IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 1, 2020, accepted November 18, 2020, date of publication November 20, 2020,
date of current version December 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3039624

Class-Incremental Learning With Deep Generative Feature Replay for DNA Methylation-Based Cancer Classification

ERDENEBILEG BATBAATAR[1], KWANG HO PARK[1], (Graduate Student Member, IEEE), TSATSRAL AMARBAYASGALAN[1], KHISHIGSUREN DAVAGDORJ[1], (Graduate Student Member, IEEE), LKHAGVADORJ MUNKHDALAI[1], (Student Member, IEEE), VAN-HUY PHAM[2], AND KEUN HO RYU[2], (Life Member, IEEE)

[1]Database/Bioinformatics Laboratory, School of Electrical and Computer Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea
[2]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam

Corresponding author: Keun Ho Ryu (khryu@tdtu.edu.vn)

This work was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT, and Future Planning, through the Basic Science Research Program, under Grant NRF-2019K2A9A2A06020672 and Grant 2020R1A2B5B02001717.

ABSTRACT Developing lifelong learning algorithms are mandatory for computational systems biology. Recently, many studies have shown how to extract biologically relevant information from high-dimensional data to understand the complexity of cancer by taking the benefit of deep learning (DL). Unfortunately, new cancer growing up into the hundred types that make systems difficult to classify them efficiently. In contrast, the current state-of-the-art continual learning (CL) methods are not designed for the dynamic characteristics of high-dimensional data. And data security and privacy are some of the main issues in the biomedical field. This article addresses three practical challenges for class-incremental learning (Class-IL) such as data privacy, high-dimensionality, and incremental learning problems. To solve this, we propose a novel continual learning approach, called Deep Generative Feature Replay (DGFR), for cancer classification tasks. DGFR consists of an incremental feature selection (IFS) and a scholar network (SN). IFS is used for selecting the most significant CpG sites from high-dimensional data. We investigate different dimensions to find an optimal number of selected CpG sites. SN employs a deep generative model for generating pseudo data without accessing past samples and a neural network classifier for predicting cancer types. We use a variational autoencoder (VAE), which has been successfully applied to this research field in previous works. All networks are sequentially trained on multiple tasks in the Class-IL setting. We evaluated the proposed method on the publicly available DNA methylation data. The experimental results show that the proposed DGFR achieves a significantly superior quality of cancer classification tasks with various state-of-the-art methods in terms of accuracy.

INDEX TERMS Computational biology, deep learning, class-incremental learning, continual learning, deep generative model, variational autoencoder, DNA methylation, cancer classification.

I. INTRODUCTION

The study of human cancers has demonstrated that cancer cells develop due to the accumulation of genetic and epigenetic alterations [1]–[3]. Epigenetics refers to mitotic changes in gene expression without modifications to the original DNA sequence. The most widely studied epigenetic mechanism in mammals is DNA methylation, where a methyl group is attached to cytosine in cytosine-phosphate-guanine (CpG) dinucleotide [4]–[6]. Microarray-based Illumina Infinium methylation assay [7] has commonly been used in epigenetics analysis due to its small sample requirement, good accuracy, high-throughput, and relatively low cost [8]. To estimate the methylation level, the Illumina Infinium assay uses a

The associate editor coordinating the review of this manuscript and approving it for publication was Nadeem Iqbal.

pair of probes to measure the intensities of methylated and unmethylated alleles at the interrogated CpG sites. Then the methylation level is estimated by measuring the intensities of this pair of probes, called beta-value, ranging from 0 to 1 [9]. The analysis of the DNA methylation level is a key ingredient in the development of cancer prognosis and personalized treatment approaches [10]–[12]. Therefore, the development of highly accurate statistical and computational techniques is required for further DNA methylation-based human cancer analysis.

In the domain of artificial intelligence (AI), the research field of machine learning (ML) has increasingly gained attention in various research fields including bioinformatics and computational biology [13]–[15]. More specifically, a critical research field within ML methods is DL that develops a biologically-inspired programming paradigm for manifold applications, such as in computer vision, natural language processing, audio recognition, speech recognition, social network filtering, bioinformatics, medical image analysis, material inspection, etc., [16]–[18]. DL technology can deliver findings in medicine comparable in some cases superior to human experts in the field of medical diagnosis of cancer and other diseases [19]. Most DL approaches for DNA methylation data focused on extracting biologically meaningful lower-dimensional, and estimating methylation status (imputation). As well as performing embeddings of CpG methylation states and classification and regression tasks [20]–[23]. Recent advances in DL, particularly unsupervised approaches, have shown promise for extracting biological knowledge through their application to genetic and epigenetic data [24]. An important advancement to DNA methylation-based DL analysis was the application of VAE [25]. It is a generative method that samples from the learned distribution of the methylation profiles to generate new data in a way that represents the original data without losing accuracy and complexity. By using these pre-trained generative models, researchers attempt to develop similar frameworks for feature extraction. That can be applied to downstream prediction tasks and identify biologically meaningful relationships revealed by VAE latent representation [26]–[31]. Although applications of DL networks to DNA methylation data have become ubiquitous, there still are challenging issues and a lack of practical methods.

Globally, there are more than 100 types of cancer, each has several subtypes [32], and an estimated 15 percent of all human cancers worldwide may be attributed to viruses [33]. New types of diseases have been increasing rapidly and their behaviors are unstable over time. For example, newly identified COVID-19 represents a significant portion of the global disease burden in 2020 [34]. Computational systems biology of cancer in the real world are exposed to continuous streams of patient information and thus are required to learn and remember multiple cancer and diseases from dynamic data distributions [35]. Therefore, developing CL, also referred to as lifelong learning [36]–[41] is highly needed in computational systems. Especially Class-IL [42]–[44], which consists of learning sets of classes incrementally, techniques for cancer classification tasks are discussed in this article.

Another practical issue in the medical system is data security and privacy [45]. CL remains a long-term challenge for DL models since the continuous accumulation of incrementally available information from non-stationary data distributions generally leads to catastrophic forgetting [46], i.e., training a new model with new information without losing previously learned knowledge. Catastrophic forgetting can be a critical issue for organizations that have to delete historic data for privacy reasons. For example, healthcare facilities might not be able to retain patient data permanently. Typical DL models require a large amount of data to learn parameters, which is a computationally expensive process, and it is needed to re-train a model repeatedly when new data comes. To train DL models efficiently without accessing past data, researchers attempted to use generative models such as VAEs or generative adversarial networks (GAN) [47] trained on past data [48]–[50]. Particularly, the deep generative replay [48] methods significantly improved the CL research in the past years by simply replaying all previous data using pre-trained generative models.

With the accumulation of high-dimensional low sample size data (HDLSS) in computational systems of real-world bioinformatics fields, Class-IL on these data is a critically important task. Traditionally, the dimension reduction or feature selection (FS) techniques are conducted as preprocessing before the classification tasks [51], [52]. After FS, some of the lower-dimensional features selected in the earlier tasks are not highly significant in the next tasks. Because some pairs of genes are common or specific for some cancer types [53], [54]. For example, assume that the highest significant 1,000 features are selected from breast and lung cancer. When other cancers come, whether some of the selected features are considered as significant or not. Most of the previous studies [48]–[50] handle a fixed number of features, e.g. image data, those are equally considered for all classes during Class-IL. Due to these unstable characteristics of the cancer data, IFS techniques are required to be developed. To our best knowledge, there are no studies conducted on incremental feature selection for high-dimensional data with deep generative models for Class-IL tasks.

To tackle the aforementioned issues, in this article, we propose a novel Class-IL method based on generative models, called DGFR, to incrementally select the most relevant features from high-dimensional DNA methylation data and then classify human cancer types when a new cancer type comes. The primary contributions of this article are highlighted as follows:

- We propose a novel DGFR method for high-dimensional DNA methylation cancer classification tasks in a Class-IL manner. DGFR is incrementally trained without accessing past data when new cancer types come.
- We also propose a novel IFS technique with deep generative replay. IFS is theoretically simple and memory efficient. It only stores mean and standard deviation (SD)

values for all features in memory and rank all features based on its variability.
- We introduce a soft replay, which is the updated version of the replay. IFS continuously updates the previously generated replays based on the newly selected features when new cancer types come. For past data, the duplicated features are kept, and not duplicated features are generated again from a normal distribution.
- Comparison of state-of-the-art continual learning methods on publicly available DNA methylation cancer datasets for class-incremental cancer classification tasks. Comprehensive experiments have demonstrated the superior quality of the proposed DGFR method. We explore the effect of the number of samples trained in different ways such as randomly, ascending, and descending orders. In real-world cases, that is important to consider the number of samples for HDLSS data. Experiments on the cancer datasets have fully demonstrated the effectiveness of the proposed DGFR method as it has significantly outperformed the baselines.

The remainder of this article is organized as follows. We first review the related works in Section II. In Section III, we formally describe the notations and explanations of the proposed DGFR method. We then describe the experimental settings in Section IV and show the experimental results, including discussions and analysis in Section V. Finally, we draw conclusions and future works in Section VI.

II. RELATED WORKS

In this section, we briefly summarize the recent research studies sequentially on FS from DNA methylation data, DNA methylation-based cancer classification, and continual lifelong learning.

A. FEATURE SELECTION

Discovering a lower number of CpG sites from high-dimensional DNA methylation data relevant to specific cancer disease could derive in more effective treatments. Selecting only a small number of CpG sites from a large number of sites strongly correlates with targeted cancer [55]. More studies suggested that only a small number of CpG sites can be sufficient markers for specific cancer [56], where the CpG sites' biological relationship concerning the target cancer can be easily identified. Generally, FS techniques could be very useful for HDLSS data problems [57] and the right FS strategy is crucially important for the classification performance [58]. There are many FS techniques; they can be divided into three categories such as filter, wrapper, and embedded; are different in the way each technique copes with a higher dimension to form a subset of features. Most of the DNA methylation-based cancer studies used variance-based filtering FS techniques to select the most variable CpG sites across several samples before performing VAE and classification algorithms [26]–[31]. The advantages of filter techniques are simple and fewer computations compared to the other two categories. The highly variable CpG sites are assumed to be biologically more meaningful than the lower variable sites. Filter techniques are performed in the selection model as a pre-processing step and can be followed by one or more classification algorithms.

B. CANCER CLASSIFICATION

Recently intensive studies of DNA methylation-based cancer analysis have been well conducted on effective training strategies for deep architectures, which are all based on an unsupervised pre-training followed by supervised fine-tuning. There is a lack of ground-truth labels in the bioinformatics domain. Therefore unsupervised DL approaches such as GAN and VAE harness the modeling power of DL without the need for accurate labels. Tybalt [26] was developed to extract biologically relevant information from cancer gene expression data with VAEs. The learned features were generally non-redundant and can reveal biologically meaningful relationships among subgroups of samples. Similar to this, an unsupervised DL framework with VAEs, applied to the DNA methylation data from three breast cancer datasets [27] and two lung cancer datasets [28], [29]. Those DNA methylation-based DL approaches have not been designed as user-friendly for execution, training, model interpretation. MethylNet [30] was developed to pre-train data, generate new data, make predictions, and discover unknown heterogeneity with minimal user supervision. However, public cancer data is rapidly increasing, there is also a lack of samples for specific cancer types in research. To alleviate this issue, methCancer-gen [31] was presented to generate a user-specified cancer type dataset by employing conditional VAE and a neural network-based generative model. It estimates the conditional probability distribution with latent variables and data and produces samples for specific cancer types.

C. CONTINUAL LIFELONG LEARNING

One of the main challenges of the computational systems, including computational systems biology, regarding continual lifelong learning is reducing catastrophic forgetting. There are numerous continual learning techniques available to handle this issue, and are distinguished into four types: regularization [39], [46], [59]–[63], dynamic architecture [60], [64], [65], rehearsal [39], [42], [66]–[74], and generative replay [48], [75]–[79]. Many approaches use combinations of these techniques to allow better performance and less computational and memory cost. Regularization defines a loss that constrains weight updates to remember past knowledge when retraining a model. In the Class-IL setting, regularization-based techniques are unable to learn the discrimination between tasks, and no regularization method can learn alone to discriminate classes from different tasks [80]. Dynamic architectures of neural networks, i.e. progressive networks, create new weights automatically when new classes come. New weights learn new tasks and old weights are frozen (not modified anymore) for keeping past information. Rehearsal strategy is another technique to mitigate catastrophic forgetting consisting of storing past

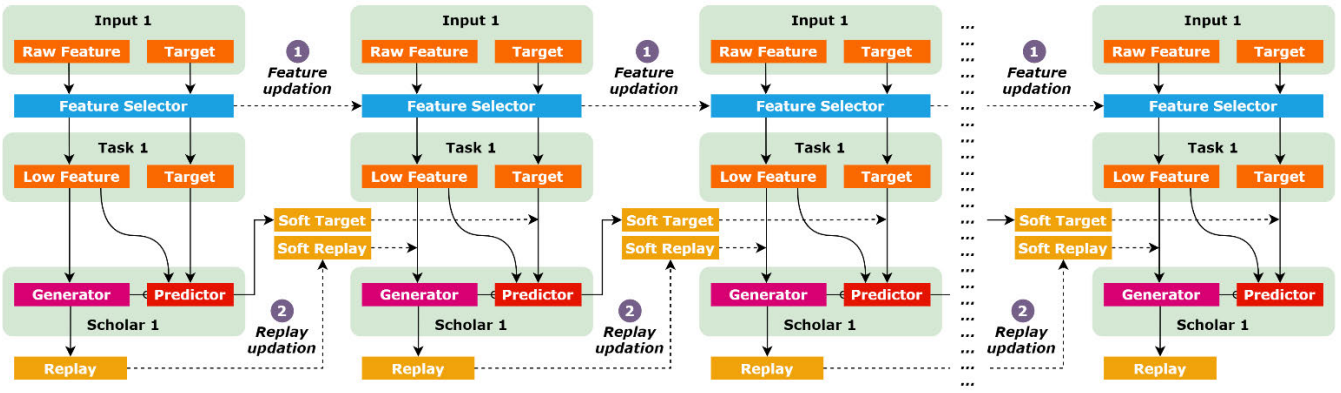

FIGURE 1. Overall structure of DGFR method.

samples and replaying them into the model while learning new information. Dynamic architecture and rehearsal techniques are effective techniques but require much memory while increasing the number of new tasks and classes. When past samples are not accessible that are common in the bioinformatics field, rehearsal techniques cannot be used anymore. Instead of storing past samples, generative replay techniques learn models that will produce artificial samples as a memory of previous knowledge.

III. PROPOSED METHOD

In this article, we propose a novel continual learning approach for high-dimensional DNA methylation data, called DGFR, which consists of a memory-efficient IFS and SN. SN uses a deep generative model as our generator and a neural network (NN) classifier as our predictor for cancer classification tasks in a Class-IL manner. IFS is used to select the most relevant features from high-dimensional features by considering all classes and SN is used to learn the distributions of the selected DNA methylation data and then predict cancer types. The overall structure of the DGFR method is shown in Figure 1.

For each task, high-dimensional DNA methylation samples (''High Feature'') and their corresponding labels (''Targets'') from new cancer types are fed into a task network of DGFR as inputs. Firstly, we perform a simple variance-based filtering FS technique (''Feature Selector'') to select the most variable CpG sites (''Low Feature'') across all samples. Secondly, we pre-train a generator network (''Generator'') to learn the distributions of inputs, and sample from it to produce pseudo-inputs. Thirdly, we train a classifier that fine-tunes the pre-trained generator network parameters, to classify cancer types on the selected features and their corresponding labels and produce pseudo-targets. When the training data for previous tasks are not accessible, pseudo-inputs (''Replay'') and pseudo-targets (''Soft-Target'') produced by a memory network can be replayed as inputs.

In practice, mostly no past information is available in bioinformatics because of their data security and privacy. For this reason and memory efficiency, we store selected feature values, and only means and standard deviations of non-selected features in each task. Here, all features are ranked by their variability. Non-selected feature values should be removed due to the above reasons. When new classes come, a mix of real inputs of new classes and pseudo-inputs of old classes are fed into a task network of DGFR. To replace the cancer-specific gene sets highly relevant to old classes with other gene sets highly relevant to new classes, we perform two updates on features (''Feature Updation'') and replay (''Replay Updation'') values, respectively. IFS generates normal distributions of new features based on their means and standard deviations if previous information is not available, and then it performs feature selection on all of the previous and new feature values incrementally. To match newly selected features, replay values are also replaced (''Soft Replay'') by their normal distributions if features are not selected.

A. NOTATIONS

Table 1 summarizes symbols and notations used in this article. Given $T = \{T_1, T_2, \ldots, T_K$ of K tasks and $D = \{D_1, D_2, \ldots, D_C$ of C datasets. For example, when $T = 6$ and $C = 12$, two datasets are considered in each task. A dataset is denoted by $D_i = \{\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_{N_i} \in \mathrm{R}^{N_i \times H_i}$, where X_{j_i} is an observation, N_i is the number of observations, and H_i is the number of high-dimensions of i-th dataset, respectively. Here, X_{j_i} consists of x_{j_i} set of instances and y_{j_i} set of labels.

In each task, we incrementally calculate and update the mean (μ) and standard deviation (σ) for each CpG, which are used for producing normal distribution (N). And so IFS selects the most relevant low-dimensional (L) features X^L from high-dimensional raw features based on their variability (σ^2). Deep generative models learn the distributions from the selected lower-dimensional data, and then fine-tuning the pre-trained model allows us to perform cancer type prediction. To achieve our goal, we perform IFS and SN networks sequentially, and they contain feature selector (Section III.B), generator (Section III.C.1), and predictor

TABLE 1. Symbols and notations.

Symbol	Description
T	Task
K	Number of tasks
D	Dataset
C	Number of datasets / classes / cancer types
$\mathcal{X}$	Data observations
N	Number of data observations
H	Number of high-dimensions
L	Number of low-dimensions
x	Data instances
y	Data labels
x'	Pseudo-input
y'	Pseudo-target
μ	Mean of CpG
σ	Standard deviation of CpG
σ^2	Variation of CpG
$\mathcal{N}$	Normal distribution
M^G	Pre-trained generative model
M^P	Predictive model
d	Dimension of latent representation
z	Latent representation of generator
f^{enc}	Encoder function
$q_\emptyset$	Approximate posterior of latent variable
$\emptyset$	Local variational parameter
f^{dec}	Decoder function
p_θ	Prior distribution of latent variable
θ	Local variational parameter
$\mathcal{L}_r$	Reconstruction loss function
D_{KL}	KL divergence of $q_\emptyset$ and p_θ
$ELBO$	Evidence (variational) lower-bound
f^{pred}	Predictor function
$\mathcal{L}_c$	Classification loss function

(Section III.C.2) functions, respectively. The algorithm of the DGFR method is explained in Table 2.

In the following sections, IFS and SN (generative and predictive models) networks are explained in detail sequentially.

B. INCREMENTAL FEATURE SELECTION

IFS is a simple variance-based filtering technique that is incrementally performed. High-dimensional data may contain a large amount of irrelevant and redundant information, which may use a lot of memories and greatly degrade the performance of learning algorithms. Therefore, we need to use flexible incremental feature selection techniques that can execute in a memory space efficiently that would be empty in the beginning and update features when new cancer types arrive. So, firstly, we calculate μ and σ incrementally and store only the calculated values instead of whole feature values. In the first task ($k = 0$), μ, and σ are calculated in Equations 1 and 2 as follows:

$$\mu_0 = \frac{1}{N_0}\sum_{i=1}^{N_0} x_i \tag{1}$$

TABLE 2. DGFR algorithm.

Algorithm 1. DGFR (K, D)

Input: Number of tasks K; Datasets D; Number of datasets C (length of D); Here, each dataset contains data samples $\mathcal{X}$ (x and y)
Output: Predicted cancer types y' for each dataset

(* *Perform each task sequentially.* *)
1. L = number of lower dimensions
2. **while** k **in** K
3. Calculate μ of all features
4. Calculate σ of all features
5. (* *Update previous μ and σ if they are calculated before and stored on memory network.* *)
6. Rank all features based on their σ^2
7. x^L = *Selector* (x, L)
8. Produce $\mathcal{N}$ of replayed class samples $\mathcal{X}'^L$ (* *If replayed data is available and feature values are not replayed.* *)
9. Mix new class samples $\mathcal{X}^L$ and replayed class samples $\mathcal{X}'^L$ (* *If replayed data is available.* *)
10. M^G, x'^L = *Generator* (x^L)
11. y' = *Predictor* (M^G, $\mathcal{X}^L$)
12. **endwhile**

Selector (x, L) (* *Select given number of highest variance features.* *)
13. Select top L features based on their σ^2
14. **return** x^L (* *End of Selector.* *)

Generator (x^L) (* *Pre-train a generative model and generate new samples.* *)
15. Pre-train a generative model M^G
16. Generate new samples x'^L from the distributions of x^L
17. **return** M^G, x'^L (* *End of Generator.* *)

Predictor (M^G, $\mathcal{X}^L$) (* *Train a predictive model and predict cancer types.* *)
18. Train a prediction model M^P by fine-tuning M^G
19. Predict cancer types y' for all samples x'^L
20. **return** y' (* *End of Predictor.* *)

$$\sigma_0 = \sqrt{\frac{1}{N_0}\sum_{i=1}^{N_0}(x_i - \mu_0)^2} \tag{2}$$

At k-th task, μ and σ are incrementally calculated without accessing past data in Equations 3 and 4 as follows:

$$\mu_k = \frac{1}{2}\mu_{k-1} + \frac{1}{2} \times \frac{1}{N_k}\sum_{i=1}^{N_k} x_i \tag{3}$$

$$\sigma_k = \sigma_{k-1} + (x_k - \mu_{k-1}) \times (x_k - \mu_k) \tag{4}$$

As shown in Figure 2, we rank all features based on their σ^2 after the calculations. In DNA methylation analysis, the overall variance of methylation across the samples can

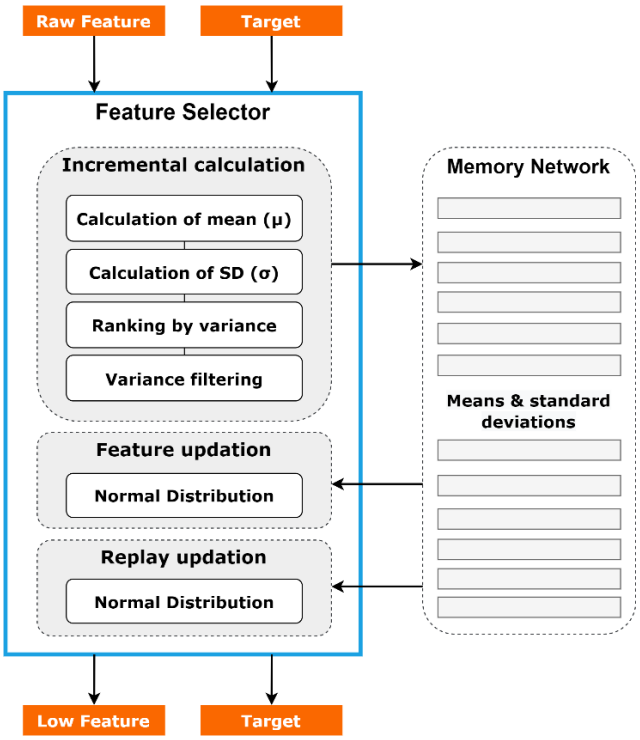

FIGURE 2. Structure of IFS.

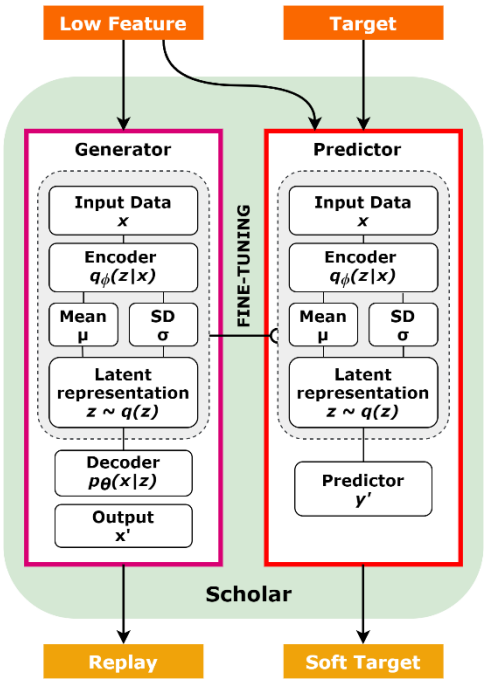

FIGURE 3. Structure of SN.

be an attractive covariate for filtering. Filtering techniques are commonly used for reducing noise for DNA methylation data with a linear time requirement and are very computationally intensive, especially if building learning models have a high computational cost. But filtering techniques rank the features by only single-feature associations with the class labels, and the number of top-ranked features is determined manually (L).

After selecting top-ranked L features, we mix new class samples into the old class samples that are replayed from previous tasks, called *hard replay*. If newly selected feature values are not replayed, we produce its normal distribution (N) using its μ and σ, called *soft replay*. For all tasks, N is calculated in Equation 5 as follows:

$$\mathcal{N}(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} exp[\frac{-\frac{1}{2}(x-\mu)^2}{\sigma^2}] \tag{5}$$

Then we perform a generator function to learn data samples for each cancer type and a predictor function to predict cancer types, sequentially. In the next sections, we explain a scholar network that can both learn the new cancer types without forgetting its knowledge.

C. SCHOLAR NETWORK

In the scholar network, introduced in [48], the generator-predictor pair learns the selected low-dimensional features and their corresponding target values, then produces the pseudo-input (replay) and the pseudo-target (soft target) pairs as shown in Figure 3. The produced pairs are mixed with new data samples to update the generator and the predictor networks. It contains a deep generative model (generator) and a NN classifier (predictor).

1) GENERATIVE MODEL

The generative model refers to any model that generates observable samples. In this article, we employ a VAE deep generative model that maximizes the likelihood of generated samples being given a real distribution. The architecture of the VAE consists of encoder and decoder components.

The encoder component comprised an input layer, fully connected encoding hidden layers, a distribution layer, and a latent space layer. Here the distribution layer produces μ and σ vectors. The latent space layer samples d-dimensional latent vectors, which are used as the extracted and learned features called latent variable (z). The encoder function (f^{enc}) can be summarized in Equation 6 as follows:

$$z = f^{enc}(x) \sim q_\emptyset(z|x) \tag{6}$$

where $q_\emptyset(z|x)$ is the approximate posterior of the latent variable z and $\emptyset$ is a local variational parameter.

The decoder component comprised fully connected decoding hidden layers, and an output layer. The output layer is used as the reconstructed input (x'). The decoder function (f^{dec}) can be summarized in Equation 7 as follows:

$$x' = f^{dec}(z) \sim p_\theta(x|z) \tag{7}$$

where $p_\theta(x|z)$ is the prior distribution of the latent variable z and θ is a local variational parameter.

The objective function of the VAE is to reconstruct the input data as much as possible, to maximize the

log-likelihood probability $p_\theta(x)$, to minimize mean squared error between original data and reconstructed data. The objective function of the generator network (reconstruction loss) is summarized in Equation 8 as follows:

$$\begin{aligned}\mathcal{L}_r &= \log p_\theta(x) \\ &= D_{KL}(q_\emptyset(z\,|\,x)\,||p_\theta(x|z)) + ELBO(\emptyset, \theta; x) \end{aligned} \quad (8)$$

where D_{KL} is the KL divergence of the approximate posterior and the prior distribution and $ELBO$ is the variational lower-bound on the marginal likelihood of each data point. The VAE network learns the knowledge about DNA methylation data from the original low-dimensional inputs and tries to reconstruct them that can be replayed for further tasks.

2) PREDICTIVE MODEL

To establish a predictive model, we employ a simple NN classifier followed by the downstream of the generator which fine-tunes the generator network's encoder part and feature extraction layers in an end-to-end manner for the task of cancer type prediction. The predictor function (f^{pred}) can be summarized in Equation 9 as follows:

$$y' = f^{pred}\left(f^{enc}(x)\right) \quad (9)$$

The objective function of the NN classifier is to predict the true class labels, to minimize the cross-entropy loss between the approximate distribution and the ground truth distribution. The objective function of the predictor network (classification loss) is summarized as shown in Equation 10:

$$L_c = -\sum y \log y' \quad (10)$$

The supervised predictor network provides predictions of cancer types for the D datasets as any of the given C cancer types among the K tasks. The predicted targets can be used as soft targets for the further processing of the prediction of cancer types in each task of Class-IL.

IV. EXPERIMENTAL SETTINGS

In this section, firstly, we describe the experimental dataset used in this article. Then we briefly introduce the baseline methods compared with the proposed DGFR method and their hyperparameters. Finally, we show the metric used for evaluating all methods.

A. DATASETS

Our experiments are conducted on twelve (C = 12) publicly available datasets obtained from the Xenabrowser (TCGA) [81] data portal, which have a total of 2,728 samples listed in Table 3. Where twelve types of cancers such as ovarian cancer (OV), kidney clear cell carcinoma (KIRC), breast cancer (BRCA), glioblastoma (GBM), colon cancer (COAD), acute myeloid leukemia (LAML), lung squamous cell carcinoma (LUSC), lung adenocarcinoma (LUAD), stomach cancer (STAD), endometrial cancer (UCEC), rectal cancer (READ), and kidney papillary cell carcinoma (KIRP) are collected. For each cancer sample, we obtained 27,578 DNA methylation data. To reduce noise, the top L features were chosen by the variance-based feature selection algorithm from DNA methylation beta values (across all cancers) on each task repeatedly. The selected L features are used as input data and sent to the SN. We performed stratified 10-fold cross-validation for model evaluation. The mean and standard deviation values are reported in the experimental results.

TABLE 3. Experimental datasets.

Dataset	# of samples	Asc	Desc	Rand
OV	616	Task 6	Task 1	Task 5
KIRC	418			
BRCA	345	Task 5	Task 2	Task 1
GBM	288			
COAD	203	Task 4	Task 3	Task 4
LAML	194			
LUSC	160	Task 3	Task 4	Task 3
LUAD	151			
STAD	141	Task 2	Task 5	Task 6
UCEC	118			
READ	73	Task 1	Task 6	Task 2
KIRP	21			

We split the experimental datasets into six ($L = 6$) tasks, where each task has two cancer classification. For example, the first task is the binary classification for BRCA and COAD cancers and incrementally added the other five tasks. To explore the effect of the number of samples among tasks, three different dataset ordering strategies, such as randomly (Rand), ascending (Asc), and descending (Desc).

In this article, we analyzed the effect of the number of selected features, notated as L, which is set as {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000}. For the feature analysis and the performance evaluation, we selected the 1,000 features with the highest variance across the 2,728 experimental data samples. Figure 4 illustrates the global density of DNA methylation among 1,000 CpG sites for all twelve cancer types, each consists of the 1,000 features of cancer samples.

As shown in this figure, there are significant differences in the methylation levels that drive the classification to good accuracy results. Theoretically, DNA methylation can be divided into three levels: low (hypomethylation), medium, high (hypermethylation) [82]. In general, the density graph shows that hypermethylation and hypomethylation are more than medium methylation for all cancers. That means that most of the CpG sites in this region are hypomethylated and hypermethylated. The density of the hypomethylation is more than the density of the hypermethylation in KIRP, LUAD, LUSC, GBM, BRCA, KIRC, and OV cancer types. By contrast, hypermethylation is more than hypomethylation in other cancers. In some cancers such as STAD, the medium methylation is more than the hypomethylation and hypermethylation levels. However, the differences between some cancers are not showing clearly, for example in LUAD and LUSC cancer types. That makes it difficult for many ML and DL techniques to differentiate between them.

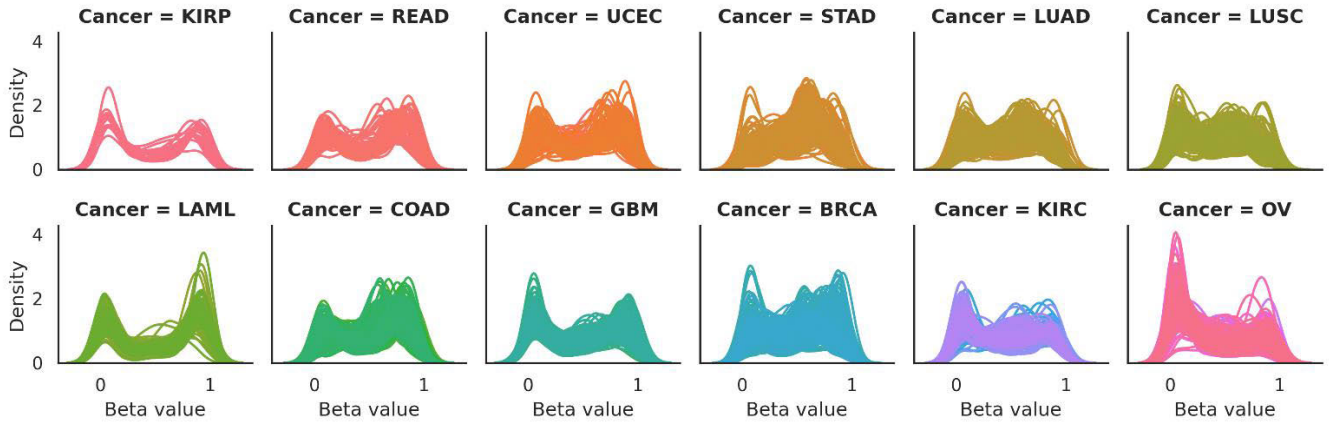

FIGURE 4. DNA methylation density.

B. BASELINE METHODS

We compared the proposed DGFR method with the state-of-the-art continual learning methods on the experimental datasets in terms of the classification accuracy. The baseline methods are divided into four categories as follows:

1. **Regularization**
   - **Elastic Weight Consolidation (EWC)** [46]: The regularization term consisting of a quadratic penalty term for each previously learned task. The number of quadratic terms grows linearly in the number of tasks.
   - **Online Elastic Weight Consolidation (Online EWC)**[62]: This method is a modification of EWC to determine weight importance by calculating the sum of the previous tasks' Fisher information matrices. Synaptic Intelligence (SI) [63]: This method is similar to online EWC to determine weight importance online during stochastic gradient descent instead of Fisher information.
2. **Dynamic architecture**
   - **Learning without Forgetting (LwF)** [60]: This is another type of regularization-based method focused on data that attempts to preserve past learning experiences from old models to a new one through knowledge distillation [59]. That means that dynamic architecture methods create new weights automatically for learning new tasks.
3. **Rehearsal**
   - **Averaged Gradient Episodic Memory (A-GEM)** [39]: This method is also another type of regularization method that uses an episodic memory. It replays the stored data as "exemplars" from given tasks to perform it computationally and memory-efficient as the regularization methods. A-GEM is an improved version of Gradient Episodic Memory (GEM) [83], by defining inequality constraints to avoid the increase in the losses.
   - **Incremental Classifier and Representation Learning (iCaRL)** [42]: This method uses an episodic memory to replay the stored data and calculate class means. Neural networks are used for feature extraction and classification is performed based on a nearest-class-mean rule [84] in that feature space.
   - **Experience Replay (ER)** [74]: A basic rehearsal method which uses an episodic memory to replay the stored data and uses them to augment the incoming data.
4. **Generative replay**
   - **Deep Generative Replay (DGR)** [48]: A dual-model architecture of a deep generative neural network model, which creates pseudo-samples that are then intermixed with recently observed data instead of using stored data. We also employed the dual-model architecture consisting of a deep generative model (generator) and a task solving model (solver).
   - **Deep Generative Replay with distillation (DGR+distill)** [78]: This method is similar to DGR, but instead of labeling the replayed inputs as the most likely class according to the previous tasks' model (hard targets), it pairs them with the predicted probabilities for all target classes (soft targets). We also employed the dual-model architecture with distillation.
   - **Replay-Through-Feedback (RtF)** [79]: The integrated architecture of the generative model and the main model with distillation by equipping it with generative feedback connections. It reduces the computational cost of generative replay.

For a fair comparison, we used the same neural network architecture for all the methods that have a multi-layer perceptron with three hidden layers of 1,000 nodes, each with ReLU non-linear activation functions. Except for iCaRL, we used a softmax function as the final output layer and the standard multi-class cross-entropy classification loss for the

predictions of the model on the current task data. All models are trained for 5,000 iterations (epochs) per task using the Adam-optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) [85] with learning rate 0.0001. For each iteration, classification loss is calculated as an average of over 64 samples (same for replayed samples) from the current task. For the generative models, symmetric VAE networks with 100-dimensional stochastic latent variable layers are pre-trained separately on all tasks. The standard normal distribution is calculated as prior. All the hyperparameters used in this article are summarized in Table 4.

TABLE 4. Hyperparameter setting.

Parameter	Value
Number of hidden layers	3
Number of nodes	1,000
Activation function	ReLU
Output layer (except for iCaRL)	softmax
Loss (except for iCaRL)	cross-entropy
Number of epochs	5,000
Optimizer	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Learning rate	0.0001
Batch size	64
Latent space dimension of VAE	100
Prior distribution of VAE	Normal

C. EVALUATION METRIC

To measure the performance of the proposed DGFR method and compare it with other baseline methods, we used classification accuracy. Accuracy is the ratio of the number of correctly predicted classes to the total number of tested samples and calculated directly from the confusion matrix, which is a specific table that is often used to describe the performance of a classification model. In the confusion matrix, true–positive (TP) and true–negative (TN) is interpreted to correct positive and negative predictions, which are actual correct predictions. False-positive (FP) and false-negative (FN) are incorrect positive and negative predictions. Accuracy is formalized in Equation 10 as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

All experiments are executed on the Intel Xeon E3 (32G memory, GTX 1080 Ti) hardware platform and the Ubuntu 18.04 computational environment. We thank the authors [86] for the great PyTorch implementations of all of the baseline methods. We used all default parameters except for not listed in Table 4. We also used the Scikit-Learn and Pytorch libraries with Python programming language for all of the analyses.

V. EXPERIMENTAL RESULTS

In this section, we illustrate some experimental results, including feature analysis that is selected by IFS, and performance evaluation that is performed by SN. We also investigate the effect of the number of selected features L to find the optimal values. We then discuss comparative analysis with other baseline methods and the efficiency of the proposed DGFR method.

A. FEATURE ANALYSIS

At each of the six tasks, we incrementally selected the predefined lower number (L) of features based on their variations using IFS. First, we discuss the descriptive statistics of the selected features concerning $L = 1,000$ as shown in Figure 5. In these figures, the number of the newly selected features after feature selection and ranking is indicated as the red rectangular bars. And the number of the kept features from the previous tasks is indicated as the blue rectangular bars. Other statistical results are listed in Appendix A.

In the first task, the initial feature sets that have 1,000 top-ranked features were selected from the first two categories (Asc = {READ, KIRP}, Desc = {OV, KIRC}, and Rand = {BRCA, GBM}). When new data comes, the means and standard deviations were re-calculated, and the variance-based feature ranking was updated. As shown in the left side of Figure 5, the kept features are increasing by small values when the number of samples is increasing in ascending order. In every new task, 250-350 features were newly selected. That shows that a small number of training samples from specific cancer types cannot express the importance of the selected features. As shown in the middle side of Figure 5, the kept features are increasing by larger values when the number of samples is decreasing in descending order. Especially in the last three tasks, only 20-50 features were newly selected. That shows how sample size influences feature selection. As shown on the right side of Figure 5, the kept and newly selected features are changed inconsistently when the number of samples is given randomly. It depends on the order of the cancer types, and it is most common in practice.

We also reported that the top-5 selected features were calculated on their variances in every task of different ordering strategies, as shown in Table 5. For example, the CpG site "cg11201229" is the most significant feature for the cancer classification task. But it is not ranked first only from the small number of samples of "READ", "KIRP" cancers. In the sixth task, 40%, 60%, and 20% of the features were re-selected from the first task, respectively, ascending, descending, and random ordering strategies. In the first task, the set of selected features are much different in the ordering strategies. In contrast, the same set of features are selected in the sixth task. That means that the importance of specific CpG sites is different for all cancers, and it is necessary to select them adaptively in each task.

B. PERFORMANCE EVALUATION

We considered four different types of state-of-the-art algorithms in terms of classification accuracy. The average accuracy results of the proposed DGFR method and the compared algorithms are shown in Table 6. We also set L as 1,000. The detailed and other performance evaluation results are listed in Appendix B. We evaluated each model by using stratified

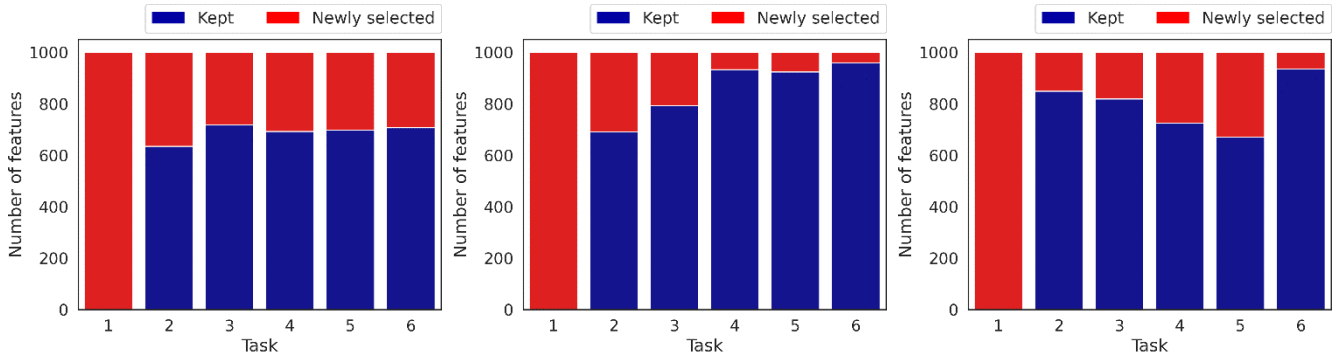

FIGURE 5. Number of newly selected/kept features on each task. The left, middle, and right sides show the number of samples is in ascending, descending, and random order, respectively.

TABLE 5. Top-5 features on each task.

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Ascending						
1	cg25600606	cg11201229	cg11201229	cg11201229	cg11201229	cg11201229
2	cg11201229	cg25600606	cg25600606	cg25600606	cg25600606	cg25600606
3	cg03853987	cg14063008	cg27592318	cg27592318	cg27592318	cg27592318
4	cg17892556	cg17872757	cg05126264	cg24992780	cg05126264	cg05126264
5	cg17872757	cg27650175	cg24992780	cg05126264	cg27557796	cg27557796
Descending						
1	cg11201229	cg11201229	cg11201229	cg11201229	cg11201229	cg11201229
2	cg25600606	cg27592318	cg27592318	cg27592318	cg27592318	cg25600606
3	cg27592318	cg25600606	cg25600606	cg25600606	cg25600606	cg27592318
4	cg23067535	cg23067535	cg20837735	cg27557796	cg05126264	cg05126264
5	cg21790626	cg21790626	cg27557796	cg05126264	cg27557796	cg27557796
Random						
1	cg11201229	cg11201229	cg11201229	cg11201229	cg11201229	cg11201229
2	cg01344452	cg01344452	cg25600606	cg27592318	cg25600606	cg25600606
3	cg00689340	cg23391785	cg27592318	cg25600606	cg27592318	cg27592318
4	cg23391785	cg25600606	cg25462291	cg27557796	cg27557796	cg05126264
5	cg27126442	cg27126442	cg01344452	cg05126264	cg05126264	cg27557796

TABLE 6. Top-5 features on each task.

Approach	Method	Asc	Desc	Rand
Lower-bound	None	16.67 (±0.00)	16.67 (±0.00)	16.37 (±0.00)
Regularization	EWC	16.67 (±0.00)	16.67 (±0.00)	16.37 (±0.00)
	Online EWC	16.67 (±0.00)	16.67 (±0.00)	16.37 (±0.00)
	SI	16.77 (±0.23)	16.67 (±0.00)	16.37 (±0.00)
Dynamic architecture	LwF	26.69 (±2.88)	20.94 (±1.19)	29.31 (±3.22)
Rehearsal	A-GEM	51.70 (±11.47)	19.55 (±3.11)	66.60 (±12.01)
	iCaRL	84.43 (±1.23)	**92.25 (±2.39)**	69.51 (±1.53)
	ER	85.85 (±0.89)	89.02 (±1.46)	89.41 (±0.98)
Generative replay	DGR	85.94 (±3.69)	86.66 (±1.57)	87.28 (±2.92)
	DGR+distill	86.88 (±1.46)	87.96 (±2.29)	90.35 (±1.43)
	RtF	87.82 (±2.45)	85.53 (±1.88)	90.18 (±0.54)
	DGFR	**92.01 (±3.41)**	88.95 (±2.79)	91.75 (±1.57)
	DGFR+distill	91.10 (±2.06)	89.77 (±1.73)	**93.48 (±1.67)**
Upper-bound	Offline	85.71 (±0.53)	88.34 (±0.76)	89.19 (±1.98)

10-fold cross-validation, then mean and standard deviation values are reported.

We used the "None" method as lower-bound, which was trained sequentially on all tasks in the standard way, also

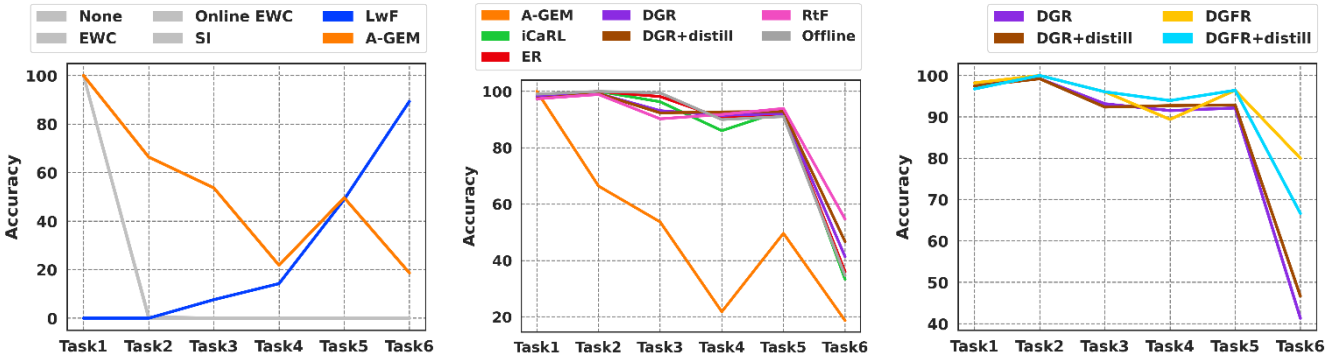

FIGURE 6. Performance evaluation on accuracy on each task. The number of samples is in ascending order.

called fine-tuning. And we used the "Offline" method as upper-bound, which was trained on the whole data in all tasks, also called joint-training. As we can see, the "None" and the regularization-based methods cannot learn the tasks except for the first tasks. Another knowledge distillation-based LwF and rehearsal A-GEM methods work better than the other regularization-based methods but not enough to be satisfied. Other methods achieved good comparable results. The compared results show that the IFS technique is much efficient and can boost classification accuracy. Firstly, when considering the ascending order, DGFR and DGFR+ distill methods achieved an average accuracy of 92.01% and 91.10%, respectively. It has greatly improved the other results of 4.19% and 3.28%, respectively. Secondly, when considering the descending order, the rehearsal-based iCaRL and ER methods achieved an average accuracy of 92.25% and 89.02%, respectively. iCaRL has greatly improved the other results by 2.48%, and ER is also comparable to the generative and the "Offline" methods. The proposed DGFR and DGFR+distill methods achieved an average accuracy of 88.95% and 89.77%, respectively. We found that iCaRL is very sensitive with different random seeds, and the other generative models show robust results on the experimental cancer datasets. Finally, when considering the random order, DGFR and DGFR+distill methods achieved an average accuracy of 91.75% and 93.48%, respectively. It has also greatly improved the other results by 1.4% and 3.13%, respectively.

We also reported the classification accuracy results of each task. As shown in Figure 6-8, the left, middle, and right sides show the results of regularization-based, rehearsal and generative, and comparison of DGR and DGFR methods, respectively. The gray color on the left side indicates the regularization-based methods which had failed on the cancer classification tasks except for the first task. The blue color on the left side indicates the LwF method, which failed in the beginning tasks and started learning the next tasks. The orange color on the left side indicates the A-GEM method, which showed satisfactory results in the beginning and failed in the next tasks. The other methods are comparative, and there are small differences shown in the figures.

As shown in Figure 6, the performances of the rehearsal and generative methods are decreasing in the last tasks when considering the number of samples is in ascending order. The reason is that the features in the first tasks were selected from a small number of samples, and those features struggled to generalize models in the last tasks. All methods failed and then showed accuracies of less than 60% in the sixth task except for DGFR. DGFR and DGFR+distill methods achieved an accuracy of 96.43% in the fifth task and 80.00% and 66.67%, respectively, in the sixth task. Compared to this, Figure 7 shows that all methods worked well and then showed accuracies of greater than 60% in all tasks except for the "Offline" method in the first task. The reason is that the features were selected from a larger number of samples in all tasks when considering the number of samples is in descending order. All the generative methods (DGR, DGR+distill, RtF, DGFR, and DGFR+distill) achieved an accuracy of 100% in the fifth task. Figure 8 shows the performance evaluation when considering the number of samples is in random order. Depends on the number of samples in random order it shows different results. For example, all methods show lower accuracies in the fifth task. Because of the cancers "OV" and "KIRC" have a large number of training samples. That means that a large set of features were newly selected, and the previously trained models fail on the new task. DGFR+distill archives an accuracy of 86.67% in the fifth task. As we can see, DGFR and DGFR+distill methods significantly improved the accuracies of the DGR and DGR+distill methods, respectively, in all tasks with different ordering strategies.

C. EFFECT OF NUMBER OF SELECTED FEATURES

For high-dimensional data, finding the optimal number of lower-dimensional features reduced by selection and transformation stages is one of the important steps. We investigated the effect of the number of selected features L, which is set as {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000}, and their optimal values in terms of classification accuracy. The detailed results are listed in Appendix B. Figure 9-11 shows the average accuracy results performed

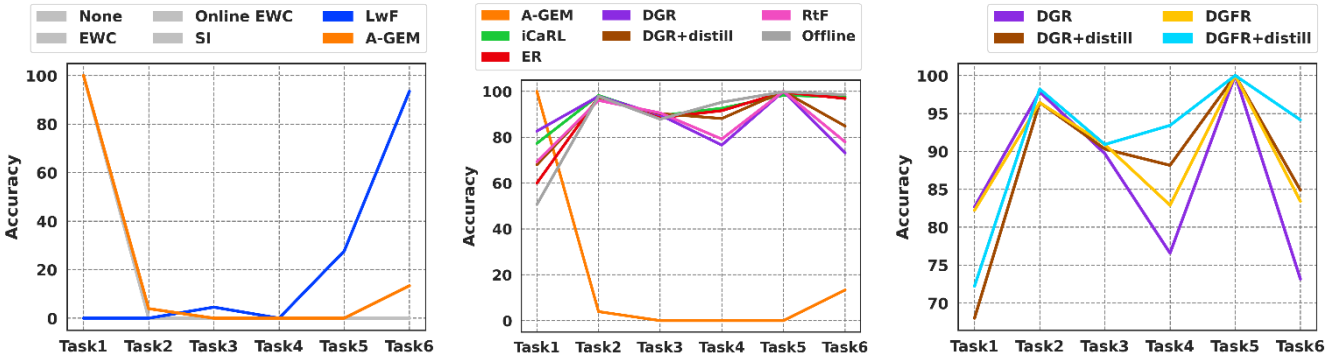

FIGURE 7. Performance evaluation on accuracy on each task. The number of samples is in descending order.

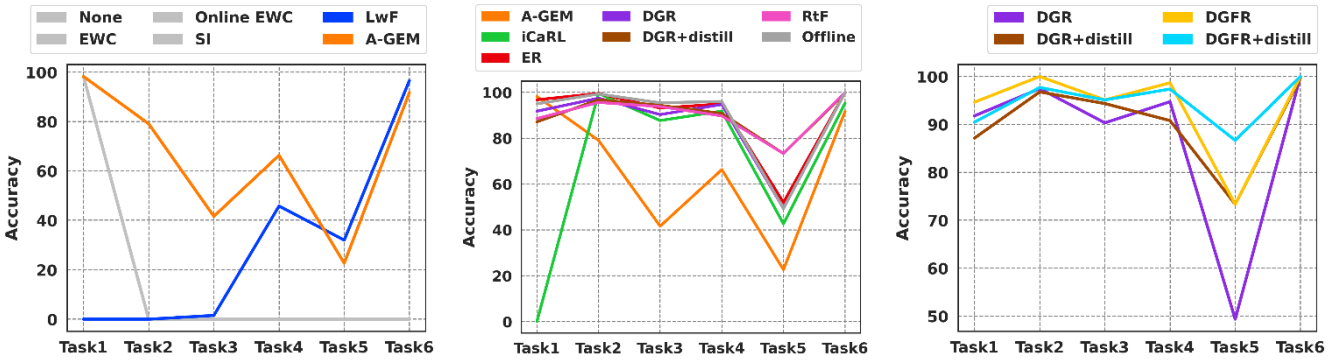

FIGURE 8. Performance evaluation on accuracy on each task. The number of samples is in random order.

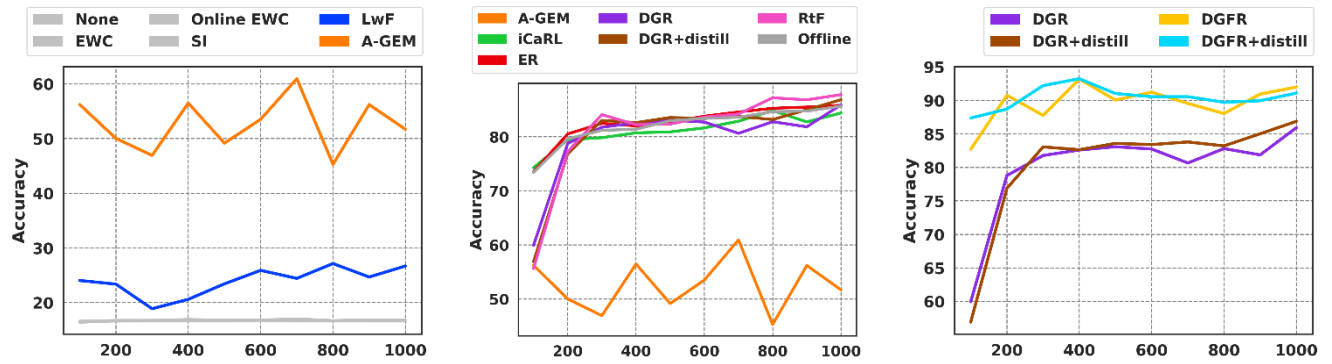

FIGURE 9. Effect of number of selected features on average accuracy. The number of samples is in ascending order.

by all baseline and proposed DGFR methods with a different number of features. As discussed above, we illustrated the results of regularization-based, rehearsal and generative, and comparison of DGR and DGFR methods on the left, middle, and right sides, respectively. And we also used the same color combinations for all methods. As shown in the figures, the regularization-based methods failed at all. In contrast, rehearsal and generative methods show satisfactory and comparative results except for A-GEM. But we found that iCaRL is very sensitive with different random seeds, especially considering the number of samples is in random order, as shown in Figure 10. DGFR and DGFR+distill methods significantly improved the accuracy of the baseline methods in all experiments. Figure 9 shows that accuracy is increasing when the number of selected features increases. RtF shows an accuracy of 87.82% with 1,000 features, which is the best result of the baseline rehearsal and generative methods. As compared to this, DGFR and DGFR+distill achieve an accuracy of 93.00% and 93.25%, respectively, which is already satisfied with only 400 features. It also has greatly improved the DGR results by approximately 10.60%. As shown in Figure 10, all the methods achieve satisfactory results with 200 features.

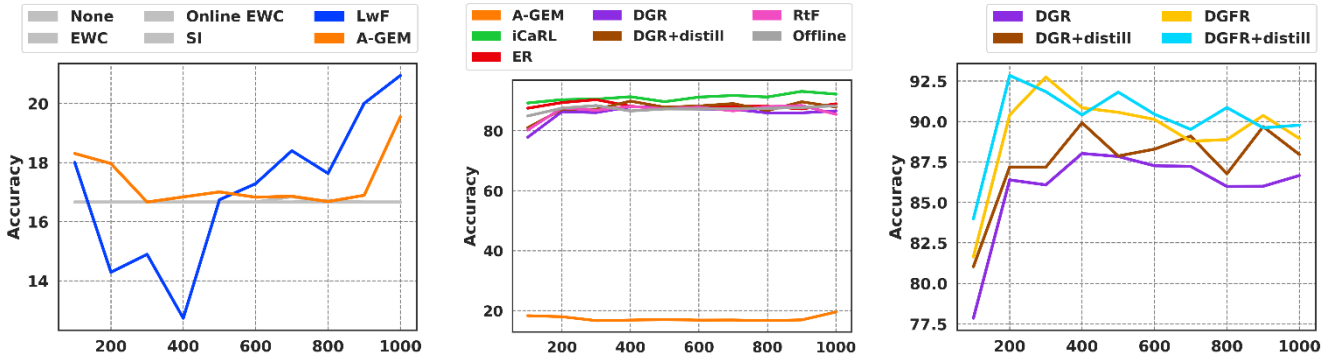

FIGURE 10. Effect of number of selected features on average accuracy. The number of samples is in descending order.

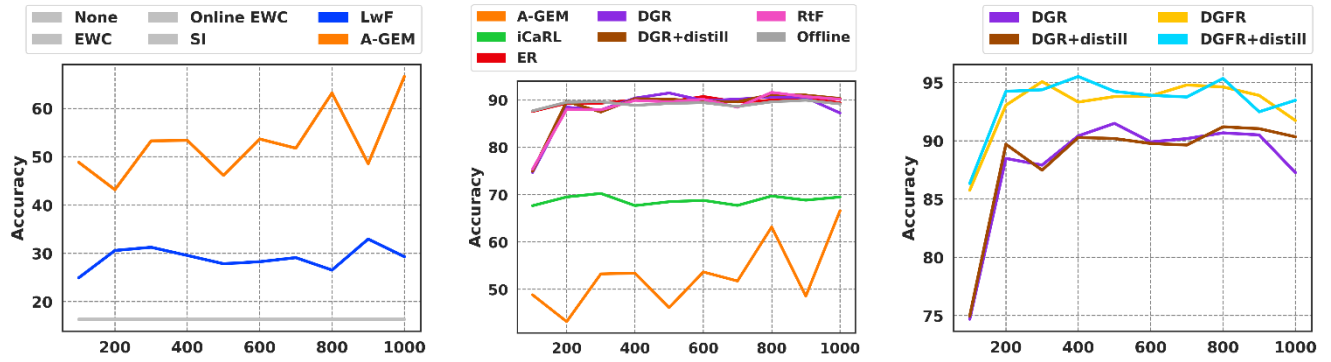

FIGURE 11. Effect of number of selected features on average accuracy. The number of samples is in random order.

For example, DGFR+distill with 200 features achieves an accuracy of 92.84%, which improved the accuracy of the previous task by 8.85%. And compared with the baseline methods, it improved the accuracy of iCaRL by 2.43%. As same as this, 200-dimensional DGFR+distill shows a satisfactory accuracy of 94.24%, which improved the accuracy of the previous task by 7.90%. It also improved the accuracy of DGR+distill by and 4.53%, as shown in Figure 11. As concluded, we found that the optimal lower number of features L is between 200 and 400. That means that these sizes of dimensions are the most convenient to reduce high dimensional data into lower-dimensional space on the experimental datasets.

D. DISCUSSIONS AND ANALYSIS

Feature selection is one of the most important steps for high-dimensional biomedical data. On the other hand, Class-IL is mandatory in the development of computational systems in bioinformatics. Most state-of-the-art Class-IL algorithms are designed for a fixed set of features, e.g., visual features. For cancer classification tasks, CpG sites can be highly significant in specific cancers and not for others. When types of cancers increase, the significance of specific CpG sites can be changed based on their variability. We found that "cg11201229", "cg25600606", and "cg27592318" CpG sites are the highest variable features, and reported the changes among the tasks in Table 5. A predefined set of features cannot express the characteristics of all cancer types, which will come in the future. So it is needed to develop an incremental feature selection algorithm that can handle previously learned features and new features adaptively. In practice, new cancer types with a lower number of samples will be added to the learning system that already learned from a higher number of samples of old cancer types. We prepared different ordering strategies such as ascending, descending, random. Then we compared the baseline and proposed methods for each ordering strategy in terms of accuracy.

In this article, we focused on feature selection from the high-dimensional DNA methylation data by taking the advantage of the current state-of-the-art algorithms. We chose the DGR method because of its generative capabilities. The other generative method is RtF, which was designed for lower computational cost of generative replay, but shows lower accuracy than DGR at most tasks. We aimed to design the proposed method to high accuracy with satisfactional computational time. We also found that iCaRL is very sensitive to different random seeds.

In bioinformatics, data security and privacy are some of the critical challenges. We tested variance-based filtering selection algorithms, which are simple but highly effective for high-dimensional data. We hope that the experimental

and analysis results give motivation to other researchers in the field of computational biology. We can see the efficiency of the proposed DGFR method in the experimental results section as we discussed above.

VI. CONCLUSION AND FUTURE WORK

In this article, we proposed a Class-IL learning method, called DGFR, which consists of an IFS and SN. SN contains a deep generative model and a neural network classifier. We used variance-based filtering as a feature selector, VAE as a generator, neural network classifier as a predictor. We ranked the features on their variabilities, and IFS adaptively selects the top-ranked features on each task. VAE pre-trained the generative models on the selected features for further analysis. Finally, we used a simple neural network to classify cancer samples into cancer categories.

We collected a total of 2,728 samples from 12 cancers from the public data portal. The state-of-the-art Class-IL algorithms are evaluated on the dataset and compared with the proposed DGFR method in terms of accuracy. To find an optimal number of features, we set it as {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000}. We chose 200-400 features as optimal values because of their satisfactional performances. The proposed DGFR and DGFR+distill methods significantly improved the accuracies of the DGR, DGR+distill, and other baseline methods. We also tested three different ordering strategies, such as ascending, descending, and random. We achieved the highest average accuracy of 93.20% (400 features), 93.25% (400 features) for ascending 92.74% (300 features), 92.84% (200 features) for descending, and 95.08% (300 features), 95.52% (400 features) for random settings, with the proposed DGFR and DGFR+distill, respectively.

In future work, we will apply the proposed method to the other high-dimensional biomedical tasks in a Class-IL way, e.g., gene expression data. The feature selection step is the most important. We will focus on developing improved feature selection algorithms in terms of performance, memory efficiency, and computational time. As well, deep generative models and classification algorithms will be considered most efficiently.

REFERENCES

[1] A. Bird, "Perceptions of epigenetics," *Nature*, vol. 447, no. 7143, pp. 396–398, May 2007.

[2] M. Esteller, "Epigenetics in cancer," *New England J. Med.*, vol. 358, no. 11, pp. 1148–1159, 2008.

[3] S. Sharma, T. K. Kelly, and P. A. Jones, "Epigenetics in cancer," *Carcinogenesis*, vol. 31, no. 1, pp. 27–36, 2010.

[4] P. M. Das and R. Singal, "DNA methylation and cancer," *J. Clin. Oncol.*, vol. 22, no. 22, pp. 4632–4642, 2004.

[5] S. B. Baylin, "DNA methylation and gene silencing in cancer," *Nature Clin. Pract. Oncol.*, vol. 2, no. S1, pp. S4–S11, Dec. 2005.

[6] M. Kulis and M. Esteller, "DNA methylation and cancer," in *Advances in Genetics*, vol. 70. New York, NY, USA: Academic, 2010, pp. 27–56.

[7] N. Touleimat and J. Tost, "Complete pipeline for Infinium human methylation 450K BeadChip data processing using subset quantile normalization for accurate DNA methylation estimation," *Epigenomics*, vol. 4, no. 3, pp. 325–341, Jun. 2012.

[8] P. Du, X. Zhang, C.-C. Huang, N. Jafari, W. A. Kibbe, L. Hou, and S. M. Lin, "Comparison of beta-value and M-value methods for quantifying methylation levels by microarray analysis," *BMC Bioinf.*, vol. 11, no. 1, Dec. 2010.

[9] S. Dedeurwaerder, M. Defrance, E. Calonne, H. Denis, C. Sotiriou, and F. Fuks, "Evaluation of the infinium methylation 450K technology," *Epigenomics*, vol. 3, no. 6, pp. 771–784, Dec. 2011.

[10] S. Kurdyukov and M. Bullock, "DNA methylation analysis: Choosing the right method," *Biology*, vol. 5, no. 1, p. 3, Jan. 2016.

[11] Y. Piao, S. K. Lee, E. J. Lee, K. D. Robertson, H. Shi, K. H. Ryu, and J. H. Choi, "CAME: Identification of chromatin accessibility from nucleosome occupancy and methylome sequencing," *Bioinformatics*, vol. 33, no. 8, pp. 1139–1146, 2017.

[12] Y. J. Kim, W. Jang, X. M. Piao, H. Y. Yoon, Y. J. Byun, J. S. Kim, and W. T. Kim, , "ZNF492 and GPR149 methylation patterns as prognostic markers for clear cell renal cell carcinoma: Array-based DNA methylation profiling," *Oncol. Rep.*, vol. 42, no. 1, pp. 453–460, 2019.

[13] M. P. Menden, F. Iorio, M. Garnett, U. McDermott, C. H. Benes, P. J. Ballester, and J. Saez-Rodriguez, "Machine learning prediction of cancer cell sensitivity to drugs based on genomic and chemical properties," *PLoS ONE*, vol. 8, no. 4, Apr. 2013, Art. no. e61318.

[14] C. Huang, R. Mezencev, J. F. McDonald, and F. Vannberg, "Open source machine-learning algorithms for the prediction of optimal cancer drug therapies," *PLoS ONE*, vol. 12, no. 10, Oct. 2017, Art. no. e0186906.

[15] G. P. Way, F. Sanchez-Vega, K. La, J. Armenia, W. K. Chatila, A. Luna, and N. Schultz, "Machine learning detects pan-cancer ras pathway activation in the cancer genome atlas," *Cell Rep.*, vol. 23, no. 1, pp. 172–180, 2018.

[16] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, p. 1, Dec. 2015.

[17] A. Aliper, S. Plis, A. Artemov, A. Ulloa, P. Mamoshina, and A. Zhavoronkov, "Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data," *Mol. Pharmaceutics*, vol. 13, no. 7, pp. 2524–2530, 2016.

[18] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2018.

[19] G. Litjens, C. I. Sánchez, N. Timofeeva, M. Hermsen, I. Nagtegaal, I. Kovacs, C. Hulsbergen- van de Kaa, P. Bult, B. van Ginneken, and J. van der Laak, "Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis," *Sci. Rep.*, vol. 6, no. 1, p. 26286, Sep. 2016.

[20] F. Li, M. Piao, Y. Piao, M. Li, and K. H. Ryu, "A new direction of cancer classification: Positive effect of low-ranking MicroRNAs," *Osong Public Health Res. Perspect.*, vol. 5, no. 5, pp. 279–285, Oct. 2014.

[21] Y. Piao, M. Piao, and K. H. Ryu, "Multiclass cancer classification using a feature subset-based ensemble from microRNA expression profiles," *Comput. Biol. Med.*, vol. 80, pp. 39–44, Jan. 2017.

[22] C. Angermueller, H. J. Lee, W. Reik, and O. Stegle, "DeepCpG: Accurate prediction of single-cell DNA methylation states using deep learning," *Genome Biol.*, vol. 18, no. 1, pp. 1–13, Dec. 2017.

[23] H. S. Shon, E. Batbaatar, K. O. Kim, E. J. Cha, and K. A. Kim, "Classification of kidney cancer data using cost-sensitive hybrid deep learning approach," *Symmetry*, vol. 12, no. 1, p. 154, 2020.

[24] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Brief Bioinform.*, vol. 18, no. 5, pp. 851–869, 2016.

[25] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114

[26] G. P. Way and C. S. Greene, "Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders," in *Proc. Pacific Symp., Pacific Symp. Biocomput.*, Kohala Coast, HI, USA, vol. 23, Jan. 2018, pp. 80–91. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/9789813235533_0008

[27] A. J. Titus, O. M. Wilkins, C. A. Bobak, and B. C. Christensen, "An unsupervised deep learning framework with variational autoencoders for genome-wide DNA methylation analysis and biologic feature extraction applied to breast cancer," *BioRxiv*, Nov. 2018, Art. no. 433763. [Online]. Available: https://www.biorxiv.org/content/10.1101/433763v5

[28] Z. Wang and Y. Wang, "Exploring DNA methylation data of lung cancer samples with variational autoencoders," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2018, pp. 1286–1289.

[29] Z. Wang and Y. Wang, "Extracting a biologically latent space of lung cancer epigenetics with variational autoencoders," *BMC Bioinf.*, vol. 20, no. S18, pp. 1–7, Nov. 2019.

[30] J. J. Levy, A. J. Titus, C. L. Petersen, Y. Chen, L. A. Salas, and B. C. Christensen, "MethylNet: An automated and modular deep learning approach for DNA methylation analysis," *BMC Bioinf.*, vol. 21, no. 1, pp. 1–15, Dec. 2020.
[31] J. Choi and H. Chae, "MethCancer-gen: A DNA methylome dataset generator for user-specified cancer type based on conditional variational autoencoder," *BMC Bioinf.*, vol. 21, no. 1, pp. 1–10, Dec. 2020.
[32] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2019," *CA: Cancer J. Clinicians*, vol. 69, no. 1, pp. 7–34, 2019.
[33] H. zur Hausen, "Viruses in human cancers," *Science*, vol. 254, no. 5035, pp. 1167–1173, Nov. 1991.
[34] P. Mehta, D. F. McAuley, M. Brown, E. Sanchez, R. S. Tattersall, and J. J. Manson, "HLH Across Speciality Collaboration," *COVID-19: Consider Cytokine Storm Syndromes and Immunosuppression*. vol. 395. London, U.K.: Lancet 2019, p. 1033.
[35] H. Kitano, "Computational systems biology," *Nature*, vol. 420, no. 6912, pp. 206–210, 2002.
[36] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3366–3375.
[37] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, "Encoder based lifelong learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1320–1328.
[38] Z. Chen and B. Liu, "Lifelong machine learning," *Synth. Lectures Artif. Intell. Mach. Learn.*, vol. 12, no. 3, pp. 1–207, 2018.
[39] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," 2018, *arXiv:1812.00420*. [Online]. Available: http://arxiv.org/abs/1812.00420
[40] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113, pp. 54–71, May 2019.
[41] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," 2019, *arXiv:1909.08383*. [Online]. Available: http://arxiv.org/abs/1909.08383
[42] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "ICaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2001–2010.
[43] E. Belouadah and A. Popescu, "IL2M: Class incremental learning with dual memory," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 583–592.
[44] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. Jay Kuo, "Class-incremental learning via deep model consolidation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1131–1140.
[45] H. Jin, Y. Luo, P. Li, and J. Mathew, "A review of secure and privacy-preserving medical data sharing," *IEEE Access*, vol. 7, pp. 61656–61669, 2019.
[46] K. James and R. Pascanu, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.
[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, "Generative adversarial nets," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680).
[48] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2990–2999.
[49] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," 2017, *arXiv:1710.10628*. [Online]. Available: http://arxiv.org/abs/1710.10628
[50] A. Seff, A. Beatson, D. Suo, and H. Liu, "Continual learning in generative adversarial nets," 2017, *arXiv:1705.08395*. [Online]. Available: http://arxiv.org/abs/1705.08395
[51] Y. Piao and K. H. Ryu, "A hybrid feature selection method based on symmetrical uncertainty and support vector machine for high-dimensional data classification," in *Proc. Asian Conf. Intell. Inf. Database Syst.* Cham, Switzerland: Springer, Apr. 2017, pp. 721–727.
[52] B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines," *Eur. J. Oper. Res.*, vol. 265, no. 3, pp. 993–1004, Mar. 2018.
[53] C. Kandoth, M. D. McLellan, F. Vandin, K. Ye, B. Niu, C. Lu, M. Xie, Q. Zhang, J. F. Mcmichael, M. A. Wyczalkowski, M. D. M. Leiserson, C. A. Miller, J. S. Welch, M. J. Walter, M. C. Wendl, T. J. Ley, R. K. Wilson, B. J. Raphael, and L. Ding, "Mutational landscape and significance across 12 major cancer types," *Nature*, vol. 502, no. 7471, pp. 333–339, Oct. 2013.
[54] E. Szczurek and N. Beerenwinkel, "Modeling mutual exclusivity of cancer mutations," *PLoS Comput. Biol.*, vol. 10, no. 3, Mar. 2014, Art. no. e1003503.
[55] C. Lövkvist, I. B. Dodd, K. Sneppen, and J. O. Haerter, "DNA methylation in human epigenomes depends on local topology of CpG sites," *Nucleic Acids Res.*, vol. 44, no. 11, pp. 5123–5132, Jun. 2016.
[56] Q. Lin, C. I. Weidner, I. G. Costa, R. E. Marioni, M. R. P. Ferreira, I. J. Deary, and W. Wagner, "DNA methylation levels at individual age-associated CpG sites can be indicative for life expectancy," *Aging*, vol. 8, no. 2, pp. 394–401, Feb. 2016.
[57] T. Thaher, A. A. Heidari, M. Mafarja, J. S. Dong, and S. Mirjalili, "Binary Harris Hawks optimizer for high-dimensional, low sample size feature selection," in *Evolutionary Machine Learning Techniques* Singapore: Springer, 2020, pp. 251–272.
[58] F. Model, P. Adorjan, A. Olek, and C. Piepenbrock, "Feature selection for DNA methylation based cancer classification," *Bioinformatics*, vol. 17, no. 1, pp. S157–S164, Jun. 2001.
[59] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531
[60] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.
[61] D. Maltoni and V. Lomonaco, "Continuous learning in single-incremental-task scenarios," *Neural Netw.*, vol. 116, pp. 56–73, Aug. 2019.
[62] J. Schwarz, J. Luketina, W. M. Czarnecki, A. Grabska-Barwinska, Y. Whye Teh, R. Pascanu, and R. Hadsell, "Progress & Compress: A scalable framework for continual learning," 2018, *arXiv:1805.06370*. [Online]. Available: http://arxiv.org/abs/1805.06370
[63] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," *Proc. Mach. Learn. Res.*, vol. 70, p. 3987, Feb. 2017.
[64] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "PathNet: Evolution channels gradient descent in super neural networks," 2017, *arXiv:1701.08734*. [Online]. Available: http://arxiv.org/abs/1701.08734
[65] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016, *arXiv:1606.04671*. [Online]. Available: http://arxiv.org/abs/1606.04671
[66] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 11849–11860.
[67] E. Belouadah and A. Popescu, "DeeSIL: Deep-shallow incremental learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 151–157.
[68] L. Caccia, E. Belilovsky, M. Caccia, and J. Pineau, "Online learned continual compression with adaptive quantization modules," 2019, *arXiv:1911.08019*. [Online]. Available: http://arxiv.org/abs/1911.08019
[69] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 831–839.
[70] R. Traoré, H. Caselles-Dupré, T. Lesort, T. Sun, N. Díaz-Rodríguez, and D. Filliat, "Continual reinforcement learning deployed in real-life using policy distillation and Sim2Real transfer," 2019, *arXiv:1906.04452*. [Online]. Available: http://arxiv.org/abs/1906.04452
[71] T. D. Bui, C. V. Nguyen, S. Swaroop, and R. E. Turner, "Partitioned variational inference: A unified framework encompassing federated and continual learning," 2018, *arXiv:1811.11206*. [Online]. Available: http://arxiv.org/abs/1811.11206
[72] R. Traoré, H. Caselles-Dupré, T. Lesort, T. Sun, G. Cai, N. Díaz-Rodríguez, and D. Filliat, "DisCoRL: Continual reinforcement learning via policy distillation," 2019, *arXiv:1907.05855*. [Online]. Available: http://arxiv.org/abs/1907.05855
[73] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 374–382.
[74] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 350–360.
[75] T. Lesort, H. Caselles-Dupré, M. Garcia-Ortiz, A. Stoian, and D. Filliat, "Generative models from the perspective of continual learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, Jul. 2019, pp. 1–8.
[76] T. Lesort, A. Gepperth, A. Stoian, and D. Filliat, "Marginal replay vs conditional replay for continual learning," in *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, Sep. 2019, pp. 466–480.

[77] C. Wu, L. Herranz, X. Liu, D. W. J. van, and B. Raducanu, "Memory replay GANs: Learning to generate new categories without forgetting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5962–5972.
[78] R. Venkatesan, H. Venkateswara, S. Panchanathan, and B. Li, "A strategy for an uncompromising incremental learner," 2017, *arXiv:1705.00744*. [Online]. Available: http://arxiv.org/abs/1705.00744
[79] G. M. van de Ven and A. S. Tolias, "Generative replay with feedback connections as a general strategy for continual learning," 2018, *arXiv:1809.10635*. [Online]. Available: http://arxiv.org/abs/1809.10635
[80] T. Lesort, A. Stoian, and D. Filliat, "Regularization shortcomings for continual learning," 2019, *arXiv:1912.03049*. [Online]. Available: http://arxiv.org/abs/1912.03049
[81] M. J. Goldman, B. Craft, M. Hastie, K. Repečka, F. McDade, A. Kamath, and J. Zhu, "Visualizing and interpreting cancer genomics data via the Xena platform," *Nature Biotechnol.*, vol. 38, pp. 675–678, May 2020.
[82] D. Stach, "Capillary electrophoretic analysis of genomic DNA methylation levels," *Nucleic Acids Res.*, vol. 31, no. 2, p. 2, Jan. 2003.
[83] D. Lopez-Paz and M. A. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017 pp. 6467–6476.
[84] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, Oct. 2012, pp. 488–501.
[85] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[86] G. M. van de Ven and A. S. Tolias, "Three scenarios for continual learning," 2019, *arXiv:1904.07734*. [Online]. Available: http://arxiv.org/abs/1904.07734

ERDENEBILEG BATBAATAR received the M.S. and Ph.D. degrees in data mining, medical informatics, and computer science from the Database and Bioinformatics Laboratory, Chungbuk National University, South Korea. He is currently a Postdoctoral Researcher of bioinformatics and computer science with Chungbuk National University. His research interests include software engineering, data mining, big data analysis, bioinformatics, machine learning, deep learning, and their applications.

KWANG HO PARK (Graduate Student Member, IEEE) received the B.S. degree in biochemistry from Chungbuk National University, Cheongju, South Korea, in 2015, and the M.Sc. degree from the Database and Bioinformatics Laboratory, in 2017, where he is currently pursuing the Ph.D. degree. His main research interests include data mining, bioinformatics, and medical informatics.

TSATSRAL AMARBAYASGALAN received the B.Sc. degree in information technology engineering from the School of Information and Technology, National University of Mongolia, Mongolia, in 2010, and the master's degree in computer science from the School of Engineering and Applied Sciences, National University of Mongolia, in 2015. She is currently pursuing the Ph.D. degree in computer science with Chungbuk National University, Republic of Korea. Her research interests include data mining, artificial intelligence, deep learning, big data, and healthcare analytics.

KHISHIGSUREN DAVAGDORJ (Graduate Student Member, IEEE) received the M.Sc. degree in industrial technology and design from the Mongolian University of Science and Technology, Mongolia, in 2013. She is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Chungbuk National University, South Korea. Her main research interests include artificial intelligence, big data, recommendation systems, and data mining.

LKHAGVADORJ MUNKHDALAI (Student Member, IEEE) received the B.Sc. degree in economics from the National University of Mongolia and the M.Sc. degree from Chungbuk National University, South Korea, where he is currently pursuing the Ph.D. degree in computer science. He has been involved in many research projects about deep learning-based time series forecasting, such as Infectious Disease Forecasting, Short-term Export and Import Forecasting, and Demand Forecasting for Postal Delivery Service. His research interests include deep learning along with their applications in domains, such as financial, medical informatics, and public health informatics. His current research interests include the development of an adaptive deep learning model for forecasting time series as well as adaptive regression models.

VAN-HUY PHAM received the M.S. degree in computer science from the University of Sciences, Ho Chi Minh City, Vietnam, in 2007, and the Ph.D. degree in computer science from Ulsan University, South Korea, in 2015. Since 2015, he has been a Lecturer and a Researcher with the Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City. His main research interests include artificial intelligence, image processing, computer vision, and deep learning applications.

KEUN HO RYU (Life Member, IEEE) received the Ph.D. degree in computer science and engineering from Yonsei University, South Korea, in 1988. He has served at Reserve Officers' Training Corps (ROTC) of the Korean Army. He was with The University of Arizona, Tucson, AZ, USA, as a Postdoctoral Researcher and a Research Scientist, and the Electronics and Telecommunications Research Institute, South Korea, as a Senior Researcher. He is currently a Professor with the Faculty of Information Technology, Ton Duc Thang University, Vietnam, an Emeritus and the Endowed Chair Researcher with Chungbuk National University, South Korea, and also an Adjunct Professor with Chiang Mai University, Thailand. He is also an Honorary Doctorate of the National University of Mongolia. He has been not only the Leader of the Database and Bioinformatics Laboratory, South Korea, since 1986, but also the Co-Leader of the Research Group, Data Science Laboratory, Vietnam, since March 2019. He is also the former Vice-President of the Personalized Tumor Engineering Research Center. He has published more than 1000 referred technical papers in various journals and international conferences, in addition to authoring a number of books. His research interests include databases, spatiotemporal databases, big data analysis, data mining, deep learning, biomedical informatics, and bioinformatics. He has been a member of the ACM since 1983. He has served on numerous program committees, including roles as the Demonstration Co-Chair of the VLDB, the Panel and Tutorial Co-Chair of the APWeb, and the FITAT General Co-Chair.