ADVERSARIAL LEARNING IN TRANSFORMER BASED NEURAL NETWORK IN RADIO SIGNAL CLASSIFICATION

Lu Zhang, Sangarapillai Lambotharan, Gan Zheng

Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University

ABSTRACT

Deep Learning has attracted significant interests in wireless communication design problems. However, recent studies discovered that the deep neural network is vulnerable to adversarial attacks in the sense that a carefully designed and imperceptible perturbation to the input of the neural network could mislead the prediction of the neural network. In this paper, motivated by attractive classification performance of the transformer based neural networks, we analyse the vulnerability and robustness of the transformer against adversarial attacks in modulation classification scenarios. Using real datasets, we demonstrate that the transformer can achieve higher accuracy as compared to a convolutional neural network in the presence of adversarial attacks.

Index Terms— deep learning, adversarial attacks, radio modulation classification, transformer based neural network

1. INTRODUCTION

Automatic modulation classification (AMC) plays an important role in signal intelligence and surveillance applications including that in cognitive radio and dynamic spectrum access to monitor the spectrum and its occupancy continuously. In the past, AMC has been accomplished using various likelihood based methods [1, 2, 3] and different machine learning methods based on carefully chosen signal features [4, 5, 6]. These techniques normally need precise estimation of different signal parameters including carrier frequency and signal power with manual calibration of threshold. However, harnessing the power of DL, the problem of handcrafted feature design can be alleviated and AMC can be achieved by training a deep neural network (DNN) using a large number of raw signal data samples and generating classification decisions with high accuracy. For example, O'Shea et al. [7] proposed a convolutional neural network (CNN) architecture for AMC which shows good classification performance as compared to the existing expert feature based modulation classifiers.

Transformers, a kind of DNN based on the self-attention mechanism, are originally used in natural language processing (NLP) tasks [8, 9, 10]. Later transformers also gained huge attention in computer vision area [11]. Due to the great success of the transformers in both NLP and computer vision, transformers have been considered as a promising technique for AMC [12, 13, 14]. However, recent studies discovered that the adversarial example could deteriorate the performance of DNN in many applications [15, 16, 17, 18, 19]. Adversarial examples are modifications of the original benign inputs through addition of small and imperceptible perturbations. Such malicious perturbation can lead to misclassification of the classifier. The adversarial attacks and the defense against them have attracted significant interests in communication design problems. For example, the work in [20] showed that adversarial examples can significantly deteriorate the classification accuracy of the CNN based modulation classifiers. A defense based on label smoothing, Gaussian noise augmentation, and neural rejection was proposed in [21] to enhance the robustness against adversarial examples in modulation classification. In this work, we investigate the robustness of transformer based neural network against adversarial examples in modulation classification. We use a particular class of adversarial attack known as white-box projected gradient descent (PGD) algorithm to generate adversarial examples. Using real datasets, we show that the transformer based neural network is more robust against PGD attack than the CNN.

2. TRANSFORMER BASED NEURAL NETWORK

We use a similar transformer architecture as that used in the vision transformer [11] as shown in Figure 1. Specifically, each input can be viewed as a 2D image ($I_1 \times I_2$) of depth one. Then the input is sent to a convolutional layer with the kernel size $I_1 \times N_k$, stride size $I_1 \times N_s$ and N_c output channels, which outputs a $N_c \times 1 \times N_o$ vector. Here $N_o = \frac{I_2 - N_k}{N_s} + 1$. After reshaping, we augment a class (CLS) token to the output which forms a vector of $(N_o + 1) \times N_c$ dimension. Then this vector is sent to a stack of N identical transformer encoder layers [10]. Each encoder layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. The hidden size for the feed-forward layer we used is $N_c \times 4$ dimension. A residual connection [22] is employed around each of the two sub-layers, followed by layer normalization [23], i.e., given the input to the sub-layer x, the output of each sub-layer is LayerNormalization$(x + Sublayer(x))$, where

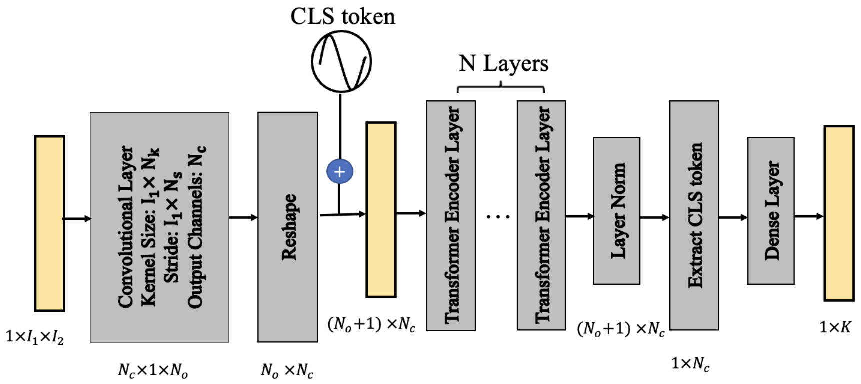

Fig. 1: The architecture of the transformer based neural network for the modulation classification.

$Sublayer(x)$ is a function implemented by the sublayer itself and layer normalization works by computing the mean and variance used for normalization from all of the summed inputs to the neurons in a layer in a single training case [23]. After 4 identical transformer encoder layers followed by another layer normalization layer, we extract the information from the added cls token which forms a $1 \times N_c$ vector. Finally this vector is fed to a dense layer which outputs a K-dimensional vector whose components provide decision scores for all the modulation types. In this work, $I_1 = 2$, $I_2 = 128$, $N_k = 32$, $N_s = 2$, $N_c = 128$, $N = 4$ and $K = 10$.

An attention function can be defined as a function that maps a query and a set of key-value pairs to an output. The output is calculated as a weighted sum of the values, with the weight allocated to each value determined by the compatibility function of the query with the associated key. Briefly, given queries and keys of dimension d_k, and values of dimension d_v, the output of the attention is computed as the dot products of the query with all keys, followed by dividing each by $\sqrt{d_k}$ and applying a softmax function to produce the weights on the values. Normally, when calculating the outputs of the attention, the queries, keys and values will be packed together into matrix Q, K, V respectively and the matrix of outputs is calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

For multi-head attention mechanism, instead of executing a single attention function, the d-dimensional keys, values and queries are linearly projected h times with different, learned linear projections to d_k, d_k and d_v dimensions, respectively. Then the attention function is performed in parallel on each of these projected queries, keys and values, giving d_v-dimensional output values. These output values are concatenated and projected, yielding the final values. In this work, we use $h = 4$ parallel heads.

3. BASICS OF ADVERSARIAL EXAMPLES

Adversarial examples can be formally described as follows. Given a trained DL classifier f and an original input data sample x, one can generate an adversarial example x' as a constrained optimization problem [24]:

$$\begin{aligned} &\min_{x'} \|x' - x\|_p \\ &s.t.\ f(x') = l',\ f(x) = l,\ l \neq l' \end{aligned} \tag{2}$$

where l and l' are the label of x and x', respectively, and $\|\cdot\|_p$ denotes the l_p-norm of the distance between two data samples, i.e., the l_p-norm of the perturbations added on the original input. Normally, we can use l_0-norm, l_∞-norm, or l_2-norm to constrain the size of the perturbation, however, in the application of radio signal classification, the l_2-norm is chosen as it represents the perturbation power [20].

In practice it is not easy to solve (2) directly, therefore, many researchers proposed various substitute algorithms to generate adversarial examples, including fast gradient method (FGM), basic iterative method (BIM), Jacobian-based saliency map attack (JSMA), DeepFool method, PGD attack, universal perturbation method [24]. For example, for FGM attack, only a one-step gradient update is performed along the direction of the gradient and the perturbation η can be calculated as $\eta = \varepsilon \cdot \nabla_x J(\theta, x, l)$, where J denotes the objective function of the DNN model, θ denotes the parameters of the classifier and ε is used to control the magnitude of the perturbation. Then the generated adversarial examples can be expressed as: $x' = x + \eta$. On the other hand, the PGD attack extends the FGM by performing a finer optimization at multiple iterations. At each iteration, the gradient update is projected onto a feasible constraint domain such that $\|x' - x\| \leqslant \varepsilon$ is satisfied.

Adversarial examples can be categoried into different scenarios according to the goal, knowledge and capability of the attacker [25]. In terms of the attacker's knowledge, attacks can be divided into perfect-knowledge white-box attacks, limited-knowledge gray-box attacks, and zero-knowledge black-box attacks. The white-box scenario means the attacker knows everything about the targeted system and black-box attacks indicate absence of any knowledge of the defense system. In this work, we consider the white-box scenario for the defender, as this set up enables one to perform a worst-case evaluation of the security of learning algorithms, establishing empirical upper bounds on the performance degradation that may be incurred by the system under attack [25].

The algorithm for generating the white-box PGD attack is shown in Algorithm 1, which is adopted from [26]. The lines 2-5 create a recursive process, specifically, the line 4 first calculates the gradient of the objective function and applies a standard gradient descent procedure. The objective function we used in this work is written as:

$$\Psi(\mathbf{x}) = s_y(\mathbf{x}) - \max_{j \neq y} s_j(\mathbf{x}), \tag{3}$$

where $s_\cdot(\mathbf{x})$ means the output of the neural network corresponding to different classes. By minimizing the objective function in (3), the attacker aims to minimize the output corresponding to the true class and maximize the output corresponding to the most competing false class. Then a projector is used to get the updated data point within the feasible constraint domain such that $||\mathbf{x}' - \mathbf{x}_0||_2 \leq \varepsilon$, where ε is calculated as:

$$\varepsilon = \sqrt{\frac{PNR \cdot \|\mathbf{x}\|_2^2}{SNR + 1}}, \tag{4}$$

where PNR denotes the perturbation to noise ratio and SNR indicates the signal to noise ratio. The projection procedure can be expressed as the following optimization:

$$\begin{aligned} &\min_{\mathbf{x}'} \|\mathbf{x}' - \mathbf{x}^*\|_2^2 , \\ &s.t. \;\; \|\mathbf{x}' - \mathbf{x}_0\|_2 \leq \varepsilon \end{aligned} \tag{5}$$

where $\mathbf{x}^*$ is the updated data sample after the standard gradient procedure $\mathbf{x}^* = \mathbf{x} - \eta \nabla \Psi(\mathbf{x})$ and $\mathbf{x}_0$ is the original input sample. The solution to (5) can be calculated as:

$$\mathbf{x}' = \mathbf{x}_0 + \frac{\mathbf{x}^* - \mathbf{x}_0}{max(\varepsilon, \|\mathbf{x}^* - \mathbf{x}_0\|_2)} \cdot \varepsilon. \tag{6}$$

However, in this work, we use (7) as the projector to force the l_2-norm of the generated perturbation equal to ε, i.e., $\|\mathbf{x}' - \mathbf{x}_0\|_2 = \varepsilon$, such that the performance of the transformer based neural network can be evaluated under certain amount of the adversarial perturbations.

$$\mathbf{x}' = \mathbf{x}_0 + \frac{\varepsilon \cdot (\mathbf{x}^* - \mathbf{x}_0)}{\|\mathbf{x}^* - \mathbf{x}_0\|_2} \tag{7}$$

Finally, the iteration will stop when the condition in line 5 is satisfied, i.e., either the objective function converges or the predicted label of generated adversarial example is unequal to the true label.

Algorithm 1 PGD-based Maximum-confidence Adversarial Examples

Input:

- the input sample $\mathbf{x}_0$ and its label y
- η: the step size
- $\Psi(\mathbf{x})$: the objective function
- $s(\mathbf{x})$: the decision score of the data sample
- Π: a projector on the l_2-norm constraint $||\mathbf{x} - \mathbf{x}'||_2 \leq \varepsilon$, where $\varepsilon = \sqrt{\frac{PNR \cdot \|\mathbf{x}_0\|_2^2}{SNR+1}}$
- t: a small positive number to ensure convergence

Output: $\mathbf{x}'$: the adversarial examples.

```
1: $\mathbf{x}' \leftarrow \mathbf{x}_0$
2: repeat
3:     $\mathbf{x} \leftarrow \mathbf{x}'$
4:     $\mathbf{x}' \leftarrow \Pi(\mathbf{x} - \eta \nabla \Psi(\mathbf{x}))$
5: until $||\Psi(\mathbf{x}') - \Psi(\mathbf{x})|| \leq t$ or $\arg\max_i s_i(\mathbf{x}') \neq y$
6: return $\mathbf{x}'$
```

4. RESULTS AND DISCUSSIONS

We illustrate and analyze the experimental results. All the algorithms are written in PyTorch and executed by NVIDIA GEforce RTX 2080 Ti GPU.

4.1. Dataset

In this work, the dataset we used is the GNU radio ML dataset RML2016.19a [27]. The GNU radio ML dataset RML2016.10a contains 220000 input samples, and each sample corresponds to one modulation scheme at a specific SNR. There are 11 modulation categories in this dataset including BPSK, QPSK, 8PSK, QAM16, QAM64, CPFSK, GFSK, PAM4, WBFM, AM-SSB, and AM-DSB. The samples are generated for 20 different SNR levels from -20dB to 18dB with a step of 2dB. Each sample has 256 dimensions, which contains 128 in-phase and 128 quadrature components. Half of the samples are used as training set and the other half are considered as testing set. To evaluate the performance of the transformer based neural network against adversarial examples, we choose 1000 data samples from testing set that correspond to SNR=10dB to generate PGD attacks.

4.2. Results

In this section, we present the accuracy performance of the transformer based neural network against PGD attack. For comparison, we generate PGD attacks for CNN as the baseline. The CNN architecture we used in this work is called

VT-CNN2 classifier which is the same as the one used in [27]. From Figure. 2, we confirm that the transformer based neural network can obtain better classification accuracy than the CNN classifier in the absence of the adversarial perturbations. The accuracy performance of the CNN and the Transformer against adversarial attacks for a range of PNR values is presented in Figure. 3. As expected, the performance of both CNN and Transformer decreases significantly as the PNR increases. Specifically, when there is no adversarial perturbation, the classification accuracy of the transformer and CNN for SNR=10dB can achieve 90.8% and 83.9% respectively, however, in the presence of adversarial attacks, the classification accuracy reduces to 39.9% and 30.6% respectively when PNR = -10dB. However, in the PNR region of -40dB to -10dB, the Transformer is able to maintain a approximately 10% performance advantage as compared to the CNN. Therefore, we conclude that the Transformer based modulation classification maintains robustness against adversarial examples for a wide range of PNR values.

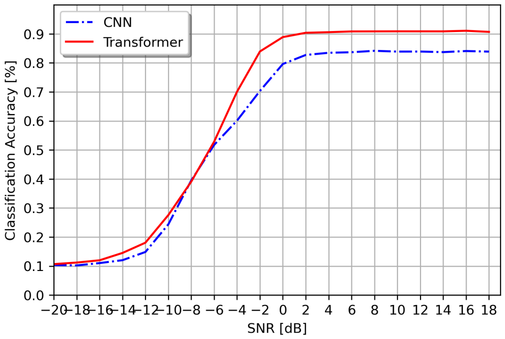

Fig. 2: Classification accuracy of normal (benign) samples for various SNR.

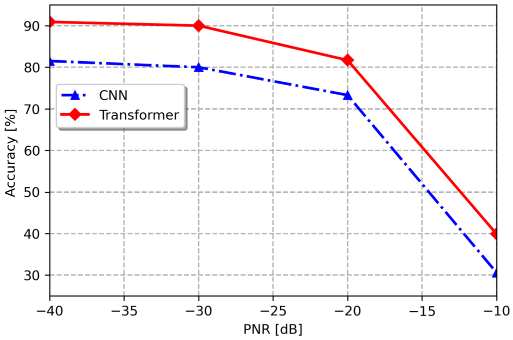

Fig. 3: The evaluation results of the transformer based neural network against PGD attack.

5. CONCLUSIONS

In this paper, using real datasets, we have shown that in radio modulation classification tasks, even though the transformer based neural network is vulnerable to the PGD attacks, it is able to maintain the performance advantage over the ordinary CNN based modulation classifiers. For a wide range of PNR values and for moderate SNR, the Transformer provides approximately 10% more classification accuracy as compared to CNN. The focus of our future work is to enhance robustness of the Transformer against a wide range of adversarial attacks.

6. REFERENCES

[1] James A Sills, “Maximum-likelihood modulation classification for PSK/QAM,” in *MILCOM 1999. IEEE Military Communications. Conference Proceedings (Cat. No. 99CH36341)*. IEEE, 1999, vol. 1, pp. 217–220.

[2] Prokopios Panagiotou, Achilleas Anastasopoulos, and A Polydoros, “Likelihood ratio tests for modulation classification,” in *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority (Cat. No. 00CH37155)*. IEEE, 2000, vol. 2, pp. 670–674.

[3] Liang Hong and KC Ho, “Antenna array likelihood modulation classifier for BPSK and QPSK signals,” in *MILCOM 2002. Proceedings*. IEEE, 2002, vol. 1, pp. 647–651.

[4] Lu Mingquan, Xiao Xianci, and Li Leming, “Cyclic spectral features based modulation recognition,” in *Proceedings of International Conference on Communication Technology. ICCT’96*. IEEE, 1996, vol. 2, pp. 792–795.

[5] Elsayed Elsayed Azzouz and Asoke Kumar Nandi, “Modulation recognition using artificial neural networks,” in *Automatic Modulation Recognition of Communication Signals*, pp. 132–176. Springer, 1996.

[6] Ananthram Swami and Brian M Sadler, “Hierarchical digital modulation classification using cumulants,” *IEEE Transactions on communications*, vol. 48, no. 3, pp. 416–429, 2000.

[7] Timothy J O’Shea, Johnathan Corgan, and T Charles Clancy, “Convolutional radio modulation recognition networks,” in *International conference on engineering applications of neural networks*. Springer, 2016, pp. 213–226.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[9] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] Moein Mirmohammadsadeghi, Samer S Hanna, and Danijela Cabric, "Modulation classification using convolutional neural networks and spatial transformer networks," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2017, pp. 936–939.

[13] Rui Zhang, Zhendong Yin, Zhilu Wu, and Siyang Zhou, "A novel automatic modulation classification method using attention mechanism and hybrid parallel neural network," *Applied Sciences*, vol. 11, no. 3, pp. 1327, 2021.

[14] Shahab Hamidi-Rad and Swayambhoo Jain, "Mcformer: A transformer based deep neural network for automatic modulation classification," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021.

[15] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille, "Adversarial examples for semantic segmentation and object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1369–1378.

[16] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proceedings of the 2016 acm sigsac conference on computer and communications security*, 2016, pp. 1528–1540.

[17] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer, "Universal adversarial perturbations against semantic image segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2755–2764.

[18] Volker Fischer, Mummadi Chaithanya Kumar, Jan Hendrik Metzen, and Thomas Brox, "Adversarial examples for semantic image segmentation," *arXiv preprint arXiv:1703.01101*, 2017.

[19] Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel, "Adversarial examples for malware detection," in *European symposium on research in computer security*. Springer, 2017, pp. 62–79.

[20] Meysam Sadeghi and Erik G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2019.

[21] Lu Zhang, Sangarapillai Lambotharan, Gan Zheng, Basil AsSadhan, and Fabio Roli, "Countermeasures against adversarial examples in radio signal classification," *IEEE Wireless Communications Letters*, vol. 10 (8), pp. 1830–1834, August 2021.

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

[23] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[24] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[25] Battista Biggio and Fabio Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[26] Angelo Sotgiu, Ambra Demontis, Marco Melis, Battista Biggio, Giorgio Fumera, Xiaoyi Feng, and Fabio Roli, "Deep neural rejection against adversarial examples," *Eurasip Journal on Information Security*, vol. 5, no. 1, 2020.

[27] Timothy J O'Shea and Nathan West, "Radio Machine Learning Dataset Generation with GNU Radio," *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, 2016.