, lines 6-9):

a First, the dummy labels for each FM (as shown in Algorithm 1) is generated because FGSM requires labels, which are not accessible to the attacker in HC-based inference. The labels (lb) are generated by obtaining the maximum of the Euclidean distance between FM_i and every other feature map (i.e. $FM_{i'}$ - where $i \neq i'$)

b Second, the loss value is calculated by taking the mean square error between each feature map and the generated label, as shown below:

$$loss_i = MSE(FM_i, lb_i) \tag{4}$$

Then the gradient of the loss function with respect to the input is obtained using the following equation:

$$gradient_i = \nabla_i(loss_i, f_{x_i}) \tag{5}$$

c After generating the gradients of the loss function, adversarial perturbations y are generated using the

[6] γ is defined as the percentage of FMs that an attacker targets.

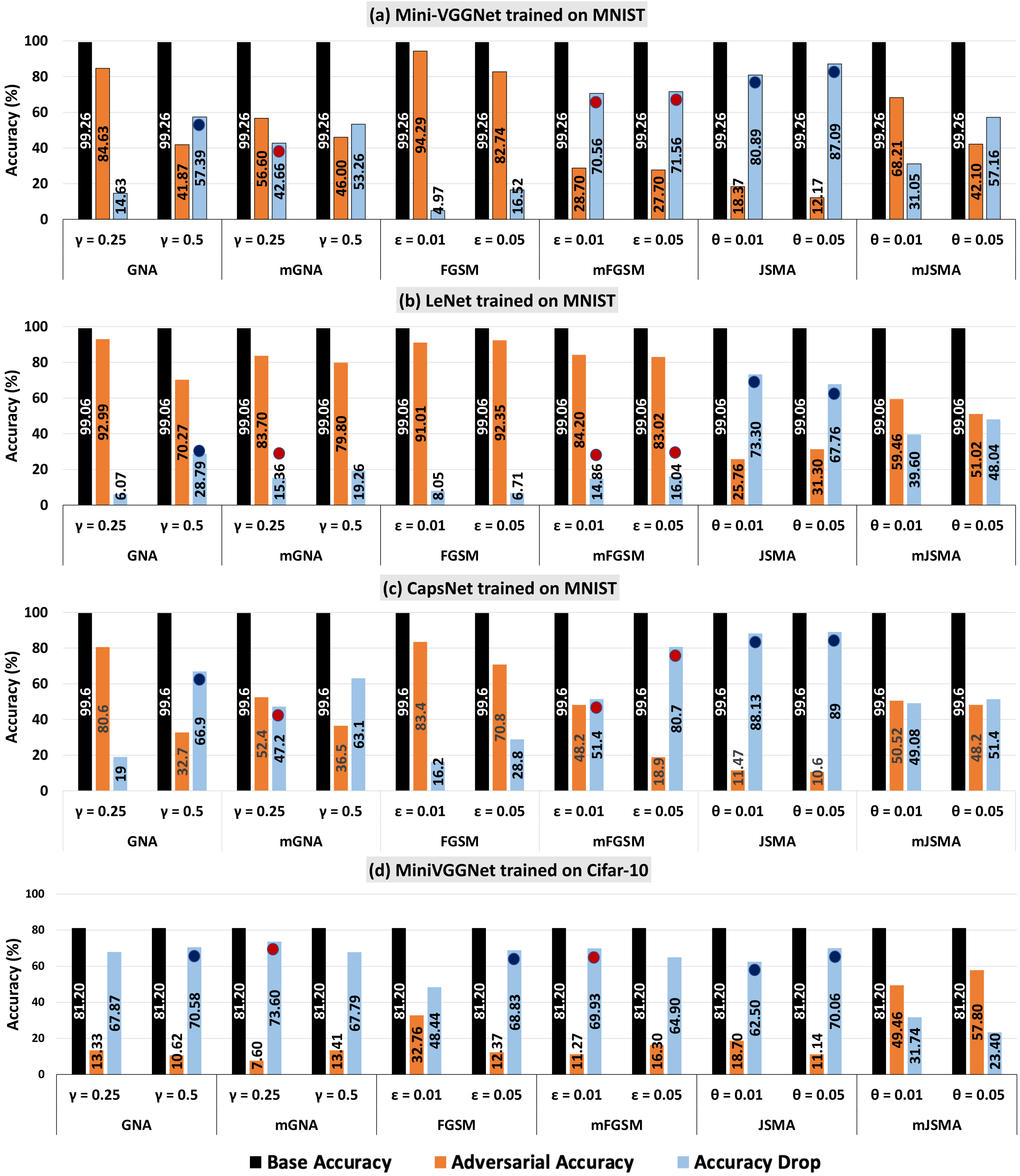

FIGURE 4. Baseline accuracy, adversarial accuracy, and accuracy drop while performing the ADAs and mADA for different CNNs trained on multiple datasets. Column tips are marked with blue circle dots if the drop in accuracy for traditional ADA is greater than the drop in accuracy for mADA and red if the drop in accuracy for mADA is greater than the drop in accuracy for traditional ADAs. Note: Adversarial accuracy is the classification accuracy of the targeted CNN during an ADA/mADA.

following equation:

$$y = \epsilon * sign(gradient) \qquad (6)$$

where ϵ represents a constant multiplier to ensure imperceptible perturbations. Finally, generate the modified FM (mFM) by adding perturbations y into the FM_t ($mFM = FM_t + y$).

1) PERFORMANCE EVALUATION

a: EXPERIMENTAL SETUP

To illustrate the effectiveness of the MAHCI for mFGSM, we evaluated it on the same experimental setup as discussed in Section IV-A1. We performed the analysis for two different values of ϵ, i.e., 0.01 and 0.05 (which are standard values used in the literature [56]). Small values of ϵ are used to ensure

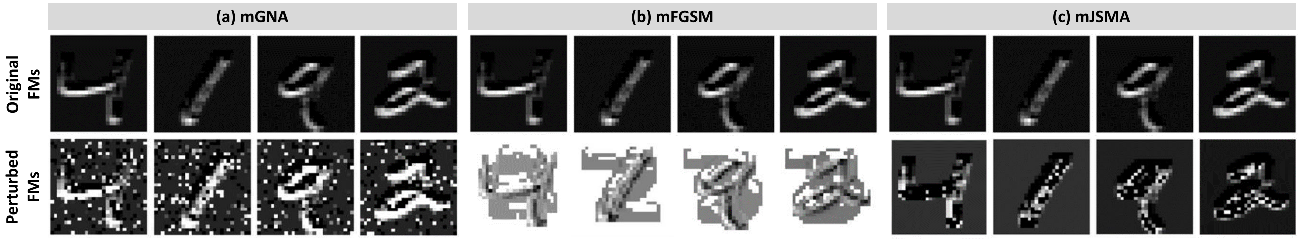

FIGURE 5. Visual examples of FMs obtained at the output of first convolution layer of LeNet before (top row) and after (bottom row) mADA on LeNet trained for MNIST. The FMs (bottom row) are perturbed with $\gamma = 0.25$ for mGNA, $\epsilon = 0.01$ for mFGSM, and $\theta = 0.01$ for mJSMA.

the adversarial perturbation is imperceptible, as described in the literature [56]. Finally, we also compared mFGSM with traditional FGSM for the same experimental setup.

Algorithm 1 Label Generation Process: Proposed Methodology to Generate Pseudo-Labels Used in Implementing mFGSM

Input: Trained CNN model, target CNN layer (l_i)
Input: OFM_i= Output FMs of l_i
Output: Pseudo-labels (lb_i)

```
function generate label
    for $i$ in $OFM$ do;
        $SumP_i$ = Sum $\sum P$; ▷ obtain sum of all pixels, P, in $i$
        for $j$ in $OFM$ do;        ▷ where $i \neq j$ is pixels
            $SumP_j$ = Sum$\sum P$
            $MaxSum$= $(SumP_i - SumP_j)^2$;        ▷ obtain square of difference between the FMs
            if $MaxSum_i < MaxSum_{i-1}$ then        ▷ stores maximum $MaxSum$
                $MaxSum_i = MaxSum_{i-1}$
            end if
        end for
        $lb_i = MaxSum_i$
    end for
    $Lb_i = [lb_0, lb_1, \ldots, lb_n]$
    return $Lb_i$
end function
```

b: RESULTS

Fig. 5(b) shows some examples of FMs that are obtained at the output of the first convolution layer before (top-row) and after (bottom-row) performing the mFGSM on LeNet trained for the MNIST dataset. Fig. 4 summarizes the results obtained from the evaluation of the traditional FGSM and mFGSM (FGSM for HC-based inference) by adding the perturbations with different imperceptibility constants at the output of the first convolution layer. By analyzing these results, we made the following key observations:

- The mFGSM is way more detrimental than the traditional FGSM in all the cases for which it has been evaluated. For example, the accuracy drop in the case of mFGSM on MiniVGGNet is 70.56% and 71.56% $\epsilon = 0.01$ and $\epsilon = 0.05$, respectively. The reason behind this observation is that mFGSM introduces more localized perturbations in FMs rather than the input of the CNNs.
- Among all the evaluated DLA, CapsNet trained for MNIST performs worst in the case of mFGSM. For example, accuracy drop in the evaluated cases are 51.40% and 80.70% for $\epsilon = 0.01$ and $\epsilon = 0.05$, respectively.
- FGSM and mFGSM are almost equally catastrophic for the MiniVGGNet model trained on Cifar-10. For example, the accuracy drop in the cases of FGSM and mFGSM for $\epsilon = 0.01$ is 48.44% and 69.90%, and for $\epsilon = 0.05$ is 68.83% and 64.90%, respectively.

C. MODIFIED JSMA (mJSMA)

Traditional JSMA works by saturating a few of the most important pixels in an image or FM to its maximum or minimum values [55], but in HC-based inference, the attacker does not have access to the input image. Hence, traditional JSMA needs to be modified for HC-based inference, leading to Modified JSMA (mJSMA). In MAHCI, the modification of JSMA for HC environment requires three steps after extracting FMs of the targeted layer(s) of CNN, as shown in Fig. 3-C:

a Firstly, the magnitude of each pixel in FMs is extracted.
b Secondly, these pixels are assorted in descending order with respect to pixel magnitude.
c Then we select a percentage of the pixels (depending on the θ value; e.g. if $\theta = 0.01$, then 1% of the highest magnitude pixels) of the FMs with respect to their magnitude and change them to zero, as described in the equation below. According to the literature, a small percentage of θ is required to ensure adversarial imperceptibility [57].

$$mFM = S^{+}(FM_{(i)}) = \begin{cases} 0 \; if \; (FM_{(i)}) > \\ top \; 1\% \; of \; input \; FM \\ \\ (FM_{(i)}) \; otherwise \end{cases} \tag{7}$$

Finally, newly generated FMs are fed as input into the next layer of DLA.

1) PERFORMANCE EVALUATION

a: EXPERIMENTAL SETUP

To elucidate the effectiveness of the mJSMA, we evaluated it on the same experimental setup as discussed in Section IV-A1. Furthermore, as stated earlier, to ensure comprehensiveness, we performed the analysis for two different values of saliency parameter,[7] θ, i.e., 0.01 (1% of the pixels in FMs are saturated) and 0.05 (5% of the pixels in FMs are saturated). Finally, we also compared mJSMA with traditional JSMA for the same experimental setup.

b: RESULTS

Fig. 5(c) shows some examples of FMs that are obtained at the output of the first convolution layer before (top-row) and after (bottom-row) performing the mJSMA on LeNet trained for the MNIST dataset. Fig 4 summarizes the results obtained from the evaluation of the traditional JSMA and mJSMA (JSMA for HC-based inference) by adding the perturbations with different saliency parameters at the output of the first convolution layer. By analyzing these results, we observed that mJSMA appears to be less lethal than the traditional JSMA across all DLAs, because traditional JSMA also directly affects the FMs (see the values in blue text of JSMA column of Fig. 4).

D. KEY INSIGHTS

1) *CapsNet trained for MNIST is the most vulnerable DLA to mADA in HC-based inference.* This could be due to the CNN model's unconventional layers, such as DigitCaps, which use dynamic routing where the lower-level capsule sends its input to the higher-level capsule that "agrees" with its input. Moreover, the adversarial noise generated by the proposed mADA is specialized to FMs.
2) None of the traditional adversarial noise attacks can be directly implemented in HC environments. All our mADA (namely, mGNA, mFGSM, and mJSMA) have been successfully deployed in HC environment. To give a perspective of the effectiveness of our attacks, with the stringent condition of our threat model, it is apparent that, as expected, our mJSMA has a lesser accuracy drop. While, counter-intuitively, mGNA and mFGSM have shown a higher accuracy drop. This is mostly attributed to the fact that traditional attacks use input images, and mADA in HC-environment use FMs, which have different dimensions.

In summary, implementation of HC-based inference can overcome the traditional ADA threats, but with a slight change in traditional ADA, mADA could be as lethal as traditional ADA in a normal setting or in some cases more lethal than traditional ADAs.

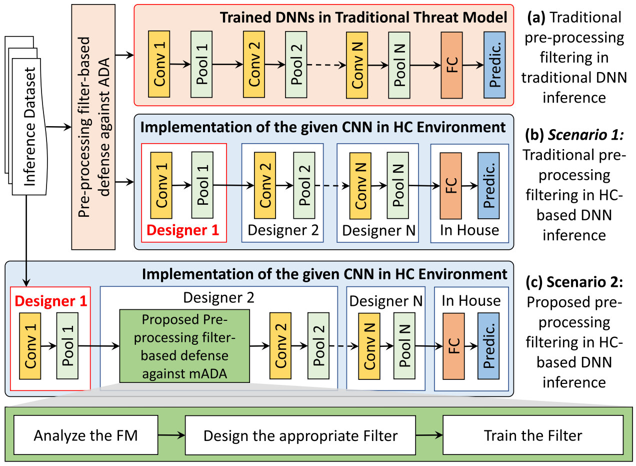
Inference Dataset
Pre-processing filter-based defense against ADA
Trained DNNs in Traditional Threat Model
Conv 1
Pool 1
Conv 2
Pool 2
Conv N
Pool N
FC
Predic.
(a) Traditional pre-processing filtering in traditional DNN inference
Implementation of the given CNN in HC Environment
Designer 1
Designer 2
Designer N
In House
(b) Scenario 1: Traditional pre-processing filtering in HC-based DNN inference
Implementation of the given CNN in HC Environment
Proposed Pre-processing filter-based defense against mADA
(c) Scenario 2: Proposed pre-processing filtering in HC-based DNN inference
Analyze the FM
Design the appropriate Filter
Train the Filter

FIGURE 6. Step-by-step procedure of the proposed pre-processing filter-based defense methodology against mADA. (a) Traditional pre-processing filters-based defense methodology in the traditional adversarial settings. (b) Scenario 1: traditional pre-processing filters-based defense methodology in the HC-based adversarial settings. (c) Scenario 2: proposed pre-processing filters-based defense methodology in the HC-based adversarial settings.

V. FILTER-BASED DEFENSE AGAINST MODIFIED ADAs IN HC ENVIRONMENTS

In literature, several techniques like Gradient masking [49], adversarial training [5] and pre-processing [33], [45] have been proposed as an adversarial defense strategy, but they cannot be directly applied in the case of HC-inference because of the following reasons:

1) Traditional pre-processing-based defenses are applied at the input of the CNN to cater the perturbations in input images (see Fig. 6(a and 5b)) [33], [45]. Whereas in mADA, adversarial perturbations can be added deeper into the DLA, e.g. Fig 5c, (the scenario 2) depicts that ADA is inserted after layer 1, hence it bypasses the traditional preprocessing filters (see Fig. 6(a and 5b)).
2) Gradient masking [49] is effective in mitigating gradient-based attacks in traditional deployment when an attacker does have access to input images in contrast to HC-environment, where an attacker cannot have access to input images. Hence, in HC-based settings where mADAs are designed to perturb FMs, gradient masking becomes ineffective as an adversary does not have the means to calculate the gradient of loss to perturb FMs.

A. PRE-PROCESSING FILTER-BASED DEFENSE

To address the above-mentioned limitations of the traditional defenses against ADA, we propose to use the pre-processing filters at the input of the subsequent trusted set of layers. The reason behind choosing the pre-processing filters is that other pre-processing functions like (e.g. quantization or pruning) complicate the DLA design and significantly impact the overall classification accuracy. Furthermore, research has shown

[7] θ is defined as the percentage of the pixels that are saturated to zero or to the maximum value in an FM.

that adversarial training on some attacks improves the robustness of CNNs on several common attacks in both white-box and black-box settings [58]. Designing the proposed pre-processing filters-based defense requires the following three steps, which is also illustrated in Fig. 6c:

1) **Analyzing the Feature Maps (FMs):** The first step of the proposed defense is to analyze the different parameters of the output FMs of the untrusted set of layers like FM size that helps to identify the appropriate filter size and kernel size.
2) **Designing the Filters:** The second step is to design the appropriate filter, which can enhance key features in the FMs and increase the robustness of the model while maintaining or increasing the accuracy. Towards this, we used an empirical approach by studying and experimenting with multiple filter configurations. Various experiments have been performed for different configurations of kernel sizes, number of kernels, and the number of convolutions layers. It is observed that the overall accuracy of the CNN improved as we increased the number of kernels till the number reached 128. Beyond which the accuracy dropped. Kernel sizes higher than 3 do not show better results either. Finally, in this work pre-processing filter with four convolution layers, having a number of kernels = 128 and kernel size = (3,3) is used. To insert our filter without disruption to the overall CNN architecture, the same padding is used for the convolution operation. This ensures the same input and output FM sizes. In addition, we use an Adam optimizer with learning rate decay. The learning rate is set to 0.001 and the decay factor at 0.1. Mean square error (MSE) is employed as the loss function in the training of the pre-processing filter for up to 20 epochs.
3) **Training the Pre-processing Filters:** The final step is to train the pre-processing filters against adversarial perturbations such as mGNA, mFGSM and mJSMA using adversarial FMs. Since, in contrast to the traditional method of deploying pre-processing filters, which aims to denoise input images, our proposed defense aims to denoise FMs of DLA layers in HC environments. Therefore, the goal of training is to obtain a denoising convolutional filter network to denoise perturbed input FMs.

1) PERFORMANCE EVALUATION

The proposed defense approach is evaluated on the same experimental setup as discussed in Section IV-A1. To give a fair perspective we compared our results with analyzed the traditional pre-processing filter-based defense in the HC-based inference settings. Fig. 7 shows the experimental analysis of the proposed defense along with the traditional pre-processing defense. By analyzing these results, we made the following key observations:

- In mGNA and mFGSM, traditional pre-processing defense either slightly improves adversarial accuracy or retains it. However, in mJSMA, it degrades it further because the traditional filter enhances features, and the enhanced features are fed to mJSMA, which effectively uses these features to develop a powerful saliency map.
- On the other-hand, the proposed pre-processing filters-based defense recovers the classification accuracy close to the baseline, which shows its generalization and effectiveness in nullifying the effects of mADAs. For example, on average, it recovers classification accuracy up to 90.5%, 90.4% and 89.29% when MiniVGGNet trained on MNIST was perturbed with mGNA (γ=0.5), mFGSM (ϵ = 0.05 and mJSMA (θ=0.05) respectively (see Fig. 7).
- On average, the implementation of the pre-processing filter results in a 5% increase in end-to-end latency for Mini-VGGNet trained on MNIST, a 4% increase for LeNet trained on MNIST, a 5% increase for CapsNet trained on MNIST, and a 6% increase for VGG16 trained on ImageNet when 200 images are used for inference, as compared to the absence of such filtering on a computer equipped with an Intel i5 processor and 8gb of RAM operating in an Ubuntu environment.

In summary, the results show that the proposed pre-processing filters can denoise the adversarial FMs and can be applied to any DLA layer(s).

B. MULTI-STRENGTH ADVERSARIAL TRAINING (MAT)

To further improve the proposed pre-processing defense, we adapted an adversarial training method that can be used for different strenghts of attacks (inspired from [59]) named as Multi-strength Adversarial Training (MAT). This involves pre-training the pre-processing convolution filter with several adversarial examples (FMs) at varying adversarial strengths before deploying it to the DLA layer(s). To illustrate the effectiveness of this method, we evaluated MAT approach on a MiniVGGNet trained on Cifar-10 dataset when a pre-processing filter is deployed against adversarial attacks on the first layer. In this evaluation, the pre-processing filter is trained using FMs generated by MiniVGGNet DLA trained on Cifar-10 dataset and perturbed with a combination of mGNA, mFGSM, and mJSMA. The choice of MiniVGGNet trained on Cifar-10 dataset is that in the previous performance evaluation, the proposed defense recovers the accuracy relatively lesser than the other cases. By analyzing these results, we observed that MAT is effective on mGNA and mJSMA, and works moderately for mFGSM attack (as shown in Fig. 8).

C. KEY INSIGHT

The traditional pre-processing filters cannot be applied in an HC-based inference because, in the case of an attack that mainly targets the FMs, e.g., JSMA, pre-processing filters enhance the most important FMs. Then these enhanced features are used to generate more effective attack maps, which leads to further degradation in the overall accuracy. On the

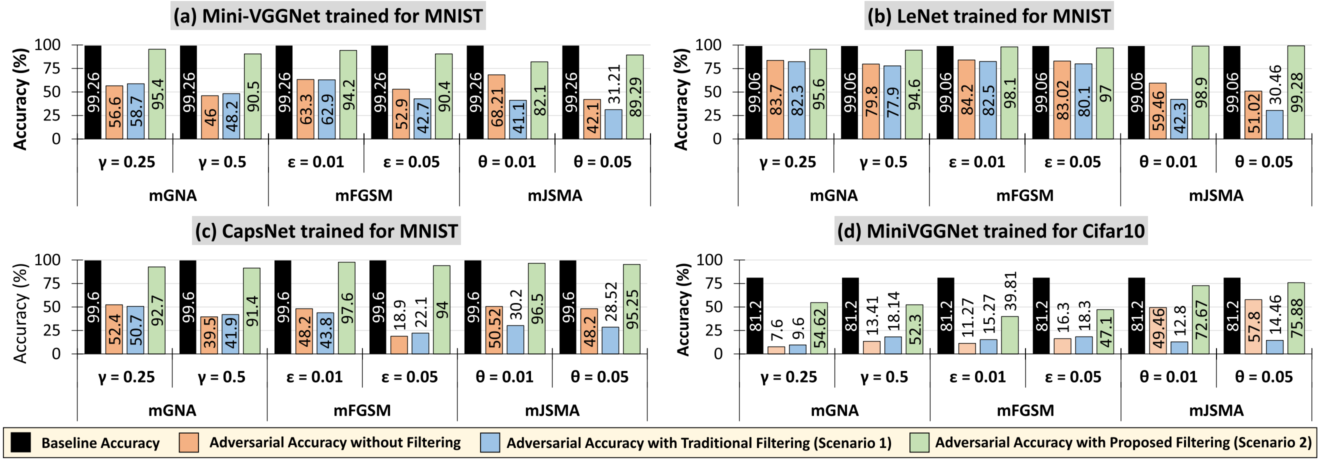

FIGURE 7. The graphs represent the performance evaluation of the proposed pre-processing filter-based defense against mADA when the attacks are performed for different CNNs trained on multiple datasets. Note, in this figure, adversarial accuracy is the classification accuracy of the targeted CNN during an ADA/mADA.

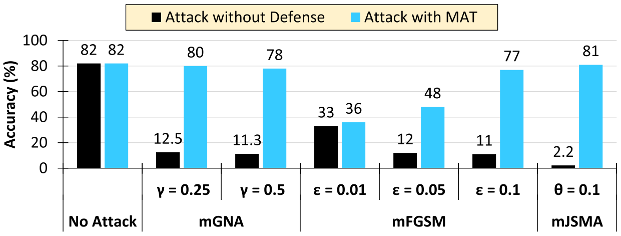

FIGURE 8. Performance evaluation of the proposed MAT defense against mADAs with different perceptibility parameters shows that in most cases, the proposed MAT recovers the classification accuracy to baseline accuracy. Note, in this figure, pre-processing filters are trained using the Cifar-10 dataset perturbed with a combination of Gaussian Noise, FGSM, and JSMA.

TABLE 1. Comparison of our defense approach with state-of-the-art.

Criteria	[60]	[61]	[62]	[63]	[64]	[65]	[66]	Ours
Req. full CNN Architecture	✓	✓	✓	✓	✓	✓	✓	X
Req. knowledge of input image	✓	✓	✓	✓	✓	✓	X	X
Req. knowledge of training params.	✓	✓	✓	✓	✓	✓	X	X
Modify classifier structure	X	✓	X	X	✓	X	✓	X

other hand, the proposed localized pre-processing filters can locally denoise the adversarial perturbations at the layer level, significantly improving classification accuracy.

VI. SCALABILITY: HOW MAHCI AND PROPOSED FILTERING WORKS FOR LARGE DLA IN HC ENVIRONMENT

To improve the accuracy of CNN models, the depth of the CNN is often increased, resulting in an increase in the number of model parameters. For example, a LeNet [67] model has 5 trainable layers with the total number of parameters is *60,000*, a CapsuleNet model [34] has 6 trainable layers with *11.6 Million*,parameters and a VGG16 model [68] has 16 trainable layers with *138 Million* parameters. Since adding more convolution layers increases the robustness of CNN models [66], it is critical to evaluate our proposed approaches on a bigger CNN model. To validate the scalability of proposed attacks and defenses, we perturbed the output FMs of the first pooling layer (L1) of a VGG16 model trained on the ImageNet dataset using mGNA, mFGSM, and mJSMA. Our choice of $L1$ stems from the findings of [66], who observed that the first layer of the CNN model is the most secure because an attack on the CNNs first layer can be neutralized by convolution operations in subsequent layers. This implies that a successful attack on the first layer of the CNN is almost certain to be successful on other layers of the CNN, making the first layer the most robust to adversarial attacks. To this end, we selected $L1$ as a case study for both attacks and defense. As shown in Fig. 9, we see a classification accuracy drop across all types of attacks, with mJSMA (θ=0.05) being the most potent, causing a 71.5% accuracy drop. With an accuracy drop of 23.5%, GNA (γ = 0.25) has the least accuracy drop. Furthermore, using the proposed convolution filter trained on adversarial FMs improved the model's top-1 classification accuracy, demonstrating the efficacy of our methodology. Fig. 9 further shows the accuracy obtained when both the traditional convolution filter and the proposed pre-processing are deployed. We note that deploying the traditional convolution filter to mitigate attacks in HC settings is ineffective. However, the pre-processing convolution filter works best in mitigating mJSMA ($\theta = 0.01$), giving a top-1 classification accuracy of 85.2%.

In addition, to demonstrate that the attacks and defenses are applicable and scalable on any layer of a CNN model, we perturbed the last convolution layer ($L17$) of the VGG16 model trained on ImageNet using mFGSM. As shown in Fig. 10, we see an accuracy drop of 65.6% for $\epsilon = 0.01$ and 73.3% for $\epsilon = 0.05$. Similarly, according to the results in Fig 9, deploying a traditional pre-processing filter has no significant impact on mitigating adversarial noise in HC environments. We also note that using the proposed pre-processing

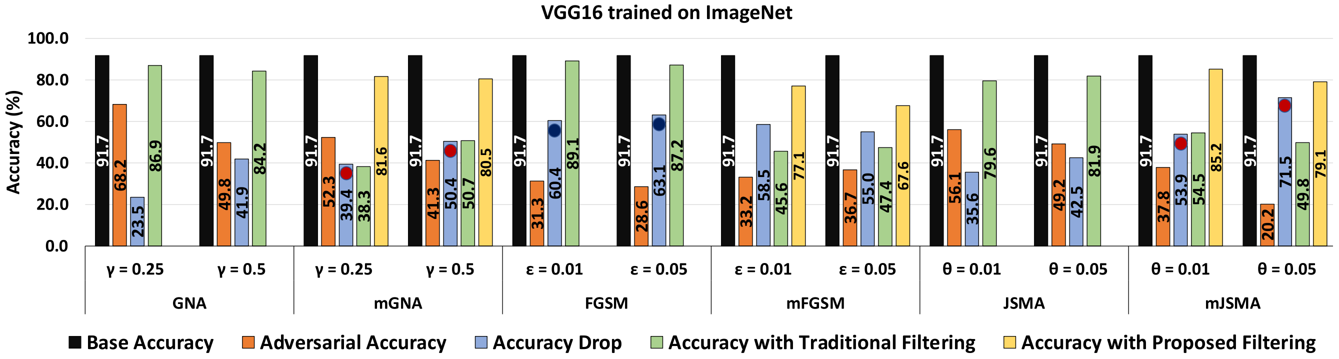

FIGURE 9. Baseline accuracy, adversarial accuracy, accuracy drop, and accuracy with the proposed filter while performing mADA for VGG16 model trained on ImageNet. Column tips are marked with blue circle dots if the drop in accuracy for traditional ADA is greater than the drop in accuracy for mADA and red if the drop in accuracy for mADA is greater than the drop in accuracy for traditional ADAs.

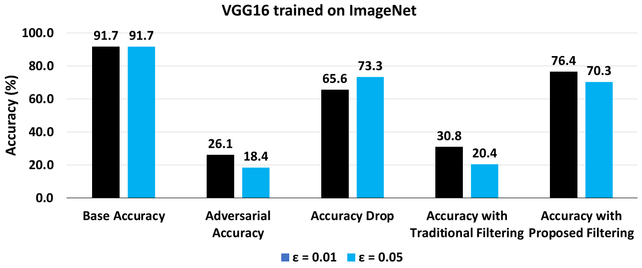

FIGURE 10. Baseline accuracy, adversarial accuracy, accuracy drop, accuracy with traditional filter, and accuracy with the proposed filter while performing mFGSM on layer 17 (*L*17) of a VGG16 model trained on ImageNet.

increases model accuracy to 76.4% for $\epsilon = 0.01$ and 70.3% for $\epsilon = 0.05$, confirming the efficacy of our methodology in HC-based environments.

VII. COMPARING DEFENSE TECHNIQUE WITH STATE-OF-THE-ART

We note that the majority of cutting-edge CNN adversarial attacks and defense methods are targeted at complete CNN architectures in white and black-box scenarios [60], [61], [62], [63], [64], [65]. In this paper, we investigated an HC environment where a trained CNN model is split and deployed to two or more edge devices (usually, with varying capabilities/configurations) successively accelerate inference. As a result, our adversarial attacks are developed under more stringent conditions in which only FMs of participating edge devices can be manipulated. Table 1 summarizes the differences between our work and other state-of-the-art approaches. The comparison is based on the amount of information available to the defender and the need to alter the CNN structure to accommodate defense techniques.

In [60], an auto-encoder (AE) is trained alongside a DNN visual classifier using the same encoding weights. During inference, the classifier works directly over the AE's internal latent representation, ensuring that any redundant information from the input image is removed, preserving prediction. This method, however, requires knowledge of the entire CNN architecture as well as knowledge of training data, but the original classifier structure is preserved. In [61] an adversarial training defense based on Generative Adversarial Networks (GAN) was proposed. This method, however, requires knowledge of the entire CNN architecture, as well as training data, and the classifier structure would be modified. In [62], an attack-agnostic adversarial defensive method that employs a novel Sparse Transformer Layer to transform images so that corresponding clean and adversarial images can be distinguished in the quasi-natural image and feature space was described. Although the original classifier structure is preserved, this method does require knowledge of the entire CNN architecture and training data. In [63], the authors proposed a technique using class activation map responses obtained for numerous top-ranking class labels to reconstruct small and carefully chosen image areas that are most important to the current classification outcome. However, this approach necessitates knowledge of the CNN architecture, training data, and classifier structure. In [64], the authors proposed a methodology to progressively align the intermediate feature representations extracted from the adversarial domain with feature representations extracted from a clean domain through domain adaptation. This method however requires the knowledge of the complete CNN architecture, and knowledge of training data, but the original classifier structure is maintained. In [65], a methodical approach to thwart adversarial attacks on graph neural networks (GNNs) is employed, using a Bayesian uncertainty technique to locate and take advantage of hierarchical uncertainties in GNNs. This technique would require altering the CNN structure during the training of the model, so this method necessitates knowledge of the entire CNN architecture as well as training data. A defense technique for detecting adversarial inputs to CNN was proposed in [66] by comparing predictions at different layers of a CNN and noting prediction inconsistency among the CNN layers. This methodology requires full knowledge of CNN architecture as well needs modification of the classifier structure. We note that our proposed methodology does not necessitate full CNN architecture, knowledge of input image, training parameters, or modification of classifier structure. This is so because an attacker's knowledge in an HC-based inference is only confined to the input and output of the device he has access to.

VIII. CONCLUSION

Broadly, this paper has investigated adversarial attacks on edge devices in HC environments. These attacks allow an adversary without knowledge of the DNN architecture or input images to construct adversarial attacks based on local information available on output FMs of a compromised edge device. Our technique provides a way of modifying the traditional state-of-the-art white-box attacks and achieves an average accuracy drop of 62% when tested on four CNN models (LeNet, CapsuleNet, MiniVGGNet, and VGG16). Furthermore, a CNN-based filter trained on adversarial FMs has been designed and deployed after empirically analyzing the parameters of DLA. This filter is strategically placed at the output of untrusted nodes. The performance evaluation showed the efficacy of the deployed convolution filter in alleviating the adverse effect of adversarial attacks in all evaluated cases. The proposed filtering approach is able to mitigate the attack by recovering the actual accuracy back to 75.1% on average, which is substantially better than traditional filtering approaches. To the best of our knowledge, this is the first study that examines the security vulnerability of DLA in the HC environment against adversarial attacks, and all three of our attacks are scalable and independent of the DLA's partition location. Similarly, we are the first to explicitly explore pre-processing noise filtering under the stringent requirements of the HC environment.

REFERENCES

[1] R. A. Khamis and A. Matrawy, "Evaluation of adversarial training on different types of neural networks in deep learning-based IDSs," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2020, pp. 1–6.

[2] T. A. Odetola, K. M. Groves, Y. Mohammed, F. Khalid, and S. R. Hasan, "2L–3W: 2-level 3-way hardware–software co-verification for the mapping of convolutional neural network (CNN) onto FPGA boards," *Social Netw. Comput. Sci.*, vol. 3, no. 1, pp. 1–25, Jan. 2022.

[3] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" 2019, *arXiv:1904.12843*.

[4] *Experimental Security Research of Tesla Autopilot NT*, Tencent Keen Security Lab, Shenzhen, China, 2019.

[5] N. Morgulis, A. Kreines, S. Mendelowitz, and Y. Weisglass, "Fooling a real car with adversarial traffic signs," 2019, *arXiv:1907.00374*.

[6] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597.

[7] M. Goldblum, L. Fowl, S. Feizi, and T. Goldstein, "Adversarially robust distillation," in *Proc. AAAI*, 2020, vol. 34, no. 4, pp. 3996–4003.

[8] L. Zeng, X. Chen, Z. Zhou, L. Yang, and J. Zhang, "CoEdge: Cooperative DNN inference with adaptive workload partitioning over heterogeneous edge devices," *IEEE/ACM Trans. Netw.*, vol. 29, no. 2, pp. 595–608, Apr. 2020.

[9] F. Xue, W. Fang, W. Xu, Q. Wang, X. Ma, and Y. Ding, "EdgeLD: Locally distributed deep learning inference on edge device clusters," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun., IEEE 18th Int. Conf. Smart City, IEEE 6th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2020, pp. 613–619.

[10] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 447–457, Jan. 2020.

[11] M. Xue, H. Wu, R. Li, M. Xu, and P. Jiao, "EosDNN: An efficient offloading scheme for DNN inference acceleration in local-edge-cloud collaborative environments," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 248–264, Mar. 2022.

[12] Z. Fu, Y. Zhou, C. Wu, and Y. Zhang, "Joint optimization of data transfer and co-execution for DNN in edge computing," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2021, pp. 1–6.

[13] X. Tang, X. Chen, L. Zeng, S. Yu, and L. Chen, "Joint multiuser DNN partitioning and computational resource allocation for collaborative edge intelligence," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9511–9522, Jun. 2021.

[14] C. Gong, F. Lin, X. Gong, and Y. Lu, "Intelligent cooperative edge computing in Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9372–9382, Apr. 2020.

[15] Y. Fang, Z. Jin, and R. Zheng, "TeamNet: A collaborative inference framework on the edge," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1487–1496.

[16] C. Hu, Y. Bai, R. Wang, C. Liu, and X. Wang, "CCIED: Cache-aided collaborative intelligence between edge devices," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun., IEEE 18th Int. Conf. Smart City, IEEE 6th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2020, pp. 668–673.

[17] J. Mao, X. Chen, K. W. Nixon, C. Krieger, and Y. Chen, "MoDNN: Local distributed mobile computing system for deep neural network," in *Proc. IEEE DATE*, Mar. 2017, pp. 1396–1401.

[18] Z. Zhao, K. M. Barijough, and A. Gerstlauer, "DeepThings: Distributed adaptive deep learning inference on resource-constrained IoT edge clusters," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2348–2359, Nov. 2018.

[19] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[20] C.-C. Hung, G. Ananthanarayanan, P. Bodik, L. Golubchik, M. Yu, P. Bahl, and M. Philipose, "VideoEdge: Processing camera streams using hierarchical clusters," in *Proc. ACM/IEEE SEC*, Oct. 2018, pp. 115–131.

[21] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *Proc. IEEE ICDCS*, Jun. 2017, pp. 328–339.

[22] A. E. Eshratifar and M. Pedram, "Runtime deep model multiplexing for reduced latency and energy consumption inference," in *Proc. IEEE ICCD*, Oct. 2020, pp. 263–270.

[23] J. Mao, Z. Yang, W. Wen, C. Wu, L. Song, K. W. Nixon, X. Chen, H. Li, and Y. Chen, "MeDNN: A distributed mobile system with enhanced partition and deployment for large-scale DNNs," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2017, pp. 751–756.

[24] A. Adeyemo, T. Sandefur, T. A. Odetola, and S. R. Hasan, "Towards enabling dynamic convolution neural network inference for edge intelligence," 2022, *arXiv:2202.09461*.

[25] D. Gragnaniello, F. Marra, G. Poggi, and L. Verdoliva, "Analysis of adversarial attacks against CNN-based image forgery detectors," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 967–971.

[26] Y. Liu, L. Wei, B. Luo, and Q. Xu, "Fault injection attack on deep neural network," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2017, pp. 131–138.

[27] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models," 2018, *arXiv:1806.01246*.

[28] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff Nets: Stealing functionality of black-box models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4954–4963.

[29] W. Hua, Z. Zhang, and G. E. Suh, "Reverse engineering convolutional neural networks through side-channel information leaks," in *Proc. 55th Annu. Design Autom. Conf.*, Jun. 2018, pp. 1–6.

[30] T. Odetola, F. Khalid, T. Sandefur, H. Mohammed, and S. R. Hasan, "FeSHI: Feature map based stealthy hardware intrinsic attack," 2021, *arXiv:2106.06895*.

[31] A. Adeyemo, F. Khalid, T. Odetola, and S. R. Hasan, "Security analysis of capsule network inference using horizontal collaboration," in *Proc. IEEE Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2021, pp. 1074–1077.

[32] H. Mohammed, T. A. Odetola, and S. R. Hasan, "How secure is distributed convolutional neural network on IoT edge devices?" 2020, *arXiv:2006.09276*.

[33] H. Ali, F. Khalid, H. A. Tariq, M. A. Hanif, R. Ahmed, and S. Rehman, "SSCNets: Robustifying DNNs using secure selective convolutional filters," *IEEE Des. Test*, vol. 37, no. 2, pp. 58–65, Apr. 2020.

[34] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," 2017, *arXiv:1710.09829*.

[35] S. Wessels and D. V. D. Haar, "Applying deep learning for the detection of abnormalities in mammograms," in *Information Science and Applications*. Cham, Switzerland: Springer, 2020, pp. 201–210.

[36] F. Khalid, M. A. Hanif, and M. Shafique, "Exploiting vulnerabilities in deep neural networks: Adversarial and fault-injection attacks," 2021, *arXiv:2105.03251*.

[37] M. Shafique, T. Theocharides, C.-S. Bouganis, M. A. Hanif, F. Khalid, R. Hafiz, and S. Rehman, "An overview of next-generation architectures for machine learning: Roadmap, opportunities and challenges in the IoT era," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2018, pp. 827–832.

[38] M. A. Hanif, F. Khalid, R. V. W. Putra, S. Rehman, and M. Shafique, "Robust machine learning systems: Reliability and security for deep neural networks," in *Proc. IEEE 24th Int. Symp. On-Line Test. Robust Syst. Design (IOLTS)*, Jul. 2018, pp. 257–260.

[39] T. A. Odetola and S. R. Hasan, "SoWaF: Shuffling of weights and feature maps: A novel hardware intrinsic attack (HIA) on convolutional neural network (CNN)," in *Proc. IEEE ISCAS*, May 2021, pp. 1–5.

[40] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," *ACM SIGARCH Comput. Archit. News*, vol. 45, no. 1, pp. 615–629, 2017.

[41] Y. Chen, S. Kar, and J. M. F. Moura, "The Internet of Things: Secure distributed inference," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 64–75, Sep. 2018.

[42] W. Rouse, N. Geddes, and R. Curry, "An architecture for intelligent interfaces: Outline of an approach to supporting operators of complex systems," *Hum.-Comput. Interact.*, vol. 3, no. 2, pp. 87–122, Jun. 1987.

[43] F. Khalid, S. Rehman, and M. Shafique, "Overview of security for smart cyber-physical systems," in *Security of Cyber-Physical Systems*. Cham, Switzerland: Springer, 2020, pp. 5–24.

[44] F. Khalid, H. Ali, H. Tariq, M. A. Hanif, S. Rehman, R. Ahmed, and M. Shafique, "QuSecNets: Quantization-based defense mechanism for securing deep neural network against adversarial attacks," in *Proc. IEEE 25th Int. Symp. On-Line Test. Robust Syst. Design (IOLTS)*, Jul. 2019, pp. 182–187.

[45] F. Khalid, M. A. Hanif, S. Rehman, J. Qadir, and M. Shafique, "FAdeML: Understanding the impact of pre-processing noise filtering on adversarial machine learning," in *Proc. IEEE DATE*, Mar. 2019, pp. 902–907.

[46] E. Raff, J. Sylvester, S. Forsyth, and M. McLean, "Barrage of random transforms for adversarially robust defense," in *Proc. IEEE CVPR*, Jun. 2019, pp. 6528–6537.

[47] M. H. Samavatian, S. Majumdar, K. Barber, and R. Teodorescu, "HASI: Hardware-accelerated stochastic inference, a defense against adversarial machine learning attacks," 2021, *arXiv:2106.05825*.

[48] R. Bao, S. Liang, and Q. Wang, "Featurized bidirectional GAN: Adversarial defense via adversarially learned semantic inference," 2018, *arXiv:1805.07862*.

[49] I. Goodfellow, "Gradient masking causes CLEVER to overestimate adversarial perturbation size," 2018, *arXiv:1804.07870*.

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

[51] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "PRADA: Protecting against DNN model stealing attacks," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Jun. 2019, pp. 512–527.

[52] D. Wu, S.-T. Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," 2020, *arXiv:2004.05884*.

[53] B. Kailkhura, V. S. S. Nadendla, and P. K. Varshney, "Distributed inference in the presence of eavesdroppers: A survey," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 40–46, Jun. 2015.

[54] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020.

[55] R. Wiyatno and A. Xu, "Maximal jacobian-based saliency map attack," 2018, *arXiv:1808.07945*.

[56] Z. Dou, S. J. Osher, and B. Wang, "Mathematical analysis of adversarial attacks," 2018, *arXiv:1811.06492*.

[57] I. Alarab and S. Prakoonwit, "Adversarial attack for uncertainty estimation: Identifying critical regions in neural networks," *Neural Process. Lett.*, vol. 54, no. 3, pp. 1805–1821, Jun. 2022.

[58] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[59] C. Song, H.-P. Cheng, H. Yang, S. Li, C. Wu, Q. Wu, and H. Li, "Adversarial attack: A new threat to smart devices and how to defend it," *IEEE Consum. Electron. Mag.*, vol. 9, no. 4, pp. 49–55, Jul. 2020.

[60] W. Liu, M. Shi, T. Furon, and L. Li, "Defending adversarial examples via DNN bottleneck reinforcement," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 1930–1938.

[61] G. Liu, I. Khalil, and A. Khreishah, "GanDef: A GAN based adversarial training defense for neural network classifier," in *Proc. IFIP Int. Conf. ICT Syst. Secur. Privacy Protection*. Cham, Switzerland: Springer, 2019, pp. 19–32.

[62] B. Sun, N.-H. Tsai, F. Liu, R. Yu, and H. Su, "Adversarial defense by stratified convolutional sparse coding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11447–11456.

[63] P. Gupta and E. Rahtu, "CIIDefence: Defeating adversarial attacks by fusing class-specific image inpainting and image denoising," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6708–6717.

[64] X. Zhang, J. Wang, T. Wang, R. Jiang, J. Xu, and L. Zhao, "Robust feature learning for adversarial defense via hierarchical feature alignment," *Inf. Sci.*, vol. 560, pp. 256–270, Jun. 2021.

[65] B. Feng, Y. Wang, and Y. Ding, "UAG: Uncertainty-aware attention graph neural network for defending adversarial attacks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 8, pp. 7404–7412.

[66] T. A. Odetola, A. Adeyemo, and S. R. Hasan, "Hardening hardware accelerartor based CNN inference phase against adversarial noises," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Jun. 2022, pp. 141–144.

[67] Y. LeCun. (2015). *LeNet-5, Convolutional Neural Networks*. [Online]. Available: http://yann.lecun.com/exdb/lenet

[68] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*.

ADEWALE A. ADEYEMO (Graduate Student Member, IEEE) received the Bachelor of Science degree in electronic and electrical engineering from the Ladoke Akintola University of Technology, Nigeria, and the master's degree in satellite communications from the Federal University of Technology, Akure, Nigeria. He is currently pursuing the Ph.D. degree with Tennessee Technological University, Tennessee, USA, where he is also continually researching ways to use machine learning to solve difficult organizational problems. His current research interests include machine learning, deep learning, and hardware security.

JONATHAN J. SANDERSON received the bachelor's degree in computer engineering from Tennessee Tech University, where he is currently pursuing the masters' degree in electrical and computer engineering. His research interests include FPGAs, machine learning, and edge computing.

TOLULOPE A. ODETOLA received the bachelor's degree in electronic and electrical engineering from Obafemi Awolowo University, Nigeria, and the master's degree in electrical and computer engineering from Tennessee Technological University, Cookeville, TN, USA, where he is currently pursuing the Ph.D. degree. He has developed hardware/software co-verification techniques for CNN deployments on hardware accelerators. His current research interests include FPGAs, hardware security, the IoT security, machine learning, and deep learning.

FAIQ KHALID (Member, IEEE) received the B.E. degree in electronics engineering and the M.S. degree in electrical engineering from the National University of Sciences and Technology (NUST), Pakistan, in 2011 and 2016, respectively. He is currently pursuing the Ph.D. degree in hardware security and machine learning security with Technische Universität Wien (TU Wien), Vienna, Austria. He was a recipient of the Richard Newton Young Fellowship Award at DAC, in 2018. His research interests include formal verification of embedded systems, hardware design security, and security for machine learning systems. He has also served as a TPC Member for FIT, WSAV, ARES, and ICONS.

SYED RAFAY HASAN (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from the NED University of Engineering and Technology, Pakistan, and the M.Eng. and Ph.D. degrees in electrical engineering from Concordia University, Montreal, QC, Canada. From 2006 to 2009, he was an Adjunct Faculty Member with Concordia University. From 2009 to 2011, he was a Research Associate with the Ecole Polytechnique de Montreal. Since 2011, he has been with the Electrical and Computer Engineering Department, Tennessee Technological University, Cookeville, TN, USA, where he is currently an Associate Professor. He has published more than 90 peer-reviewed journals and conference papers. His current research interests include hardware design security in the Internet of Things (IoT), hardware implementation of deep learning, deployment of convolution neural networks in the IoT edge devices, and hardware security issues due to adversarial learning. He received the SigmaXi Outstanding Research Award, the Faculty Research Award from Tennessee Technological University, the Kinslow Outstanding Research Paper Award from the College of Engineering, Tennessee Tech University, and the Summer Faculty Fellowship Award from the Air force Research Laboratory (AFRL). He has received research and teaching funding from NSF, ICT-funds UAE, AFRL, and Intel Inc. He has been a part of several funded research projects, as a PI or a co-PI, worth more than $1.1 million. He has been the Session Chair and a Technical Program Committee Member of several IEEE conferences, including ISCAS, ICCD, MWSCAS, and NEWCAS, and a regular reviewer for several IEEE Transactions and other journals.