PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Unsupervised training dataset curation for deep-neural-net RF signal classification

George Sklivanitis, Jose Sanchez Viloria, Konstantinos Tountas, Dimitris Pados, Elizabeth Serena Bentley, et al.

George Sklivanitis, Jose A. Sanchez Viloria, Konstantinos Tountas, Dimitris A. Pados, Elizabeth Serena Bentley, Michael J. Medley, "Unsupervised training dataset curation for deep-neural-net RF signal classification," Proc. SPIE 12522, Big Data V: Learning, Analytics, and Applications , 125220C (13 June 2023); doi: 10.1117/12.2665151

SPIE.

Event: SPIE Defense + Commercial Sensing, 2023, Orlando, Florida, United States

Unsupervised Training Dataset Curation for Deep-Neural-Net RF Signal Classification

George Sklivanitis[a], Jose A. Sanchez Viloria[a], Konstantinos Tountas[a], Dimitris A. Pados[a], Elizabeth Serena Bentley[b], and Michael J. Medley[b]

[a]Center for Connected Autonomy and AI, Florida Atlantic University, Boca Raton, FL, USA
[b]Air Force Research Laboratory, Rome, NY, USA

ABSTRACT

We consider the problem of unsupervised (blind) evaluation and assessment of the quality of data used for deep neural network (DNN) RF signal classification. When neural networks train on noisy or mislabeled data, they often (over-)fit to the noise measurements and faulty labels, which leads to significant performance degradation. Also, DNNs are vulnerable to adversarial attacks, which can considerably reduce their classification performance, with extremely small perturbations of their input. In this paper, we consider a new method based on L1-norm principal-component analysis (PCA) to improve the quality of labeled wireless data sets that are used for training a convolutional neural network (CNN), and a deep residual network (ResNet) for RF signal classification. Experiments with data generated for eleven classes of digital and analog modulated signals show that L1-norm tensor conformity curation of the data identifies and removes from the training data set inappropriate class instances that appear due to mislabeling and universal black-box adversarial attacks and drastically improves/restores the classification accuracy of the identified deep neural network architectures.

Keywords: Modulation classification, AI/ML, data curation, tensor decomposition

1. INTRODUCTION

Rapid radio spectrum characterization in congested (and sometimes contested) environments plays an important role toward autonomous spectrum management and enforcement of policy/regulations for future spectrum sharing applications. In parallel, high-quality spectrum analytics (at either the waveform/modulation or the network protocol or the device level) offers an opportunity to recognize unlicensed spectrum/interfering users, malfunctioning equipment and take action. Existing approaches require expensive, high-maintenance expert systems that rely on prior knowledge of signal properties, features and decision statistics and focus on energy detection, localization and classification of spectrum activity under simplified hardware, propagation and radio environment models. Additionally, characterization of spectrum activity and of the corresponding radio devices requires tuning to the band and signal of interest to perform comparisons with existing baseline signal databases, thus incurring significant computational power and implementation/deployment cost before taking further action.

Over the past few years, large public image repositories and emerging high-performance graphics processing units (GPUs) have accelerated the adoption of deep neural networks (DNNs) as means to carry out visual object detection and image classification. However, the application of DNNs to raw physical-layer in-phase (I)/quadrature (Q) samples for radio-frequency (RF) signal classification[1] and device-level fingerprinting[2] has yet to be conclusively demonstrated. Deep learning relies on back-propagation with stochastic gradient descent to optimize large parametric neural network models. However, when neural networks train on noisy or mislabeled data, they often (over-)fit to the noise measurements and faulty labels, which leads to significant performance degradation. Also, it has been shown that DNNs are highly vulnerable to adversarial data attacks, which raises major security and robustness concerns. Adversarial data are malicious inputs that are obtained by slightly perturbing an original input, in such a way that the deep learning algorithm misclassifies them. The adversarial attacks can be divided into white-box and black-box attacks, based on the amount of knowledge that the

Further author information: (Send correspondence to George Sklivanitis.) George Sklivanitis:
E-mail: gsklivanitis@fau.edu, Telephone: 1 561 297 1163

Big Data V: Learning, Analytics, and Applications, edited by Panos P. Markopoulos, Bing Ouyang, Vagelis Papalexakis, Proc. of SPIE Vol. 12522, 125220C
© 2023 SPIE · 0277-786X · doi: 10.1117/12.2665151

adversary has about the model. In white-box attacks, the adversary has the full knowledge of the classifier, while in black-box attacks the adversary does not have any knowledge (or has limited knowledge) of the classifier. In this paper, we consider a new method based on L1-norm principal-component analysis (PCA) to improve the quality of labeled wireless data sets that are used for training a convolutional neural network (CNN),[3] and a deep residual network (ResNet)[1] - that typically lead to classification accuracy values around 90% for high-signal-to-noise ratio (NSR) RF signals. We test experimentally the classification accuracy of the CNN and ResNet DNNs using synthetically generated *IQ* data that are publicly available in the GNU radio machine learning (ML) dataset.[4] The dataset includes eleven (11) classes of digital and analog modulated signals (BPSK, QPSK, 8PSK, 4PAM, 16QAM, 64QAM, GFSK, CPFSK, AM-SSB, AM-DSB and WBFM signals). Random processes for carrier frequency offset, sample-rate offset, additive white Gaussian noise, multi-path and fading introduce different hardware and channel impairments during data generation. Examples of labeled signals per class are organized in a three way tensor. The conformity of the tensor data per class is evaluated through iterative projections on robust, high confidence data characterizations per class that are returned by L1-norm tensor subspaces.[5–8] Non-conforming tensor slabs are likely to be contaminated by excessive noise or mislabeled examples due to mistakes made during data annotation or due to black-box adversarial attacks and are automatically removed from the dataset.

2. ACE: AUTONOMOUS CONFORMITY EVALUATION

We consider organizing *IQ* data from each class of signals $c \in \{1, 2, \ldots, 11\}$ in a 3-mode tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, where I_1 denotes the signal duration in samples, $I_2 = 2$ includes the *IQ* components of each signal, and I_3 is the number of signal examples per class. Autonomous conformity evaluation (ACE) converts the original tensor data $\boldsymbol{\mathcal{X}}$, to a new tensor of the exact same dimensions $\boldsymbol{\mathcal{W}}$, where each new tensor entry measures the conformity of that entry with respect to all other data points. The conformity metric takes values from the $[0, 1]$ set of real numbers, with conformity values close to 0 indicating "misbehaving" data points, and values close to 0 corresponding to nominal data points. This is achieved, by utilizing iteratively refined L1-norm (absolute-error) data subspaces.[5,6,8] Detection of non-conforming data entries enables the identification of contaminated tensor data slabs.

With respect to the i-th data tensor, $i \in 1, 2, \ldots, 11$ and for its n-th mode unfolding for all $n \in \{1, 2, 3\}$, we create data matrices $\mathbf{X}_{(1)} \in \mathbb{R}^{I_1 \times I_2 I_3}$, $\mathbf{X}_{(2)} \in \mathbb{R}^{I_2 \times I_1 I_3}$, $\mathbf{X}_{(3)} \in \mathbb{R}^{I_3 \times I_1 I_2}$ and we calculate the r_1, r_2, r_3 L1-norm principal components $\mathbf{Q}_1^{(0)} \in \mathbb{R}^{I_1 \times r_1}$, $\mathbf{Q}_2^{(0)} \in \mathbb{R}^{I_2 \times r_2}$, $\mathbf{Q}_3^{(0)} \in \mathbb{R}^{I_3 \times r_3}$ by solving the following problems[5,6,9]

$$\mathbf{Q}_1^{(0)} = \underset{\substack{\mathbf{Q} \in \mathbb{R}^{I_1 \times r_1}, \\ \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_{r_1}}}{\operatorname{argmax}} \left\| \mathbf{X}_{(1)}^T \mathbf{Q} \right\|_1, \tag{1}$$

$$\mathbf{Q}_2^{(0)} = \underset{\substack{\mathbf{Q} \in \mathbb{R}^{I_2 \times r_2}, \\ \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_{r_2}}}{\operatorname{argmax}} \left\| \mathbf{X}_{(2)}^T \mathbf{Q} \right\|_1, \tag{2}$$

$$\mathbf{Q}_3^{(0)} = \underset{\substack{\mathbf{Q} \in \mathbb{R}^{I_3 \times r_3}, \\ \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_{r_3}}}{\operatorname{argmax}} \left\| \mathbf{X}_{(3)}^T \mathbf{Q} \right\|_1. \tag{3}$$

The resulting bases emphasize the subspaces spanned by the nominal (uncorrupted) entries of the original tensor $\boldsymbol{\mathcal{X}}$. Tensor entries that are contaminated with anomalous data are not spanned by the resulting bases. Data conformity for the 1-st mode of the tensor is calculated by projecting all columns of $\left[\mathbf{X}_{(1)}\right]_{:,i}, i = 1, 2, \ldots, I_2 I_3$ on the calculated subspace $\mathbf{Q}_1^{(0)}$ as

$$d_{1,i}^{(1)} = \left\| \mathbf{Q}_1^{(0)} \mathbf{Q}_1^{(0)^T} \left[\mathbf{X}_{(1)}\right]_{:,i} \right\|_2^{-1}, \quad \forall i = 1, 2, \ldots, I_2 I_3. \tag{4}$$

Similarly, data conformity for the 2-nd mode of the tensor is calculated as

$$d_{2,i}^{(1)} = \left\| \mathbf{Q}_2^{(0)} \mathbf{Q}_2^{(0)^T} \left[\mathbf{X}_{(2)}\right]_{:,i} \right\|_2^{-1}, \quad \forall i = 1, 2, \ldots, I_1 I_3, \tag{5}$$

and conformity for the 3-rd mode of the tensor is calulcated as

$$d_{3,i}^{(1)} = \left\| \mathbf{Q}_3^{(0)} \mathbf{Q}_3^{(0)^T} \left[\mathbf{X}_{(3)} \right]_{:,i} \right\|_2^{-1}, \quad \forall i = 1, 2, \ldots, I_1 I_2. \tag{6}$$

Small $d_{n,i}^{(1)}$ values are expected if $\left[\mathbf{X}_{(n)}\right]_{:,i}$, $n \in \{1,2,3\}$ is an anomalous data vector and large $d_{n,i}^{(1)}$ values if $\left[\mathbf{X}_{(n)}\right]_{:,i}$, $n \in \{1,2,3\}$ is a nominal data vector, or conforming. After the calculation of the projections, we fold the conformity values to a tensor form as follows

$$\boldsymbol{\mathcal{W}}_1^{(1)} = \text{tensorization}\left(\left[d_{1,1}^{(1)}, \ldots, d_{1,I_2 I_3}^{(1)} \right]^T \odot \mathbf{1}_{I_1 \times I_2 I_3}, 1 \right), \tag{7}$$

$$\boldsymbol{\mathcal{W}}_2^{(1)} = \text{tensorization}\left(\left[d_{2,1}^{(1)}, \ldots, d_{1,I_1 I_3}^{(1)} \right]^T \odot \mathbf{1}_{I_2 \times I_1 I_3}, 1 \right), \tag{8}$$

$$\boldsymbol{\mathcal{W}}_3^{(1)} = \text{tensorization}\left(\left[d_{3,1}^{(1)}, \ldots, d_{1,I_1 I_2}^{(1)} \right]^T \odot \mathbf{1}_{I_3 \times I_1 I_2}, 1 \right), \tag{9}$$

where $\mathbf{1}_{I_1 \times I_2 I_3}$, $\mathbf{1}_{I_2 \times I_1 I_3}$, $\mathbf{1}_{I_3 \times I_1 I_2}$ stand for all-ones matrices of dimension $I_1 \times I_2 I_3$, $I_2 \times I_1 I_3$, and $I_3 \times I_1 I_2$, respectively, and the *tensorization*$(\cdot)$ operation converts the unfolded matrix to the original three-mode tensor form (reverting the unfolding process). Tensor $\boldsymbol{\mathcal{W}}_1^{(1)}$ contains the conformity values corresponding to each column of the original tensor $\boldsymbol{\mathcal{X}}$. We repeat the above process for the rest of the modes of the original data tensor, and calculate the conformity tensors $\boldsymbol{\mathcal{W}}_2^{(1)}$ and $\boldsymbol{\mathcal{W}}_3^{(1)}$. We calculate the final individual entry conformity tensor $\boldsymbol{\mathcal{W}}^{(1)}$ by combining the above calculated tensors in an additive weighting fashion (according to assumed relative "importance") and max-min normalization so that each element is in the $[0,1]$ range

$$\boldsymbol{\mathcal{W}}^{(1)} = \frac{\sum_{n=1}^{3} \alpha_n \boldsymbol{\mathcal{W}}_n^{(1)} - \min\left(\sum_{n=1}^{3} \alpha_k \boldsymbol{\mathcal{W}}_n^{(1)} \right)}{\max\left(\sum_{n=1}^{3} \alpha_n \boldsymbol{\mathcal{W}}_n^{(1)} \right) - \min\left(\sum_{n=1}^{3} \alpha_n \boldsymbol{\mathcal{W}}_n^{(1)} \right)}, \tag{10}$$

where the weighting parameters $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}^+, \sum_{n=1}^{3} \alpha_n = 1$ measure the importance of the n-th tensor mode $\min(\cdot)$ returns the minimum element of its tensor argument, and $\max(\cdot)$ returns the maximum element. The normalization in (10) leads to value 0 for the least conforming elements and value 1 for the most conforming ones. The final conformity tensor $\boldsymbol{\mathcal{W}}^{(1)}$ enables element-wise conformity of the original tensor data.

Next, the original tensor dataset is globally weighted through the conformity tensor $\boldsymbol{\mathcal{W}}^{(1)}$ by element-by-element multiplication of $\boldsymbol{\mathcal{X}}$ with $\boldsymbol{\mathcal{W}}^{(1)}$. The refined L1-norm tensor bases are calculated by means of L1-HOOI[9] on $\boldsymbol{\mathcal{X}}^{(1)} = \boldsymbol{\mathcal{X}} \odot \boldsymbol{\mathcal{W}}^{(1)}$, where $\odot$ is the element-wise or Hadamard product. Conformity tensors $\boldsymbol{\mathcal{W}}^{(1)}, \boldsymbol{\mathcal{W}}^{(2)}, \ldots,$ are iteratively generated until numerical convergence to $\boldsymbol{\mathcal{W}}$ is observed i.e.,

$$\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{W}}^{(l)}, \text{ such that } \|\boldsymbol{\mathcal{W}}^{(l)} - \boldsymbol{\mathcal{W}}^{(l-1)}\|_{\mathrm{F}} < \epsilon, \tag{11}$$

for some small $\epsilon > 0$. To identify tensor slabs that are contaminated by anomalous/mislabeled *IQ* signal examples, we calculate the mean data conformity value per slab as follows

$$\bar{w}_k = \frac{1}{I_1 * I_2} \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \left[\boldsymbol{\mathcal{W}} \right]_{i,j,k}, k = 1, 2, \ldots, I_3. \tag{12}$$

For each class, tensor slabs with high conformity values contain nominal *IQ* signal examples, while low-conformity slabs contain contaminated *IQ* signal samples for this class. We choose to remove slabs with conformity value $\bar{w}_n$ below a pre-defined threshold $t_{\text{cutoff}} \in [0,1]$ from the received tensor $\boldsymbol{\mathcal{X}}$, thus recovering a new curated tensor $\tilde{\boldsymbol{\mathcal{X}}} \in \mathbb{R}^{I_1 \times I_2 \times N}$, where $N < I_3$. Our objective is to minimize the number of anomalous/mislabeled *IQ* signal examples contained in the new tensor $\tilde{\boldsymbol{\mathcal{X}}}$.

Algorithm 1 Autonomous conformity evaluation (ACE) algorithm for tensor data curation

Input: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$; init. $\{\mathbf{Q}_n^{(0)}\}$ (*e.g.*, arbitr. or HOSVD[10]); $\{r_n\}$, $\{\alpha_n\}$, $n \in \{1,2,3\}$; $p, \epsilon > 0, t_{\text{cutoff}} \in [0,1]$

1: $\boldsymbol{\mathcal{W}} \leftarrow \text{ACE}\left(\boldsymbol{\mathcal{X}}, \left\{\{r_n\}, \{\alpha_n\}, \{\mathbf{Q}_n^{(0)}\}\right\}_{n\in\{1,2,3\}}\right)$

2: $\bar{w}_k \leftarrow 1/(I_1 * I_2) \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} [\boldsymbol{\mathcal{W}}]_{i,j,k}, \quad k = 1, 2, \ldots, I_3$

3: Sort (descendingly) $\bar{\mathbf{w}}$ and remove the bottom p elements where $\bar{w}_k < t_{\text{cutoff}}, k = 1, 2, \ldots, p$

4: Keep the remaining $N < I_3$ indices of the original dataset and create the new training dataset $\bar{\boldsymbol{\mathcal{X}}} \in \mathbb{R}^{I_1 \times I_2 \times N}$

Function: $\text{ACE}\left(\boldsymbol{\mathcal{X}}, \left\{\{r_n\}, \{\alpha_n\}, \{\mathbf{Q}_n^{(0)}\}\right\}_{n\in\{1,2,3\}}\right)$[11]

1: Init. $l \leftarrow 1$; $\boldsymbol{\mathcal{W}}^{(0)} \leftarrow \mathbf{1}_{I_1 \times I_2 \times I_3}$; $a \leftarrow \left\|\boldsymbol{\mathcal{W}}^{(0)}\right\|_F$; $\boldsymbol{\mathcal{X}}^{(0)} \leftarrow \boldsymbol{\mathcal{X}}$

2: **while** $a > \epsilon$ **do**

3: $\quad \{\mathbf{Q}_n^{(l)}\}_{n\in\{1,2,3\}} \leftarrow \text{L1} - \text{HOOI}\left(\boldsymbol{\mathcal{X}}^{(l-1)}, \{\mathbf{Q}_n^{(l-1)}\}_{n\in\{1,2,3\}}\right)$[9]

4: $\quad$ **for** $n \in \{1,2,3\}$ **do**

5: $\quad\quad I_n = \{\prod_{k\in\{1,2,3\}\setminus n} I_k\}$

6: $\quad\quad$ **for** $i \in \{I_n\}$ **do**

7: $\quad\quad\quad d_{n,i}^{(l)} \leftarrow \left\|\mathbf{Q}_n^{(l)} \mathbf{Q}_n^{(l)^T} \left[\mathbf{X}_{(n)}\right]_{:,i}\right\|_2^{-1}$

8: $\quad\quad \boldsymbol{\mathcal{W}}_n^{(l)} \leftarrow$tensorization$\left(\mathbf{1}_{I_n} \odot \left[d_{n,1}^{(l)}, \ldots, d_{n,I_n}^{(l)}\right]^T, n\right)$

9: $\quad \boldsymbol{\mathcal{W}}_{\text{temp}} \leftarrow \sum_{n\in\{1,2,3\}} \alpha_n \boldsymbol{\mathcal{W}}_n^{(l)}$; $\mu \leftarrow \min(\boldsymbol{\mathcal{W}}_{\text{temp}})$; $\xi \leftarrow \max(\boldsymbol{\mathcal{W}}_{\text{temp}})$;

10: $\quad \boldsymbol{\mathcal{W}}^{(l)} \leftarrow \frac{\boldsymbol{\mathcal{W}}_{\text{temp}} - \mu}{\xi - \mu}$; $\boldsymbol{\mathcal{X}}^{(l)} \leftarrow \boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{W}}^{(l)}$

11: $\quad a \leftarrow \left\|\boldsymbol{\mathcal{W}}^{(l)} - \boldsymbol{\mathcal{W}}^{(l-1)}\right\|_F$

12: $\quad l \leftarrow l + 1$

3. TRAINING-TIME ATTACKS

3.1 Mislabeling

For the mislabeling event, we considered the case of human errors during annotation/labeling of signal examples. In practice, since most of the annotation/labeling of signals is done manually by a human analyst, the probability of error is high. In this experiment, we consider random selection of 25% of the samples from each class (from the 11 classes of signals in our dataset), and we use these samples as our pool of outlier/mislabeled *IQ* signal examples. Then, we contaminate each class with samples from our pool without repetition i.e., we avoid "contaminating" one class with samples from the same class. Thus, the dataset is reorganized to contain 25% mislabeled signal examples across all classes. We also consider the case where only part of the dataset is contaminated, i.e., either low or high SNR signal examples are contaminated only. In all cases, the total amount of signal examples per class remains as before we introduced mislabeled data.

3.2 Adversarial Attacks

For adversarial attack events, let us denote the signal received at a receiver as $\mathbf{x}$. When an attacker is present, it also transmits a signal to create a low-power perturbation $\mathbf{r}_{\text{x}}$ at the receiver. Therefore, the received signal is written as $\mathbf{x}_{\text{adv}} = \mathbf{x} + \mathbf{r}_{\text{x}}$. The attacker's target is to design $\mathbf{r}_{\text{x}}$ in such a manner that it causes misclassification for the underlying DNN at the receiver.

Many algorithms have been proposed in the literature for designing such perturbations. In[12] the authors present an algorithm that utilizes the entire input $\mathbf{x}$ to the DNN classifier model and the corresponding modulation label to develop a computationally efficient method for crafting adversarial perturbations. The main drawbacks of such an algorithm are that it requires a-priori knowledge of the entire input. Also, each element of $\mathbf{x}$ is perturbed by its corresponding element in $\mathbf{r}_{\text{x}}$, i.e., the attacker must be synchronous with the transmitter, and it is assumed that the attacker has perfect knowledge of the underlying model. Another common method

of creating such perturbations was presented in.[13] The authors present an iterative -computational expensive- algorithm that in each iteration generates an adversarial perturbation of each input.

In this experiment, we consider a low-computational complexity PCA-based approach that was originally proposed in[12] for generating perturbations that provide a better fooling rate on the dataset compared to.[13] Given a subset of the inputs $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ to the DNN classifier, where $N < I_3$, and their associated perturbation directions $\{\mathbf{n}_{\mathbf{x}_1}, \mathbf{n}_{\mathbf{x}_2}, \ldots, \mathbf{n}_{\mathbf{x}_N}\}$, where $\mathbf{n}_{\mathbf{x}_i} = \nabla_{x_i} L(\theta, \mathbf{x}_i, \mathbf{y}^{\text{true}}) / \|\nabla_{\mathbf{x}_i} L(\theta, \mathbf{x}_i, \mathbf{y}^{\text{true}})\|_2$, where θ is the set of model parameters, $L(\cdot)$ is the loss function of the model, and $\mathbf{y}^{\text{true}}$ is the true label, the algorithm can craft a universal adversarial perturbation that can fool the model with high probability, independently of the input applied to the model. In the ML literature, such a perturbation is called a black-box universal adversarial perturbation (UAP).[13] To achieve this, we use as the direction of the UAP the direction of the first principal component of $\mathbf{X}^{N \times p} = [\mathbf{n}_{\mathbf{x}_1}, \ldots, \mathbf{n}_{\mathbf{x}_N}]^T$ where p is the dimension of the input.

4. EXPERIMENTAL STUDIES

The RadioML 2016.10A[3,14] -a synthetic dataset generated with GNU Radio– contains more that 1200000 complex *IQ* signal samples, where each sample is associated with one modulation scheme at a specific SNR.[1] The dataset contains 11 different modulations: OOK, 4ASK, BPSK, QPSK, 8PSK, 16QAM, AM-SSB-SC, AM-DSB-SC, FM, GMSK, OQPSK. Signal examples are generated for 20 different SNR levels from −18 dB to 20 dB with a step of 2 dB. Each signal example is a vector of size 256 elements, which corresponds to 128 in-phase and 128 quadrature samples. We split our data to 70% for our training set and 30% for our test set. Experiments are performed with two NVIDIA GeForce RTX 2080 Ti GPUs.

Regarding training-time attacks, for the mislabeling event, we randomly select 25% of the data from each one of the 11 classes. Each class is then contaminated with samples belonging to the other 10 classes, e.g., the OOK class is contaminated with random signal examples from 4ASK, BPSK, QPSK, 8PSK, 16QAM, AM-SSB-SC, AM-DSB-SC, FM, GMSK, OQPSK. The same procedure is carried out for the low-SNR and high-SNR mislabeling events, but in those cases only data with SNR below −4 dB and above −4 dB are contaminated, respectively. Regarding the adversarial attack event, we used 20% of the training dataset for generating the attack and introduce perturbations -as these are generated by Algorithm 2 in[12]- to 20% of the training set.

We first organized the data in each class in a 3-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{128 \times 2 \times N_c}$, where the first dimension 128 denotes the signal length, the second dimension 2 denotes the *IQ* components of the signal and N_c denotes the number of signal examples per class. The α parameters for ACE were set to $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}$. On each case, after calculating the average conformity per slab, the conformity values were sorted descendingly and 20% of the data was dropped. This threshold of 20% was selected after experimenting with different thresholds and it was the one that produced the best classification results.

Figures 1, 2 and 3 present the average classification accuracy of the CNN across the 11 classes versus SNR for the mislabeling corruption event. Fig. 1 shows that classification performance is severely affected, especially in the high-SNR regime, where the performance drops from 84% to 71%. By curating the data with ACE, we are able to restore average accuracy to 80%. The same holds for the high-SNR mislabeling event (Fig. 2), where only samples above −4 dB were affected by contamination. After data corruption, the average accuracy is 67%, while after data curation, the accuracy returns to 81%. The low-SNR corruption event (Fig. 3) is not as visible as the high-SNR case because the network is not able to distinguish the signals in the low-SNR regime. Still, Fig. 3 shows that for −6 dB, the average performance of the CNN trained on the corrupted dataset is 20%, while for training on nominal data average performance is 37%. After applying ACE, average classification accuracy is 30%.

Figures 4, 5, and 6 present the average classification accuracy of the ResNet DNN across the 11 signal classes versus SNR for the mislabeling event. Similar to the CNN network, ResNet classification performance is severely affected (Fig. 4), especially in the high-SNR regime, where the performance drops from 90% to 83%. Upon applying ACE, we only get 1% gain in the high-SNR regime, while in the low-SNR regime, we get even better performance than training on the original, non-corrupted version of the dataset. For the high-SNR mislabeling event (Fig. 5), ResNet average accuracy is 66%, and after data curation with ACE, the accuracy is restored to 80%. Low-SNR corruption is more prevalent in the ResNet network. The accuracy is reduced significantly in

the low-SNR regime. We observe that training with high-SNR samples (Fig. 6) is affected as well. After data curation, we are able to remove most of the corrupted signal samples and average classification performance returns to acceptable levels.

Figures 7 and 8 present the average classification accuracy of the CNN and ResNet networks across the 11 classes versus SNR for the adversarial attack event. It is clear that while ACE is not able to recover the original performance for CNN, we are still able to have 25% performance gain, compared to training with the attacked dataset. For ResNet, data curation with ACE provides 20% in performance gain, compared to the performance we get after training with the attacked dataset.

5. CONCLUSIONS

We present a blind, unsupervised way for calculating the element-by-element conformity of tensor data sets that contain *IQ* signal examples from different modulation classes. The original tensor data are transformed into tensors of the same dimensions where each new tensor entry measures the conformity of that entry through iterative projections on refined L1-norm tensor subspaces. Data with low conformity values are removed from the *IQ* signal dataset to improve training of DNNs. We show that ACE could restore DNN classification accuracy at acceptable levels when training data is contaminated by mislabeled signal examples and black-box universal adversarial perturbations of the input data.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grants EEC-2133516, ECCS-2030234, ITE-2226392 and CNS-2117822 and the U.S. Air Force Research Laboratory under Grant FA8750-20-C-1021 and FA8750-21-F-1012. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of AFRL.

REFERENCES

[1] O'Shea, T. J., Roy, T., and Clancy, T. C., "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing* **12**(1), 168–179 (2018).

[2] Jian, T., Rendon, B. C., Ojuba, E., Soltani, N., Wang, Z., Sankhe, K., Gritsenko, A., Dy, J., Chowdhury, K., and Ioannidis, S., "Deep learning for rf fingerprinting: A massive experimental study," *IEEE Internet of Things Magazine* **3**(1), 50–57 (2020).

[3] OShea, T. J., Corgan, J., and Clancy, T. C., "Convolutional radio modulation recognition networks," in [*Proc. Int. Conf. Eng. Appl. Neural Netw.*], 213–226 (2016).

[4] O'Shea, T. J. and West, N., "Radio machine learning dataset generation," in [*GNU Radio Conference*], (2016).

[5] Markopoulos, P. P., Karystinos, G. N., and Pados, D. A., "Optimal algorithms for L1-subspace signal processing," *IEEE Trans. Signal Process.* **62**, 5046–5058 (Oct. 2014).

[6] Markopoulos, P. P., Kundu, S., Chamadia, S., and Pados, D. A., "Efficient L1-norm principal-component analysis via bit flipping," *IEEE Trans. Signal Process.* **65**, 4252–4264 (Aug. 2017).

[7] Tountas, K., Pados, D. A., and Medley, M. J., "Conformity evaluation and L_1-norm principal-component analysis of tensor data," in [*SPIE Big Data: Learning, Analytics, and Applications Conf., SPIE Defense and Commercial Sensing*], (Apr. 2019).

[8] Tountas, K., Sklivanitis, G., Pados, D. A., and Medley, M. J., "Tensor data conformity evaluation for interference-resistant localization," in [*Proc. Asilomar*], 1582–1586 (Pacific Grove, CA, Nov. 2019).

[9] Chachlakis, D. G., Prater-Bennette, A., and Markopoulos, P. P., "L1-norm tucker tensor decomposition," *IEEE Access* **7**, 178454–178465 (2019).

[10] Kolda, T. G. and Bader, B. W., "Tensor decompositions and applications," *SIAM Review* **51**(3), 455–500 (2009).

[11] Tountas, K., Chachlakis, D. G., Markopoulos, P. P., and Pados, D. A., "Iteratively re-weighted l1-pca of tensor data," in [*2019 53rd Asilomar Conference on Signals, Systems, and Computers*], 1658–1661 (2019).

[12] Sadeghi, M. and Larsson, E. G., "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Communications Letters* **8**(1), 213–216 (2019).

[13] Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P., "Universal adversarial perturbations," in [*2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 86–94 (2017).

[14] O'Shea, T. J., Corgan, J., and Clancy, T. C., "Unsupervised representation learning of structured radio communication signals," *CoRR* **abs/1604.07078** (2016).

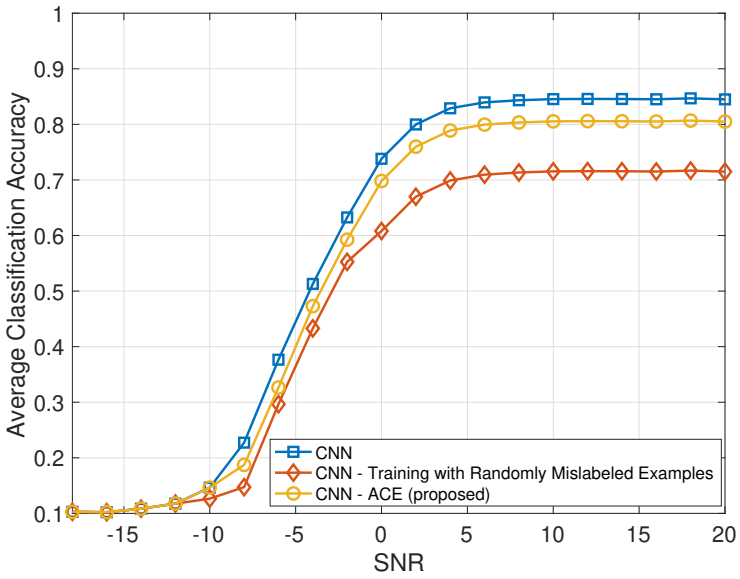

Figure 1: Average classification accuracy vs. SNR for CNN - Random mislabeling corruption.

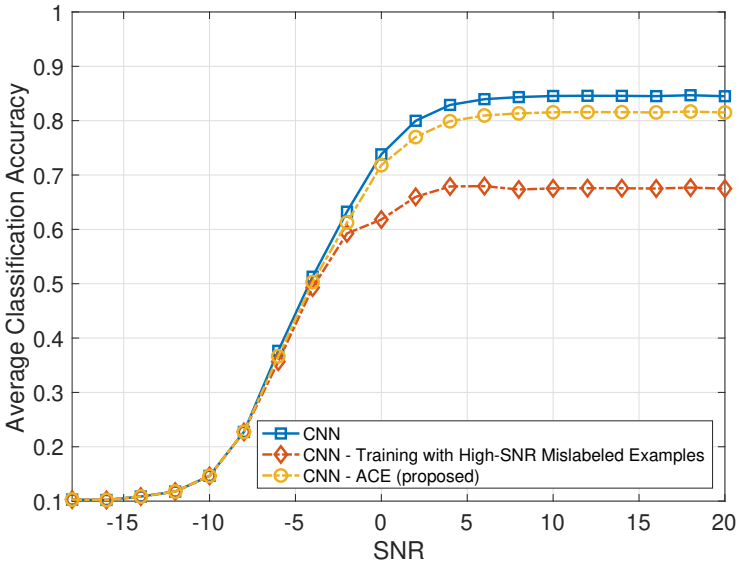

Figure 2: Average classification accuracy vs. SNR for CNN - High SNR mislabeling corruption.

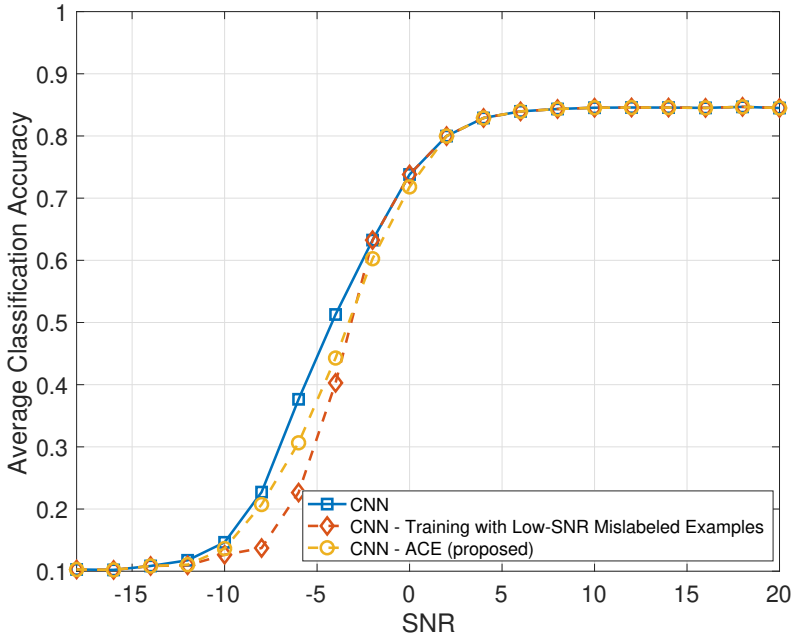

Figure 3: Average classification accuracy vs. SNR for CNN - Low SNR mislabeling corruption.

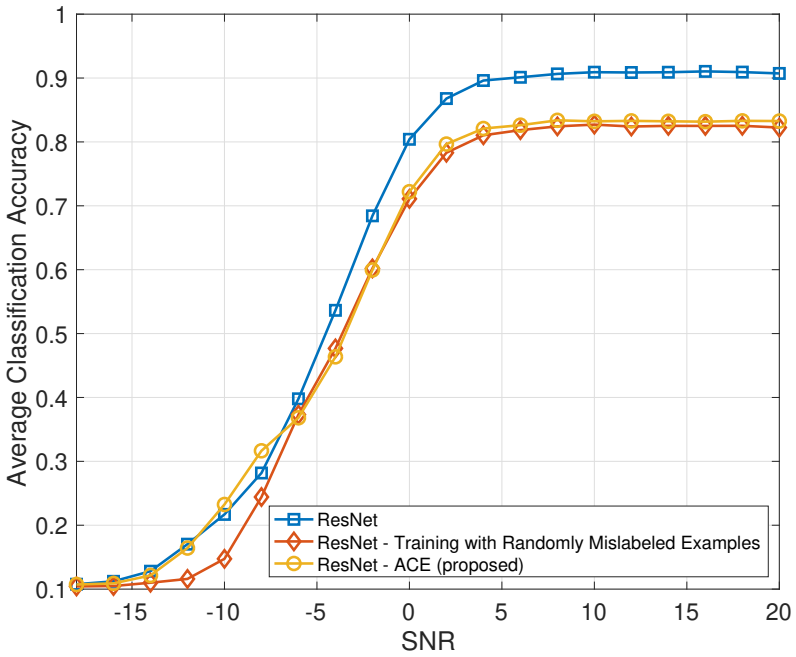

Figure 4: Average classification accuracy vs. SNR for ResNet - Random mislabeling corruption.

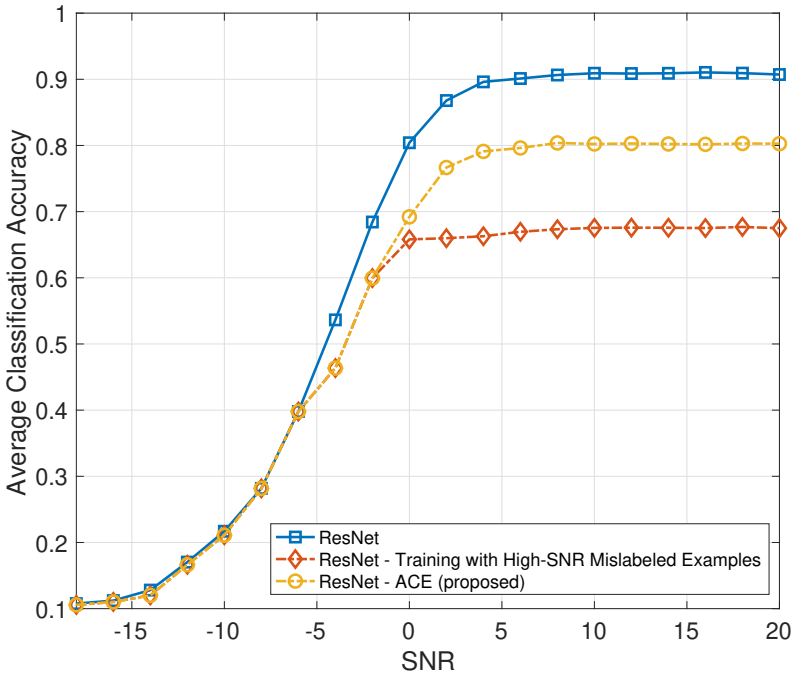

Figure 5: Average classification accuracy vs. SNR for ResNet - High SNR mislabeling corruption.

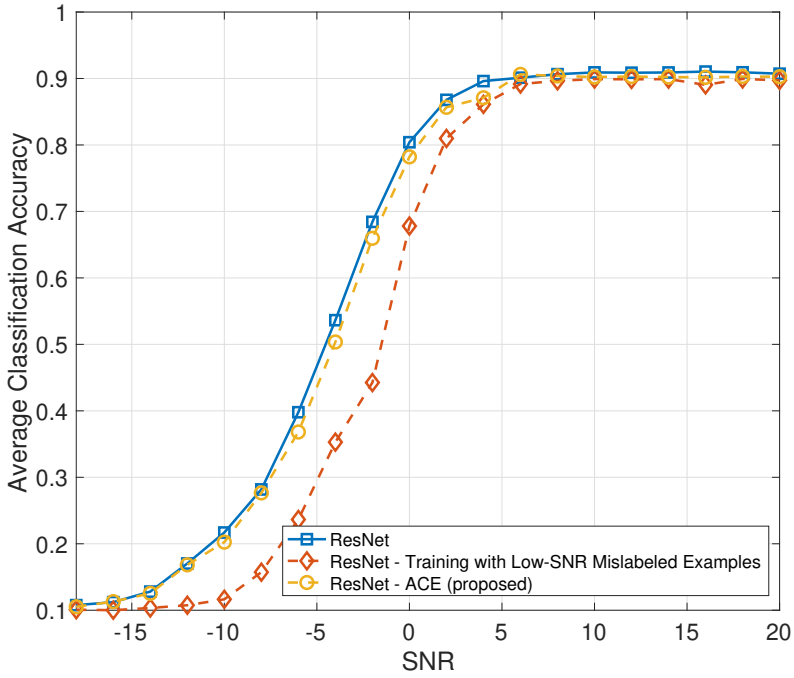

Figure 6: Average classification accuracy vs. SNR for Resnet - Low SNR mislabeling corruption.

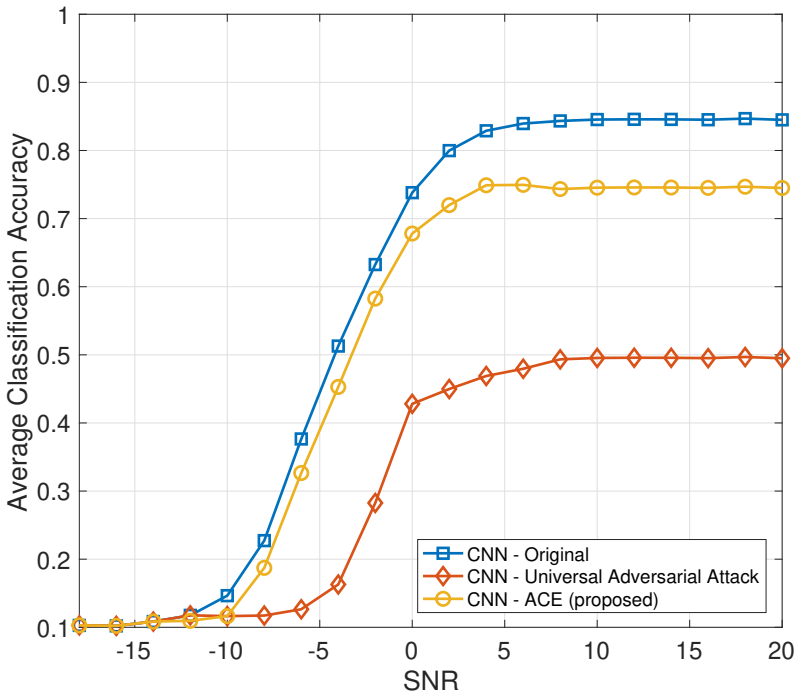

Figure 7: Average classification accuracy vs. SNR for CNN - Universal adversarial attack.

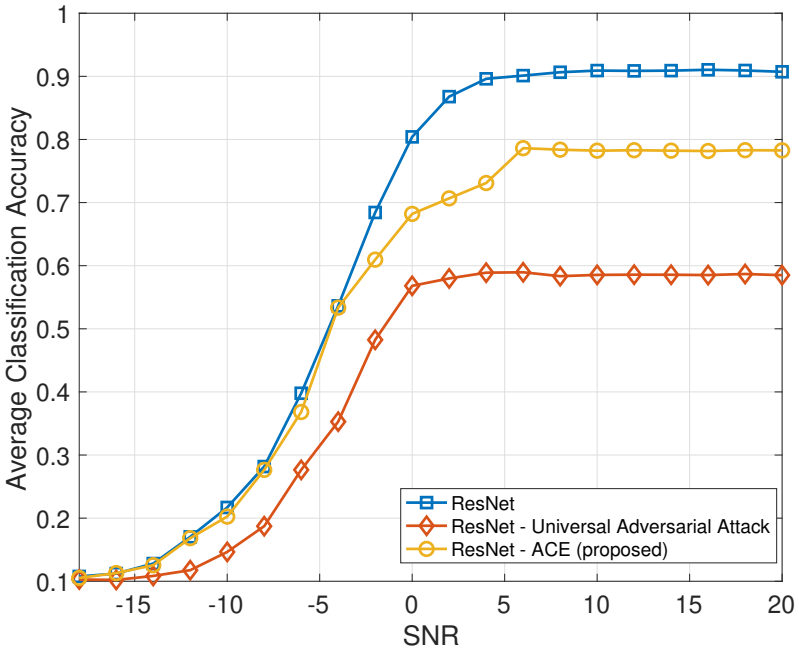

Figure 8: Average classification accuracy vs. SNR for ResNet - Universal adversarial attack.