Dynamic spectrum sharing using deep learning and generative artificial intelligence models.

Amanda Sheron Gamage

The Graduate School

Yonsei University

School of Electrical and Electronic Engineering

Dynamic spectrum sharing using deep learning and generative artificial intelligence models.

A Dissertation Submitted to the
School of Electrical and Electronic Engineering
and the Graduate School of Yonsei University
in partial fulfillment of the
requirements for the degree of
Master of Science

Amanda Sheron Gamage

June 2025

This certifies that the dissertation of Author Nameis approved.

_____

Thesis Supervisor: Prof. Kim Seong Lyun

_____

Prof.

_____

Prof. Dr. Committee03

Graduate School

Yonsei University

June 2025

ACKNOWLEDGEMENTS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Name

Department of ...

The Graduate School, Yonsei University

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Keywords: Keyword 1, Keyword 2, Keyword 3

REMARK

Material from:

Some passages of this dissertation have been reprinted from the above sources.

CHAPTER 1

INTRODUCTION AND BACKGROUND

Spectrum sharing is a critical approach in modern wireless communications that allows multiple users, systems, or services to access the same frequency bands dynamically or concurrently while minimizing interference. Traditionally, spectrum allocation followed a static model where regulatory bodies, such as the Federal Communications Commission (FCC) or International Telecommunication Union (ITU), assigned exclusive frequency bands to specific services, such as cellular networks, broadcasting, and military communications. However, with the exponential growth of wireless technologies—such as 5G, 6G, IoT, and satellite-based internet—this rigid allocation model has led to spectrum underutilization in some bands and congestion in others. Spectrum sharing mitigates this inefficiency by enabling more flexible and intelligent use of the available spectrum.

There are various models of spectrum sharing, including Licensed Shared Access (LSA), Dynamic Spectrum Access (DSA), and unlicensed spectrum sharing. LSA allows secondary users to access licensed spectrum under strict regulatory agreements, while DSA employs cognitive radio technology to detect and utilize underutilized spectrum opportunistically. In unlicensed sharing, multiple users operate in shared bands (e.g., Wi-Fi in the 2.4 GHz and 5 GHz bands) with interference management protocols. Techniques such as spec-

trum sensing, cooperative communication, and machine learning-based allocation further enhance the effectiveness of spectrum sharing by allowing real-time adaptation to spectrum availability.

The need for spectrum sharing is driven by increasing spectrum scarcity, growing wireless traffic, and the demand for more efficient spectrum utilization. It enables cohabitation between different services, such as military and commercial applications, or between terrestrial and satellite networks, improving overall network capacity and reliability. Additionally, spectrum sharing reduces the need for costly spectrum auctions and facilitates innovation by allowing emerging technologies to access frequency bands that would otherwise remain unused. As wireless ecosystems evolve, spectrum sharing will play a pivotal role in ensuring sustainable and scalable connectivity while balancing regulatory, technical, and economic considerations.

Generative AI models have revolutionized machine learning by enabling the synthesis of high-quality data across various domains, including image generation, speech synthesis, and text generation. Unlike traditional discriminative models that learn decision boundaries, generative models aim to approximate complex data distributions, allowing them to generate realistic samples from learned representations. CNNs have been the backbone of deep learning for visual data processing, excelling in tasks such as image classification, object detection, and segmentation. By leveraging hierarchical feature extraction through convolutional layers, CNNs capture spatial patterns, making them highly effective for structured data like images.

1.1 Citizens Broadband Radio Service (CBRS)

CBRS in the United States refers to the unlicensed 150MHz spectrum in the 3.5 GHz band. CBRS allows dynamic and efficient allocation of spectrum resources, fostering both commercial and public interest applications. CBRS operates under a three-tiered hierarchy designed to prioritize and protect high-priority users while ensuring equitable access for lower-priority users: Incumbent, Priority Access License (PAL), and General Authorized Access (GAA).

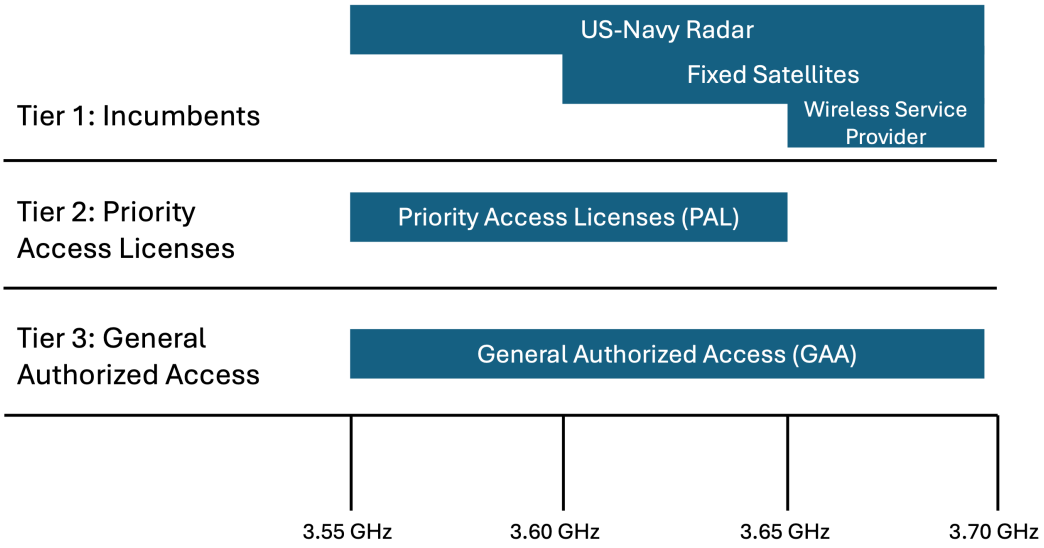


Figure 1.1: The CBRS spectrum

Incumbent users include federal and military radar systems and fixed satellite stations. These users have the highest priority and are granted uninterrupted access to the spectrum. Typically, these systems operate along coastal regions and require significant protection from interference.

PAL Users are licensed commercial users who acquire spectrum rights

through an auction process. PAL licenses are granted for up to 10 MHz in a given geographic area and provide a level of protection from interference by lower-tier users. Common applications include private LTE/5G networks for enterprises, utilities, and healthcare.

GAA users are the unlicensed users who can access the spectrum opportunistically in areas where it is not in use by Incumbent or PAL users. GAA users operate under strict rules to prevent interference with the higher-priority tiers. Figure 1.1 illustrates the three tiers of the CBRS spectrum.

Since these tiers simultaneously use the CBRS spectrum, the FCC of the USA mandates that GAA users must not cause interference with PAL or incumbent users, and PAL users must not cause interference with incumbent users. A Spectrum Access System (SAS) is required to manage this potential interference [1].

SAS is a cloud-based, automated frequency management system mandated by the FCC to enforce CBRS rules and ensure efficient use of the spectrum. SAS acts as a dynamic coordinator, regulating how the spectrum is shared and preventing interference between users across the three tiers.

The SAS continuously monitors the spectrum usage and allocates channels dynamically to users based on their tier. Incumbent users are always given precedence, followed by PAL users, and then GAA users. To prevent collision scenarios, particularly involving Incumbent users, SAS currently employs several mechanisms, including continuous monitoring, spectrum reallocation for non-priority users, and preemption for Incumbent users.

Preventing interference is crucial in CBRS operations. SAS enforces a priority hierarchy, ensuring GAA users access spectrum only when PAL and In-

cumbent users are inactive. If a higher-priority user requires access, SAS reallocates GAA users dynamically. PAL users are similarly protected through exclusive spectrum assignments [2].

1.2 Background

GANs and diffusion-based generative models have gained significant traction for various applications, including spectrogram augmentation and audio synthesis. In [3], GAN-based models were employed for Radar spectrogram augmentation, leveraging a Joint Distribution Learning (JDL) module for injecting diversity into synthetic data. While effective for enhancing dataset variability, the reliance on adversarial training often poses challenges in terms of stability and convergence, particularly in complex spectrogram domains.

The work in [4] conducted a multimodal comparison of latent Diffusion Models (DDPMs) and GANs for medical image synthesis. Their results demonstrated the superiority of latent DDPMs over Least Squares GAN and Variational Autoencoder GAN, particularly in generating high-quality, diverse images. While this comparison offers insights into the potential of diffusion models, its application was limited to medical imaging and did not address the challenges of dynamic spectrum scenarios, such as those found in CBRS.

The U-Net architecture, proposed in Imagen [5] and later adapted in [6], has shown promise for spectrogram-based high-quality audio synthesis. These approaches exploit the hierarchical encoding-decoding structure of U-Nets to generate fine-grained audio spectrograms but have not been extended to the domain of spectrum-sharing systems or collision detection in CBRS.

Effective CBRS management relies on robust collision detection, yet traditional rule-based methods struggle in dynamic environments due to scalability and adaptability challenges. Deep learning-based classification offers a promising alternative, but the scarcity of collision data limits training [7].

To address this, we propose a novel spectrum-sharing system that generates spectrogram images of collision scenarios using GAN [8], DDPM [9], and VQ-VAE [10]. These images will train a CNN to detect collisions and alert SAS.

We evaluate the generative models by comparing their impact on CNN classification accuracy and data diversity. Then we simulate a CBRS environment using MATLAB which uses DRL for instead of a SAS to assign channels to users in a rapidly changing environment. While GANs and VAEs have been used to address data limitations, diffusion models have shown state-of-the-art results in natural image generation [4]. However, a direct use generative AIs with CNN for interference detection and DRL for channel allocation is unexplored.

This work employs conditional GAN, conditional DDPM, and VQ-VAE models to generate visualized CBRS datasets and compares their performance based on CNN classification accuracy.

The key contributions of this study are:

1. Introduce a scalable and generalizable spectrum-sharing system for a dynamic CBRS environment.

2. Use Generative models to generate visualized waveforms of a CBRS system.

3. Make a quantitative comparison of three of the most popular genera-

tive models currently available, in generating frequency spectrogram images.

4. Use DRL based channel allocation system with a CNN based collision detection system to manage the CBRS environment.

This document consists of the following: .....

CHAPTER 2

METHODOLOGY

2.1 Dataset

The dataset utilized for training our generative models and the CNN model was obtained from the research conducted by [7] called the ViC dataset. This dataset comprises 2400 images belonging to 12 different scenarios for two channels.

Each channel can have four possible states: empty, primary (RADAR), secondary (LTE), and collision. Since this spectrum-sharing system manages two tiers of users and two channels, there are 16 possible cases depending on how each user occupies each channel. However, four cases are discarded since the Citizens Broadband Radio Service Device (CBSD) is assumed to use only one channel simultaneously. Thus, the two channels could be in 12 different states. Figure 2.1 illustrates the spectrogram images corresponding to the channel states across the 12 classes. We used 2000 images to train the generative models and the other 400 for the CNN training.

The class names presented in Figure 2.1 will be used throughout this paper when discussing the results. Waveforms were visualized using Short Time Fourier Transform (STFT) with a size of 64×64 matrix.

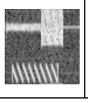
	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 9	Class 10	Class 11
Channel 1	Empty	Empty	Radar	Radar	Empty	LTE	Radar	LTE	Collision	LTE	Collision
Channel 2	Empty	Radar	Empty	Radar	LTE	Empty	LTE	Radar	Empty	Collision	Radar
Spectrogram Image											

Figure 2.1: Training ViC dataset containing the 12 classes [7]

2.2 VQ-VAE

The VQ-VAE is a generative model that combines continuous latent representations with discrete, interpretable codes by introducing a quantization mechanism. As illustrated in Figure 2.2 In VQ-VAE, the encoder maps the input x to a continuous latent representation $z_e(x)$ which is then quantized to the closest embedding $z_q(x)$ from a predefined discrete codebook $e_1, e_2, ..., e_k$. This process ensures that the latent space is discrete and semantically meaningful. During training, the quantization mechanism minimizes the difference between $z_e(x)$ and $z_q(x)$ using a gradient-through approximation. The decoder then reconstructs the input x from the quantized representation $z_q(x)$. The use of codebook $e_1, e_2, ..., e_k$ enables the model to structure the latent space compactly, making it highly effective for tasks like image generation and speech synthesis. This discrete representation, denoted by $q(z|x)$, improves reconstruction quality by focusing on meaningful latent structures and allows efficient scaling for large datasets and complex generative tasks. The innovation of replacing continuous latent spaces with quantized embeddings results in a model that bridges the gap between interpretability and high-quality gener-

ation.

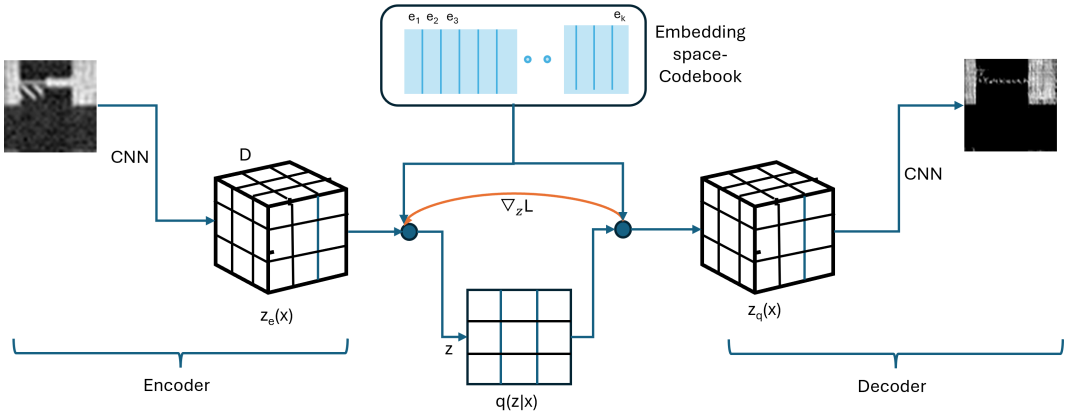


Figure 2.2: The VQ-VAE architecture

In the architecture of the conditional VQ-VAE we used, the encoder is a CNN designed to extract features from the input grayscale images. It has three convolutional layers to progressively reduce spatial dimensions while increasing the number of feature channels. The encoder also incorporates a residual stack, which consists of multiple residual layers to enhance feature extraction by allowing information to bypass intermediate transformations, thereby improving gradient flow and capturing complex patterns.

After encoding, the latent representation is passed through a 1x1 convolution layer to adjust the feature dimensions to match the input requirements of the vector quantizer. This mapping is performed using the Euclidean distance between the latent vectors and the codebook embeddings. To ensure effective learning, the vector quantizer minimizes a combination of the reconstruction loss, a codebook commitment loss, and a latent quantization loss.

The loss function can be expressed as follows:

$$L = \mathbb{E}[\|x - \hat{x}\|^2] + \beta \cdot \mathbb{E}[\|z_{\text{enc}} - \text{sg}(z_{\text{quant}})\|^2] \tag{2.1}$$

where x is the input image, $\hat{x}$ is the reconstructed image, z_{enc} is the latent vector from the encoder, z_{quant} is the quantized vector, and $\text{sg}(\cdot)$ denotes the stop-gradient operator.

The decoder is another CNN that takes the quantized latent representations and reconstructs the original input images. It mirrors the encoder's structure, starting with a convolutional layer followed by the residual stack to refine features, and ends with transposed convolutional layers to upsample the latent representation back to the original input dimensions. The decoder ensures that the reconstructed images retain high fidelity to the input.

We implemented a conditional VQ-VAE model, enabling class-specific image generation. The encoder compresses grayscale input images using three convolution layers followed by residual blocks, while the decoder mirrors this process using transposed convolutions to reconstruct the data. The quantization layer is also class-conditioned, leveraging a codebook with 512 embeddings, each of 64 dimensions.

2.3 GAN

GANs, consist of two neural networks: a generator that creates data samples and a discriminator that evaluates their authenticity. The generator and discriminator engage in a game-theoretic scenario where the generator aims to produce indistinguishable fake data, while the discriminator strives to iden-

tify real from generated data. This adversarial process allows GANs to produce highly realistic images, audio, and other data forms [11]. The loss functions used by the Generator and the Discriminator can be written as follows. Figure 2.3 illustrates this architecture. L_D and L_G are the loss functions for

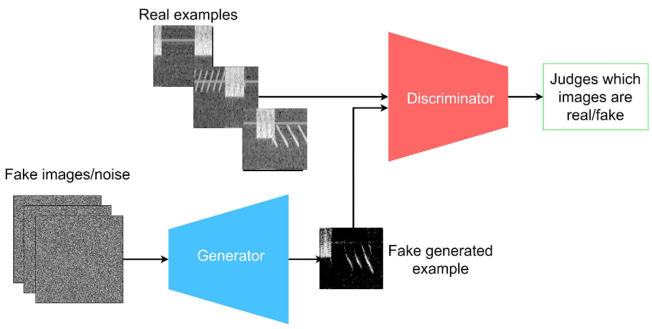


Figure 2.3: The GAN architecture

the Discriminator and the Generator for image x, respectively.

$$L_D = \mathbb{E}[max(0, 1 - D(x_{real}))] + \mathbb{E}[max(0, 1 + D(G(z_{fake})))] \tag{2.2}$$

$$L_G = \mathbb{E}[D(G(z_{fake}))] \tag{2.3}$$

We used a conditional GAN with a Generator $\mathcal{G}$ and a Discriminator $\mathcal{D}$ which were trained simultaneously through a min-max game. For the Generator, we used an input noise vector z of 100 dimensions and an embedding layer to convert class labels into embeddings with five convolution layers and ReLU activations in hidden layers. Tanh activation was used in the output layer to produce pixel values in the range [-1, 1]. In the Discriminator, five convolution layers were used to down-sample the images with Leaky ReLU

activations in hidden layers. We used Sigmoid activation in the output layer to produce the probability.

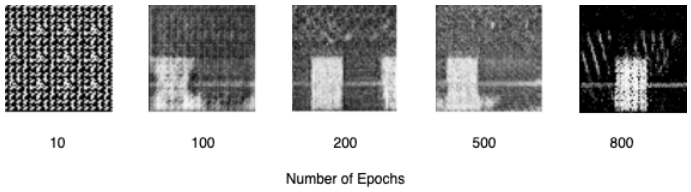


Figure 2.4: Spectrogram images generated by the Generator of the GAN

We generated 1200 images with 100 images per class for the twelve classes and used the pre-trained CNN model to classify the images.

2.4 DDPM

DDPMs are generative models designed for image generation using variational inference. They operate through a Markovian process that involves a finite sequence of steps, denoted as T. Training involves two main stages: the forward process, where noise is added to data in a controlled manner, and the reverse process, where the model learns to denoise this data step-by-step. Each step in this process acts as a denoising operation, aiming to refine the image quality progressively[9]. Figure 2.5 illustrates the diffusion process of the DDPM.

During the forward process, the clean image y_0 is sampled with Gaussian noise of variance $\beta_1, ..., \beta_T$ is added over T time steps.

$$\begin{aligned} q(y_t \mid y_0) &:= \mathcal{N}(y_t; \sqrt{\bar{\alpha}_t} y_0, (1 - \bar{\alpha}_t) I) \\ &= \sqrt{\bar{\alpha}_t} y_0 + \epsilon \sqrt{1 - \bar{\alpha}_t}, \epsilon \sim \mathcal{N}(0, I) \end{aligned} \tag{2.4}$$

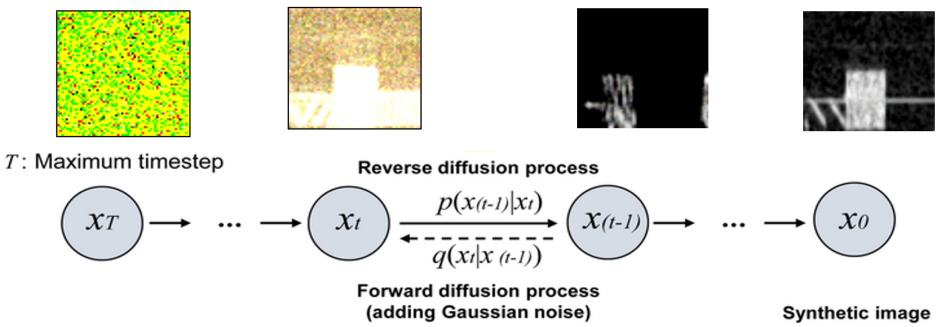


Figure 2.5: Diffusion process of the DDPM

Where $\alpha_t = 1 - \beta_t$

In the reverse process, the added noise is removed step by step to recover the image y_0.

$$p(y_T) = \mathcal{N}(0, I) \tag{2.5}$$

$$p(y_{(t-1)} \mid y_t) = \mathcal{N}(y_{(t-1)}; \mu_\theta(y_t, t), \sqrt{\beta_t} I) \tag{2.6}$$

We used a conditional DDPM, and when training the DDPM we enabled mixed precision training and multithreaded data loaders. We sampled the images regularly during the training and we could see the learning rate gradually decreasing with the time steps. For training our model we sampled a timestep $t \sim U[1, T]$ with $T = 2000$ and a 5×10^{-3} learning rate. After training the model, we used the trained weights to generate 1200 images with 100 images per class and used the pre-trained CNN model to classify the images to their respective classes.

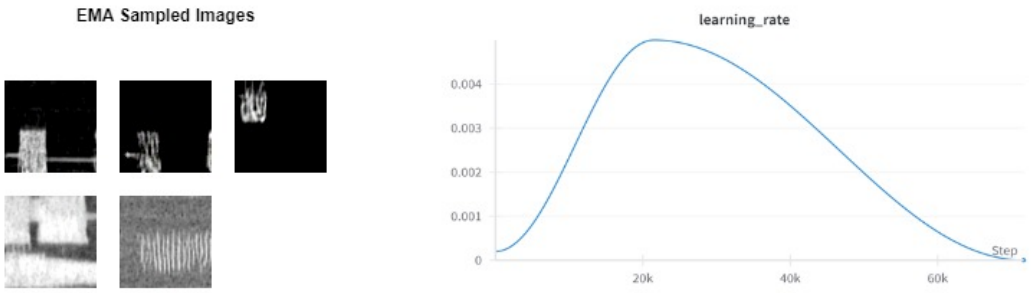


Figure 2.6: EMA sampled images and the change of the learning rate with each time step when training the DDPM

2.5 CNN

CNNs are a type of deep learning model widely used for image analysis tasks. They are designed to automatically and adaptively learn spatial hierarchies of features from input data through layers of convolutional filters. These filters detect patterns such as edges, textures, and more complex features in the images, making CNNs highly effective for tasks like image classification.

We trained a CNN model that can detect collision scenarios using the generated spectrogram data from the three models. Since the spectrogram images generated by the DDPM, GAN, and VQ-VAE models contained some noise, which could have led to misclassification by conventional CNN architectures, particularly when distinguishing fine-grained features, we employed an enhanced CNN architecture specifically designed to effectively capture and differentiate subtle details within the images, thereby mitigating the impact of noise on classification accuracy.

We used a CNN architecture based on a modified ResNet-50 model, en-

hanced with a Convolutional Block Attention Module (CBAM) to improve its feature extraction and classification capabilities. The architecture comprised convolutional layers for basic feature extraction, residual blocks for efficient gradient propagation, Global Average Pooling (GAP) for summarizing spatial information, and a Fully Connected (FC) layer adapted for classifying the 12 spectrogram image classes.

The model was trained on spectrogram images resized to 448×448 for capturing finer details. The training used the Adam[12] optimizer for adaptive gradient updates and a learning rate scheduler that reduced the learning rate every five epochs to ensure stable convergence.

2.6 DRL

In this work, we employed a DRL agent to learn optimal spectrum access strategies in a simulated CBRS environment. The agent interacts with an environment that mimics the dynamic characteristics of the CBRS band, including incumbent radar signals and LTE-based priority access users. The environment is modeled as a Markov Decision Process (MDP) where the agent observes the occupancy of two communication channels and selects an action from a discrete set: idle, use channel 1, or use channel 2.

We implemented a Deep Q-Network (DQN) agent that approximates the Q-value function using a neural network. The agent receives as input a two-dimensional observation vector representing the current channel state, where each element indicates whether a channel is occupied (1) or free (0), as predicted by the pre-trained CNN model that analyzes the spectrogram of the re-

ceived signal. The agent's goal is to maximize a cumulative reward by learning a policy that avoids interference with higher-tier users while efficiently utilizing the available spectrum. The reward structure is defined such that selecting an idle channel yields a positive reward, selecting an occupied channel results in a penalty, and remaining idle results in a small negative reward to discourage inactivity.

The neural network used by the DQN agent consists of a feature input layer followed by two fully connected layers with ReLU activations, and a final output layer that estimates Q-values for each possible action. During training, we enabled GPU acceleration to improve performance. The training utilized an experience replay buffer and a target network, and was configured with Double DQN and a smooth target update factor of 10^{-3} to stabilize learning. We set the discount factor $\gamma = 0.99$, and trained the model for 1000 episodes with a maximum of 100 steps per episode.

The environment generates new signal observations by randomly simulating different channel conditions, including 'empty', 'radar', 'LTE', and 'collision'. These synthetic signals are converted into spectrogram images, resized to 224×224 pixels, and classified by the CNN. This predicted class is used to update the channel occupancy vector, which forms the observation space for the DRL agent.

2.7 System Architecture

We evaluated classification performance across different generative models and computed class-wise and overall accuracy, along with FID scores for each

class. Then, we used a pre-trained CNN model trained with a dataset comprising both original data and the data generated by the three generative models to detect incumbent users. Figure 2.7 illustrates this scenario.

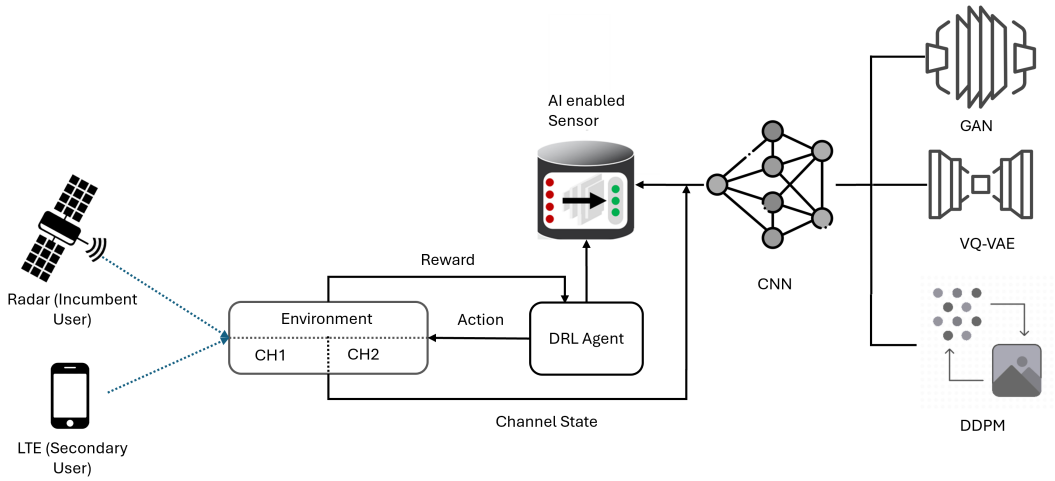


Figure 2.7: System Architecture

2.8 Experimental Setting

2.8.1 VQ-VAE

The conditional VQ-VAE model was designed for class-specific image generation, with an encoder that compresses grayscale inputs through three convolutional layers and residual blocks to enhance feature extraction. The quantization layer, class-conditioned, uses a codebook of 512 embeddings (64 dimensions each) to discretize the latent space. A 1×1 convolution adjusts the encoder's output to fit the quantizer's input requirements. The decoder, struc-

tured similarly to the encoder, uses transposed convolutions and residual stacks to reconstruct high-fidelity images from quantized representations. The model generated 1,200 images, with 100 per class across 12 classes, which were classified using a pre-trained CNN.

2.8.2 GAN

The conditional GAN used in this study consists of a generator and discriminator trained in a min-max game. The generator takes a 100-dimensional noise vector and class embeddings, passing them through five convolutional layers with ReLU activations and a Tanh output to generate pixel values in the range [-1, 1]. The discriminator uses five convolutional layers with Leaky ReLU activations to downsample input images, and a Sigmoid output provides a probability score. The model generated 1,200 images, with 100 per class across 12 classes, which were classified using a pre-trained CNN.

2.8.3 DDPM

The conditional DDPM used in this study generates images through a two-stage Markovian process: a forward process, where Gaussian noise is added to the clean image over T=2000 timesteps, and a reverse process, where the model removes the noise step-by-step to reconstruct the original image. Mixed precision training and multithreaded data loaders were utilized for efficiency, and images were sampled during training to monitor progress. The model was trained with a learning rate of 5×10^{-3}, sampling timesteps $t \sim U[1, T]$ After training, 1,200 images (100 per class) were generated and classified using a pre-trained CNN.

2.8.4 CNN

A modified ResNet-50 architecture with a CBAM was used to classify 12 spectrogram image classes, effectively capturing fine-grained details despite noise. The model was trained on 448×448 resized images using the Adam optimizer and a learning rate scheduler to ensure stable convergence. For training, we used a combination of the original dataset and the generated data from each model. The combined dataset was divided into 80% for training and 20% for inference, and the results are based on three separately trained models, each incorporating generated data from one of the three-generation models.

We compared the class-wise and overall accuracy of the CNN classification model trained with data generated by different models for the inference set. Then, we calculated the FID score for each class.

2.8.5 FID

FID is a widely used metric for evaluating the quality of images generated by generative models. It quantitatively compares the distributions of features extracted from real and generated images in a high-level feature space. These features are typically obtained from a pre-trained neural network, such as the Inception-v3 model. The FID score is lower when the two distributions are closer, indicating higher quality and diversity of the generated images [21].

Let μ_r and Σ_r represent the mean and covariance of the features extracted from real images, and μ_g and Σ_g represent the mean and covariance of the features extracted from generated images. The FID score is calculated using the following formula:

$$\text{FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}}) \tag{2.7}$$

Usually, to compute the FID, real and generated images are passed through the pre-trained Inception-v3 network, and features are extracted from a specific layer (often the "pool3" layer). The means and covariances of these features are then computed for both datasets. However, in this paper, we used a ResNet18 model with the final classification layer replaced instead of Inception-v3. The ResNet18 model employed for FID calculation is distinct from the CNN model used for classification. While the ResNet18 was trained exclusively on the original dataset to extract consistent feature representations for real and generated images, the classification CNN was trained on a combined dataset comprising both real and synthetic data. This separation ensures that the FID evaluation remains unbiased by the influence of synthetic data used in the classification task. Finally, the FID score is calculated by substituting these statistics into equation 2.7.

A lower FID Score indicates that the distributions of real and generated features are more similar, implying better quality and diversity of generated images whereas a higher FID score indicates that the generated images are less similar to real images, suggesting poorer quality or lack of diversity.

2.8.6 PCA

PCA is a statistical technique used for dimensionality reduction, facilitating the analysis and visualization of high-dimensional data. By transforming the original variables into a new set of uncorrelated variables, PCA captures the

directions of maximum variance in the data. The first principal component accounts for the largest possible variance, with each succeeding component accounting for the remaining variance under the constraint of being orthogonal to the preceding components. This method simplifies complex datasets while preserving their essential patterns and structures [13].

2.8.7 DRL

The DRL agent observes the current state (e.g., channel occupancy or spectrogram predictions), selects an action (e.g., choose Channel 1, Channel 2, or remain idle), and receives a reward based on the outcome (e.g., successful transmission or interference avoidance). Over repeated episodes, the agent learns to maximize its long-term performance, such as throughput or interference avoidance, by training a neural network to approximate optimal policies. Figure 2.8 depicts Incumbent signals generated by the MATLAB simulation while Figure 2.9 illustrates the DRL agent's training process, where the agent learns to select the best channel based on the spectrogram images generated by the three models. The agent's training involved 1000 episodes, with each episode consisting of 100 steps, and it was trained using a reward structure that incentivized efficient spectrum access while avoiding interference with higher-tier users.

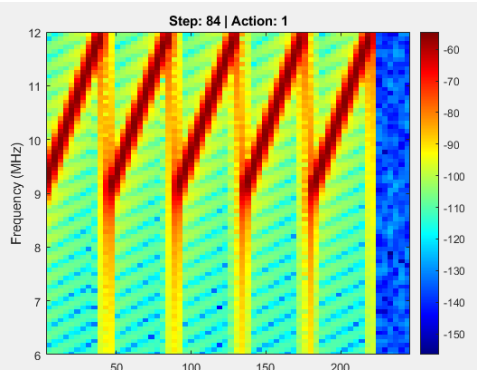


Figure 2.8: MATLAB simulation

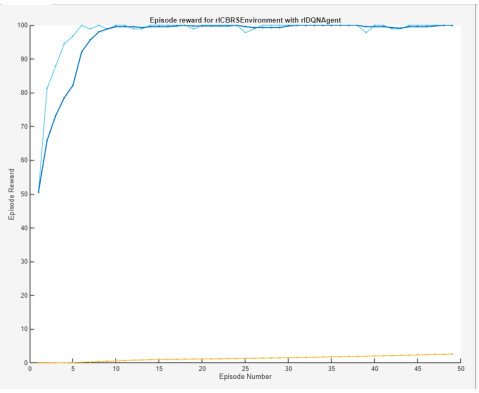


Figure 2.9: DRL agent training process

CHAPTER 3

RESULTS

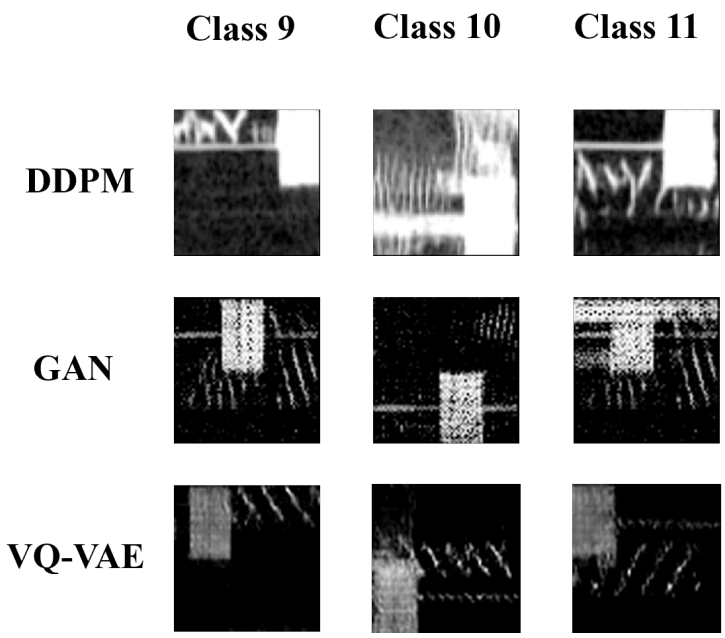


Figure 3.1: A side-by-side comparison of some of the collision scenarios generated by the DDPM, the GAN, and the VQ-VAE

Classes depicted in Figure 3.1: class 9, class 10, and class 11 are all collision scenarios (Refer to Figure 2.1), which were generated by the DDPM, the GAN, and the VQ-VAE.

Figure 3.2 illustrates the change in FID score across the classes for the 3 models. The GAN, DDPM, and VQ-VAE models achieved average FID scores of

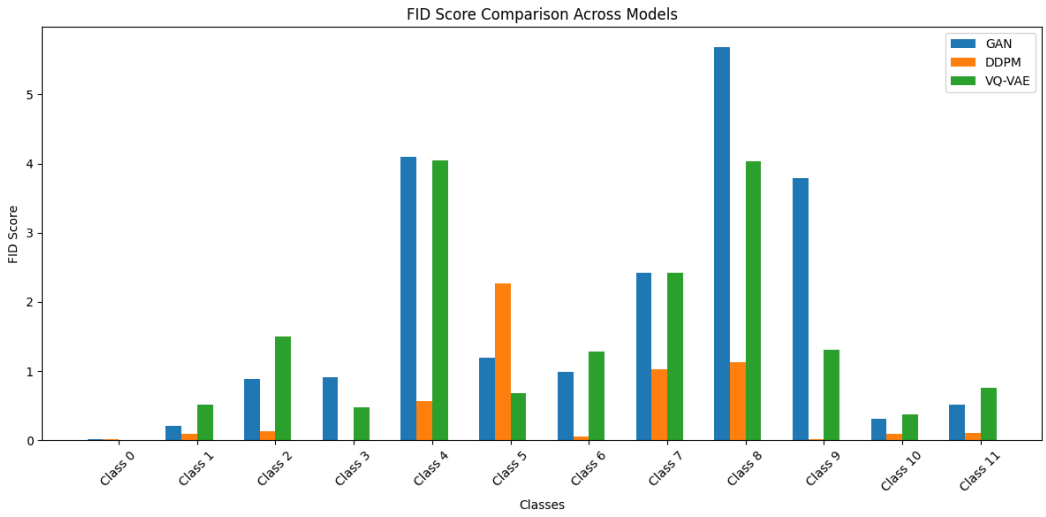


Figure 3.2: FID Score Comparison Across Models for Different Classes.

1.944, 0.504, and 1.580, respectively. We used a ResNet18 model with the final classification layer replaced to extract features instead of using a pre-trained InceptionV3 network due to the specific nature of our dataset.

Since our dataset lacks the complex textures, patterns, and color information typically found in natural image datasets, the differences between real and generated spectrograms are smaller in the feature space. As a result, the FID scores we obtained are smaller than usual for all three models.

Class 0 consistently shows near-zero FID scores across all three models due to its purely noise-like nature. As observed in the graph, the classes representing collision scenarios—Class 9, Class 10, and Class 11—exhibit significantly lower FID scores for images generated by the DDPM. This suggests that the spectrogram images with collision scenarios generated by the DDPM are more natural and resemble the original dataset better than the other two

models. The other two models also have relatively low FID scores for Class 10 and Class 11.

Among the three models, the DDPM also has the lowest average FID score, suggesting that DDPM is the most effective model overall for generating spectrograms that closely resemble the real dataset, as evidenced by its consistently low FID scores across most classes.

The GAN model's high FID scores across many classes suggest model collapse in its generated spectrograms for this dataset, to which GANs are susceptible. Given that our main focus is on generating a diverse dataset to improve the accuracy of a visualized interference classification system, having a wide variety of data is essential.

VQ-VAE model, on the other hand, shows varied performance, outperforming GAN in many cases (e.g., Classes 1, 2, 6, 9) but falling behind DDPM in most.

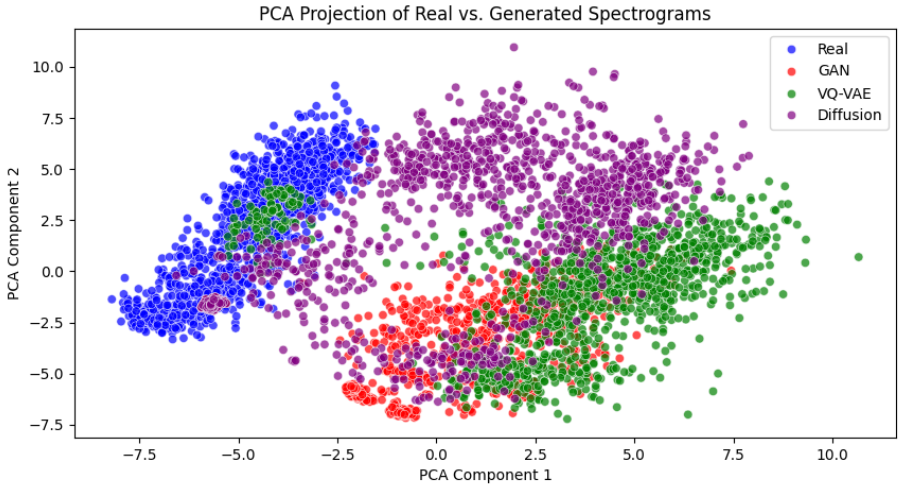


Figure 3.3: PCA scores for the GAN, DDPM and VQ-VAE

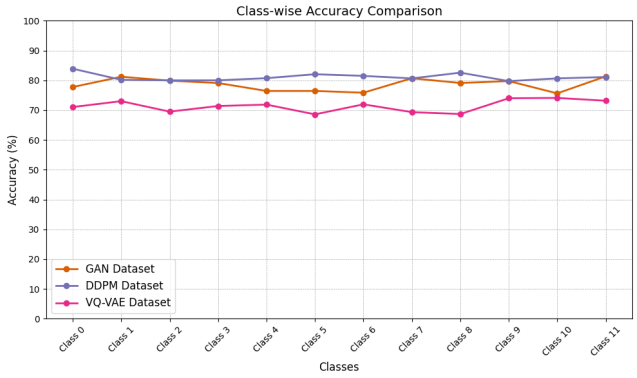


Figure 3.4: Class-wise classification accuracy for GAN, DDPM and VQ-VAE generated data

The PCA projection in Figure 3.3 indicate that real and generated spectrograms form separate yet overlapping clusters, suggesting that generative models do not simply reproduce existing samples but introduce new variations. The spread of generated samples in the PCA projection indicates that the models create additional variations, potentially improving classifier generalization. The PCA analysis provides an independent validation of data distribution differences, confirming that generative models add meaningful diversity. However, the PCA projection also shows that the GAN model's generated samples are more clustered and less spread out than the other two models, indicating that the GAN model may not be as effective at generating diverse samples as the DDPM and VQ-VAE models. The DDPM model's generated samples are more spread out than the VQ-VAE model's, suggesting that the DDPM model may be more effective at generating diverse samples than the VQ-VAE model.

Figure 3.4 illustrates the change in the classification accuracy of the CNN

model for the spectrogram images generated by the three generative models for the twelve different data classes. Meanwhile, the spectrogram images generated by the DDPM generally outperformed the images generated by other models. Even though we used a CNN model specifically trained to distinguish fine-grained features ignoring the noise, the high noise and blurry features in the VQ-VAE model seem to have affected the accuracy of the CNN model than the other two models.

Cases	Model	Average Accuracy	FID Score
1	DDPM	81.9%	0.504
2	GAN	78.5%	1.944
3	VQ-VAE	71.3%	1.580

Table 3.1: Average accuracy for GAN, DDPM, and Original data

Table 1 depicts the average classification accuracy and the FID scores of the spectrogram images, for the three different models and the original data. Although the average accuracy is not far apart for the three generative models, due to the lack of diversity of the generated data by the GAN, and the noisy images by VQ-VAE we can conclude that the DDPM performs better than the GAN or VQ-VAE when generating spectrogram images for a robust spectrum-sharing system.

CHAPTER 4

DISCUSSION

Equation is cited as Equation (4.1) and Equation (4.3) Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat **ref1**. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur **ref2**. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum **ref3**.

$$F = ma \tag{4.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \tag{4.2}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{4.3}$$

CHAPTER 5
CONCLUSION

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat **ref1**. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur **ref2**. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum **ref3**.

Appendices

APPENDIX A
EXPERIMENTAL EQUIPMENT

In **??**, Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

APPENDIX B

DATA PROCESSING

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Monospace font Test

`PPO Algorithm in Pseudocode.`

Bold and Monospace Test

`PPO Algorithm in Pseudocode.`

Algorithm 1 PPO

for $iteration = 1, 2, \ldots$ **do**
  for $actor = 1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{old}}$ in environment for T time steps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt. θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{old} \leftarrow \theta$
end for

REFERENCES

[1]HPE Aruba Networking, “What is cbrs?”, https://www.arubanetworks.com/faq/what-is-cbrs/.

[2]Federal Communications Commission, “Amendment of the commission’s rules with regard to commercial operations in the 3550-3650 mhz band”, FCC Report and Order (2015).

[3]Y. Yang, Y. Zhang, Y. Lang, B. Li, S. Guo, and Q. Tan, “Gan-based radar spectrogram augmentation via diversity injection strategy”, IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1–12 (2022).

[4]P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis”, CVPR (2021).

[5]C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, “Photorealistic text-to-image diffusion models with deep language understanding”, Advances in Neural Information Processing Systems, 35:36479–36494 (2022).

[6]G. Zhu, Y. Wen, M.-A. Carbonneau, and Z. Duan, “Edmsound: spectrogram based diffusion models for efficient and high-quality audio synthesis.”, https://arxiv.org/pdf/231 (2023).

[7]H. Nam, K. Han, J. Park, H. Ch, and S.-L. Kim, “A spectrum sharing system: visualized interference classification and cnn approach”, 2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Los Angeles, CA, USA (2021).

[8]I. J. 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).

[9]J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models”, NeurIPS 2020 (34th Conference on Neural Information Processing Systems) (2020).

[10]A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning”, NIPS’17: Proceedings of the 31st International Conference on Neural Information Processing Systems (2018).

[11]A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: an overview”, IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53-65 (2018).

[12]D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization”, 3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA, May 2015. (2015).

[13]F. L. Gewers, G. R. Ferreira, H. F. de Arruda, F. N. Silva, C. H. Comin, D. R. Amancio, and L. da F. Costa, “Principal component analysis: a natural approach to data exploration”, ACM Comput. Surv. 54, 4, Article 70 (2022).