CWGAN-GP Augmented CAE for Jamming Detection in 5G-NR in Non-IID Datasets

Samhita Kuili, Mohammadreza Amini, Burak Kantarci
School of Electrical and Computer Engineering
University of Ottawa
Ottawa, Canada
Emails: {skuil016, mamini6, burak.kantarci}@uottawa.ca

***Abstract*—In the ever-expanding domain of 5G-NR wireless cellular networks, over-the-air jamming attacks are prevalent as security attacks, compromising the quality of the received signal. We simulate a jamming environment by incorporating additive white Gaussian noise (AWGN) into the real-world In-phase and Quadrature (I/Q) OFDM datasets. A Convolutional Autoencoder (CAE) is exploited to implement a jamming detection over various characteristics such as heterogenous I/Q datasets; extracting relevant information on Synchronization Signal Blocks (SSBs), and fewer SSB observations with notable class imbalance. Given the characteristics of datasets, balanced datasets are acquired by employing a Conv1D conditional Wasserstein Generative Adversarial Network-Gradient Penalty(CWGAN-GP) on both majority and minority SSB observations. Additionally, we compare the performance and detection ability of the proposed CAE model on augmented datasets with benchmark models: Convolutional Denoising Autoencoder (CDAE) and Convolutional Sparse Autoencoder (CSAE). Despite the complexity of data heterogeneity involved across all datasets, CAE depicts the robustness in detection performance of jammed signal by achieving average values of 97.33% precision, 91.33% recall, 94.08% F1-score, and 94.35 % accuracy over CDAE and CSAE.**

***Index Terms*—Data augmentation, Deep learning, Jamming detection, Convolutional autoencoder, 5G NR.**

I. Introduction

In recent years, 5G-NR wireless communication has been booming with a significant increase in wireless devices, for instance, smartphones, tablets, IoT, and massive IoT devices. With the advent of telecommunication infrastructure, wireless technologies encompass massive multiple input multiple output (MIMO) [1], millimeter-wave (mmwave) [2], carrier aggregation [3], learning-based resource allocation [4] which provision for end-to-end service connectivity between a 5G cellular network and end-users. On the contrary, a 5G-NR wireless cellular network is also susceptible to security attacks, notably jamming attacks, which intentionally disrupt signal-to-noise ratio, and bit error rate of the transmitted signals, degrading the communication quality. Jamming attacks target physical layer downlink channels and downlink signals of 5G NR, exploiting the inherent vulnerabilities in Synchronization Signal Blocks (SSBs), which contain vital components like Primary and Secondary Synchronization Signals (PSS and SSS) responsible for cell identification and user association with gNodeB (gNB) [5].

A critical problem in 5G-NR networks is the heterogeneous data distribution from diverse user devices, as data

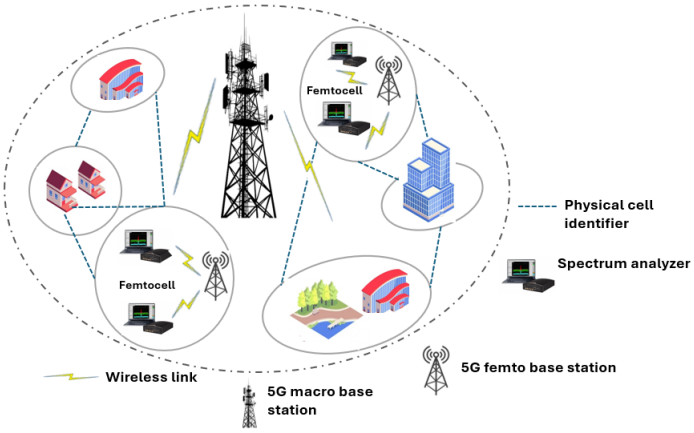

Fig. 1. Jamming detection in a 5G-NR cellular network within a femtocell

is typically non-independent and identical distributed (non-IID) due to diverse geographical location. This causes the user datasets to vary significantly in size and data distribution across multiple users [6]. As the 5G-NR network expands, jamming detection necessitates machine learning techniques [7], [8] and deep learning on physical layer (PHY) to comprehend the underlying patterns of a propagated received signal. Existing deep learning-based detection methods assume uniform data distribution, which may not perfectly align with a real world 5G deployments where non-independent and identical distributed data is prevalent. We propose a jamming detection paradigm that takes into account for heterogeneous data obtained from each user while addressing class imbalance issues in real-world datasets. Varotto et al. [9] trains a convolutional autoencoder (CAE) only on non-jammed signals and proposes security strategies against attacks in orthogonal frequency-division multiplexing (OFDM)-based 5G signals. Additional models, such as the double-threshold deep neural network (DT-DDNN) [10] enable detection of wider types of jammers with lower false positive and miss detection rates by transforming I/Q samples into 2D images. Almazrouei et al. [11] propose a data-driven deep learning approach to denoise radio signals of IEEE 802.11 protocols without relying on expert knowledge by using convolutional denoising autoencoder and highlight an improvement in classification accuracy by exploiting both decoder and classifier. Luo et al.

[12] propose a convolutional sparse autoencoder to sparsify the feature maps by integrating max-pooling into efficient feature leaning. These learned features are further used to propose a image classification strategy using the CSAE by integrating convolutional neural network.

Jamming signals are rare, leading to significant class imbalance that results in poor deep learning performance in classifying non-jammed and jammed SSB signals. Varying channel conditions and interference levels alleviate the learning process. To address this challenge, our framework comprises Conditional Wasserstein Generative Adversarial Network with Gradient Penalty (CWGAN-GP) for augmenting minority class observations and mitigate data imbalance while CAE enhances feature extractions to improve classification performance. Usage of Generative Adversarial Network (GAN) has been promising in effectively generating synthetic observations that closely resemble with the real data distribution and elevating the number of observation in the data. Chapaneri and Shah [13], [14] discuss a reliable technique to enable data augmentation by exploiting a variant of GAN: Wasserstein GAN (WGAN) to improve the minority attack classification problem caused by cyber-attacks in network traffic. Chen et al. [15] use conditional Wasserstein generative adversarial network with gradient penalty (CWGAN-GP) based data augmentation to detect winding deformation in power transformers, and shows promising improvements over conventional Artificial intelligence (AI)-based fault diagnosis models. A visual representation of femtocells in 5G-NR cellular network is shown in Fig. 1. The main contributions of the paper are highlighted below:

1) A two-stage jamming detector tailored for 5G networks in RF domain is implemented by capturing In-phase and Quadrature (I/Q) samples collected from over-the-air real-world 5G signals across multiple locations.
2) Unlike prior works which deal with uniform distribution and balanced datasets, we adopt CWGAN-GP to augment limited SSB observations focusing on non-IID datasets to mitigate the concern of class imbalance and ensuring more representative training distribution.
3) The augmented datasets are further trained with CAE, which jointly executes both reconstruction and classification-based jamming detection, improving detection ability while addressing data heterogeneity across femtocells.

Our work advances the existing state-of-the-art methods by adopting proposed framework and assessing the performance over benchmark models [11], [12] in identifying jammed signals while training on both non-jammed and jammed signals of a time domain dataset. The organization of this paper is as follows. Section II elaborates on CWGAN-GP data augmentation technique for jamming detection. Section III discusses about the system model adopted for jamming detection. Section IV presents the experimental setup with simulation results in Section V, and Section VI summarizes the work in this article.

II. CWGAN-GP Augmented-based jamming detection

The objective of this work is to define an augmented ML-based approach which takes into account the dataset heterogeneity for each dataset collected at different geographical locations. This heterogeneity is identified by presence of non-IID data representing the attribute skewness, difference in quantity of SSB observations (training samples) across datasets, and imbalanced class distribution of jammed and non-jammed signals. The proposed framework deals with the stages of data collection and preprocessing to simulate a jammed 5G RF environment.

A. Data collection

Data is obtained with the help of spectrum analyzer which collects received signal waveform over-the-air, shared between telecommunication operators: Telus Communication Inc. and Rogers Communication Inc. Additionally, these received waveforms are acquired by setting a specific center carrier frequency and bandwidth over the available transmission cellular networks advocating various 5G-NR bands and bandwidths, respectively.

B. Data preprocessing

The collected received signal is transformed into spectrogram which coherently reflects the useful information of channel resource blocks. Only specific SSBs from resource blocks is extracted in the form of complex I/Q samples. Given $\mathcal{N}$ different geographical locations, $\mathcal{N}$ I/Q datasets are generated, each containing diverse training SSB observations. We assume the absolute values for I/Q samples which is effective for power-based jamming detection, where phase of the signal is ignored in the computation. Moreover, these absolute values are normalized across all datasets keeping a high-dimensional feature space. Furthermore, the incorporation of AWGN as jammed signal is simulated by varying the signal-to-noise (SNR) ratio to a suitable range for all the datasets. This provides information on the training SSBs with imbalanced class distribution of non-jammed and jammed signals across all datasets. Our proposed framework is not limited to AWGN but can also be leveraged for other types of jamming signals.

C. Data augmentation to tackle the class imbalance

To tackle the data augmentation technique, a CWGAN-GP is chosen to generate more SSB observations as an oversampling approach. However, the oversampling is employed on both minority (non-jammed) and majority (jammed) signals to obtain a balanced binary classification problem. Additionally, augmentation facilitates CAE from becoming biased towards one class of signals. GAN consists of two neural networks (generator and discriminator) as proposed by Goodfellow et al. [16]. The generator aims to leverage a Gaussian noise to obtain synthetic observations which resemble to the real data distribution. The objective function of a GAN follows a min-max game as formulated as,

$$\min_G \max_D V(D,\ G) = \mathbb{E}_{x\sim p_{data}(x)}(log(D(x)) + \mathbb{E}_{z\sim p_z(z)}(log(1-D(G(z))) \tag{1}$$

The generator L_G and the discriminator L_D losses are represented as follows:

$$L_G = -\mathbb{E}_{z\sim p_z(z)}(D(G(z)))) \tag{2}$$

$$L_D = -\Big[\mathbb{E}_{x\sim p_{data}(x)}\Big(logD(x)\Big) + \mathbb{E}_{z\sim p_z(z)}\Big(log(1-D(G(z))\Big)\Big] \tag{3}$$

where $p_{data}(x)$ denotes the real data distribution; $p_z(z)$ represents Gaussian distribution noise z; $G(\cdot)$ represents the generator function; E$(\cdot)$ represents the expected function; $D(\cdot)$ represents the discriminator function. The computation of L_D takes into account both real and generated data while distinguishing between them as in (3). WGAN and WGAN-GP leverage a metric *Earth-Mover* (EM) distance as the measure of the distance between real data distribution and generated data distribution which is better than *Jensen-Shanon* (JS) divergence followed in conventional GANs. WGAN is highly effective in circumventing the issue of mode collapse. The EM distance is expressed as,

$$W(\mathbb{P}_r,\mathbb{P}_g) = \inf_{\gamma\in\Pi(\mathbb{P}_r,\mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x-y\|\right] \tag{4}$$

where $\Pi(\mathbb{P}_r,\mathbb{P}_g)$ denotes the entire joint probability distribution $\gamma(x,y)$ of real distribution $\mathbb{P}_r$, and generated data distribution $\mathbb{P}_g$. Moreover, $W(\mathbb{P}_r,\mathbb{P}_g)$ depicts the minimum cost required to transfer the mass while converting the distribution $\mathbb{P}_r$ into $\mathbb{P}_g$. Furthermore, EM distance is relatively useful in obtaining meaningful gradients for gradient descent training. The objective function between the generator (G) and the critic (C) (known as the discriminator) for WGAN is defined as,

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_r}[\log D(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_g}[\log(1-D(\mathbf{x}))] \tag{5}$$

On the contrary, WGAN still fails to converge due to the weight clipping factor in WGAN. Therefore, Gulrajani [17] introduces WGAN-GP, an extension of WGAN which penalizes the norm of the gradient of the critic concerning its input. This enables WGAN-GP to be appropriate for stable training with almost no hyperparameter tuning. The modified objective function of WGAN-GP is defined as,

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_r}[D(\mathbf{x})] - \mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_g}[D(\hat{\mathbf{x}})] - \lambda\mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}\left[(\|\nabla_{\hat{\mathbf{x}}}D(\hat{\mathbf{x}})\|_2-1)^2\right] \tag{6}$$

where λ is the gradient penalty coefficient $\hat{\mathbf{x}}$ is the sampling distributions between real distribution $\mathbb{P}_r$ and generated distribution $\mathbb{P}_g$ shown in (7):

$$\hat{\mathbf{x}} = \epsilon\mathbf{x} + (1-\epsilon)\tilde{\mathbf{x}}, \quad \epsilon\sim \text{Uniform}[0,1], \quad \mathbf{x}\sim\mathbb{P}_r, \quad \tilde{\mathbf{x}}\sim\mathbb{P}_g \tag{7}$$

On the contrary, CWGAN-GP ensures auxiliary conditioned information $\mathbf{y}$; class label to both the critic and the generator.

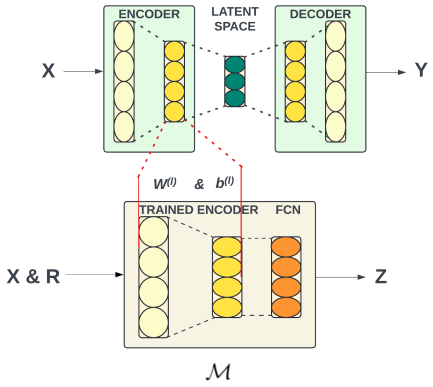

Fig. 2. Architecture of convolutional autoencoder (CAE).

Formally, the objective value function that minimizes the loss function for the critic and the generator is expressed in (8), (9) and (10).

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_r}[D(\mathbf{x}|\mathbf{y})] - \mathbb{E}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[D(\tilde{\mathbf{x}}|\mathbf{y})] - \lambda\mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}\left[(\|\nabla_{\hat{\mathbf{x}}}D(\hat{\mathbf{x}}|\mathbf{y})\|_2-1)^2\right] \tag{8}$$

$$L(D) = -\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_r}[D(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[D(\tilde{\mathbf{x}}|\mathbf{y})] + \lambda\mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}\left[(\|\nabla_{\hat{\mathbf{x}}}D(\hat{\mathbf{x}}|\mathbf{y})\|_2-1)^2\right] \tag{9}$$

$$L(G) = -\mathbb{E}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[D(\tilde{\mathbf{x}}|\mathbf{y})] \tag{10}$$

III. Jamming Detection with Convolutional Autoencoder

The CAE is employed for one-class classification or jamming detection. The 2D temporal correlation in the augmented dataset is useful for undergoing a convolutional operation of the high-dimensional I/Q samples. Moreover, unlike other autoencoders where CAE is usually trained as a reconstruction, we intend to use CAE as both a reconstructor and a classifier. As illustrated in Fig. 2, CAE takes input array $\mathbf{X}$ of dimension $\mathbf{P}$ by $\mathbf{Q}$; where $\mathbf{P}$ being SSB observations and $\mathbf{Q}$ is high-dimensional I/Q samples. The output for CAE is $\mathbf{Y}$, which is the same size as $\mathbf{X}$ due to the reconstruction characteristic of the model. The CAE comprises $\boldsymbol{L}$ layers ℓ = 1,..., $\boldsymbol{L}$. The output of the final layer of encoder is obtained as (11). The decoder comprises transpose Conv1D layers, which form the reconstructed input from the encoded representation through compressed latent space. The output of the final layer of decoder is obtained as (12).

$$\mathbf{U}^{(\ell)} = f\left(\mathbf{C}^{(\ell)} * \mathbf{U}^{(\ell-1)} + \mathbf{b}^{(\ell)}\right) \tag{11}$$

$$\mathbf{V}^{(\ell)} = f\left(\mathbf{D}^{(\ell)} * \mathbf{V}^{(\ell-1)} + \mathbf{d}^{(\ell)}\right) \tag{12}$$

where $\mathbf{U}^{(\ell)}$ and $\mathbf{V}^{(\ell)}$ are the outputs of the ℓ^{th} layer of encoder and decoder respectively, $f(\cdot)$ is the non-linear activation function, typically ReLU in this case. $\mathbf{C}^{(\ell)}$ and $\mathbf{D}^{(\ell)}$ are the convolutional weights at layer ℓ, convolutional operation $*$ with $\mathbf{U}^{(\ell-1)}$ and $\mathbf{V}^{(\ell-1)}$, and $\mathbf{b}^{(\ell)}$, $\mathbf{d}^{(\ell)}$ as bias

at layer ℓ. The input of the first layer is $\mathbf{X} \in \mathbb{R}^{\mathbf{P}\times\mathbf{Q}}$,and the output of the last layer $\boldsymbol{L}$ is $\mathbf{Y} = \mathbf{V}^{(\boldsymbol{L})}$.

To implement jamming detection, our CAE is trained by compressing the input $\mathbf{X}$, representing the I/Q features of both jammed and non-jammed signals, using latent representation. The goal is to train the model in unsupervised learning to minimize the mean-square error (MSE) between $\mathbf{X}$ and $\mathbf{Y}$ as obtained in (13). However, the reconstructed weights $\boldsymbol{W}_{\mathrm{e}}^{(\ell)}$ and biases $\boldsymbol{b}_{\mathrm{e}}^{(\ell)}$ from the trained encoder of CAE are captured from the ℓ^{th} layer of encoder. These weights and biases are transferred to the fully connected neural network (FCN); transforming the CAE to act as a classifier by combining trained encoder and FCN (added to the head of the encoder) into a new updated model $\mathcal{M}$ as shown in (14) and (15) respectively.

$$\bar{\Gamma} = \mathbb{E}[\Gamma], \quad \Gamma = \|\mathbf{X} - \mathbf{Y}\|^2. \tag{13}$$

$$\boldsymbol{W}_{\mathcal{M}}^{(\ell)} = \boldsymbol{W}_{\mathrm{e}}^{(\ell)}, \quad \forall \ell \in \{1, 2, \ldots, L\}. \tag{14}$$

$$\boldsymbol{b}_{\mathcal{M}}^{(\ell)} = \boldsymbol{b}_{\mathrm{e}}^{(\ell)}, \quad \forall \ell \in \{1, 2, \ldots, L\}. \tag{15}$$

The detection ability of $\mathcal{M}$ is ensured by taking input $\mathbf{X}$ and ground truth $\mathbf{R}$, train it over 80% train data and evaluate on 20% test data with a suitable threshold γ.

IV. Experimental Setup

An experimental setup is implemented within the 5G n71 band. As Per 3GPP specifications, this band spans a downlink frequency range from 617 MHz to 652 MHz, offering a total bandwidth of 35 MHz [18]. The frequency range is divided between two operators, TELUS and Rogers, each allocated 10 MHz of bandwidth. TELUS operates with a center frequency of 632 MHz, while Rogers operates at 622 MHz. The setup, depicted in Fig. 3, features a ThinkRF RTSA R5500 spectrum analyzer serving as the receiver with two different antennas to capture Over-The-Air (OTA) 5G signal from TELUS network.

Sampling occurs at a frequency of 15.36 MHz across various environments, including indoor locations and outdoor scenarios (encompassing both Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS) conditions). The gathered samples are saved in CSV format using the PyRF4 API and are subsequently processed. To obtain accurate information from the SSB, it is essential to estimate both the time offset (TO) and carrier frequency offset (CFO). Since the exact center frequency is unknown, a blind search approach is required. To precisely determine the TO and CFO, we leverage the PSS correlation properties and the cyclic prefix from the Cyclic Prefix Orthogonal Frequency Division Multiplexing (CP-OFDM) 5G waveform to align with the gNB signal.

The optimization problem for estimating the CFO is expressed as,

$$\hat{\omega}_{CFO} = \arg\max_{\omega_i} \left[\sum_{\tau} y(\tau) e^{j\frac{\omega_i}{f_s}\tau} x_{pss}(t-\tau) \right], \tag{16}$$

where x_{pss} is the primary synchronization signal, the first OFDM symbol in SSB and f_s is the sampling frequency. For obtaining time offset to the SSB, Schmidl & Cox approach [19] is used. Hence, the following optimization problem (17) is solved numerically where $\mathcal{P}(t)$ and $\mathcal{R}(t)$ are represented as (18) and (19), where $\hat{L}$ is one-half of the number of samples in one OFDM symbol.

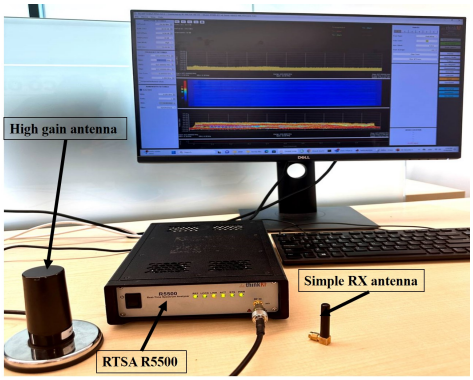

Fig. 3. Experimental set-up for jamming detection.

TABLE I
Information on Datasets

Dataset ID	Location & propagation conditions	SSB observation count	Class Imbalance
1	Banchory (Outdoor, NLOS, LOS)	826	(1) : 793 (0) : 33
2	Legget (Outdoor, LOS)	544	(1) : 518 (0) : 26
3	Indoor_2 (Indoor, LOS)	971	(1) : 933 (0) : 32
4	Indoor_3 (Indoor, NLOS)	1038	(1) : 998 (0) : 40
5	Indoor_4 (Indoor, NLOS)	877	(1) : 839 (0) : 38
6	Indoor_5 (Indoor, NLOS)	989	(1) : 945 (0) : 44
7	Neighbor_2 (Outdoor, LOS, NLOS)	805	(1) : 771 (0) : 34
8	Neighbor_3 (Outdoor, NLOS)	923	(1) : 886 (0) : 37
9	Neighbor_1 (Outdoor, LOS)	749	(1) : 719 (0) : 30
10	Park Shirley (Outdoor, LOS, NLOS)	833	(1) : 799 (0) : 34
11	Shirin Market (Outdoor, LOS)	664	(1) : 638 (0) : 27
12	Stop Sign (Outdoor, LOS)	978	(1) : 937 (0) : 41

$$\hat{T}_{off} = \arg\max_{t} M(t) = \frac{|\mathcal{P}(t)|^2}{\mathcal{R}(t)^2}, \tag{17}$$

$$\mathcal{P}(t) = \sum_{n=0}^{\hat{L}-1} y^*(t+n) y(t+n+\hat{L}), \tag{18}$$

$$\mathcal{R}(t) = \sum_{n=0}^{\hat{L}-1} |y(t+n+\hat{L})|^2 \tag{19}$$

V. Experimental Results

The simulation is performed on 12 heterogeneous datasets, each comprising fewer SSB observations with a significant class imbalance of jammed (1) and non-jammed (0) signals. The information on each dataset is summarized in Table I.

A. Data augmentation using CWGAN-GP

We adopt CWGAN-GP, which handles heterogeneity on each dataset by augmenting the number of observations to a fixed amount, for instance, 5000 observations; enforcing 2500 jammed and 2500 non-jammed signals. The entire class imbalance for each dataset is assumed to be the training set prior to oversampling using CWGAN-GP. The architecture of CWGAN-GP follows a five-layer Conv1D neural networks for C and two Conv1D neural networks for G. CWGAN-GP is trained over a few epochs with a fixed batch size [20] to

TABLE II
CWGAN-GP Parameters and Hyperparameters

Parameter/Hyperparameter	Value/Details
Model Architecture	C: 32-512 units, G: 128-64 units
Latent Vector Dimension	128
Dropout	C: 0.5, G: None
Batch Normalization	C: None, G: Yes
Activation Functions	C and G hidden: LeakyReLU, G output: tanH
Batch Size	64
Training Epochs	20
Optimizer	α: 0.0001, β_1: 0.5, β_2: 0.9
Gradient Penalty Coefficient	20
Critic Training	7

TABLE III
PARAMETERS AND HYPERPARAMETERS OF AUTOENCODERS

Parameter/Hyperparameter	Value/Details		
	CAE	CDAE	CSAE
Number of Layers (Encoder)	3	3	3
Number of Layers (Decoder)	3	3	3
Sparsity probability	-	-	0.05
Sparsity factor	-	-	0.01
Noise factor	-	0.3	-
Activation	ReLU	ReLU	ReLU
Dropout	0.2	0.2	0.2
Batch size	200	200	200
Learning rate	0.0001	0.0001	0.0001
Epochs	30 (Autoencoder & Classifier)	15 (Autoencoder), 30 (Classifier)	15 (Autoencoder), 30 (Classifier)
Optimizer	Adam (Autoencoder & Classifier)	Adagrad (Autoencoder), Adam (Classifier)	SGD (Autoencoder), Adam (Classifier)
Loss function	MSE and BCE	MSE and BCE	MSE and BCE

TABLE IV
JAMMING DETECTION OUTCOME COMPARISON ON 80:20 TRAINING SET/TESTING SET

Dataset ID	CAE					CDAE					CSAE				
	Precision	Recall	F1-Score	FAR	MDR	Precision	Recall	F1-Score	FAR	MDR	Precision	Recall	F1-Score	FAR	MDR
1	100	82	90	0	17.8	83	98	90	19.9	2	97	95	96	2.7	5
2	97	92	95	2.5	8	64	88	74	47.6	12	88	98	93	12.5	2
3	97	81	88	2.7	19	85	96	90	15.5	4	93	92	92	7	8
4	97	95	96	3.1	5	91	97	94	10.6	3	93	89	91	7.2	11
5	100	99	99	0.4	1	84	98	91	18	2	94	97	96	6	3
6	92	95	94	8.1	5	98	82	90	1.8	18	87	88	87	14.1	12
7	100	99	99	0.4	1	94	90	92	6.2	10	98	98	98	2	2
8	99	92	95	1.1	8	97	84	90	2.7	16	90	94	92	9.9	6
9	92	68	78	6.4	32	97	95	96	2.6	5	95	97	96	5.1	3
10	98	97	98	1.6	3	99	86	92	1	14	51	65	57	68.6	35
11	100	99	100	0.1	1	92	91	91	7.6	9	98	95	96	1.9	5
12	96	97	97	4.3	3	92	96	94	9.5	4	95	93	94	5.21	7

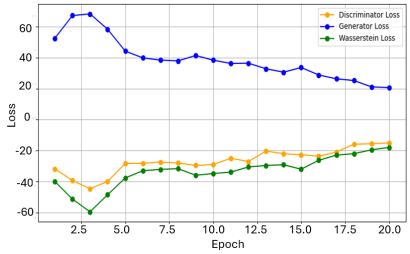

Fig. 4. Training Loss Curves in CWGAN-GP.

generate 250 generated observations i.e. 5000 observations; which comprise 2500: jammed and 2500: non-jammed SSB observations. We choose the default values for optimizer Adam, set gradient penalty coefficient, and train critic a few times unlike the default values used in [17]. Table II presents the details on parameters and hyperparameters for CWGAN-GP. The CWGAN-GP model shows convergence over the training epochs (see Fig. 4), depicting critic loss stabilizes along with Wasserstein loss. However, generator loss spikes during the early stages of training, highlighting that the generated samples are far from real samples, and gradually stabilizes over time to generate more realistic samples.

B. Training with CAE, CDAE and CSAE

CAE is trained on each dataset ID to showcase detection performance in terms of classification metric precision, recall, F1-score, and accuracy of the model. However, jamming detection requires other metrics, for instance, False Alarm Rate (FAR) and Missed Detection Rate (MDR) to comprehend the real-world deployment effectiveness. FAR and MDR metrics are critical for measuring false alarms and potential indications of compromising network security. Moreover, CAE is first trained in an unsupervised learning algorithm while assuming 8:2 as training and validation sets. During the first training process, CAE captures the weights and biases of the trained encoder and is transferred to the fully connected layer; acting as a classifier, and subsequently trained in a supervised learning manner. The parameters and hyperparameters for the CAE model are highlighted in Table III. The jamming detection performance of the classifier using the trained weights showcases promising accuracy, precision, recall, and f1-score obtained for each dataset ID while considering $\gamma = 0.5$. However, Dataset ID 9 achieves lower recall and F1-score of 68% and 78% respectively as compared to other datasets. This signifies that a larger proportion of true jammed signals are incorrectly detected as false negatives or non-jammed signals. In addition, the missed detection rate is 0.32 which depicts that 32% of jammed signals are identified as non-jammed signals. Moreover, the false alarm rate is 0.064 or 6.4% of true non-jammed signals are incorrectly identified as jammed signals.

On the contrary, CDAE [11] and CSAE [12] are trained unsupervised and compute reconstruction errors between the input samples and the decoded output. Only the reconstruction errors are used at the input to the trained encoder and fully connected layer to obtain the classification performance with the same threshold unlike the similar training followed for CAE. However, the weights/biases are captured by CDAE and CSAE and forwarded to FCN similar to CAE. The detection ability of CDAE shown in Table IV highlights promising performance across all datasets but Dataset ID 2; achieving a precision, recall, and F1-score of 64%, 88%, and 74%, respectively. The low value of precision depicts the presence of high false positives. The lower false negative provides a direct hint of obtaining a higher recall. In addition, the missed detection rate for Dataset ID 2 shows that 12% of jammed signals are identified as non-jammed signals and the false alarm rate of 47.6% of non-jammed signals are incorrectly identified as jammed signals; causing more false positives. On the contrary, CSAE performs satisfactorily well across all the datasets but Dataset ID 10 with precision, recall, F1-score, and accuracy are shown in Table IV. The poor detection performance coherently indicates high false negatives and high false positives responsible for acquiring low precision and recall, respectively. In terms of missed detection rate and false alarm rate, 35% of the jammed signals are distinguished as non-jammed signals, and 68.6% of the non-jammed signals are mistaken as jammed signals. The performance differences across all datasets are evident due to varying propagation and

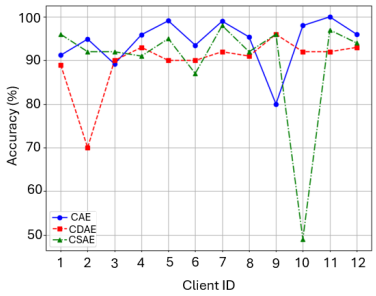

Fig. 5. Accuracy comparison of each dataset.

TABLE V
AVERAGE CLASSIFICATION PERFORMANCE METRICS OF MODELS

Models	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
CAE	97.33	91.33	94.08	94.35
CDAE	89.67	91.75	90.33	89.93
CSAE	89.92	91.75	90.67	89.92

channel conditions of jamming power at different locations. In addition, the accuracy comparison for the models across all the datasets highlights CAE outperforms CDAE and CSAE shown in Fig. 5. Moreover, a comparison showcases notable performance differences by assuming the proposed CAE over the other benchmark models: CDAE and CSAE. The average of precision, F1-score, and accuracy highlight that the proposed CAE model outperforms the benchmark models with a significant difference shown in Table V.

VI. Conclusion and Future Work

We have proposed an augmented-based jamming detection against 5G-NR networks while assuming various factors: data heterogeneity across multiple femtocells, limited SSB observations, and the presence of class imbalance across all datasets. Our approach employs the exploitation of CWGAN-GP to generate more synthetic SSB observations and obtain balanced datasets; comprising an equal amount of jammed and non-jammed signals. To ensure high classification performance and detection of jammed attacks, we employ CAE and train the model in both unsupervised and supervised learning on IQ signals of a 5G-NR cellular network. The results depict that the detection ability of CAE outperforms other benchmark models: CDAE and CSAE in terms of metrics: precisions, acceptable recall, F1-score, and accuracy. However, a detailed comparison of CAE model over benchmark models across all datasets showcases that the proposed approach performs better by achieving an accuracy of at least 90% without the involvement of reconstruction errors in the training process unlike CDAE and CSAE. The detection performance of CAE relies on the quality of augmented samples of CWGAN-GP, which might impact the performance if there is frequent fluctuation of generator loss without converging over time. Our ongoing work aims to address computational complexity and optimization strategies to improve scalability by assuming more femtocells in a 5G-NR network.

Acknowledgment

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under the Discovery and CREATE TRAVERSAL Programs.

References

[1] E. Björnson and L. Sanguinetti, "Scalable cell-free massive mimo systems," *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4247–4261, 2020.

[2] X. Shen, Y. Liu, L. Zhao, G.-L. Huang, X. Shi, and Q. Huang, "A miniaturized microstrip antenna array at 5g millimeter-wave band," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 8, pp. 1671–1675, 2019.

[3] A. Goyal and K. Kumar, "Lte-advanced carrier aggregation for enhancement of bandwidth," in *Advances in VLSI, Communication, and Signal Processing: Select Proceedings of VCAS 2018*. Springer, 2020, pp. 341–351.

[4] P. Yu, F. Zhou, X. Zhang, X. Qiu, M. Kadoch, and M. Cheriet, "Deep learning-based resource allocation for 5g broadband tv service," *IEEE Transactions on Broadcasting*, vol. 66, no. 4, pp. 800–813, 2020.

[5] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A tutorial on beam management for 3gpp nr at mmwave frequencies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 173–196, 2018.

[6] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Trans on neural networks and learning systems*, vol. 31/9, pp. 3400–3413, 2019.

[7] P. Lohan, B. Kantarci, M. Amine Ferrag, N. Tihanyi, and Y. Shi, "From 5g to 6g networks: A survey on ai-based jamming and interference detection and mitigation," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 3920–3974, 2024.

[8] M. Hachimi, G. Kaddoum, G. Gagnon, and P. Illy, "Multi-stage jamming attacks detection using deep learning combined with kernelized support vector machine in 5g cloud radio access networks," in *Intl. Symp. on networks, computers and communications*. IEEE, 2020, pp. 1–5.

[9] M. Varotto, S. Valentin, and S. Tomasin, "Detecting 5g signal jammers with autoencoders based on loose observations," in *2023 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2023, pp. 160–165.

[10] G. Asemian, M. Amini, B. Kantarci, and M. Erol-Kantarci, "Dt-ddnn: A physical layer security attack detector in 5g rf domain for cavs," *ArXiv*, vol. abs/2403.02645, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:268249163

[11] E. Almazrouei, G. Gianini, C. Mio, N. Almoosa, and E. Damiani, "Using autoencoders for radio signal denoising," in *Proceedings of the 15th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, 2019, pp. 11–17.

[12] W. Luo, J. Li, J. Yang, W. Xu, and J. Zhang, "Convolutional sparse autoencoders for image classification," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 7, pp. 3289–3294, 2017.

[13] R. Chapaneri and S. Shah, "Enhanced detection of imbalanced malicious network traffic with regularized generative adversarial networks," *J. of Network and Computer Applications*, vol. 202, p. 103368, 2022.

[14] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[15] Y. Chen, Z. Zhao, J. Liu, S. Tan, and C. Liu, "Application of generative ai-based data augmentation technique in transformer winding deformation fault diagnosis," *Engineering Failure Analysis*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267641919

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[18] 3GPP, "5G; NR; Base Station (BS) radio transmission and reception," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.104, 02 2024, version 17.12.0.

[19] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for ofdm," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.

[20] S. McKeever and M. S. Walia, "Synthesising tabular datasets using wasserstein conditional gans with gradient penalty (wcgan-gp)," *Technological University Dublin*, 2020.