

# Continual learning on audio scene classification using representative data and memory replay GANs

Ibnu Daqiqil ID<sup>1</sup>, Masanobu Abe<sup>2</sup>, Sunao Hara<sup>2</sup>

<sup>1</sup>Computer Science, Faculty of Mathematic and Natural Science, Universitas Riau, Pekanbaru, Indonesia

<sup>2</sup>Graduate School of Interdisciplinary Science and Engineering in Health Systems, Okayama University, Okayama, Japan

## Article Info

### Article history:

Received Jan 2, 2024

Revised Aug 29, 2024

Accepted Sep 28, 2024

### Keywords:

Audio scene classification

Continual learning

Generative adversarial model

Memory replay

Representative memory

## ABSTRACT

This paper proposes a methodology aimed at resolving catastrophic forgetting problem by choosing a limited portion of the historical dataset to act as a representative memory. This method harnesses the capabilities of generative adversarial networks (GANs) to create samples that expand upon the representative memory. The main advantage of this method is that it not only prevents catastrophic forgetting but also improves backward transfer and has a relatively stable and small size. The experimental results show that combining real representative data with artificially generated data from GANs, yielded better outcomes and helped counteract the negative effects of catastrophic forgetting more effectively than solely relying on GAN-generated data. This mixed approach creates a richer training environment, aiding in the retention of previous knowledge. Additionally, when comparing different methods for selecting data as the proportion of GAN-generated data increases, the low probability and mean cluster methods performed the best. These methods exhibit resilience and consistency by selecting more informative samples, thus improving overall performance.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Ibnu Daqiqil ID

Computer Science, Faculty of Natural Science, Universitas Riau

Kampus Bina Widya KM. 12,5, Simpang Baru, Kota Pekanbaru, Riau 28293, Indonesia

Email: daqiqil@s.okayama-u.ac.jp

## 1. INTRODUCTION

In practical applications, it is crucial for a machine learning model to continuously update itself with new data in order to incorporate the newly acquired knowledge [1]. Nevertheless, when we retrain a model by adjusting its parameters using the most recent data, the model tends to excel primarily on the recently acquired information. This occurs due to the fact that the model's parameters are exclusively fine-tuned for the latest data and may neglect the optimization that was initially achieved during earlier learning phases. This problem is commonly referred to as catastrophic forgetting or interference [2].

In the context of audio scene classification (ASC), it is customary for sounds within a given scene to undergo frequent changes. When recording audio data for training purposes, the captured sounds may deviate from the actual environmental conditions. For instance, in the scenario of a park scene, if an audio recording is conducted in a park during daylight hours, it may capture the sounds of children playing as an example. Nonetheless, if the recording is conducted during nighttime, it might encompass the sounds of insects and the serenity of the nocturnal environment. Thus, variations in ASC can also be contingent upon factors such as location, time, and recording conditions. Consequently, a machine learning model must possess the capability to adapt to these fluctuations. Furthermore, when there is an incorporation of new labels or categories pertaining to recording locations.

Recent research endeavors have been directing their attention towards rehearsal-based strategies, which entail the utilization of a limited subset of previous data. These approaches exhibit potential in mitigating the issue of catastrophic forgetting [3], [4]. The rehearsal methods, although valuable, frequently encounter the problem of overfitting since the quantity of data they retain is considerably smaller than the incoming new data. As a result, these stored samples can either contribute to overfitting or be disregarded during training due to their limited volume. While increasing memory storage for new data may appear to be a straightforward solution, it doesn't align with the constraints of limited memory space in practical scenarios. The challenge lies in the efficient preservation of crucial old information using a restricted number of samples.

This paper is an extension of our work presented in [5] and in previous research [6], [7], it has been observed that samples generated through a generative method can sufficiently retain the acquired knowledge. Nonetheless, the effectiveness of this approach is greatly contingent upon the caliber of the generator utilized. Therefore, our proposed methodology entails the storage of a single generator with the capability to generate representations for all past samples. Furthermore, we employ a distillation technique that harnesses the prior generator and engage in retraining with memory to counteract any biases in the generator's output.

The remainder of this paper is organized as: section 2 provides an overview of related research that has influenced this research. Section 3 outlines our proposed method. The experimental setup is elaborated upon in section 4, and section 5 offers an exhaustive analysis and discussion of the experimental outcomes. Finally, section 6 concludes this paper by summarizing the primary outcomes and presenting concluding remarks.

## 2. RELATED WORK

### 2.1. Audio scene classification in the concept drift situation

ASC refers to the procedure of identifying and categorizing environmental sounds within a particular context or environmental category [8]. ASC can find application in various tasks, including but not limited to surveillance [9], urban planning, and enhancing user experiences in multimedia [10]. The primary challenge in ASC stems from concept drift, which pertains to the alteration of sound distribution within an environment over time. This can have a notable impact on the efficacy of classification models, as the acoustic attributes within an environment may undergo changes due to fluctuations in weather, urban development, or daily and seasonal activity patterns.

For instance, when capturing sounds in a garden, the timing of the recording can influence the occurrence and volume of sounds, such as birds chirping at sunrise. Additionally, the specific location within the park is of significance; a pond may feature the sounds of water and ducks, whereas a playground will be accompanied by the sounds of children playing. As time passes, the introduction of new factors, such as construction activities in nearby parks or seasonal variations, can result in alterations in sound distributions. A model that has been trained with park sounds during the morning hours may not be capable of recognizing the acoustic environment during the evening due to shifts in ambient noise levels and activity patterns. This phenomenon is referred to as concept drift.

Concept drift is a phenomenon in machine learning wherein the data distribution undergoes changes over time [11]-[13]. This impacts the performance of models that were previously trained. Such alterations can be triggered by a variety of factors, including environmental shifts, modifications in user behavior, or transformations in the data source. Concept drift can pose a substantial challenge in the administration of machine learning models, particularly when these models are employed in applications demanding high levels of accuracy and performance consistency. Figure 1 illustrates concept drift as the shift in feature distribution over time.

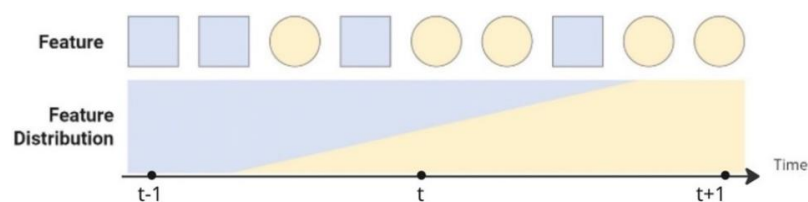


Figure 1. Feature and feature distribution change illustration in concept drift

Introducing a new class into an existing machine learning model constitutes one manifestation of concept drift. The inclusion of the new class can lead to alterations in the data distribution, potentially jeopardizing the performance of the pre-existing model. Hence, it holds significance to monitor and tackle concept drift when incorporating new classes into the model, ensuring that the model can sustain its accuracy and efficacy as time progresses. Within this context, strategies for handling concept drift and integrating new classes assume paramount importance in preserving the quality of machine learning models [14], [15].

Recent studies in the field of ASC have concentrated on the creation of resilient models to confront these difficulties. Methods such as transfer learning, which involves the application of knowledge from one domain to another, have been employed to mitigate concept drift. Moreover, incremental learning techniques, enabling the model to acquire knowledge in an ongoing manner, have been suggested to address the problem of catastrophic forgetting.

## 2.2. Continual learning

Concept drift and catastrophic forgetting represent substantial challenges within the realm of machine learning. Concept drift denotes the situation in which a model must adapt to changing data patterns to uphold its performance. On the other hand, catastrophic forgetting describes the occurrence in which a model, upon assimilating new data, unintentionally erases previously acquired knowledge. To tackle these challenges, approaches such as preserving raw data, generating data abstractions, or retaining specific model components that encapsulate prior learning are utilized. Effective strategies are imperative to safeguard crucial historical knowledge for utilization in new contexts. Learning algorithms are typically developed with a focus on regularization to restrain modifications to learned information (regularization-based methods), architectural adjustments to accommodate new learning tasks (architecture-based methods), or rehearsal to reinforce prior knowledge concurrently with new learning (rehearsal-based methods). The strategies mentioned aim to establish a balance between preserving existing knowledge and integrating new information. This helps avoid the problems of concept drift and catastrophic forgetting.

Regularization-based methods [16]-[19] constrain the update velocity of parameters vital to previously learned tasks to regulate the model's updating process. While these methods can mitigate catastrophic forgetting by not directly storing instances of past data, their efficacy may diminish in more challenging scenarios [20] or when applied to complex datasets [21].

Architecture-based approaches aim to evolve or modify the model's structure (network or components) during the incremental training process [22]-[26]. Some strategies involve the use of masks to selectively activate segments of the network [27]-[29], while others modify existing components within the model [8]. For instance, one approach discussed in [22] extending the network by augmenting each layer with a fixed number of neurons for new tasks, while preserving the parameters of the established layers to avoid the loss of previously learned information. Nevertheless, such methods generally result in heightened complexity and size of the network. Furthermore, certain techniques require knowledge of task identity to condition the network during inference, which could limit the network's architectural adaptability and potentially hinder performance improvement [19].

Rehearsal-based methods have demonstrated successful outcomes compared to other approaches. Rebuffi *et al.* [30] presented iCarl, which utilizes a technique known as herding to select and store exemplars that are representative of previous learning sessions. His method integrates distillation and classification losses to update a classifier incrementally, applying the nearest mean of exemplars as a rule for classification. Yoon *et al.* [3] investigated an approach for online core-set selection, focusing on selecting samples from previous tasks that are most representative of the data distribution. This method has shown superiority over state-of-the-art techniques like EWC [17], A-GEM [31], and ER-Reservoir [32]. introduced Gdumb, a method that greedily retains samples while ensuring class balance. Moreover, generative-based techniques like generative feature replay [33], memory replay-generative adversarial networks (MER-GAN) [7] have showcased the utilization of GANs for knowledge preservation. However, these approaches typically require additional storage to maintain information from previous learning experiences.

## 3. PROPOSED METHOD

This article presents an incremental learning mechanism founded on rehearsal, where the utilized data encompasses not only representative data but also a blend of various other data. This mix includes three critical types: the actual task-specific data ( $\mathcal{D}_{task}$ ), a curated subset that represents the previous data distributions ( $\mathcal{D}_{rep}$ ), and synthetically generated data that serves to fill in gaps or extend the training regimen ( $\mathcal{D}_{gan}$ ). The procedure behind this approach is influenced by the learning techniques observed in educational settings. Typically, learners frequently employ repetitive research methods to reinforce their comprehension and memory of information. This can be accomplished through various repetitive strategies, such as rote

learning, repetition, note-taking, or emphasizing crucial sections of the text by underlining or highlighting. These strategies are recognized as advantageous for students to improve memory retention [34]. We incorporate these rehearsal-based learning techniques into computer neural networks to enhance their performance.

In the realm of computational neural networks, the rehearsal process can be effectively modeled by capturing and preserving data from prior training sessions. This accumulated data is subsequently intelligently reused to strengthen the knowledge of the trained network and facilitate the incorporation of new information [35]. By employing this rehearsal-based framework, we can leverage the pre-existing knowledge within the network while consistently accommodating new knowledge. This, in turn, enables more robust and adaptive learning capabilities.

We contemplate a sequential training setup comprising a sequence of tasks represented as  $\mathcal{T} = (\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_N)$  where  $N$  represents the total number of tasks. Each individual task,  $\mathcal{T}_t$  is associated with a specific dataset  $\mathcal{D}_{task}^t = \{x_n^t, y_n^t\}_{n=1}^{N_t}$  containing  $N_t$  data points along with their corresponding labels. Importantly, within task  $\mathcal{T}_t$ , we assume that  $y^t$  consists of unique classes that do not overlap with the label sets from previous tasks,  $y^t \cap \{y^0 \dots y^{t-1}\} = \emptyset$ . Our objective is to minimize the standard loss function, usually denoted as the cross-entropy loss, during the training of the model on task  $\mathcal{T}_t$ . This guarantees that the model acquires proficiency in the particular task under consideration while sustaining its performance on tasks encountered previously. By adhering to this sequential training setup and minimizing the standard loss function, we can efficiently train the model to address each task independently while retaining the acquired knowledge from prior tasks. This approach empowers the model to progressively acquire new skills and adjust to the specific demands of each task, resulting in improved performance across a variety of tasks within the training sequence.

The initial phase of this framework involves training the classifier  $M$ , selecting representative data  $\mathcal{D}_{representative}$ , and training the generator  $G$ . Once this initial training is completed, the next step is to conduct incremental training to enhance the knowledge in  $M$ . The details of the process are:

### 3.1. Classifier

In our experiment, we employ a convolutional neural network (CNN) model for the task of ASC. The CNN architecture consists of multiple layers, particularly convolutional layers, which have a vital role in capturing local patterns within the input feature map. Each convolutional layer is equipped with kernels that convolve with the feature map, enabling the network to extract pertinent features. The classifier, denoted as  $M$ , consists of two main components: the feature extractor  $F$  and the classifier head  $C$ . When presented with data  $x$ , the feature extractor  $F$ , parameterized by  $\theta$ , extracts the feature vector  $u$ . This feature vector is then fed into the classifier head  $C$ , which employs a projection matrix  $V$  to transform  $u$  into class scores using a softmax function denoted as  $\mathcal{A}$ . In other words, the classifier computes  $C = \mathcal{A}(Vu)$ , to classify the input feature,  $M(F(x; \theta); \mathcal{A}(Vu))$ .

The feature extractor  $F$  consists of six convolutional layers, drawing inspiration from VGG-like CNNs. Within each convolutional block, there are two convolutional layers with a kernel size of  $3 \times 3$ . To enhance training stability and efficiency, batch normalization is applied between each convolutional layer, and we employ the rectified linear unit (ReLU) for nonlinearity. Furthermore, in each convolutional block, we employ average pooling with a size of  $2 \times 2$  to reduce the image dimensions. To obtain fixed-length feature vectors, we add a fully connected layer on top of the feature extractor  $F$ . This layer extracts high-level features from the convolutional outputs. For a comprehensive depiction of the network architecture, please consult Figure 2.

### 3.2. Representative data

Data representative ( $\mathcal{D}_{representative}^t$ ) refers to a subset of training data ( $\mathcal{D}_{task}^{t-1}$ ) from previous episodes or steps that is carefully chosen and retained for subsequent learning. This data is stored within the data reservoir  $R^t$ .  $\mathcal{D}_{representative}^t$  was selected using several methods:

- High or low probability: this approach employs the value generated from  $\mathcal{A}$  to indicate the probability level of the classification result. A higher value signifies a greater probability of correct classification by the classifier. Opting for a higher logit value indicates a preference for retaining training data that is highly likely to be correctly categorized. Conversely, opting to prioritize data with a lower probability suggests that we are preserving data that is more susceptible to misclassification.
- Mean clustering: mean clustering utilizes the mean of feature  $u$  to decide which data should be incorporated into the representative dataset. A smaller distance from the mean suggests that the chosen data is frequently found in the dataset.
- Barycenter: the concept behind this approach is similar to mean clustering, but it involves selecting samples whose  $u$  values are closest to their moving barycenter distance [30].

- Random selections: this method entails randomly selecting samples from the dataset currently in use.

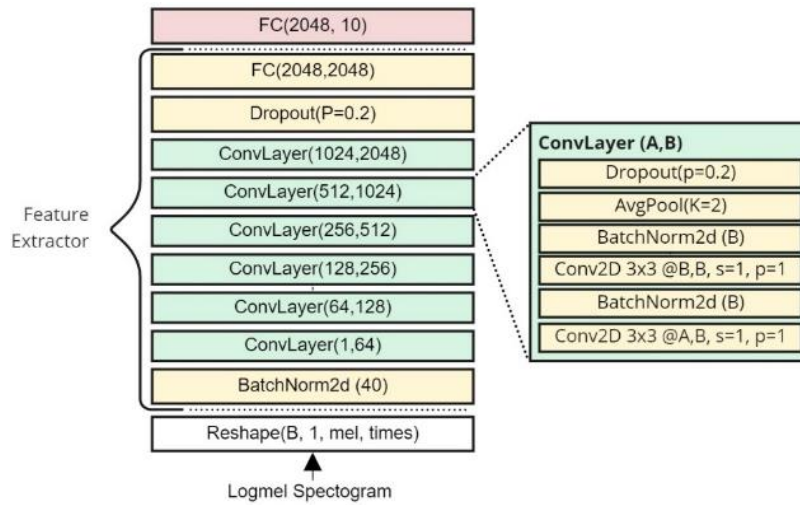


Figure 2. Audio scene classifier architecture

### 3.3. Pseudo-rehearsal data

Pseudo-rehearsal data refers to training data that includes artificially generated samples used in the model's retraining process. This research introduces two methods for generating pseudo-data: the first method involves augmenting existing data, and the second method employs generative techniques. To enhance our data, we utilize four audio augmentation techniques: Gaussian noise addition, TimeStretch, PitchShift, and Shift [36]. Incorporating Gaussian noise aids our model in adapting to slight variations and noise interference. This noise, introduced randomly, is determined by the transformation probability and is influenced by the amplitude factor (ranging from 0.2 to 0.7). TimeStretch enables us to modify the duration of audio without affecting its tonal quality, allowing us to make it faster or slower as required. PitchShift alters the pitch while preserving the tempo, achieved through time stretching and resampling. However, it's important to acknowledge that phase vocoding can occasionally impact audio quality by modifying certain aspects of the sound. Lastly, shift entails shifting audio samples forward or backward, creating a smooth transition. At the end, when using augmentation, the dataset  $\mathcal{D}_{rehearsal}^t$  contain  $\mathcal{D}_{task}^t$ ,  $\mathcal{D}_{representative}^{t-1}$  and  $\mathcal{D}_{augmentation}^{t-1}$ .  $\mathcal{D}_{augmentation}^{t-1}$  is the augmentation result of  $\mathcal{D}_{representative}^{t-1}$ .

In the generative approach, we employ (MER-GANs) [7] generator  $G^t$  to generate dataset  $\mathcal{D}_{gan}^t$ . GANs, which map low-dimensional latent spaces to the intricate distributions of samples, have gained prominence for their data generation capabilities. Consisting of two networks, Generators and Discriminators, GANs function within a zero-sum game framework, where each network's goal is to surpass the other.

The MER-GAN is composed of three primary components: a generator  $G$ , a discriminator  $D$ , and a classifier  $C$ . The discriminator and classifier share nearly the same network, with the exception of the final layer, which is tailored to the task (referred to as the task-specific layer). The generator uses a set of parameters,  $\theta^G$ , to create sample  $\tilde{x} = G_{\theta^G}(z, c)$  when a latent vector  $z$  and a class  $c$  given. The discriminator, with its parameters,  $\theta^D$ , attempts to discern whether the samples are genuine or generated by the improving generator, which is becoming more proficient at producing realistic samples. Besides, there's a classifier with parameters,  $\theta^C$ , that assigns a class to the samples. This aids the generator in producing samples that closely resemble those belonging to the correct class.

In this paper, to train the MER-GAN model, we use a rehearsal dataset,  $\mathcal{D}_{rehearsal}$ , that contain task dataset  $\mathcal{D}_{task}$ , representative dataset,  $\mathcal{D}_{representative}$  and generated dataset from augmentation or generation by  $G$ ,  $\mathcal{D}_{aug}$  or  $\mathcal{D}_{gan}$ . Then we employ a joint-retraining approach to train  $G$ ,  $D$  and  $C$ . The generator ( $G$ ) actively reproduces data from previous tasks through generative sampling and is employed in the learning process of the current task to prevent forgetting. The depiction of the retraining process using GAN can be found in Figure 3.

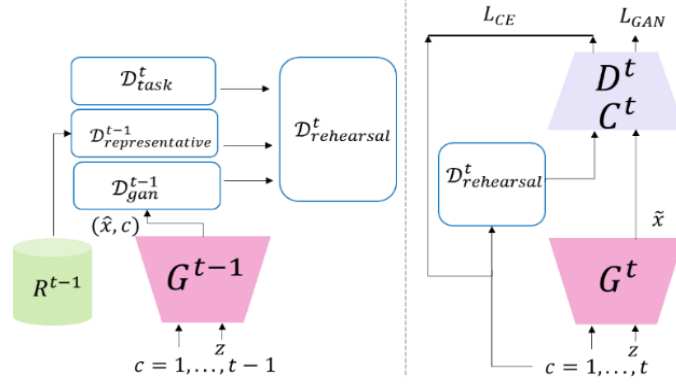


Figure 3. MER-GAN joint training architecture

Firstly, we generate dataset  $\mathcal{D}_{gan}^{t-1}$  contain generated sample from all previous tasks from task 0 to  $t-1$ . Then we combine  $\mathcal{D}_{gan}^{t-1}$ ,  $\mathcal{D}_{rep}^{t-1}$ , and  $\mathcal{D}_{task}^t$  as retrain dataset  $\mathcal{D}_{rehearsal}^t$ . Once the rehearsal dataset has been formed, the network is trained through joint training:

$$\min_{\theta_t^G} (L_{GAN}^G(\theta_t, \mathcal{D}_{rehearsal}^t) + \lambda_{CLS} L_{CLS}^G(\theta_t, \mathcal{D}_{rehearsal}^t)) \quad (1)$$

$$\min_{\theta_t^D} (L_{GAN}^D(\theta_t, \mathcal{D}_{rehearsal}^t) + \lambda_{CLS} L_{CLS}^D(\theta_t, \mathcal{D}_{rehearsal}^t)) \quad (2)$$

$$L_{GAN}^G(\theta_t, \mathcal{D}_{rehearsal}^t) = -\mathbb{E}_{z \sim p_z, c \sim p_c} [D_{\theta^D}(G_{\theta^G}(z, c))] \quad (3)$$

$$L_{CLS}^G(\theta_t, \mathcal{D}_{rehearsal}^t) = -\mathbb{E}_{z \sim p_z, c \sim p_c} [y_c \log C_{\theta^G}(G_{\theta^G}(z, c))] \quad (4)$$

$$L_{GAN}^D(\theta_t, \mathcal{D}_{rehearsal}^t) = -\mathbb{E}_{(x,c) \sim S} [D_{\theta^D}(x)] + \mathbb{E}_{z \sim p_z, c \sim p_c} [D_{\theta^D}(G_{\theta^G}(z, c))] + \lambda_{GP} \mathbb{E}_{x \sim S, z \sim p_z, c \sim p_c, \epsilon \sim p_\epsilon} [(\|\nabla D_{\theta^D}(\epsilon x + (1-\epsilon)G_{\theta^G}(z, c))\| - 1)^2] \quad (5)$$

$$L_{CLS}^D(\theta_t, \mathcal{D}_{rehearsal}^t) = -\mathbb{E}_{(x,c) \sim S} [C_{\theta^D}(G_{\theta^G}(z, c))] \quad (6)$$

The MER-GAN loss employs the WGAN formulation with gradient penalty, denoted as  $L_{GAN}^G(\theta_t, \mathcal{D}_{rehearsal}^t)$  and  $L_{CLS}^G(\theta_t, \mathcal{D}_{rehearsal}^t)$  are loss for generator and the cross entropy loss for classification, respectively,  $p_c = U(1, t)$ ,  $p_z = N(0,1)$  are the sampling distributions (uniform and Gaussian, respectively),  $y_c$  is the one-hot encoding of  $c$  for computing the cross-entropy,  $\epsilon$  are parameters of the gradient penalty term sampled as  $p_\epsilon = U(0,1)$  and the last term of  $L_{GAN}^D$  is the gradient penalty.

#### 4. EXPERIMENT SETUP

In this experiment, we divide the research into three stages:

##### 4.1. Feature extraction

We utilize the TAU urban acoustic scenes 2019 dataset [37] to evaluate our approach. This dataset is a collection of audio recordings captured in various urban environments. These recordings include sounds commonly heard in cities, such as traffic noise, people talking, and street sounds. The dataset is organized into different acoustic scenes, allowing researchers to study and analyze the acoustic characteristics of urban environments.

The initial phase in both the training and incremental processes involves feature extraction. In this study, we employ normalized mel-frequency cepstral coefficients (MFCCs) to represent the short-term power spectrum of audio in the Mel scale frequency domain. MFCCs are widely utilized as features in tasks related to audio processing and speech recognition. Initially, pre-emphasis is applied to amplify the energy content in high frequencies. Following this, the signal is windowed, and fast Fourier transformation is performed to convert the sample from the time domain to the frequency domain. The resulting frequencies are then mapped onto a Mel scale, and inverse discrete cosine transform (DCT) is applied. Finally, each MFCC undergoes normalization using mean and variance normalization techniques.

## 4.2. Dataset splitting for experiment scenario

The dataset contains 10 classes, which are further divided into 5 tasks, with each task encompassing two distinct classes. Each task consists of two distinct classes. In our experiment, we assessed the efficacy of integrating pseudo-data with GAN in comparison to using GAN alone, employing representative datasets of various sizes denoted as  $\mathcal{D}_{mem}$ . In total, we conducted nine distinct experimental scenarios, which included GAN alone (GAN-Alone), Small representative data combined with GAN, small representative data with augmentation (SmallRep+AUG), Medium representative data combined with GAN, and Medium representative data with augmentation. Additionally, we examined Large representative data combined with GAN and large representative data with augmentation (LargeRep+AUG). Table 1 provides an overview of the experimental scenarios and their respective data size configurations.

Table 1. Experiment scenario

Experiment scenario	$\mathcal{D}_{representative}$	$\mathcal{D}_{gan}$	$\mathcal{D}_{aug}$
GAN-alone	-	100%	-
Small representation data with GAN	10%	90%	-
Small representation data with augmentation	10%	-	90%
Medium representation data with GAN	50%	50%	-
Medium representation data with augmentation	50%	-	50%
Large representation data with GAN	75%	25%	-
Large representation data with augmentation	75%	-	25%

## 4.3. Model training and evaluation

We utilize average accuracy and backward transfer (BWT) as metrics to evaluate our proposed approach. BWT is commonly computed in the context of incremental or continual learning to gauge the extent to which knowledge acquired from previous tasks either endures or diminishes following the learning of the subsequent task. BWT is defined as the alteration in performance on earlier tasks after the model has been trained on the next task. A positive BWT value signifies that learning the subsequent task has a beneficial impact on the preceding tasks, whereas a negative value denotes “catastrophic forgetting” where learning the new task leads to a decline in performance on the earlier tasks. In (7) delineates the computing average accuracy, while (8) furnishes for BWT. In these equations,  $T$  represents the total number of tasks, and  $Acc_{i,i}$  is the test accuracy score for task  $j$  after the model learned task  $i$ .

$$ACC = \frac{1}{N} \sum_{i=1}^N Acc_{N,i} \quad (7)$$

$$BWT = \frac{1}{N-1} \sum_{i=1}^{N-1} Acc_{N,i} - Acc_{i,i} \quad (8)$$

In all of our experiments, the training process consists of 300 epochs for the classifier and 500 epochs for the GAN. We employ the Adam optimizer with a learning rate of 1e-4 for both the classifier and the GAN.

## 5. RESULT AND DISCUSSION

Using GANs without representative data can initially help reduce catastrophic forgetting. However, this approach faces challenges in maintaining its effectiveness over time as the number of tasks increases. The average accuracy achieved in this experiment is 0.733. Nevertheless, this accuracy consistently decreases with each subsequent task. At the outset, the model demonstrates a notable high accuracy of 0.9852 in the initial task, but this figure declines to 0.7947 in the second task when GAN-generated data is introduced. Subsequently, the accuracy continues to deteriorate, reaching 0.5863 by the fifth task. However, it is important to highlight that a significant decrease in accuracy is observed when the training data transitions to GAN-generated data. As illustrated in Figure 4, the model exhibits robust performance when dealing with newly introduced classes trained using real data. For example, on the third task, the accuracy for the newly introduced classes reaches 0.8405, while for the previously established classes, it achieves lower accuracies of 0.6108 and 0.5387.

While GANs possess the capability to generate diverse synthetic data and mitigate catastrophic forgetting, their effectiveness in capturing novel patterns and class variations is constrained. When we advance to the subsequent incremental stage, where the dataset  $\mathcal{D}_{retrain}$ , consists only of a combination of new task data  $\mathcal{D}_{task}$  and data generated by the GAN generator, the potential for mode collapse becomes more pronounced. Consequently, GANs tend to generate samples with lower diversity. This constraint

diminishes the New Generator's ability to preserve previously acquired knowledge while incorporating new information. In simpler terms, the model's capacity to learn and represent new data becomes less effective during the subsequent learning stage.

N	1	2	3	4	5	Average
1	0.9852					0.9852
2	0.8499	0.7396				0.79475
3	0.6108	0.5387	0.8409			0.6634667
4	0.5881	0.5523	0.6644	0.7941		0.649725
5	0.5236	0.4901	0.6238	0.5872	0.7068	0.5863

Figure 4. Model performance using GAN only

The use of  $\mathcal{D}_{rehearsal}$ , which is a combination of  $\mathcal{D}_{representative}$  with synthetic data generated through augmentation processes ( $\mathcal{D}_{aug}$ ) a GAN generator ( $\mathcal{D}_{gan}$ ), has been demonstrated that using GANs is more effective in mitigating the issue of catastrophic forgetting when compared to utilizing data solely from a GAN generator. Table 2 shows that the use of GANs consistently yields superior results in all types of experiments compared to data augmentation.

Table 2. Experiment results using  $\mathcal{D}_{representative}$

Data representative selection method	Task id	Augmentation data size			GAN data size		
		25%	50%	90%	25%	50%	90%
High probability	1	0.9863	0.9863	0.9908	0.9963	0.9963	0.9713
	2	0.7884	0.6214	0.5766	0.8250	0.8034	0.7313
	3	0.7326	0.5970	0.5904	0.7730	0.7449	0.7162
	4	0.6682	0.6710	0.5562	0.7153	0.7199	0.6676
	5	0.6739	0.5898	0.5560	0.7627	0.7396	0.7333
	Avg	0.7699	0.6931	0.6540	0.8145	0.8008	0.7639
Low probability	1	0.9863	0.9876	0.9908	0.9926	0.9890	0.9675
	2	0.6905	0.5767	0.5730	0.8122	0.7979	0.8368
	3	0.7034	0.6298	0.6094	0.8145	0.8137	0.7168
	4	0.6447	0.6061	0.5069	0.8320	0.8100	0.7005
	5	0.6648	0.6173	0.4923	0.8039	0.8006	0.6811
	Avg	0.7379	0.6835	0.6345	0.8510	0.8422	0.7805
Random	1	0.9897	0.9823	0.9815	0.9908	0.9907	0.9809
	2	0.6671	0.7064	0.7491	0.8618	0.8188	0.7117
	3	0.7219	0.6516	0.5217	0.8229	0.8049	0.7227
	4	0.7131	0.6875	0.5537	0.8019	0.7483	0.6980
	5	0.6837	0.6355	0.4899	0.8134	0.7335	0.6616
	Avg	0.7551	0.7327	0.6592	0.8581	0.8192	0.7550
Barry centre	1	0.9897	0.9825	0.9889	0.9945	0.9925	0.9790
	2	0.6990	0.7479	0.6854	0.8589	0.8470	0.7820
	3	0.7426	0.5788	0.6444	0.8375	0.8275	0.7605
	4	0.7467	0.6426	0.5815	0.7910	0.7580	0.6421
	5	0.6897	0.6050	0.5879	0.8019	0.7681	0.6168
	Avg	0.7735	0.7114	0.6976	0.8567	0.8386	0.7561
Mean cluster	1	0.9932	0.9987	0.9760	0.9945	0.9943	0.9771
	2	0.8073	0.7018	0.6877	0.7721	0.7432	0.7514
	3	0.7036	0.6438	0.6273	0.7965	0.7641	0.7662
	4	0.7084	0.5988	0.5825	0.7969	0.7107	0.6869
	5	0.6491	0.6225	0.5984	0.7834	0.6998	0.6682
	Avg	0.7723	0.7131	0.6944	0.8287	0.7824	0.7700

The use of augmentation data produced good results at large representative data size (75%  $\mathcal{D}_{representative}$  and 25%  $\mathcal{D}_{aug}$ ) and medium representative data size (50%  $\mathcal{D}_{representative}$  and 50%  $\mathcal{D}_{aug}$ ). However, when the amount of augmentation data was reduced to low representative data size (10%  $\mathcal{D}_{representative}$  and 90%  $\mathcal{D}_{aug}$ ), there was a significant decrease in accuracy. This phenomenon is attributed to overfitting, which is induced by the noise present in the augmented data, rendering it ineffective in enhancing the model's performance. Figure 5 illustrates the performance of the model across different sizes of representative data using various data selection methods. For large and medium representative data, as



shown in Figures 5(a) and (b), respectively, the random and Barry Centre selection methods yielded superior results. The random method, which selects samples without considering classification probabilities or the broader distribution of representative data, provides adequate diversification for generalization purposes. Meanwhile, the Barry Centre method selects samples based on their proximity to the cluster center, utilizing both correctly and incorrectly classified samples to generate new centers that better represent existing and future data. Conversely, low probability selection methods, where most selected samples are misclassified, exhibit poor performance across different data sizes. In Figure 5(c), the use of small representative data shows that the results of all methods vary significantly. Some methods even have accuracies similar to those without using representative data (GAN only). This indicates that selection methods do not have a substantial impact when the data size is small, and their influence is not significant.

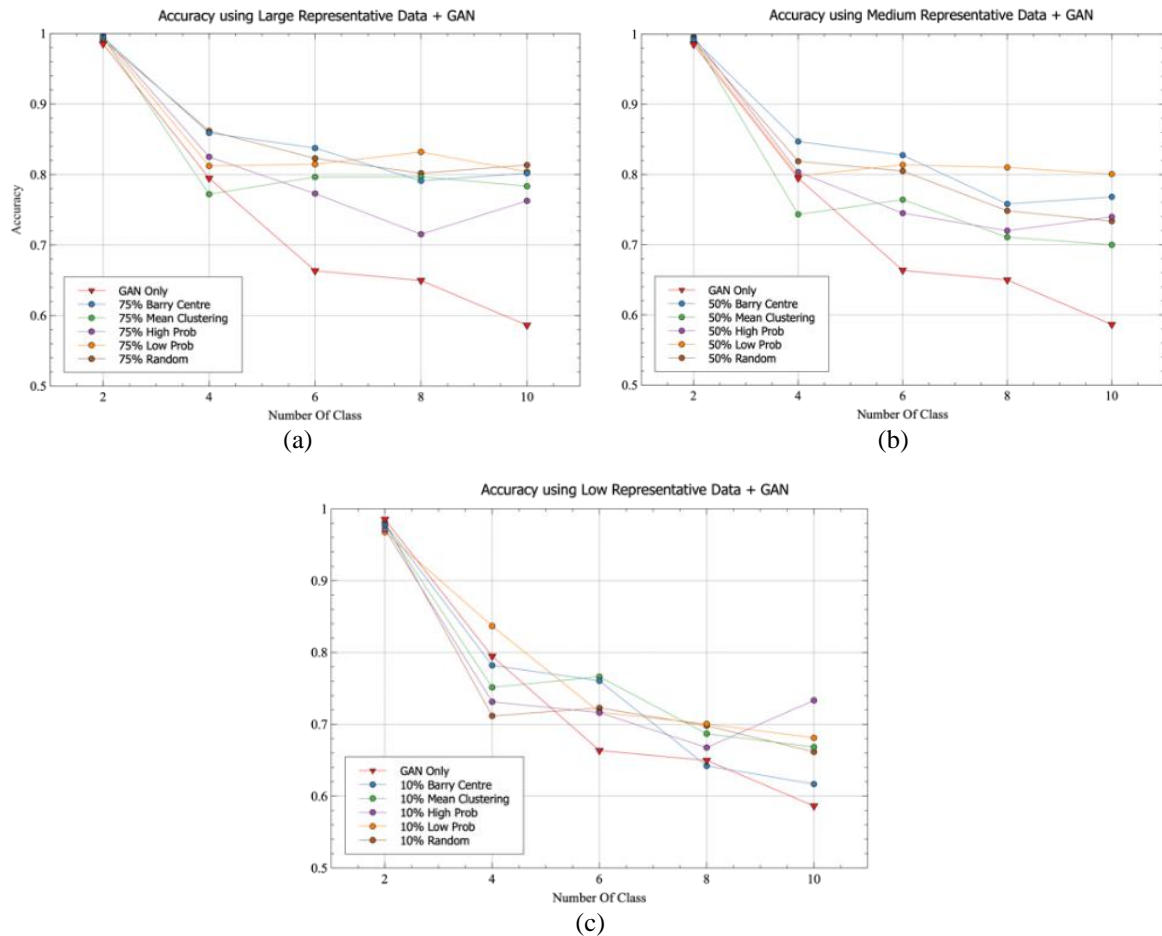


Figure 5. The performance of model; (a) large representative data and GAN, (b) medium representative data and GAN, and (c) small representative data and GAN

In experiments involving GAN data ( $\mathcal{D}_{gan}$ ), a reduction in accuracy was observed as the size of the representation data decreased. However, there was no significant decrease between large representative data (75%  $\mathcal{D}_{representative}$  and 25%  $\mathcal{D}_{gan}$ ) and small representative data (10%  $\mathcal{D}_{representative}$  and 90%  $\mathcal{D}_{gan}$ ) in all selection methods. The low probability and mean cluster selection methods exhibited greater stability as the size of  $\mathcal{D}_{gan}$  increased to 90%. This stands in contrast to the other methods, which experienced a steep decline at the 90% level. Meanwhile, in terms of achieving optimal performance, the high probability and random selection method displayed a significant improvement when transitioning from a 25% to a 50%  $\mathcal{D}_{gan}$  size, highlighting the positive impact of increasing the GAN data size. Nonetheless, both methods encounter a decline in performance at the 90% level, suggesting a saturation point in the incorporation of  $\mathcal{D}_{gan}$ . In terms of consistency, the barry centre method showed relatively good consistency, with a smaller drop in accuracy compared to the other methods when moving from 50% to 90%  $\mathcal{D}_{gan}$  size. Finally, in the context of resistance to overfitting, mean cluster and low probability exhibit greater resilience to overfitting

as larger  $\mathcal{D}_{gan}$ , datasets are incorporated, as evidenced by the more gradual changes in accuracy across various  $\mathcal{D}_{gan}$  sizes.

Figure 6 show the detail performance in random selection method using both GAN and data Augmentation. In Figures 6(a), (c), and (e) offer a comprehensive visualization of the model’s accuracy for each tested task using data augmentation using random selection. Figure 6(a) demonstrates that there is no significant difference in accuracy between the old and new tasks, suggesting that the model possesses strong generalization capabilities. Nevertheless, in Figures 6(c) and (e), a noticeable trend emerges: the model consistently exhibits superior performance on new tasks. For instance, in Figure 6(e), particularly for the third task (with  $N=3$ ), the accuracy for the test data in the first task is 0.3133, for the second task is 0.5236, while for the third task ( $N=3$ ) it reaches 0.7281. This shows that the high accuracy only exists for the new tasks that contain real data ( $\mathcal{D}_{task}$ ). However, an intriguing and noteworthy observation arises: when we examine the second task ( $N=2$ ), the accuracy for the test data specific to that task was previously 0.8309. This suggests a notable decline in accuracy when the model encounters the subsequent task, which could potentially exemplify an issue known as “catastrophic forgetting,” wherein the model loses the capability to execute the prior task after undergoing training for a new task.

Furthermore, in Figures 6(b), (d), and (f), it is evident that there is an enhancement in performance for several specific tasks. As an example, in Figure 6(f), pertaining to the 5th task ( $N=5$ ), a notable performance improvement is observed in the test data for the 3rd task, increasing from 0.7179 to 0.7722. This phenomenon illustrates the performance improvement that takes place in successive tasks, even with a relatively modest amount of representation data (10%). Additionally, on datasets with larger representations (Figures 6(b) and (d)), it becomes apparent that this enhancement occurs more frequently, and the decline in performance is less pronounced.

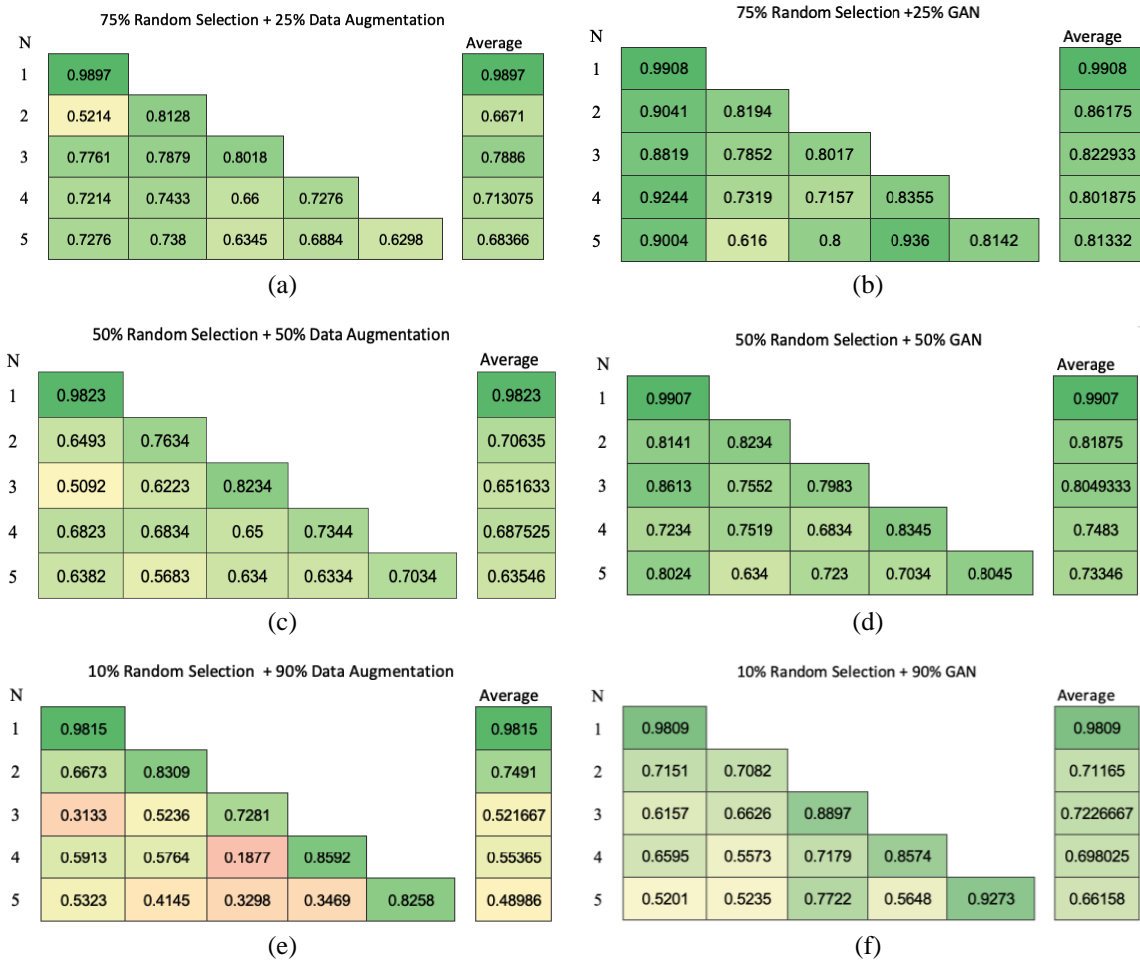


Figure 6. Detail accuracy of the random selection method in every task (N); (a) large representative data and augmentation, (b) large representative data and GAN, (c) medium representative data and augmentation, (d) medium representative data and GAN, (e) low representative data and augmentation, and (f) low representative data and GAN

### 5.1. Positive backward transfer result

In this experiment, the backward transfer in tasks 1 and 2 is consistently 0 because only  $\mathcal{D}_{task}$  so is used as data in the first two tasks, resulting in no improvement. Table 3 show the detail result of positive BWT for  $\mathcal{D}_{representative}$  and  $\mathcal{D}_{gan}$ . Improvement is only observed in task 3 and subsequent tasks. Regarding backward transfer, low probability and mean cluster methods seem to be the most promising, particularly when the size of the representative data is medium or small. These two methods appear to be more effective in leveraging the knowledge acquired from new tasks to enhance the model's comprehension of previous tasks. High probability tends to be less effective in achieving this objective, whereas random and barry centre exhibit more diverse and context-dependent outcomes.

Table 3. Positive BWT result

Data size	Task	Data representative selection method				
		High probability	Low probability	Random	Barry centre	Mean cluster
Large representative (75% $\mathcal{D}_{representative}$ and 25% $\mathcal{D}_{gan}$ )	1	0	0	0	0	0
	2	0	0	0	0	0
	3	0.0637	0.0295	0	0.0314	0.0513
	4	0	0.0184	0.0142	0	0.0068
	5	0.0538	0	0.0281	0.0451	0.0018
	Total	0.1175	0.0479	0.0423	0.0765	0.0598
Medium representative (50% $\mathcal{D}_{representative}$ and 50% $\mathcal{D}_{gan}$ )	1	0	0	0	0	0
	2	0	0	0	0	0
	3	0.0215	0.0419	0.0236	0.0295	0.0513
	4	0.0161	0.0181	0	0	0.0048
	5	0.0261	0	0.0395	0.0376	0.0147
	Total	0.0637	0.0599	0.0631	0.0671	0.0708
Small representative (10% $\mathcal{D}_{representative}$ and 90% $\mathcal{D}_{gan}$ )	1	0	0	0	0	0
	2	0	0	0	0	0
	3	0	0	0	0.0058	0.0157
	4	0.0048	0.0430	0.0146	0	0
	5	0.0353	0.0122	0.0181	0.0250	0.0285
	Total	0.0401	0.0553	0.0327	0.0307	0.0442

In terms of storage usage, there are variations between the original dataset, representative data, and the size of the GAN model. The original dataset requires approximately 3.9 gigabytes (GB) of audio data per class. For the large-scale, medium-scale, and small-scale representations, 73.6 megabytes (MB), 54.6 MB, and 16 MB per class are needed, respectively. Our generator, denoted as G, consistently consumes 82.85 MB of storage space for all classes. Consequently, when utilizing the generator, the storage size remains constant, even as the number of trained models increases over time.

## 6. CONCLUSION

The main objective of this study is to handle the catastrophic forgetting problem of data that changes over time. The observation from this study suggests that the integration of retraining data and GANs emerges as a promising solution, offering improved preservation of prior knowledge compared to relying solely on GANs. Furthermore, the superiority of certain data selection methods, such as the low probability and mean cluster methods, highlights the importance of robust strategies in handling increasing proportions of GAN-generated data.

Initially, the experimental results indicate that the utilization of GANs demonstrated initial effectiveness in mitigating catastrophic forgetting, with the initial model accuracy reaching 0.9852. Nevertheless, as additional tasks were incorporated, the efficacy of the GANs declined, resulting in an accuracy decrease to 0.5863 by the fifth task. Despite the capability of GANs to generate diverse synthetic data, the model encounters challenges in preserving previous knowledge when exposed to novel patterns and class variations. This issue is particularly pronounced when the training data becomes increasingly dominated by GAN-generated data.

To address this issue, the integration of retraining data, which comprises a blend of representative data and GAN-generated data, exhibited superior results and mitigated the impacts of catastrophic forgetting more effectively than relying solely on GANs. This hybrid approach appears to offer a more comprehensive training context that contributes to the preservation of prior knowledge. Moreover, when assessing the stability of different data selection methods amidst a rising proportion of GAN-generated data, the low probability and mean cluster methods exhibited superior performance. These methods demonstrate resilience

and consistency as they are capable of selecting more informative samples, thereby enhancing generalization. Lastly, an examination of storage efficiency highlights another advantage of GANs. Despite the growth in the number of classes and data representations, the storage requirement for the GAN model remains constant, rendering it an appealing choice for large-scale continuous learning applications.

## ACKNOWLEDGEMENTS

We gratefully acknowledge the funding and support provided by the DIPA of Universitas Riau, with Grant No. 8320/UN19.5.1.3/AL.04/2023, which enabled the successful completion of our research.




## REFERENCES

- [1] I. D. Id, M. Abe and S. Hara, "Concept Drift Adaptation for Acoustic Scene Classifier Based on Gaussian Mixture Model," *2020 IEEE REGION 10 CONFERENCE (TENCON)*, Osaka, Japan, pp. 450-455, 2020, doi: 10.1109/TENCON50793.2020.9293766.
- [2] M. McCloskey and N. J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem," *Psychology of Learning and Motivation*, vol. 24, pp. 109-165, 1989, doi: 10.1016/S0079-7421(08)60536-8.
- [3] J. Yoon, D. Madaan, E. Yang, and S. J. Hwang, "Online Coreset Selection for Rehearsal-based Continual Learning," in *International Conference on Learning Representations (ICLR)*, 2021, pp. 1-16.
- [4] A. Prabhu, P. H. S. Torr, and P. K. Dokania, "GDumb: A Simple Approach that Questions Our Progress in Continual Learning," - *ECCV 2020: 16th European Conference*, Glasgow, UK, 2020, vol. 12347, pp. 524-540, doi: 10.1007/978-3-030-58536-5\_31.
- [5] I. D. Id, M. Abe and S. Hara, "Incremental acoustic scene classification using rehearsal-based strategy," *2022 IEEE 11th Global Conference on Consumer Electronics (GCCE)*, Osaka, Japan, pp. 606-610, 2022, doi: 10.1109/GCCE56475.2022.10014339.
- [6] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual Learning with Deep Generative Replay," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, pp. 2994-3003, May 2017.
- [7] C. Wu, L. Herranz, X. Liu, Y. Wang, J. van de Weijer, and B. Raducanu, "Memory Replay GANs: learning to generate images from new categories without forgetting," in *32nd International Conference on Neural Information Processing*, pp. 1-12, Sep. 2018.
- [8] I. D. Id, M. Abe, and S. Hara, "Acoustic Scene Classifier Based on Gaussian Mixture Model in the Concept Drift Situation," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 5, pp. 167-176, Sep. 2021, doi: 10.25046/aj060519.
- [9] S. Chandrakala and S. L. Jayalakshmi, "Environmental Audio Scene and Sound Event Recognition for Autonomous Surveillance," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1-34, 2019, doi: 10.1145/3322240.
- [10] M. K. Jones, "Acoustic Scene Classification using Convolutional Neural Networks on Multivariate Audio," M.S. thesis, Universitat Pompeu Fabra, Barcelona, 2019.
- [11] T. Cerquitelli, S. Proto, F. Ventura, D. Apiletti, and E. Baralis, "Automating concept-drift detection by self-evaluating predictive model degradation," *arXiv*, 2019, doi: 10.48550/arXiv.1907.08120
- [12] M. U. Togbe, Y. Chabchoub, A. Boly, M. Barry, R. Chiky, and M. Bahri, "Anomalies detection using isolation in concept-drifting data streams," *Computers*, vol. 10, no. 1, pp. 1-21, 2021, doi: 10.3390/COMPUTERS10010013.
- [13] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Incremental Clustering for the Classification of Concept-Drifting Data Streams," [Online] Available: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7c736036ed1965351962ab001d444571048c2dd4>.
- [14] G. Ditzler, M. Roveri, C. Alippi and R. Polikar, "Learning in Nonstationary Environments: A Survey," in *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12-25, Nov. 2015, doi: 10.1109/MCI.2015.2471196.
- [15] R. Elwell and R. Polikar, "Incremental Learning of Concept Drift in Nonstationary Environments," in *IEEE Transactions on Neural Networks*, vol. 22, no. 10, pp. 1517-1531, Oct. 2011, doi: 10.1109/TNN.2011.2160459.
- [16] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory Aware Synapses: Learning what (not) to forget," *arXiv*, Oct. 2018, doi: 10.48550/arXiv.1711.09601.
- [17] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521-3526, Mar. 2017, doi: 10.1073/pnas.1611835114.
- [18] Z. Li and D. Hoiem, "Learning without Forgetting," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935-2947, 1 Dec. 2018, doi: 10.1109/TPAMI.2017.2773081.
- [19] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," *34th International Conference on Machine Learning, ICML 2017*, vol. 8, pp. 6072-6082, 2017.
- [20] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online Continual Learning in Image Classification: An Empirical Survey," *Neurocomputing*, Jan. 2021, doi: 10.1016/j.neucom.2021.10.021.
- [21] Y. Wu *et al.*, "Large Scale Incremental Learning," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 374-382.
- [22] A. A. Rusu *et al.*, "Progressive Neural Networks," *arXiv*, Jun. 2016, doi: 10.48550/arXiv.1606.04671.
- [23] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong Learning with Dynamically Expandable Networks," *ICLR 2018 Conference Track 6th International Conference on Learning Representations* Aug. 2017.
- [24] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong, "Learn to Grow: A Continual Structure Learning Framework for Overcoming Catastrophic Forgetting," *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:3925-3934, 2019.
- [25] N. Loo, S. Swaroop, and R. E. Turner, "Generalized Variational Continual Learning," *arXiv*, 2020, doi: 10.48550/arXiv.2011.12328.
- [26] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, "Embracing Change: Continual Learning in Deep Neural Networks," *Trends in Cognitive Sciences*, vol. 24, no. 12, pp. 1028-1040, Dec. 2020, doi: 10.1016/j.tics.2020.09.004.
- [27] M. De Lange *et al.*, "A Continual Learning Survey: Defying Forgetting in Classification Tasks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366-3385, 1 July 2022, doi: 10.1109/TPAMI.2021.3057446.
- [28] A. Mallya and S. Lazebnik, "PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 7765-7773, 2018, doi: 10.1109/CVPR.2018.00810.




- [29] Z. Wang, T. Jian, K. Chowdhury, Y. Wang, J. Dy and S. Ioannidis, "Learn-Prune-Share for Lifelong Learning," *2020 IEEE International Conference on Data Mining (ICDM)*, Sorrento, Italy, pp. 641-650, 2020, doi: 10.1109/ICDM50108.2020.00073.
- [30] S. A. Rebuffi, A. Kolesnikov, G. Sperl and C. H. Lampert, "iCaRL: Incremental Classifier and Representation Learning," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 5533-5542, doi: 10.1109/CVPR.2017.587.
- [31] D. Sculley *et al.*, "Machine Learning: The High-Interest Credit Card of Technical Debt," *NIPS 2014 Workshop on Software Engineering for Machine Learning (SE4ML)*, pp. 1-9, 2014, doi: 10.1007/s13398-014-0173-7.2.
- [32] A. Chaudhry *et al.*, "On Tiny Episodic Memories in Continual Learning," *arXiv*, Feb. 2019, doi: 10.48550/arXiv.1902.10486.
- [33] X. Liu *et al.*, "Generative Feature Replay for Class-Incremental Learning," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020, pp. 226-227.
- [34] L. M. Joseph, "Incremental Rehearsal: A Flashcard Drill Technique for Increasing Retention of Reading Words," *The Reading Teacher*, vol. 59, no. 8, pp. 803-807, May 2006, doi: 10.1598/RT.59.8.8.
- [35] D. Muñoz, C. Narváez, C. Cobos, M. Mendoza, and F. Herrera, "Incremental learning model inspired in Rehearsal for deep convolutional networks," *Knowledge-Based Systems*, vol. 208, pp. 1-21, Nov. 2020, doi: 10.1016/j.knosys.2020.106460.
- [36] D. S. Park *et al.*, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Interspeech 2019*, pp. 2613-2617, Sep. 2019, doi: 10.21437/Interspeech.2019-2680.
- [37] A. Mesaros, T. Heittola, and T. Virtanen, "TUT Acoustic scenes 2017, Development dataset," Mar. 2017, doi: 10.5281/ZENODO.400515.

## BIOGRAPHIES OF AUTHORS






**Ibnu Daqiqil ID**    is a faculty member in the Department of Computer Science at Universitas Riau. He holds a Ph.D. in Interdisciplinary Science and Engineering in Health System from Okayama University, a master's degree in Information Technology from the University of Indonesia, and a bachelor's degree in Computer Science from Brawijaya University. His academic interests lie in machine learning and signal processing, particularly in the areas of semi-supervised learning, and nonstationary learning. He can be contacted at email: [ibnu.daqiqil@lecturer.unri.ac.id](mailto:ibnu.daqiqil@lecturer.unri.ac.id).



**Masanobu Abe**    is a Professor in the Graduate School of Interdisciplinary Science and Engineering in Health Systems at Okayama University. Additionally, He serves as the Executive Director for Digital Transformation and Green Transformation, Senior Vice President at Okayama University in Japan. His research interests are primarily in the area of speech synthesis, ubiquitous computing, digital speech processing, audio signal processing, and human centered computing. He can be contacted at email: [abe-m@okayama-u.ac.jp](mailto:abe-m@okayama-u.ac.jp).



**Sunao Hara**    is an Associate Professor at the Faculty of Environmental, Life, Natural Science, and Technology at Okayama University. He holds B.E., M.E., and Ph.D. degrees from Nagoya University, Nagoya, Japan. His research interests include speech processing, particularly in spoken dialog systems, automatic speech recognition, and speech synthesis, as well as real-world audio analysis, such as acoustic event detection and acoustic scene classification. He can be contacted at email: [hara@okayama-u.ac.jp](mailto:hara@okayama-u.ac.jp).