MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM) | 979-8-3503-7423-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/MILCOM61039.2024.10773813

HyperAdv: Dynamic Defense Against Adversarial Radio Frequency Machine Learning Systems

Milin Zhang[×], Michael De Lucia[‡], Ananthram Swami[‡],
Jonathan Ashdown[*], Kurt Turck[*] and Francesco Restuccia[×]
[‡] DEVCOM Army Research Laboratory, United States
[*] Air Force Research Laboratory, United States
[×] Institute for the Wireless Internet of Things, Northeastern University, United States

***Abstract*—Radio Frequency Machine Learning Systems (RFMLS) have attracted increasing interest over the past few years. However, it has been demonstrated that RFMLS are vulnerable to Adversarial Machine Learning (AML). While AML has been extensively investigated in traditional domains, current state of the art often compromises the performance on benign data or introduces excessive computational overhead. As such, it cannot meet the strict requirements of tactical RFMLS. In this paper, we propose a novel defense approach based on dynamic adaptation of Deep Neural Network (DNN). Specifically, we leverage a hypernetwork to dynamically generate diverse parameters for a target DNN during inference. In addition, an ensemble learning and multi-stage training framework is proposed to train such a hypernetwork. Experimental results show that the proposed defense can increase the accuracy on adversarial examples by 48% and 16% in comparison to naturally trained DNN and defensive training strategies, respectively.**

I. Introduction

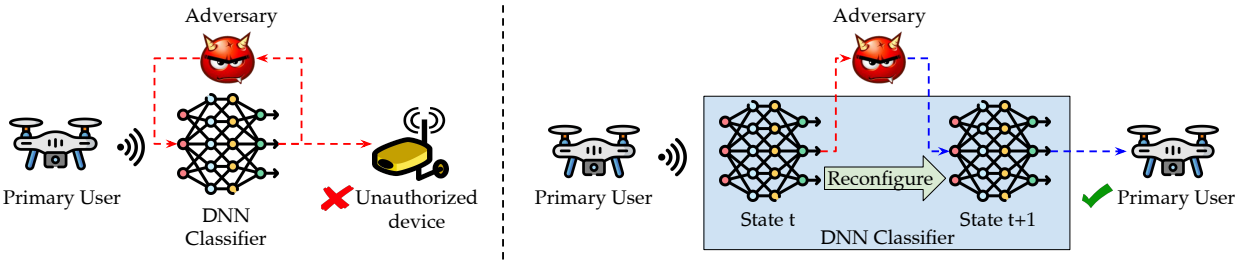
DNNs have achieved significant success in many tactical Radio Frequency Machine Learning Systems (RFMLS) such as signal classification [1], spectrum sensing [2], and radio fingerprinting [3], among others. However, it was demonstrated in [4] that adding malicious perturbations to input data can result in a significant performance loss for DNNs. This aspect has been investigated in the literature as Adversarial Machine Learning (AML), which aims at revealing the vulnerabilities of DNNs as well as improving robustness to adversarial perturbations. While a generalized framework of AML in wireless has been investigated [5], there does not exist a generalized approach to improve adversarial robustness for wireless tasks. On the other hand, current state-of-the-art defense approaches [6–8] for computer vision tasks cannot meet the needs of RFMLS. For example, although Adversarial Training (AT) [6] leverages malicious inputs to improve robustness during training, it suffers significant performance loss on benign data [9]. Other approaches such as certified robustness [7] and input purification [8] require additional computation cost that can lead to excessive latency for the tactical wireless domain.

In contrast to the conventional defense mechanisms that train a static robust DNN classifier [6, 7] or utilize static denoising DNN [8], we investigate AML from a dynamic perspective. As depicted in Fig. 1, a powerful adversarial attack such as Projected Gradient Descent (PGD) [6] can often compromise DNN robustness by iteratively updating the perturbation based on the gradient information of the DNN. To this end, a feasible defense approach to improve DNN robustness can be achieved by dynamically changing the parameters of the DNN, thus resulting in different gradients. Consequently, adversarial updates based on the previous DNN gradient may not be effective for the new DNN model.

We propose HyperAdv, a novel dynamic DNN framework based on hypernetworks [10], which generates different parameters for the DNN during inference. The changing DNN parameters enhance adversarial robustness by varying gradient direction at each iteration, hence posing a challenge for attackers to find an effective adversarial gradient update. Moreover, a novel ensemble learning approach is proposed to diversify DNN parameters. The proposed approach first projects the logits of the DNN to a different space via random affine transformations. Then, parallel ensemble learning is used to optimize the projected logit space. To this end, even if ensemble training learned a similar decision boundary for different projected logit spaces, original DNN mappings remain different, hence having a different gradient landscape. We evaluate our defense approach on the publicly available RadioML 2018.01A dataset [11]. Experimental results demonstrate that our approach can improve adversarial accuracy by up to 48% compared to naturally trained DNN without compromising clean accuracy. Moreover, HyperAdv can also be integrated with existing static defenses. Compared to AT [6], our approach improves robustness by over 16% and clean accuracy by approximately 8%. The key contributions are summarized as follows:

- We propose a novel dynamic DNN framework named HyperAdv that dynamically generates different weights for a target network during inference. Such dynamic design can improve adversarial robustness by changing the gradient update of adversaries without compromising performance;
- We propose a novel ensemble training approach to encourage the hypernetwork to generate diverse model parameters. In addition, we propose a multi-stage training approach to decrease the model complexity of hypernetwork. Our training approach can effectively improve the end-to-end performance as well as reduce the model size of HyperAdv;
- We evaluate our defense strategy with publicly available wireless dataset [11], demonstrating 48% improvement in robustness for naturally trained DNN and 16% improvement in

Approved for Public Release; Distribution Unlimited: AFRL-2024-2492.



(a) Static Adversarial Defense **(b) Dynamic Adversarial Defense: HyperAdv**

Fig. 1: (a) In static defense, adversaries can craft effective perturbations by iteratively querying the Deep Neural Network (DNN). (b) In dynamic `HyperAdv`, the adversarial perturbation at DNN state t is not effective for state $t+1$.

robustness as well as 8% improvement in clean accuracy compared to static defensive training [6]. Code and trained DNNs are shared at https://github.com/Restuccia-Group/HyperAdv.

Paper Organization. Section II introduces background and related work on AML and hypernetwork. Section III describes the design of our dynamic defense. Section IV presents experiment results while Section V presents conclusions and discusses potential future directions.

II. Background and Related Work

Adversarial Machine Learning. Without loss of generality, we investigate adversarial evasion attack in multi-class classification problems, such as modulation classification and radio fingerprinting. Formally, the goal of the adversary is to find a minimum perturbation δ such that

$$f(x+\delta) = y' \quad y' \neq y \tag{1}$$

where $f(\cdot)$, x, and y', y are the DNN classifier, input, DNN output, and groundtruth label respectively. It was demonstrated in [4] that one-step gradient can be used to generate effective adversarial examples while PGD [6] enhanced the effectiveness by iteratively updating adversarial examples with multiple steps of gradient information, that is

$$x_{t+1} = x_t + \alpha \cdot \frac{\nabla \mathcal{L}(f_W(x_t), y)}{||\nabla \mathcal{L}(f_W(x_t), y)||_p} \tag{2}$$

where x_t denotes the adversarial example at the t-th iteration, W denotes the weights of DNN, and $\nabla \mathcal{L}(f_W(x_t), y)$ denotes the gradient of the cross-entropy loss w.r.t. DNN output $f_W(x_t)$ and groundtruth y. $||\cdot||_p$ denotes the L_p norm and α is the step size of the adversarial update.

In the black-box setting where the gradient of DNN cannot be accessed, the attacker can train a surrogate model based on outputs of the victim DNN. It was demonstrated that adversarial examples against the surrogate model can be effectively transferred to the original model [12–14].

To improve robustness to such gradient-based attacks, Madry *et al.* [6] trained DNNs with adversarial examples, which can be modeled as a min-max optimization problem,

$$\min_W \mathbb{E}\{\max_\delta \mathcal{L}(f_W(x+\delta), y)\} \tag{3}$$

where the inner maximization problem denotes adversarial attack and outer minimization problem denotes AT.

While AT [6] significantly enhances DNN robustness, it compromises performance on clean data. TRADES [9] optimizes the trade-off between clean and robust accuracy by incorporating the Kullback-Leibler divergence (KLD) between the clean output and adversarial output into the min-max optimization problem. Equation 3 is refined as

$$\min_W \mathbb{E}\{\mathcal{L}(f_W(x), y) + \lambda \cdot \max_\delta D_{KL}(f_W(x)||f_W(x+\delta))\} \tag{4}$$

where $D_{KL}(\cdot)$ denotes the KLD and $\lambda \geq 0$ denotes a trade-off between clean and robust performance.

Adversarial ML in RFMLS. Recently, AML has been investigated in a RFMLS setting [5, 15, 16], revealing that well-crafted adversarial examples can lead to a significant loss in performance in RFMLS. For example, *exploratory attacks* try to train a surrogate model to imitate the functionality of the DNN [17–19], while *evasion attacks* leverage gradient-based methods to craft adversarial inputs [20–22]. In *spoofing attacks*, synthetic signals are generated to impersonate a legitimate transmitter [23–25]. To tackle AML attacks, the model can be trained with adversarial examples [6, 9] or other steps taken to prevent the adversary from building an accurate surrogate model [26]. Existing AML in RFMLS considers only *static* settings, which is in stark contrast to our approach.

Hypernetworks [10] are frameworks that utilize a DNN to generate parameters for another DNN. Specifically, the framework consists of a *hypernet* and a *target network*. Formally, let $H(\Psi, c) = W_c$ denote the hypernet, with learnable parameters Ψ, that generates parameters W_c of the target DNN based on a given context c. The target network $f(W_c, x) = y$ will take the weight W_c and data x as input, and generate an output y. During training, Ψ is end-to-end optimized with context c and output y of the target network. Then, the target network can be dynamically generated at runtime. Hypernetworks have been investigated in many tasks such as continual learning [27], federated learning [28] and multi-object optimization [29]. Recently, hypernetworks have been also utilized for robust DNN such as adversarial robustness [30] and out-of-distribution robustness [31]. Authors of [30, 31] consider input statistics as context c and the hypernet is used to adapt the input. In contrast, we consider randomly generated c, independent of x, making it fundamentally different from [30, 31].

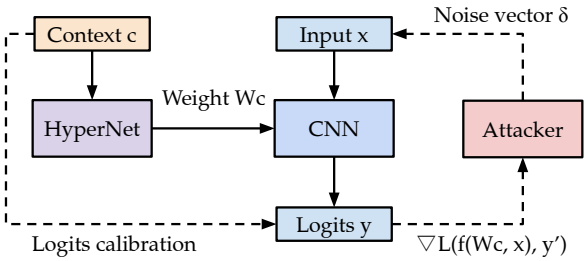


Fig. 2: Overview of HyperAdv.

III. The HyperAdv Framework

The proposed dynamic defense is based on hypernetwork where a hypernet is used to generate parameters dynamically for another Convolutional Neural Network (CNN) during inference. The overall system consists of a hypernet $H(\cdot)$, a target CNN $f(\cdot)$, and a set of randomly generated context vectors $\{c_i\}_{i=1}^n$. During training, the hypernet $H(\Psi, c)$ will take n context vectors as input and generate multiple context-aware CNN weights $\{W_c^i\}_{i=1}^n$. These parameters are used for the target CNN $f(W_c, x)$ to generate n outputs for each input x. Unlike the conventional end-to-end training aiming at learning the optimal parameters for a single CNN, HyperAdv learns to generate multiple target CNNs with a single hypernet $H(\cdot)$. During inference, the context vector is dynamically changed for each query, thus resulting in a different target CNN for each input. The changing W_c generates diverse gradient information at each step, making it more difficult to find effective adversarial samples.

A fundamental question in this dynamic defense framework is how to train a hypernet $H(\cdot)$ so that, for each context c, it can produce a different W_c with a unique landscape in hyperspace. As $H(\cdot)$ is end-to-end optimized based on its input c and output y of the target CNN, it may learn an universal solution W_c, making all target CNNs output the same y. To diversify W_c, we use the context c in an affine transformation which projects y to a new space y'. Then y' is treated as the ultimate output of the system in both training and testing. In this case, while calibrated result y' may be the same due to the end-to-end optimization, the original output y is distinct for different target CNN, making the W_c unique.

The overall defense mechanism is depicted in Fig. 2. First, for each input x, a context c will be randomly chosen from the predefined context set $\{c_i\}_{i=1}^n$. Then, the hypernet $H(\Psi, c)$ will generate a set of context-aware parameters W_c for the target network. Subsequently, the target CNN $f(W_c, x) = y$ will perform inference and generate an output y based on the given W_c. y is further calibrated by the context vector and mapped to y'. For each query, HyperAdv will have a different W_c and y'. Thus, the perturbation δ given by the previous gradient $\nabla\mathcal{L}(f(W_c, x), y')$ may not be effective for the new target CNN. Next we discuss the details of each component in HyperAdv.

• *Target Network.* We utilize 1-dimensional CNN whose effectiveness has been demonstrated in wireless signal classification tasks [5]. The target CNN consists of 6 1-d CNN layers whose kernel size is 1×3 with ReLU activations. Channel sizes of CNNs are 64, 64, 128, 128, 256, and 256 respectively. Maxpooling layers are utilized after each CNN layer for downsampling features. A global average pooling as well as a linear layer are leveraged to decode extracted features and output raw logits y. The total number of parameters in the target CNN is ~ 0.4 million.

• *Hypernetwork.* One challenge in the proposed framework is the complexity of the hypernet. To address the resource constraint in many RFMLS scenarios, the size of a hypernet that can generate n target CNNs should be equal to or less than n times of the target CNN's size. However, the small size of hypernet may hamper the end-to-end performance of target CNNs. To this end, we initially train a large hypernet (i.e., the teacher) and then train a smaller hypernet (i.e., the student) to learn the output of the teacher.

The teacher model consists of 14 independent linear hyper blocks to generate weight and bias of 6 1-d CNN layers and 1 linear layer in target CNNs. A hyper block that takes a context vector as input and generates corresponding parameters is defined with 2 linear layers. A ReLU activation is used after the first layer for non-linear transformation. The hidden layer has 256 units and the output dimension is the size of W_c. To reduce the model size, the student model decreases the hidden dimension to 56. In addition, the second linear layer in each hyper block is divided into 8 chunks, with independent linear mappings applied only within each chunk. The ultimate size of Ψ is ~ 3.1 million.

• *Context.* Training a hypernetwork is intrinsically a model ensemble learning problem. Pang *et al.* [32] pointed out that naive ensemble learning can generate a similar decision boundary for different DNN in the hyperspace, making the ensemble vulnerable to transferable adversarial attacks. To increase the diversity of generated DNN, the input of hypernetwork c is also utilized to map the raw output y to a new space y' that is used for the final inference task. For simplicity, we define such mapping as an affine transformation.

$$y' = \alpha \odot y + \beta \tag{5}$$

where $\odot$ represents element-wise multiplication, α and β are vectors with the same dimension as y. In practice, a set of $\{\alpha_i\}_{i=1}^n$ and $\{\beta_i\}_{i=1}^n$ are randomly generated with a uniform distribution $U(-1, 1)$. Then, the i-th context vector $c_i = \{\alpha_i, \beta_i\}$ is created by concatenating α_i and β_i. Experimental results demonstrate that the calibration significantly enhances the diversified ensemble learning.

• *Learning Strategy.* We leverage parallel ensemble learning to train HyperAdv. In forward propagation phase, the hypernet will take all context $\{c^i\}_{i=1}^n$ and generate parameters $\{W_c^i\}_{i=1}^n$ for all target models $\{f^i(\cdot)\}_{i=1}^n$. Then, $\{f^i(\cdot)\}_{i=1}^n$ are utilized to get output $\{y^i\}_{i=1}^n$ for each input x. $\{y'^i\}$ is transformed by $\{c^i\}_{i=1}^n$ with Equation 5. The loss of parallel

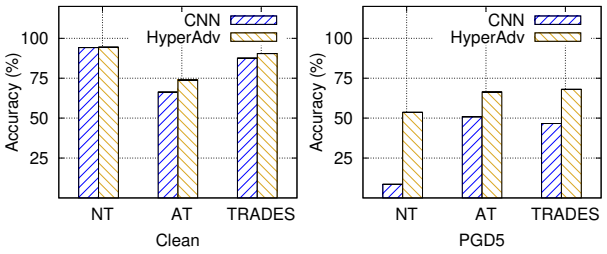


Fig. 3: Trade-off between clean and adversarial accuracy. (Left) baseline CNN and HyperAdv performance on clean data with different training approaches; (Right) baseline CNN and HyperAdv performance on PGD-distorted data with different training approaches.

ensemble learning is defined as

$$\mathcal{L} = \frac{1}{n}\sum_{i} \mathcal{L}^i(f^i(W_c^i, x), y'^i) \tag{6}$$

where $\mathcal{L}^i(\cdot)$ denotes the loss function for the i-th target model $f^i(\cdot)$. In backward propagation phase, the hypernetwork $H(\Psi, c)$ is optimized with gradient descent based on Equation 6. In our experiments, we set n to 8.

As directly training the small hypernet will compromise the classification performance, we use a multi-stage training approach: i) train a teacher hypernet using Equation 6; ii) train a student hypernet by minimizing $\frac{1}{n}\sum_i^n ||W_c^{t_i} - W_c^{s_i}||_2^2$, where $W_c^{t_i}$ and $W_c^{s_i}$ are weights of i-th target CNN generated by teacher and student hypernets, respectively; and iii) finetune the student hypernet using Equation 6.

IV. Performance Evaluation

Experimental Setup. In this work, we evaluate our defense based on a multi-class modulation classification task. We leverage the RadioML 2018.01A dataset [11] that consists of 24 different modulation classes with Signal to Noise Ratio (SNR) range from -20 dB to 30 dB. As Deep Learning (DL) methods do not achieve satisfactory classification performance on wireless signals with low SNR [11], we only train using signals with SNR greater than 10 dB. The utilized dataset consists of 1.08 million signals, each comprising 1024 I/Q samples. The dataset is split into training and testing set with a ratio of 0.8 to 0.2.

To compare the improvement of HyperAdv, we train a CNN which has the same architecture as the target network. As HyperAdv can be incorporated with other static defense approaches, we also train HyperAdv and the baseline CNN with two defensive training methods [6, 9]. Models trained with conventional cross-entropy loss are denoted as "Natural Training (NT)" while models trained with [6] and [9] are denoted as AT and TRADES, respectively. Models are trained on all training data with a mixed SNR range using the Adam optimizer. Baseline CNNs and teacher hypernetworks are trained for 50 epochs with a learning rate of 0.0001. Student hypernets are initially trained to regress the weights W_c generated by the teachers using a learning rate of 0.001, and then fine-tuned for 1 epoch with a learning rate of 0.0001.

TABLE I: Accuracy of HyperAdv and its static counterparts.

	NT		AT		TRADES	
	Clean	PGD	Clean	PGD	Clean	PGD
CNN	94.20	3.72	66.28	50.42	87.55	43.16
HyperAdv-R	95.20	**45.02**	74.01	**63.98**	90.37	**60.92**
HyperAdv-S	95.11	23.70	72.74	50.10	91.27	43.08
HyperAdv-E	**96.34**	27.88	**78.36**	61.34	**92.79**	49.50

We consider l_∞ PGD attack with a perturbation $\delta \leq 0.05$ in the white-box setting where the attacker can access the weights W_c of target CNN at each step.[1] Note that this attack model is more severe than the generalized wireless AML setting in [5] as we don't add path loss and fading to perturbations. The attacker has complete gradient information of the target CNN as well as a perfect wireless propagation channel. Thus, results in this paper present a worst-case scenario of robustness. In real-world applications, HyperAdv can provide better robustness as attackers have limited knowledge of the victim model and also face non-ideal wireless channel conditions.

Robustness Trade-off. Fig. 3 shows the classification performance of baseline CNN and our HyperAdv on clean and PGD-distorted data where the number of PGD iterations is set to 5 (The performance as a function of iterations is investigated in Fig 4.) The naturally trained CNN achieves 94.20% accuracy on clean data and 8.54% accuracy on adversarial data. Although AT and TRADES improve the adversarial accuracy to 50.88% and 46.62% respectively, the clean accuracy is reduced to 66.28% and 87.55% respectively. This is because AT and TRADES trained with only adversarial data may suffer *adversarial overfitting* [33]. While they increase classification performance on adversarial examples, there is considerable loss of accuracy on benign data. To this end, such static defense approaches are not suitable for reliable RFMLS. On the other hand, HyperAdv achieves 56.30% accuracy on PGD attack, 47.76% improvement compared to CNN-NT. The performance on benign data is 95.20%, which is comparable to CNN-NT. Thus, HyperAdv improves adversarial robustness without sacrificing clean accuracy.

In addition, HyperAdv can be applied to other defense to further enhance robustness. By incorporating HyperAdv with AT and TRADES, the adversarial robustness increases by 15.52% and 22.00%, respectively. Interestingly, HyperAdv also improves the AT and TRADES performance on clean data by 7.73% and 2.82%. This is because ensemble learning intrinsically augments the adversarial samples with different models during training, thereby mitigating the overfitting of adversarial data and improving performance [34].

Effect of Dynamic Inference. To comprehensively evaluate the enhanced robustness introduced by the dynamic design of HyperAdv, we also perform PGD attack on its static counterparts. First, we consider using a single context consistently during inference, denoting this model as HyperAdv-S. In this case, the attacker consistently updates adversarial examples

[1]PGD may not always create realistic, demodulatable wireless samples. Therefore, we choose a relatively small δ for high SNR signals in the dataset.

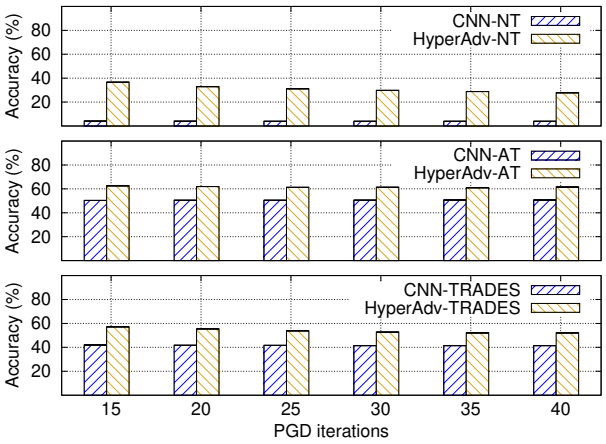


Fig. 4: Robust accuracy as a function of PGD iterations. (Top) Naturally trained CNN and HyperAdv; (Middle) Adversarially trained CNN and HyperAdv; (Bottom) CNN and HyperAdv trained with TRADES algorithm.

against a single set of weights. Thus, HyperAdv-S represents the static robust performance for single target model. In addition, we also consider using the ensemble of all target models for inference, without dynamically changing the model parameters. The inference output is the average of projected output y' of all target CNNs. In this case, the adversarial gradient information can be backpropagated through all target models. This scenario, denoted as HyperAdv-E, describes the static robustness of overall target models. The original HyperAdv with randomly changed context is denoted as HyperAdv-R.

Table I shows performance of HyperAdv-R and its static counterparts on both clean and adversarial data. The naive CNN without hypernetworks is also reported as a baseline. We increase the number of PGD iterations to 10 for more comprehensive assessments. HyperAdv-E achieves slightly better performance on clean data compared to others due to the effect of ensemble inference. HyperAdv-E also has better performance on adversarial data compared to HyperAdv-S and baseline CNN due to the same reason. Compared to CNN-NT, HyperAdv-S can improve the robust accuracy by roughly 20%, which indicates that the diversified ensemble learning can improve adversarial robustness to some extent [32]. However, for AT and TRADES, HyperAdv-S exhibits no difference in performance on adversarial examples in comparison to CNN, meaning that the ensemble learning without dynamics on adversarial examples is less effective when combined with powerful defense such as AT and TRADES. On the other hand, HyperAdv-R achieves best accuracy on adversarial data compared to other two static HyperAdv, indicating that the dynamic inference mechanism can effectively mitigate the iterative gradient search of adversarial attacks.

Robustness as a Function of Iterations. We investigate the robustness as a function of PGD iterations in Fig. 4. For NT, the adversarial robustness of HyperAdv decreases from 36.62% to 27.74% with an increase in # iterations. This indicates that HyperAdv requires more computation

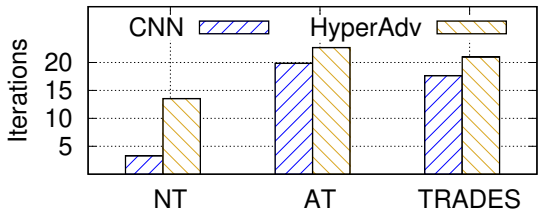


Fig. 5: Computational cost of PGD attack.

resources for attackers to find effective gradient information. Moreover, the worst-case robustness (with maximum PGD iterations) of HyperAdv is 23.82% higher than that of the basic CNN, indicating that the robustness is barely degraded by increasing computation. This is because multiple target CNNs have diverse parameters which results in distinct gradient landscape. Attacks searching the gradient across multiple target CNNs will result in a sharp overlapping landscape, making gradient descent to be trapped at an ineffective local minimum. This observation is further supported by the results of AT and TRADES in Fig. 4. For AT and TRADES, HyperAdv constantly outperforms basic CNN by 11.08% and 12.28% on average, which means the sharpness of the overall gradient landscape significantly increases the robustness.

Computational Cost of PGD Attack. In reality, latency is often a critical need of many RFMLS systems. Therefore, an effective adversarial attack with large number of iterations may not be realistic for AML in RFMLS. To this end, we also assess our defense strategy with the computational cost of PGD. Fig. 5 shows the average of the number of iterations that PGD spends to craft adversarial examples (the maximum number of iteration is considered as 40). The average number of PGD iterations against CNN-NT, -AT and -TRADES are 3.31, 19.83 and 17.64, respectively. Compared to baseline CNN, HyperAdv increases the average number of iterations to 13.53, 22.66 and 21.01 for NT, AT and TRADES respectively. This improvement is due to the dynamic nature of HyperAdv which can generate diverse gradients against the adversary. With an increasing number of iterations, the effective perturbation may not be found within the time limit, hence resulting in a computationally robust system.

V. Conclusion

We proposed a novel defense for AML in RFMLS that dynamically alters DNN parameters during inference, making it challenging for adversaries to obtain effective gradient information for attacks. Experiments demonstrate that our dynamic defense enhances the adversarial robustness of naturally trained DNNs by 48% without compromising performance on clean data. Furthermore, our approach can be combined with other static defenses to further improve performance. Integrating our method with static adversarial training increases adversarial robustness by 16% and improves performance on benign data by 8%. Future work could explore the defense to a more generalized wireless AML setting such as spoofing attack and exploratory attack.

Acknowledgment of Support and Disclaimer

This work has been funded in part by the National Science Foundation under grants CNS-2134973, ECCS-2229472, CNS-2312875 and ECCS-2329013, by the Air Force Office of Scientific Research under contract number FA9550-23-1-0261, by the Office of Naval Research under award number N00014-23-1-2221, and by the Air Force Research Laboratory via *Open Technology and Agility for Innovation (OTAFI)* under transaction number FA8750-21-9-9000 between SOSSEC, Inc. and the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of U.S. Air Force, U.S. Navy or the U.S. Government.

References

[1] F. Restuccia and T. Melodia, "Big Data Goes Small: Real-Time Spectrum-Driven Embedded Wireless Networking Through Deep Learning in the RF Loop," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pp. 2152–2160, 2019.

[2] D. Uvaydov, S. D'Oro, F. Restuccia, and T. Melodia, "DeepSense: Fast Wideband Spectrum Sensing Through Real-Time In-the-Loop Deep Learning," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–10, 2021.

[3] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, K. Chowdhury, S. Ioannidis, and T. Melodia, "Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pp. 646–655, 2020.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

[5] F. Restuccia, S. D'Oro, A. Al-Shawabka, B. C. Rendon, K. Chowdhury, S. Ioannidis, and T. Melodia, "Generalized Wireless Adversarial Deep Learning," *Computer Networks*, vol. 216, p. 109264, 2022.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[7] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified Adversarial Robustness via Randomized Smoothing," in *Proceedings of International Conference on Machine Learning (ICML)*, pp. 1310–1320, PMLR, 2019.

[8] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion Models for Adversarial Purification," in *Proceedings of International Conference on Machine Learning (ICML)*, pp. 16805–16827, PMLR, 2022.

[9] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically Principled Trade-off between Robustness and Accuracy," in *Proceedings of International Conference on Machine Learning (ICML)*, pp. 7472–7482, PMLR, 2019.

[10] D. Ha, A. M. Dai, and Q. V. Le, "HyperNetworks," in *International Conference on Learning Representations (ICLR)*, 2017.

[11] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air Deep Learning Based Radio Signal Classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[12] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical Black-Box Attacks Against Machine Learning," in *Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security (ASIA CCS)*, pp. 506–519, ACM, 2017.

[13] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples," *arXiv preprint arXiv:1605.07277*, 2016.

[14] N. Papernot, P. McDaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," in *IEEE Military Communications Conference (MILCOM)*, pp. 49–54, IEEE, 2016.

[15] D. Adesina, C.-C. Hsieh, Y. E. Sagduyu, and L. Qian, "Adversarial Machine Learning in Wireless Communications Using RF Data: A Review," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 77–100, 2022.

[16] L. Zhang, S. Lambotharan, G. Zheng, G. Liao, A. Demontis, and F. Roli, "A Hybrid Training-Time and Run-Time Defense Against Adversarial Attacks in Modulation Classification," *IEEE Wireless Communications Letters*, vol. 11, no. 6, pp. 1161–1165, 2022.

[17] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu, and J. H. Li, "Adversarial Deep Learning for Cognitive Radio Security: Jamming Attack and Defense Strategies," in *Proceedings of IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, IEEE, 2018.

[18] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep Learning for Launching and Mitigating Wireless Jamming Attacks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 1, pp. 2–14, 2018.

[19] Z. Luo, S. Zhao, Z. Lu, J. Xu, and Y. Sagduyu, "When Attackers Meet AI: Learning-Empowered Attacks in Cooperative Spectrum Sensing," *IEEE Transactions on Mobile Computing*, vol. 21, no. 5, pp. 1892–1908, 2020.

[20] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating Adversarial Evasion Attacks in the Context of Wireless Communications," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1102–1113, 2019.

[21] M. Z. Hameed, A. György, and D. Gündüz, "The Best Defense Is a Good Offense: Adversarial Attacks to Avoid Modulation Detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1074–1087, 2020.

[22] M. Sadeghi and E. G. Larsson, "Adversarial Attacks on Deep-Learning Based Radio Signal Classification," *IEEE Wireless Communications Letters*, vol. 8, pp. 213–216, Feb 2019.

[23] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Generative Adversarial Network in the Air: Deep Adversarial Learning for Wireless Signal Spoofing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 294–303, 2020.

[24] Y. Shi, K. Davaslioglu, and Y. E. Sagduyu, "Generative Adversarial Network for Wireless Signal Spoofing," in *Proceedings of the ACM Workshop on Wireless Security and Machine Learning (WiseML)*, pp. 55–60, ACM, 2019.

[25] K. Davaslioglu and Y. E. Sagduyu, "Generative Adversarial Learning for Spectrum Sensing," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2018.

[26] Y. Sagduyu, Y. Shi, and T. Erpek, "Adversarial Deep Learning for Over-the-Air Spectrum Poisoning Attacks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 306–319, 2019.

[27] J. von Oswald, C. Henning, B. F. Grewe, and J. Sacramento, "Continual Learning with Hypernetworks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[28] A. Shamsian, A. Navon, E. Fetaya, and G. Chechik, "Personalized Federated Learning using Hypernetworks," in *Proceedings of International Conference on Machine Learning (ICML)*, pp. 9489–9502, PMLR, 2021.

[29] A. Navon, A. Shamsian, E. Fetaya, and G. Chechik, "Learning the Pareto Front with Hypernetworks," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2021.

[30] H. Gong, M. Dong, S. Ma, S. Camtepe, S. Nepal, and C. Xu, "Parameter-Saving Adversarial Training: Reinforcing Multi-Perturbation Robustness via Hypernetworks," *arXiv preprint arXiv:2309.16207*, 2023.

[31] T. Volk, E. Ben-David, O. Amosy, G. Chechik, and R. Reichart, "Example-based Hypernetworks for Out-of-Distribution Generalization," *arXiv preprint arXiv:2203.14276*, 2022.

[32] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving Adversarial Robustness via Promoting Ensemble Diversity," in *Proceedings of International Conference on Machine Learning (ICML)*, pp. 4970–4979, PMLR, 2019.

[33] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness May Be at Odds with Accuracy," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[34] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble Adversarial Training: Attacks and Defenses," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.