Classification with Synthetic Radio Data for Real-life Environment Sensing

Soumeya Kaada, Sid Ali Hamideche, Chloe Daems, Marie Line Alberi Morel

Classification with Synthetic Radio Data for Real-life Environment Sensing

1st Soumeya KAADA
Nokia Paris Saclay
Massy, France
soumeya.kaada@nokia.com
@etudiant.univ-rennes1.com

2nd Sid Ali Hamideche
Nokia Paris Saclay
Massy, France
sid.hamideche@nokia.com
@etudiant.univ-rennes1.com

3rd Chloe Daems
Nokia Paris Saclay
Massy, France
chloe.daems@nokia.com
chloe.daems@student-cs.fr

4th Marie Line Alberi Morel
Nokia Paris Saclay
Massy, France
marie_line.alberi-morel
@nokia-bell-labs.com

***Abstract*—In sensing-enabled mobile infrastructure, the network itself acts as a whole sensor by leveraging radio data or signals collected within Base Stations (BSs). This data is exploited for the development of data-driven machine learning solutions to augment network's capabilities. Nevertheless, large-scale qualitative data is required for achieving high accuracy learning. However, their training phase leads to prohibitive cost and heavy constraints on data collection and storage that are not desirable for network. To overcome this problem, we propose to use synthetic data instead of real data for training machine learning models to avoid high cost data sharing/storage. In this paper, we are interested in real-life *Environment Sensing Network* in a context of limited data amount sharing. We focus on Indoor-Outdoor Detection (IOD) using unsupervised machine learning classification models. For this purpose, experiments are conducted following the paradigm of Training on Synthetic data and Testing on Real Data (TSTR). We conduct a comparative study of four well-known generative models, that are able to generate synthetic 3GPP radio data with similar distribution than the source data. We investigate the quality of these synthetic generated radio data according to three dimensions: distribution similarity, data variability and detection capability. The classification models trained with synthetic generated data are tested in real-life context to infer whether a user connected to the network is inside or outside a building. The study shows convincing results with an Indoor/Outdoor unsupervised classification performance up to 80% of $F1-score$ like in real-life data training scenarios.**

***Index Terms*—Environment sensing, Indoor/Outdoor detection, Generative models, Unsupervised classification models, Data augmentation**

I. Introduction

With 5G-advanced network and beyond, Machine Learning (ML) and Deep Learning (DL) are expected to be introduced in many parts of the network. They show a unique ability to achieve better performances in operations such as *Environment Sensing* in Radio Access Network (RAN). The radio signals or data gathered by RAN become then a source of situational information related to user's location (e.g. indoor or outdoor location) and the network infrastructure acts itself then as a sensor [1]. This new sensing capability inside the network can augment the efficiency of wireless network operations, in terms of better QoE for video applications [2] or accurate user localisation detection [3], and slice selection to switch from a slice with more flexible resources to a resilient one [4]. In this context, the network will become a center of collection and processing of situational radio data. The radio signals or data are gathered by devices and Base Stations (BSs) and processed inside the network. However, due to the ever-growing proliferation of mobile communicating devices connected to next generation networks, data collection can become an extremely expensive process when an accurate view of what is happening in the field covered by BSs is desired. Furthermore, high learning accuracy of radio data-driven ML solutions requires large and qualitative training datasets. Specifically, in the case of *Environment Sensing*, gathering a large number of various situations is required to get an accurate estimation of environmental situations. So, the challenge for mobile infrastructure with constrained bandwidth, computational costs and storage ability is to accommodate such data volumes without significantly increasing capital and operational expenses (CAPEX/OPEX).

In this work, we are interested in answering the following research question: is it possible to do accurate *Environment Sensing* in RAN based on ML/DL algorithms in a context of lack/absence of data sharing? In the literature, solutions referred to as *Data Augmentation* techniques have been investigated to expand limited datasets [5]. The generative model is a well-known method of data augmentation that has the advantage of directly learning the statistical characteristics of source data in an unsupervised way. It ensures the generation of additional data to take advantage of big data capabilities and thus, to improve the accuracy of machine learning models. [6, 7] have recently demonstrated the effectiveness of Generative Adversarial Networks (GANs) and Variationnal Auto-Encoders (VAEs) in overcoming the problems associated with limited dataset in many domains such as image analysis, natural language processing or positioning applications. In [8] deep generative models are used for positioning applications in indoor/outdoor situations based on cell ID and Received Signal Strength (RSS) to train ML algorithms. [9] proposes an improved GAN based on Wasserstein distance and a gradient penalty to generate fake anomaly samples and improve anomaly detection accuracy. This work investigates ML models trained on both real data and synthetic data obtained from data augmentation. However, in these scenarios, data is collected in a given network node and ML training is performed in another network node. This requires transmitting source data over the network. Thus, to overcome the cost

This work is supported by French government funding within the France 2030 framework through the INFLUENCE project.

issues and heavy constraints of collecting or storing data, we are interested in exploring the possibility of training ML models on only synthetic data unlike related works. This solution would avoid carrying the original source data but rather the generative model only. To do so, synthetic data should mirror real-world data accurately. In this paper, we investigate whether a synthetic training dataset generated using well-known generative models can yield the same results for *Environment Sensing* case as using an original real dataset without compromising the sensing accuracy. We focus on Indoor-Outdoor Detection (IOD) using radio features. Large-scale experiments are done on two 3GPP radio features. We compare four well known generative models to *Produce On Demand* synthetic radio data. These are GANs, Wasserstein GANs (WGANs), VAEs and Gaussian Mixture Models (GMMs). Data quality is assessed in terms of distribution similarity and variability using density and performance based metrics. Finally, we validate the detection capability of generated or synthetic data by tracking $F1-score$ on four Indoor-Outdoor Detection (IOD) clustering algorithms. These are K-means, GMMs for clustering, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) and Deep Embedding Clustering (DEC) algorithms. Our main contributions are:

- Show that synthetic data can be used and supported to train ML based solutions for user *Environment Sensing*.
- Propose an automatic framework that learns, generates and evaluates the quality of synthetic radio data using density similarity and variability metrics. Experimentally show that they are good indicators of the expected IOD performance.
- Provide a comparative study of four distinct generative models and four unsupervised IOD models, and show the impact of limiting the training dataset size on the quality of generated data.

The rest of the paper is organized as follows: *Section II* lists the background on generative models. *Section III* details the methodology of user *Environment Sensing* based on synthetic data through IOD use case. The framework is also presented and the problem is formulated. Experiments and results are discussed in *Section IV* and a conclusion is given in *Section V*.

II. Background on Generative Models

Generative models are unsupervised learning methods for estimating density function. They received a lot of attention in the last ten years, with the creation of GANs and VAEs [10, 11]. Since then, improved versions were proposed such as WGANs or Bayesian GANs to get rid of training problems [12, 13]. Besides, lesser complex ML algorithms like GMMs succeeded in learning multi-modal distributions [14]. Unlike basic data augmentation techniques (e.g translation, rotation, flipping), generative models are known for their effectiveness in learning directly the density distribution of input data and produce unlimited amounts of samples from it.

A. Generative Adversarial Networks

GAN is based on an adversarial training of two neural networks referred as generator g and discriminator d with weights and biases parameters denoted as ω and θ respectively. The generator function *g(z,ω)* samples data from a random latent distribution *P(z)* represented by a noise input vector z, and attempts to approximate to original distribution P_{data} of input real data x. The discriminator function *d(x,θ)* estimates if a sample is coming from original dataset rather than from g by outputting a binary value, either 0 for fake or 1 for real. The loss of g and d is a minmax game:

$$\min_g \max_d \underset{x\sim P_{data}}{E} [\log d(x;\theta)] + \underset{z\sim P_g}{E} [\log (1 - d(g(z;\omega);\theta))] \tag{1}$$

where P_g is the generated data distribution. d maximizes the probability of assigning the correct label to input data while g is trained to minimize $\log(1 - d(g(z)))$.

B. Wasserstein GANs

WGAN is a derivative of original GAN that uses the Wasserstein distance metric expressed in eq. 9 ($\rho = 1$). The metric measures the distance between P_{data} and P_g. The discriminator function referred to as f is called a 'critic'. As in GAN, f and g play the following minmax game:

$$\min_g \max_{||f||_{L<1}} \underset{x\sim P_{\text{data}}}{E} [f(x;\theta)] - \underset{z\sim P_{\text{g}}}{E} [f(g(z;\theta))] \tag{2}$$

The original WGAN improves training stability and prevents mode collapse compared with GAN. It sometimes generates still poor samples or fails to converge. Hence we use WGAN-GP that adds a Gradient Penalty (GP) to the loss to meet the Lipschitz constraint ($||f||_{L<1}$) and to improve the model stability.

$$\mathcal{L} = \underset{\tilde{x}\sim P_g}{E} [f(\tilde{x})] - \underset{x\sim P_{data}}{E} [f(x)] + \lambda \underset{\hat{x}\sim P_{\hat{x}}}{E} \left[\left(\|\nabla_{\hat{x}} f(\hat{x})\|_2 - 1 \right)^2 \right] \tag{3}$$

where $\hat{x} = t\tilde{x} + (1-t)x$ with $t \in [0, 1]$.

C. Variational Auto Encoders

VAE is based on an auto encoder method that uses dimension reduction capacity to encode the input data distribution and to learn explicitly its density parameters: the mean μ and the co-variance matrix σ. Those parameters are used in the decoder as parameters of a Gaussian distribution, from which we sample the latent space z to learn the input distribution.

$$\mathcal{L} = \|x - d(z)\|^2 + D_{KL} \left[N\left(\mu_x, \sigma_x\right), N(0, I) \right] \tag{4}$$

Where $d(z)$ is the output of the decoding scheme to be minimised, and D_{KL} is the Kulback-Leibler Divergence (7) measured between the generated distribution (compressed version of the encoding scheme) N and a standard Gaussian distribution with mean 0 and variance 1.

D. Gaussian Mixture Models

GMM is an unsupervised ML algorithm that can be used for multiple purposes like clustering, classification and generation, starting with the postulate that the data distribution is composed of a combination of multiple Gaussians with unknown parameters. It is based on a weighted sum of N gaussian components densities as follows:

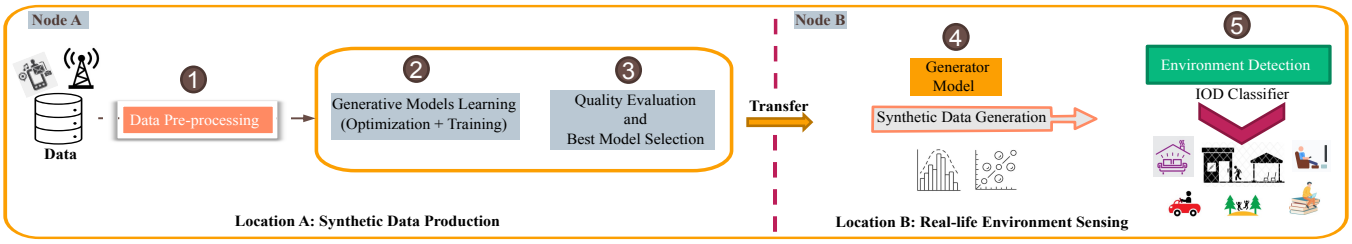


Fig. 1: Synthetic Data Generation Framework: Synthetic Data Production and Real-life *Environment Sensing*

$$P(x|\lambda) = \sum_{i=1}^{N} w_i g(x, \mu_i|\sigma_i) \tag{5}$$

where x are the observations, w_i with $i = 1, ..., N$ with the constraint that $\sum_{i=1}^{N} w_i = 1$ is the mixture weights and $\Sigma_i g(x, \mu_i)$ are the component Gaussian densities. λ are GMM parameters (w_i, μ_i, σ_i) to be estimated and that maximize the likelihood of GMM given the training data. λ are estimated using the iterative expectation-maximization algorithm.

RF Conditions		*RSRP* (dBm)	*RSRQ* (dB)
	Excellent	-44 to -80	-3 to -10
	Good	-80 to -90	-15 to -10
	Mid Cell	-100 to -90	-20 to -15
	Cell Edge	-140 to -100	<-20

(a)

(b)

Fig. 2: (a) Radio Frequency conditions - (b) Pearson correlation

III. Synthetic Data for User Environment Sensing

In this section we present our methodology of *Environment Sensing* aided with synthetic data.

A. Problem formulation and proposed framework

The environment represents one of the main contextual components of user behavior. Different user behaviors would result in a significant difference in traffic generation and service usage, which directly influence the base station performance. Thus, analyzing the user behavior is widely considered [1] to improve 5G services and network performances. One of the applications consists in detecting user environment (IOD) when he/she is connected to the network, whether the user is indoor (Buildings, Home, office, Mall...) or outdoor (Transport, Pedestrian ...).

To perform IOD using synthetic data, we propose an automatic framework depicted in Fig. 1 that (I) learns generative models and evaluates synthetic data quality to select the best generator model at a node A. Afterwards, using the transferred generative model to another node B to (II) produce synthetic data and then to perform real-life *Environment Sensing*. To investigate the accuracy of the generative process executed by the framework, we conduct a comparative study of four different generative models. For this purpose, we evaluate generated data quality in terms of distribution similarity and variability using generic metrics. Finally, generated data is fed into different classification models of IOD to validate the detection capability of synthetic data. So, the framework is executed so as to maximise the performance of the IOD classifier trained on generated data regarding the one trained on original data such that the performance is less or equal than a certain threshold ϵ. We formulate the problem as:

$$\begin{gathered} Perf(\{f_{\omega_G^*}(x_v)\}, \{y_v\}) - Perf(\{f_{\omega_x^*}(x_v)\}, \{y_v\}) \leq \epsilon \\ s.t \;\; f_{\omega_G^*} = f(x_{tG}, \omega^*) \;\; where \;\; x_{tG} \in \{g(\theta^*, z)\} \\ f_{\omega_O^*} = f(x_t, \omega^*) \;\; where \;\; x_t \sim P_{data} \\ (x_v, y_v) \in V \;\; where \;\; x_v \sim P_{data} \end{gathered} \tag{6}$$

Where $Perf(\{f_{\omega_G^*}(x_v)\}, \{y_v\})$ is the performance of the best IOD model on the validation dataset denoted as V. The set is fed with testing data from original dataset. The IOD classifier function with parameters ω_G^*, denoted as $f_{\omega_G^*}$, is trained on synthetic data x_{tG} sampled from P_g. They are generated by the best generator function $g(\theta^*, z)$ with θ^* and the latent vector z. Besides, $Perf(\{f_{\omega_O^*}(x_v)\}, \{y_v\})$ is the performance of the best IOD model on V with parameters ω_O^* trained on original data x_t sampled from P_{data}. $y_v \in \{0, 1\}$ represents the label for indoor and outdoor environments.

B. Synthetic Data Production

This section details the first part of the framework (Fig. 1). We start by describing data features used for the training, formulate the generative learning problem and define the metrics for evaluating data quality .

1) Radio Data Features

For environment detection, many physical layer measurements may be used like GPS coordinates, the cellular signals, the magnetic intensity, or measurements from inertial sensors. In this work, we focus on two 3GPP standard radio features already employed in [3], namely the signal strength - Reference Signal Received Power ($RSRP$) and the signal quality - Reference Signal Received Quality ($RSRQ$). These performance indicators are commonly used in mobile networks context for reflecting the behavior of users. The radio features are as follows:

- $RSRP$: average received power of a single Reference Signal (RS) resource element.
- $RSRQ$: ratio between $RSRP$ and the total power of received signal (RSSI).

Data cleaning and verification methods as described in [1] are applied to correct erroneous environment labels as much as possible. Thus, our whole data-set is composed of a vector of two features ($RSRP$, $RSRQ$) described in Fig. 2.

2) Generative models learning

The entry of generative models denoted as $x_t = (x_1, x_2)$ represents a pair of two samples from two radio measurements ($RSRP$, $RSRQ$). x_t is a multivariate variable which joint probability distribution is denoted as P_{data}. The training dataset is made of N training samples $\{x_{tn}\}_{1 \leq n \leq N}$. We start from the assumption that the training data samples contain a good approximation of the true distribution so that an empirical estimation can be made with no explicit representation of P_{data}. The input data is first cleaned and pre-processed, then an optimization of the model hyper-parameters is performed to effectively train afterwards the generative models. We investigate four generative algorithms that are GAN, VAE, WGAN and GMM.

3) Data quality evaluation

The generator model G has to produce radio data with the best quality. To evaluate the quality of generated data, the visual inspection of the samples is not a reliable practical guide for automated systems. Consequently, (1) density similarity and (2) variability and variance metrics suitable for evaluating quality of generated data are used by the system in the framework.

(1) Density distribution based metrics that measure the distance between original and generated distribution.

- **Jenson Shannon Divergence (JSD):** is a distance measure based on the Kullback Leibler Divergence (KLD). It measures how much information is lost when the distribution P_g is approximated to original data distribution P_{data}. KLD is written as:

$$D_{KL}(P_{data} \| P_g) = \mathbb{E}_{P_{data}} \left[log(\frac{P_{data}}{P_g}) \right] = \sum_{i=1}^{N} P_{data} \cdot \log \frac{P_{data}}{P_g} \tag{7}$$

  Unlike KLD, JSD is a symmetric metric that outputs smooth values. It is relevant when comparing two distributions that are not overlapping.

$$\text{JSD}(P_{data} \parallel P_g) = \frac{1}{2} D_{KL}(P_{data} \parallel M) + \frac{1}{2} D_{KL}(P_g \parallel M) \tag{8}$$

  where $M = \frac{1}{2}(P_{data} + P_g)$.

- **Wasserstein Distance:** is a similarity distance measure between probability distributions on a given metric space M [15]. It is also referred as Kantorovich–Rubinstein metric or earth mover's distance. It measures an optimal transport plan with least cost between two probability distributions μ and ν with a $\rho - moment$ as follows:

$$W_\rho(\mu, \nu) := \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{M \times M} d(x,y)^\rho d\gamma(x,y) \right)^{1/\rho} \tag{9}$$

  where $\Gamma(\mu, \nu)$ is the collection of all measures on M.

(2) Data variance and variability evaluation based on quality metrics that measure quality or/and quantity of data points sampled from the distribution. The data variability reflects the importance of "extreme" data or outlier data present in end tails of the distribution. To have a numerical estimation of data variability, we implement the Precision and Recall for distribution and extract the medians and standard deviations.

- **Precision and Recall for distribution Area under the curve (PRD-AUC):** gives the Area Under the Curve (AUC) of all the Precision and Recall [16]. Precision measures how much of distribution Q can be generated by a part of reference distribution P while Recall measures how much of P can be generated by a part of Q. They are expressed by:

$$Precision(P_g | P_{data}) = \frac{unique(P_{data}) \cap unique(P_g)}{unique(P_g)} \tag{10}$$

$$Recall(P_g | P_{data}) = \frac{unique(P_{data}) \cap unique(P_g)}{unique(P_{data})} \tag{11}$$

- **Medians, means and Standard Deviation (STD)**

C. Real-life Environment Sensing

Having the best generator, the second framework part of Fig. 1 consists in learning IOD model using synthetic data then applying it to detect real life user environment using real data. We formulate the unsupervised classification learning problem then, we discuss its performance evaluation. The original dataset is split into a training dataset of length p and a testing/validation dataset of length m (with $p + m = N$). The synthetic dataset serves as a training data.

IOD ML algorithm training: Given a set of training data denoted as $T = \{\mathbf{x_{t_1}}, \cdots, \mathbf{x_{t_p}}\}$ where $\mathbf{x_{t_i}} \in \mathbb{R}^l$ is the i-th data vector of length l with $i \leq p$. Our IOD model contains $l = 2$ features and its training is done over T. When the IOD model is trained with original or generated data, $\mathbf{x_{t_i}}$ is filled respectively with original or generated training samples of ($RSRP$,$RSRQ$). Since generated data is not annotated, we use unsupervised/clustering algorithms to check the detection capability of synthetic data compared with original ones. We investigate four clustering algorithms for our binary unsupervised classification indoor/outdoor that are K-means, GMM for clustering, BIRCH, and DEC.

IOD ML algorithm evaluation: Given a set of validation or testing data denoted as $V = \{[\mathbf{x_{v_1}}\ y_{v_1}], \cdots, (\mathbf{x_{v_m}}\ y_{v_m})\}$ where $\mathbf{x_{v_i}} \in \mathbb{R}^l$ is the i-th data vector of length l with $i \leq m$. $y_{vi} \in \{0, 1\}$ represents the label for indoor and outdoor environments. For the experiments, we perform a benchmark training over original dataset and a training over synthetic dataset. Then, the IOD classification performance is evaluated over V that contains labeled real data collected from real-life environment context. This allows to evaluate the detection capability of synthetic samples.

IV. Experimental results

In this section, we experimentally evaluate the effectiveness of the proposed framework to generate high quality synthetic data of $RSRP$ and $RSRQ$ for an unsupervised IOD. Starting

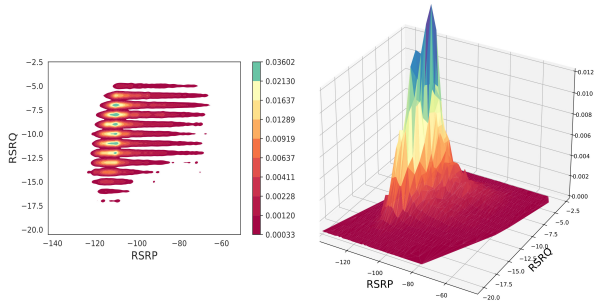


Fig. 3: Joint probability distributions of original $RSRP$ and $RSRQ$: 2D (left) and 3D (right)

from an original dataset collected in Paris region mainly, precisely in both indoor and outdoor environments, we describe the experiments setup and follow the steps of the framework depicted in Fig. 1. We collect 233K samples with various situations.

(A) Synthetic Data Production: performs data preprocessing, hyperparameter optimization and training of the four generative models. Then, the quality of synthetic data is evaluated using density similarity based metrics (JSD and Wasserstein distance) and variability based metrics (PRD, median and STD). Finally, the best generator is selected.

(B) Real-Life Environment Sensing: receives the best generator, trains the IOD clustering algorithms using either original or synthetic data. Then, their detection capability is validated based on $F1-score$ computed using the validation set V. The dataset collected is split into 80% (187K) for the training dataset and 20% (46K) for the testing/validation dataset. The original joint distribution P_{data} is shown in Fig. 3.

A. Experimental setup

We are also interested in studying the impact of reducing the training dataset size on the quality of the data produced and on the IOD classification performance. So, the training is done considering multiple sizes of training dataset, $\{100\%, 50\%, 20\%, 10\%, 5\%, 0.45\%\}$.

First, we discuss the training performance of joint synthetic probability P_g comparing to original P_{data}. An analyze of the training time is done. Then, we compare density similarity and variability metrics for the four algorithms to assess data quality. The impact of reducing the size of the training datasets on the JSD is also studied. In the second part, we evaluate the $F1-score$ of the four IOD algorithms trained on original data. We then compare them to the $F1-score$

(1) Parameter/Algorithm	**GAN**	**WGAN**	**VAE**
Number of layers	1	3	1
Optimizer	Adam	Adam	Adam
Loss	Binary cross entropy	Wasserstein distance + GP	KLD + Reconstruction loss
Epochs	10000	100	100
Batch size	128	128	128
Training time (s) on 100%	2900	600	700

(2) Algorithm /Parameter	Number of components	Covariance type	Initial params	Tolerance	Training time (s) on 100%
GMM	10	Full	K-means	1e-3	4

(3) Training dataset size	100%	50%	20%	10%	5%	0.45%

TABLE I: Training configuration of generative models

obtained when classifying synthetic data. The study goal is to investigate the reproducibility of classification using only synthetic data instead of the original data. We investigate the best generator and clustering algorithm in the context of high size $\{100\%, 50\%, 20\%\}$ or low size $\{10\%, 5\%, 0.45\%\}$ of training datasets. Optimizations and experiments, including training and evaluations, are performed on a Linux machine equipped with Intel Core i7-5930 CPU and GPU, Nvidia GTX Titan X graphic card and 64 GB of RAM.

B. Synthetic data production

(1) Data pre-processing: As a first step, we clean and rescale data using standardization because this technique provides a more stable and accurate training for our experiments.

(2) Generative models learning (hyper-parameter optimization and training): We train the four generative models to learn P_{data} using the optimal hyper-parameters given in Table I. The hyper-parameters (e.g., activation functions or learning rate) are optimized for deep generative models (GAN, VAE and WGAN). For this purpose, we use the Tree-structured Parzen Estimator (TPE) optimization algorithm which is a Bayesian optimization approach. It looks for the best choices of neural network configuration among a plurality of possibilities in order to optimize its learning performance. Fig. 4 represents the joint 2D probability distribution obtained with the four generative models. They all look like the original drawn in Fig. 3. But it is difficult to visually observe the small differences.

(3) Data quality evaluation and best model selection: During the training, we track the evolution of the density, variability and variance of the similarity metrics over 5 distinct trainings and over a significant number of iterations. For accurate analysis, the metrics are averaged every 10 iterations over 5 generated data sequences. The table II shows the minimum average values of JSD, Wasserstein distance, PRD, median and STD. These are quantified for each generative model studied. The table also details the training execution time for each case.

We observe that GMM learning takes less time to con-

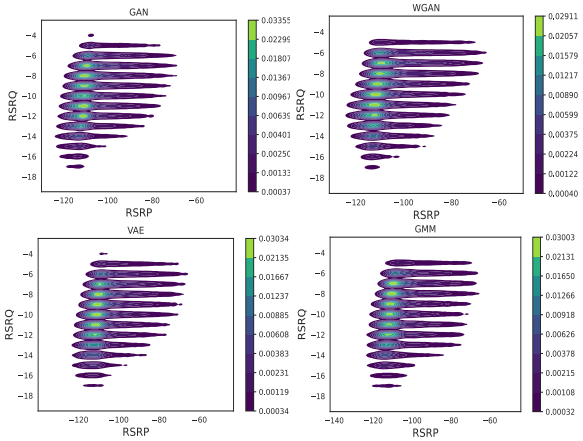


Fig. 4: Joint probability distributions of synthetic $RSRP$ and $RSRQ$ - GAN, WGAN, VAE and GMM - 2D

	JSD	Wasserstein distance	PRD	Median$\pm STD$ Orgin.: -36.5±48.88	Time (s)
WGAN	0.019	0.795	0.991	-31.79±48.92	600
GAN	0.015	0.621	0.992	-31.5±48.97	2900
VAE	0.014	0.608	0.993	-34.25±48.93	700
GMM	0.012	0.182	0.996	-32.6±48, 89	4

TABLE II: Performance comparison on density similarity, variability and variance metrics and training time

verge compared to deep generative model learning and overall achieves better results. The convergence rapidity can be explained by the fact that we are trying to generate a two-dimensional joint probability distribution that seems visually easier to model with a mixture of Gaussians. In addition, we note that the GAN training requires more epochs to converge, which was expected from the state of the art since we train the generator and the discriminator simultaneously. However, using the Wasserstein distance results in a noticeable decrease in the convergence speed of the GAN, as expected. VAE has a similar runtime as WGAN.

When the density similarity metrics show low values, they reveal high similarity between the distributions of the generated data and the original data. Therefore, they reflect the success of the generative model learning. On the other hand, when they show a high PRD, the generative model is able to better generate the "extreme" data present in the distribution tails. The generation of endline data is increased, which improves the variability of the data and thus, the range of observed data that covers more space. According to the metrics (JSD, Wasserstein distance and PRD), GMM and VAE are the best models for learning P_{data}. These good results can be explained by the small range of possible data for the radio indicators and the fact that the joint probability has a bell shape typical of Gaussians. Both models are based on the Gaussian assumption. To deeply analyze the impact of training data volume on the quality of the generated data, Fig. 5 plots the JSD metric as a function of the size of the training data set. We notice that the curve increases with the percentage of the training data size. The JSD remains roughly stable until 10% and then starts to increase significantly. This is because with generative models, a minimal volume of training data is required to accurately learn the joint probability and produce high quality data. The threshold at which the JSD degrades is between 5% and 10%. Nonetheless, in the context of a small data set size, generative models can continue to reproduce the data. The results empirically show that GMM is the best at generating high quality data in all cases according to JSD.

The following subsection compares IOD models in the case of *Environment Sensing*. To select the best generator to transfer to the remote node, we focus on using density similarity metrics. Indeed, outlier data (measured with PRD) have a very small impact on the performance of the IOD task. Specifically, we chose to focus on JSD rather than Wasserstein distance because it is less complex and faster to compute. The framework selects then the best model weights and biases that match the minimum average JSD and transfers them to node B.

C. Real-life Environment Sensing

(4 and 5) Environment detection with IOD classifier: After selecting the best model weights and biases at node A and forwarding them, the generative model is extracted at the remote node B. Synthetic data is then produced to train the unsupervised ML clustering algorithms. Then, their performance is evaluated on V for each best generative model and compared with the performance of IOD models that are trained with original data. Figures 6 show the average $F1-score$ as a function of the training dataset size for both cases of training data (original and synthetic). We notice that $F1-score$ decreases with the size of the training dataset in the majority of cases. The evolution of $F1-score$ is correlated with the evolution of JSD presented in Fig. 5. Indeed, with fewer training data samples, the joint distribution is less well learned by the generative models, which has a negative impact on the classification results. However, we observe that the GMM increases slightly to 0.45% in Fig. 6a due to overfitting.

The comparative study in fig 6a between the four IOD algorithms trained on original data reveals that GMM for clustering and K-means are the best in high dataset size context ($F1-score$ = 80%). K-means and BIRCH are the best in low dataset size context ($F1-score$ = 70%). So, the simple ML clustering techniques (K-means, BIRCH, GMM clustering) are enough accurate for IOD case. Besides, their training time is relatively small (less than a minute) with K-means, BIRCH or GMM clustering compared to DEC (10 minutes). DEC is the worst one in terms of classification and time performance because it is a deep learning technique. Note that in case of low dataset size context GMM clustering suffers from the lack of data. However, when training IOD on synthetic data only, we observe in 6b and 6c that in the majority of cases, the $F1-score$ is similar to the ones obtained with the original data. In a high size context, K-means and GMM Clustering learned on synthetic or original data give similar $F1-scores$. GMM Clustering with VAE and WGAN performs better in classification for a $F1-score$ of 81BIRCH and DEC trained on synthetic data perform less well, as mentioned above, but achieve the same performance as with the original data. In a small context, all IOD models still provide $F1-scores$ similar and low $F1-scores$ except for GMM clustering and DEC. In these cases, IOD models trained on synthetic data perform better than IOD models trained on the original data.

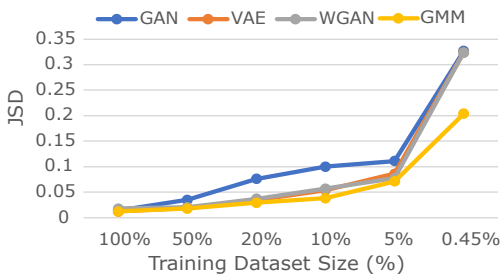
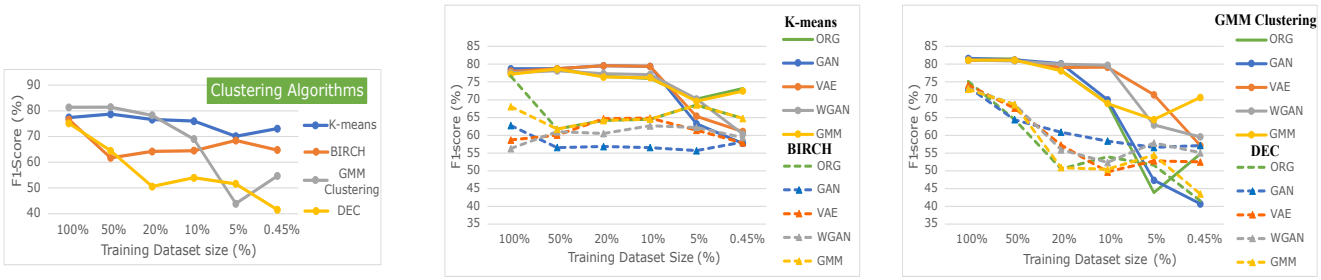


Fig. 5: JSD versus training dataset size - GAN, WGAN, VAE and GMM



(a) (b) (c)

Fig. 6: Mean $F1 - score$ versus training dataset size: (a) Clustering algorithms trained with original data - (b) K-means and BIRCH trained with synthetic data - (c) GMM Clustering and DEC trained with synthetic data

Outcome discussion: Based on the similarity of the $F1 - scores$ observed in the studied cases, we can conclude that the generated data of high quality can achieve similar performance as the original data. These results validate the good detection ability of the generated data, which can be as good as the original data. In practice, engineers can choose the most appropriate quality metric or combination of quality metrics to use in selecting and transmitting the best generator for the use cases being studied. In the context of massive data, transmitting generative models over the network instead of the source data would help overcoming storage costs or data transport issues over the network. When the cost of data transfer is not high but data transfer cannot be performed for other reasons, generative models can then overcome the challenge of data privacy or augmentation. Thus, this work presents a first application to enable environment detection based on synthetic data. Nevertheless, other challenges remain. Indeed, to ensure data saving at the lowest cost, the use of an accurate and low complexity generative model is a prerequisite. In addition, models must deliver data with labels and account for spatial and temporal dependencies of input features. Their use should be extended to other use cases and other data features with more complex probability distributions.

V. Conclusion

The main objective of this paper is to explore the possibility of using only synthetic radio data for machine learning training algorithms in a context of absence of data sharing. We propose to use generative models to provide high quality synthetic data for 3GPP radio data. We conduct a comparative study of four popular generative models on large-scale experiments with real data. Using metrics based on density similarity and variability, we highlight the importance of automatic quality assessment of synthetic data. Finally, the detection capability of synthetic data has been validated on four different IOD classification algorithms. We experimentally demonstrate that these metrics are good indicators of expected performance. We also show that generating high quality synthetic data can achieve similar performance to the original data. For the future, we propose to test these methods on 3GPP data such as Channel Quality Indicator (CQI) or Timing Advance (TA) and to investigate generative time series models.

References

[1] M. L. Alberi Morel, I. Saffar, K. Singh, S. A. Hamideche, and C. Viho. 'Improving User Environment Detection Using Context-aware Multi-Task Deep Learning in Mobile Networks'. *IEEE Transactions on Cognitive Communications and Networking*, 2022.

[2] P. Monogioudis A. Ray, S. Deb. 'Localization of LTE measurement records with missing information'. In *IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2016.

[3] S. Valentin S. Mekki, T. Karagkioules. 'HTTP adaptive streaming with indoors-outdoors detection in mobile networks'. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2017.

[4] F. Mannweiler E. Pateromichelakis, F. Moggio et al. 'End-to-End Data Analytics Framework for 5G Architecture'. *IEEE Access*, page 7, 2019.

[5] K. Maharana, S. Mondal, and B. Nemade. 'A review: Data pre-processing and data augmentation techniques'. *Global Transitions Proceedings*, Vol 3:pages 91–99, 2022.

[6] T. M. Khoshgoftaar C. Shorten. 'A survey on Image Data Augmentation for Deep Learning'. *Springer Open : Journal of Big Data*, 2019.

[7] J. Gazda T. Maksymyuk M. Ruzicka, M. Volosin. 'The extension of existing end-user mobility dataset based on generative adversarial networks'. In *International Conference on Radioelektronika*. IEEE, 2019.

[8] M. Youssef H. Rizk, A. Shokry. 'Effectiveness of Data Augmentation in Cellular-based Localization Using Deep Learning'. In *IEEE Conference on Wireless Communications and Networking*. IEEE, 2019.

[9] K. Qian X. He K. Wang H. Lu, M. Du. 'GAN-Based Data Augmentation Strategy for Sensor Anomaly Detection in Industrial Robots'. *IEEE Sensors Journal*, Vol 22, 2022.

[10] I. J. Goodfellow et al. 'Generative Adversarial Nets'. In *NIPS*. ACM, 2014.

[11] D. P. Kingma and M. Welling. 'Auto-Encoding Variational Bayes'. In *International Conference on Learning Representations (ICLR)*, 2014.

[12] M. Arjovsky V. Dumoulin A. Courville I. Gulrajani, F. Ahmed. 'Improved Training of Wasserstein GANs'. In *NIPS*, 2017.

[13] Y. Saatchi and A. G. Wilson. 'Bayesian GAN'. In *NIPS*, 2017.

[14] D. A. Reynolds. 'Gaussian Mixture Models'. *Encyclopedia of Biometrics*, page 659–663, 2009.

[15] F. L. Hitchcock. 'The Distribution of a Product from Several Sources to Numerous Localities'. *Journal of Mathematics and Physics*, Vol 20:pages 224–230, 1941.

[16] M. Lucic O. Bousquet S. Gelly M. S. M. Sajjadi, O. Bachem. 'Assessing Generative Models via Precision and Recall'. In *NIPS*. ACM, 2018.