Soft Computing
https://doi.org/10.1007/s00500-019-04602-2

METHODOLOGIES AND APPLICATION

Data augmentation using MG-GAN for improved cancer classification on gene expression data

Poonam Chaudhari[1] · Himanshu Agrawal[2] · Ketan Kotecha[2]

Abstract

Molecular biology studies on cancer, using gene expression datasets, have revealed that the datasets have a very small number of samples. Obtaining medical data is difficult and expensive due to privacy constraints. Accuracy of classifiers depends greatly on the quality and quantity of input data. The problem of small sample size or small data size has been addressed by augmentation. Owing to the sensitivity of synthetic data samples for the cancer data classification for gene expression data, this paper is motivated to investigate data augmentation using GAN. GAN is based on the principle of two blocks (generator and discriminator) working in a collaborative yet adversarial way. This paper proposes modified generator GAN (MG-GAN) where the generator is fed with original data and multivariate noise to generate data with Gaussian distribution. As the generated data lie within latent space, we reach saddle point faster. GAN has been widely used in data augmentation for image datasets. As per our understanding, this is the first attempt of using GAN for augmentation on gene expression dataset. The performance merit of proposed MG-GAN was compared with KNN and Basic GAN. As compared to KNN and GAN, MG-GAN improves classification accuracy by 18.8% and 11.9%, respectively. The loss value of the error function for MG-GAN is drastically reduced, from 0.6978 to 0.0082, ensuring sensitivity of the generated data. Improved classification accuracy and reduction in the loss value make our improved MG-GAN method better suited for critical applications with sensitive data.

Keywords Data augmentation · Generative adversarial network · Gene expression dataset · Cancer detection · Modified generator GAN · Multivariate noise · Gaussian distribution · Latent space · Saddle point

1 Introduction

1.1 Background of gene expression microarray data and need of data augmentation

One extremely important use of machine learning is in prognosis and diagnosis in the medical domain. Some diseases like cancer are extremely complex and show a large variation in severity, duration, location, sensitivity and resistance to drugs, cell differentiation and origin and understanding of pathogenesis (Wu et al. 2012).Gene expression profiling is used widely to understand and analyse the conditions of cells and their response to different stimuli/conditions, which helps in pathogenesis of diseases (Wang et al. 2018). Gene expression microarrays can be used for diagnostic purposes as well as for providing insights into biology. Gene expression data have a high number of features and comparatively very few samples (Chaudhari and Agarwal 2018). Training the classifier becomes difficult, as a lower number of samples can lead to over fitting. It therefore becomes extremely important to add more samples to train the classifier better and help with classification. Data augmentation is the process of generating additional data from existing data to solve this problem.

Communicated by V. Loia.

✉ Poonam Chaudhari
poonam92chaudhari@gmail.com

[1] Gokhale Education Society's R. H. Sapat College of Engineering, Management Studies and Research, Nashik, India

[2] Symbiosis Institute of Technology, Pune, India

Published online: 13 December 2019

1.2 Motivation

For some applications, acquiring a sufficient amount of appropriate training data is difficult. Especially in medical imaging, the acquisition of labelled training data is very time-consuming and costly, as a trained expert needs to manually annotate every image in the training set. Fang Liu suggested using data augmentation to solve the problem of a small sample size (Liu et al. 2019a, b). In the biomedical field, data augmentation becomes especially important as there is a great dearth of easily available data (as medical records are sensitive data, and it is difficult to get consent). Moreover, it is very important to protect the semantics of the data, which in turn means that there are many limitations on mathematical transformations that can be applied to the data (Konidaris et al. 2018).The use of deep neural networks (DNN) has brought advances in machine learning by improving algorithms to perform even better. The identification of genetic determinants underlying potential cancer is a herculean task (Colllins 2002). Typical data augmentation techniques can result in data that vary greatly from the original. However, in the area of cancer classification, it is crucial to maintain the sensitivity of the data, i.e. for the augmented data to be as close as possible in behaviour to the original data. We therefore need to augment the data using more sophisticated approaches.

A promising approach for training a model that synthesizes data is known as generative adversarial networks (GANs).GANs have gained great popularity, and different variations of GANs were recently proposed for generating high-quality realistic natural images (Frid-Adar et al. 2018). In the area of artificial intelligence also, GANs have made their mark owing to their outstanding abilities in the area of data generation (Pan et al. 2019). The author states that GANS can be used for data augmentation very effectively even in the case of tasks where traditional techniques of data augmentation do not produce good results or are completely ineffective (Namozov and Im Cho 2018).

1.3 Contributions

From our interaction with industry experts, we realised that samples are not easily available. The samples are expensive; also sample collection from beta customers is difficult due to privacy constraints. Generating new, correct sample data from existing samples therefore becomes all the more important. The specific contributions made through modified generator GAN are as stated:

- We conducted a detailed survey on data augmentation using GAN. Our work is motivated by the recent developments in GANs for generating new data. Use of GAN for classifying gene expression data has not been attempted by any research group. As per our knowledge, this is the first attempt of using GAN in connection with cancer classification using microarray gene expression dataset.
- The conventional GAN consists of generator (for generating synthetic data) and discriminator (discriminate between original data and synthetic data). The feedback from discriminator is used to train the generator to generate synthetic samples as close as possible to original data. Traditionally, the generator block is supplied with noise to generate synthetic data. As the generator starts the generation process with very little information of the original data, the feedback cycles from discriminator to generator are high. Thus, the computational time and cost required to generate acceptable samples are high. Secondly, as we deal with sensitive application area of cancer classification, it is important to maintain the sensitivity of the data, i.e. for the augmented data to be as close as possible in behaviour to the original data. We need to augment the data using more sophisticated approaches.

  We propose an improvisation on GAN, called as modified generator GAN (MG-GAN), where the generator is fed with original data along with minimalistic multivariate noise as input. Inclusion of original data helps the generator to understand the upper and lower bound values of each feature. The latent space of working guides the generator to generate synthetic samples which lie within the boundary values of the original data. As the synthetic data are extremely close to the original data, the feedback loop completes in relatively shorter time with acceptable augmented data.
- Our experimental results using data augmented with the MG-GAN technique show improvement in classification accuracy while maintaining a high sensitivity value. The accuracy is improved by 18% as compared to traditional augmentation technique, KNN and 11% with respect to Basic GAN.

2 Related work

The related work is broadly classified into three subsections dealing with the application and usability of the algorithm in various domains, few challenges confronted and ongoing research to overcome the same and lastly, scope for future advancements.

2.1 GAN and its variants for various application domains

Goodfellow et al. (2014) introduced GAN in 2014, where he used antagonist to improve the performance of machine learning algorithms. The algorithm gained momentum from 2017, where researchers started exploring the capacity of GAN in wide variety of applications such as image transformation, MRI scans, video, biomedical data, gear safety in the automobile sector (Li et al. 2019), pearl classification (Xuan et al. 2018), underwater machine vision (Chen et al. 2017), remote sensing and generation of ground-level views using overhead imagery (Deng et al. 2018) and even the fashion industry for shoe designing (Deverall et al. 2017).

Radford et al. (2015) proposed convolution neural networks (CNNs), as used in deep learning, both for the generator as well as the discriminator and applied it for image representations. Deep CNNs suffer from over fitting as the number of learnable parameters in deep models is very large. To overcome this drawback, Zhu et al. (2018) suggested the use of GAN for hyper-spectral images. Yu et al. (2019) extended the research to deal with the textural details of the image content. The concept of edge-aware GAN (EA-GAN) for synthesis of MR images was introduced. An interesting application of preserving original identity in an image after ageing was studied by Antipov et al. (2017). Marchesi (2017) investigated GAN for generating high-quality mega pixel images, where limited data were used as opposed to thousands of images used by the other researchers. Eghbal-zadeh and Widmer (2017) introduced general likelihood estimation for assessing the quality of generated images using GAN; the method is independent of GAN architecture as well as method of training. Li et al. (2017) brought out the theoretical basis of GAN and explained the collapse of discriminator with a proof. Vertolli and Davies (2017) introduced a method of training and evaluation of GAN generated images and stated that different distance metrics in loss function could capture different parameters in images. Compressed sensing MRI, a recent technique to reduce time of acquisition of MRI images, was presented by Quan et al. (2018).

Creswell and Bharath (2018) showed that representations learnt by GANs can be compared quantitatively by using an inversion approach. Antoniou et al. (2017) proposed data augmentation by applying existing data to novel unseen classes of data. Luc et al. (2016) proposed a GAN-based approach on a hybrid loss term consisting of multiclass cross-entropy.

Huang et al. (2017) used stacked GAN with pre-trained discriminator and hierarchical arranged GANs. Lu et al. (2018) proposed an idea called Bi-GAN, in which two generators are used. As extended research, Wang et al. (2019) suggested using a group of generators to work as adversaries with one discriminator.

Li et al. (2018a, 2018b) used GAN in the area of CPS (cyber-physical systems). Lucas et al. (2019) showed that a significant performance increase can be achieved by training a DNN with the appropriate loss function and architecture. GANs were applied for speech denoising as well (Dutt and Premchand 2017). Tembine (2019) suggested the use of a Bregman discrepancy to speed up learning and achieve a higher rate of convergence, without the need to use a second derivative of the objective function.

An application of GANs for cosmological data was reported by Mustafa et al. (2019) in the area of gravitational lensing to sense dark energy in galaxies. Chan and Elsheikh (2017) applied GAN to generate geological data in sub-surface fields. Gong et al. (2017) stated through the results that GAN has been used very effectively in Geosciences.

Shang et al. (2017) handled imputations of missing data of MISNST using CycleGAN. Gurumurthy et al. (2017) proposed DeLiGAN, a novel GAN-based architecture for diverse and limited training data scenarios like handwritten digits, objects and hand-drawn sketches.

2.2 Challenges faced by GAN

Although GAN has been widely used in a fleet of application domains, researchers also drew attention to few challenges faced by GAN.

Li et al. (2018a, b) provided at least partial solutions to some outstanding GAN problems such as measuring convergence, controlling distributional diversity and maintaining the equilibrium between the discriminator and the generator. Metz et al. (2016) solved the common problem of mode collapse, stabilized training of GANs with complex recurrent generators and increased diversity and coverage of the data distribution by the generator. Arjovsky et al. (2017) came up with a new algorithm called WGAN, which showed improved stability of learning, got rid of problems like mode collapse and provided meaningful learning curves useful for debugging and hyper-parameter searches. Oliehoek et al. (2018) worked on local Nash equilibrium to solve the mode collapse issue.

2.3 Analysis for future direction

An in-depth literature review on GAN helped us understand the usefulness and versatility of GANs. The study also helped us identify an unexplored area of GAN dealing with genetic data. Very recently (from 2017 to 2018), there have been attempts to use GAN for genetic data.

Marouf et al. (2018) tried GAN on gene sequencing data. Wan et al. (2018) looked into the role of landmark genes in prediction and identified different relevance patterns, which provided insights into the relations among gene regulatory networks. Ghasedi et al. (2018) used a model based on GAN to approximate the joint distribution of landmark and target genes, and an inference network to learn the conditional distribution of target genes given the landmark genes.

The study of this literature inspired us to explore the possibility of building on the good work ahead and using GAN for augmentation in gene expression datasets.

3 Preliminaries

3.1 *K*-nearest neighbourhood rule

The nearest neighbourhood rule is a nonparametric decision rule. This means that no prior knowledge of distribution is required. Synthetic samples can be generated without knowing intrinsic details of the original data. In the *k*-nearest neighbour (KNN) technique, the process of choosing the right value of *k* is critical for accuracy. The research work of Chaudhari and Agarwal (2019) states that as we increase the value of *k*, the diversity of the synthetic sample increases. However, if the value of *k* is increased too much, it leads to samples which do not match to the original data. If *k* is kept too low, then the augmented data do not have enough variation. Hence, the thumb rule defines the value of *K* as the square root of the number of samples in the dataset. The Euclidean distance formula is computed to generate new samples. Matlab documentation for classification using nearest neighbours (2019) is shown in Fig. 1.

3.2 Generative adversarial network

GAN is much more complex than previously used deep learning algorithms. In deep learning, we train the algorithm using available data and then try to classify, cluster or predict. In GAN, we create new things. For example, when GAN is applied to an image dataset, we can create new images; when it is applied to musical data, we can create new tunes/notations.

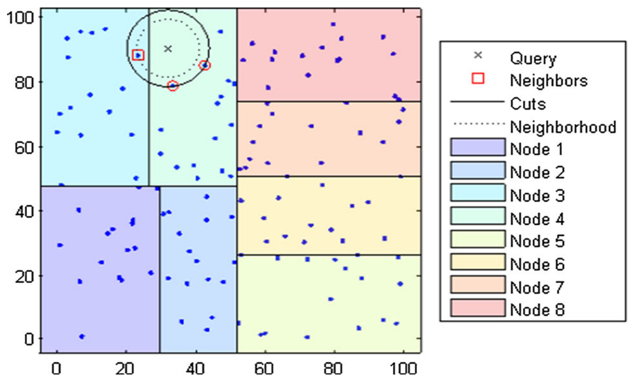


Fig. 1 *K*-nearest neighbourhood approach

GAN is a machine learning algorithm that works on a strictly competitive zero-sum principle. GAN has two major building blocks as shown in Fig. 2 (Gharakhanian 2017):

- The generator, which generates samples without learning the features of the input dataset, i.e. without understanding the semantics of the data.
- The discriminator, which tries to discriminate between the original sample and the generated sample. The discriminator uses both the training samples (original data) and generated samples separately.

In GAN, we train both the generator and the discriminator using feed forward network and dropout algorithm. The two blocks work together to improve each other, but in an adversarial way. The discriminator tries to learn the original samples very keenly and guides the generator by sending feedback about the generated synthetic samples. The generator learns from the feedback and tries to generate new samples which are very close to the original dataset. The model converges till the discriminator can no more differentiate the samples.

The two blocks have different objective functions. The discriminator works on maximum likelihood factor; the generator works on the maximum error rate of discriminator (Hui 2018).

Formula for Maximum Likelihood Factor:

$$\mathrm{lik}(\theta) = \prod_{i-1}^{n} f(x_i|\theta) \tag{1}$$

where n is the number of samples provided as input to the discriminator and x is the sample generated by the generator and i is used as loop counter. The two samples are used to compute the closeness which in turn decides the acceptability of the generated sample.

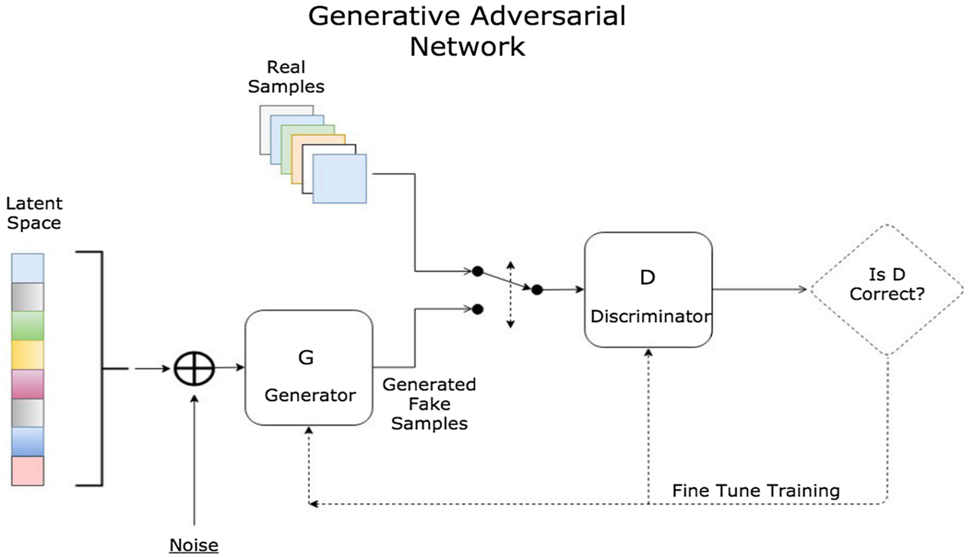
Formula for Maximum Error Rate:

$$\text{Standard error of difference} = \sqrt{\frac{(p+q)^2-(p-q)^2}{n}} \tag{2}$$

where p, q are the data points which help calculate the difference between generated sample and original sample from the dataset with n samples.

Two metrics are normally used to calculate expected probability distribution. The traditional formula is Kullback–Leibler divergence (KL). KL divergence is asymmetric. GANs use Jensen–Shannon divergence (JSD), which calculates the similarity between two probability distributions. The use of JSD, which is symmetric in

Fig. 2 Generative adversarial network architecture



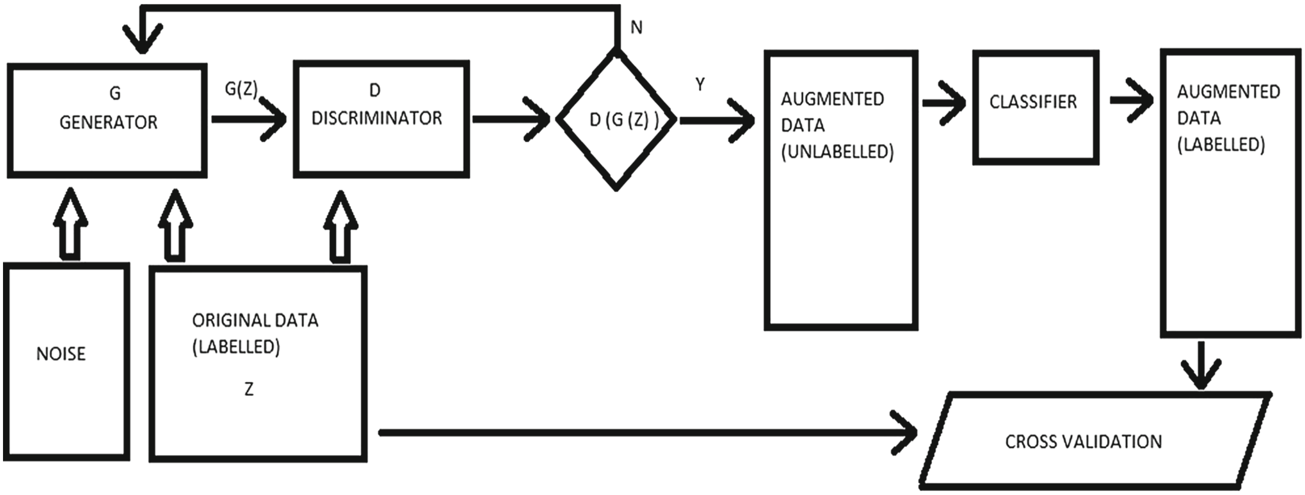


Fig. 3 MG-GAN system architecture

nature, is one of the reasons for the success of GANs (Ledig et al. 2017; Weng 2017; Huszár 2015).

The JSD Distance formula is:

$$\mathrm{JSD}(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M) \tag{3}$$

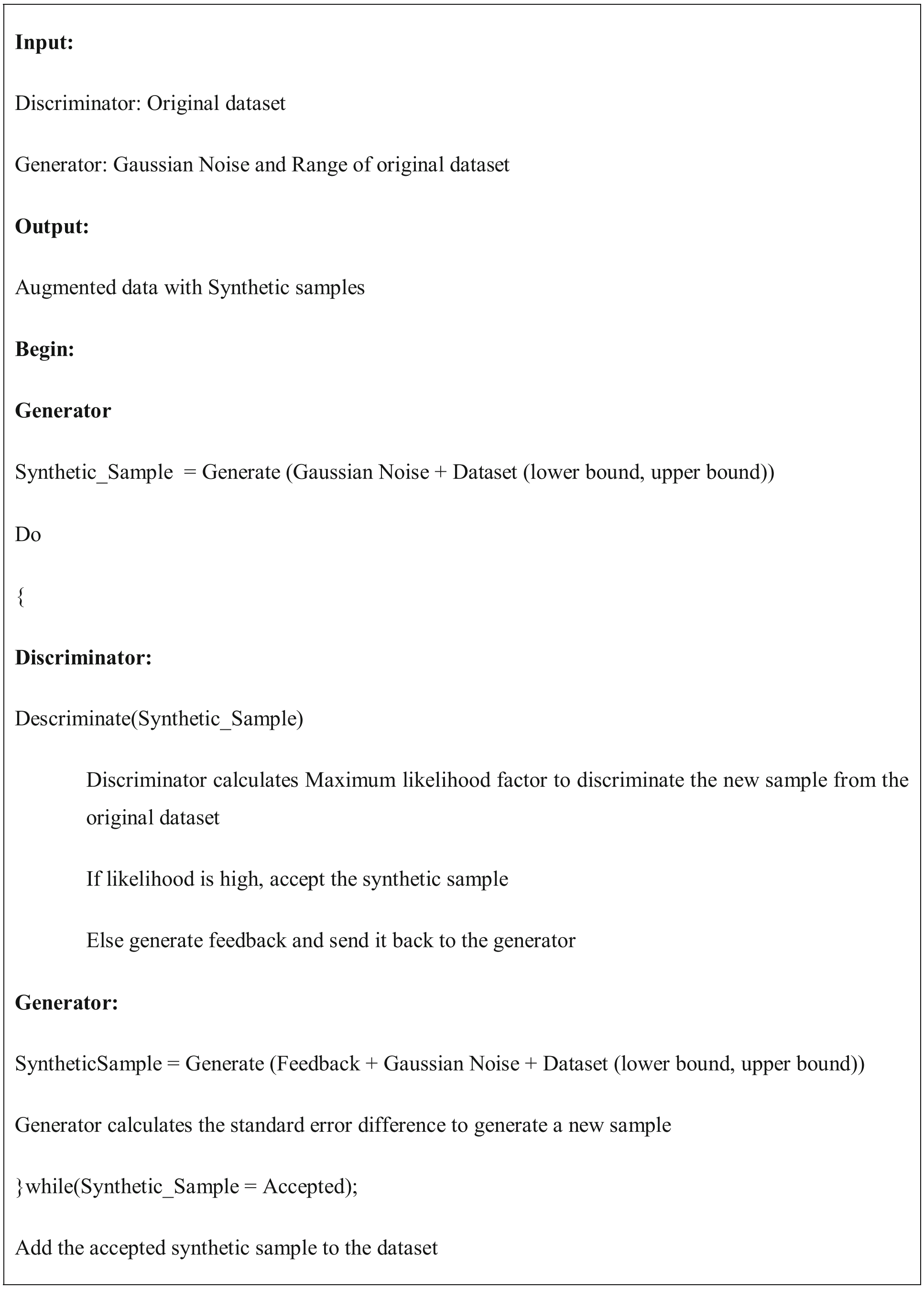
where P and Q are the probability distributions and M is the mean of P and Q. $D(P||M)$ represents the divergence of probability distribution P from the mean.

GAN has been successful in balancing datasets. However, the application of Basic GAN for classifying data with a small sample size has not been studied in great detail (Liu et al. 2019a, b). Traditional GAN adds random noise to the original dataset to create synthetic samples. This is not desirable in the sensitive application area of cancer classification.

4 Proposed modified generator—GAN

4.1 MG-GAN system architecture

Figure 3 explains the system architecture. The input to the generator is a combination of Gaussian noise and the range of the original data, stating the upper limit and the lower limit. The generator generates a synthetic sample which is executed by a function generate, G(Z). The output generated by the generate function is given as an input to the discriminator. The discriminator has a function called discriminate, D(). The discriminate function takes the input from the generator D(G(Z)) and tries to find the likelihood between the generated sample and the original sample. The factor decides the status of the generated sample. If the sample is rejected, the feedback is sent back to the generator based on which the generator computes the error difference and updates with a new synthetic sample.



```
Input:
Discriminator: Original dataset
Generator: Gaussian Noise and Range of original dataset
Output:
Augmented data with Synthetic samples
Begin:
Generator
Synthetic_Sample  = Generate (Gaussian Noise + Dataset (lower bound, upper bound))
Do
{
Discriminator:
Descriminate(Synthetic_Sample)
        Discriminator calculates Maximum likelihood factor to discriminate the new sample from the
        original dataset
        If likelihood is high, accept the synthetic sample
        Else generate feedback and send it back to the generator
Generator:
SyntheticSample = Generate (Feedback + Gaussian Noise + Dataset (lower bound, upper bound))
Generator calculates the standard error difference to generate a new sample
}while(Synthetic_Sample = Accepted);
Add the accepted synthetic sample to the dataset
```

4.2 Model description

As we deal with sensitive gene expression data, we need to ensure that the model is stable enough for generating synthetic data which lie in proximity to the original data. The application data are extremely compelling and sentient; thus, we cannot generate synthetic data by addition of random noise. The objective primarily focuses on optimizing the loss function and generating synthetic data acceptable by the discriminator.

The traditional GAN uses noise as an input to the generator. With this input, the generator makes an attempt to generate synthetic data which could be approved by the discriminator. However, the discriminator is pre-trained using the original data. This clever machine learning unit is stringent with approval and does not surpass fake samples. In the process of improvement, the discriminator sends back feedback to the generator. The generator is a slow learner as it has to start learning from scratch.

We propose modified generator GAN, where the discriminator is trained with original data, just as in traditional GAN. However, contradictory to the basic traditional GAN, the generator is fed with the original data along with multivariate noise with distribution $P(n)$ as input. We use Gaussian distribution with covariance. Let $P(o)$ and $P(s)$ be the probability distribution of original data and synthetic data. If we consider x and y as integers, then $\pi(P(o), P(s))$ is the set of joint distribution of (x, y). The dimension used for this distribution helps to demark the latent space. The latent space specifies the innate scope of variation. This helps the generator to generate samples which lie within the permissible latent space. As the new sample lies within the circumscribed area and the sample is very similar to the original data, the discriminator accepts the synthetic data as real. The feedback loop need not iterate multiple times to train the generator. This helps to save computational time and cost. These two adversarial blocks work on min–max approach. The numerical methods used to deal with this approach are approximation and optimization. We used ADAM optimization function for generating stable solutions and reaching a saddle point in finite time.

The deep neural network classifier is pre-trained using the original data. The unlabelled augmented data are given to deep neural network classifier. Deep neural network classifier uses a sophisticated mathematical modelling to process data in complex ways. Training of deep neural network requires four decision steps: selection of model type with algorithm, selection of the architecture of the network, assignment of training parameters and learning of model parameters.

We build the network using a sequential model. ReLU is used as an activation function within the hidden layers. Batch normalization is used to normalize the input to the layers by scaling the activation function. The batch size is 32. The network has five hidden layers. The classifier assigns labels to the augmented data. The labelled augmented data are cross-validated to compute the accuracy of the classifier.

5 Results and discussion

5.1 Dataset

We made use of the gene expression microarray data from datasets publicly available on the NCBI repository (https://www.ncbi.nlm.nih.gov/). The NCBI repository supports both microarray gene expression data and sequence data. Experimental computations were performed on 20 gene expression datasets representing different categories of tumours.

Generally, there are only two classes (cancerous and non-cancerous), but in a couple of cases, we found a third class (suspected but unconfirmed cancer). The number of samples in this class was very low, so we call it “micro class”. As our work focuses on binary classification, we used a technique called one-vs.-one to absorb data from the micro-class into the first two classes. As the amount of data in the micro-class is very small, it does not have a noticeable impact on the overall results is shown in Tables 1 and 2.

Representative datasets from each category of cancer are described below.

- Lung cancer (GDS2771): Epithelial cells from regular cigarette smokers were collected. Data are segregated into two categories as smokers without cancer and with suspected lung cancer. Results provide insights into the feasibility of using gene expression to detect early stage lung cancer in smokers. The dataset contains 192 samples (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE4115).
- Prostate Cancer (GDS2545): Tissues adjacent to the tumour were collected. The classification is done in two classes as normal cancer or metastatic cancer. The dataset consists of 171 samples (88 classified as primary prostate cancer & 83 as metastatic cancer) (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE6919).
- Leukaemia (GDS2118): Myelodysplastic syndrome leads to leukaemia. This dataset with 66 samples classifies the patient as cancerous or non-cancerous (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE4619).
- Breast Cancer (GDS27567): Peripheral mononuclear blood samples were collected. The dataset includes 162 samples (94 cancerous; 68 non-cancerous) (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE27567).
- Colon cancer (GDS3268): Transcriptional profiling of colon epithelial biopsies from ulcerative colitis patients and healthy control donors. This dataset has 202 samples (44 cancerous; 158 non-cancerous) (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE11223).

Table 1 Description of datasets

#	Cancer dataset	GDS	Samples	Attribute count
1	Lung	GDS2771	192	22,215
2	Prostate	GDS2545	171	12,625
3	Leukaemia	GDS2118	66	12,646
4	Breast	GDS27567	162	54,675
5	Colon	GDS3268	202	54,673

Table 2 Dataset with sample count within classes

#	Cancer dataset	GDS	Class label 1 (count)	Class label 2 (count)
1	Lung	GDS2771	MPM(103)	ADCA(89)
2	Prostate	GDS2545	N-88	PC-83
3	Leukaemia	GDS2118	AML(38)	ALL(28)
4	Breast	GDS27567	N(94)	BC(68)
5	Colon	GDS3268	AML(44)	ALL(158)

Table 1 shows the count of the samples(row) and the attributes(columns), whereas Table 2 depicts the number of samples in each class.

5.2 State-of-art methods

We study the following state-of-the-art methods of augmentation:

- *K*-nearest neighbourhood rule:
  As a general rule of thumb, value of k is computed as square root of the total samples. Euclidean metric is used for calculating the distance between line segments considering adjacent k-nearest neighbours to generate synthetic samples.
- Basic GAN
  GAN is an augmentation technique, which works with generator and discriminator in a conflicting manner, to improve both of them and achieve optimal solution.
- Modified generator GAN
  This is the proposed approach where the generator is also trained with the help of original data, to generate synthetic samples adjacent to original samples.

The data generated through each of the augmentation process are classified using a deep neural network classifier to compute classification accuracy, precision and recall.

5.3 Performance metrics

Four different kinds of performance metrics can be used to compute the efficiency of the augmentation techniques.

- *True Positive* (*TP*) Positive samples that were correctly labelled by the classifier,
- *True Negative* (*TN*) Negative samples that were correctly labelled by the classifier,
- *False Positive* (*FP*) Negative samples that were incorrectly labelled as positive,
- *False Negative* (*FN*) Positive samples that were incorrectly labelled as negative,

Using these four parameters, we calculated precision, recall and accuracy. As we have added synthetic samples, maintaining sensitivity is important; hence, we emphasize the importance of recall value. Metrics are calculated using these formulae:

- $$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{\text{Total}} \tag{4}$$
  where Total = TP + FP + FN + TN
- $$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{5}$$
- $$\text{Recall(also called as sensitivity)} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{6}$$

5.4 Results for loss function in MG-GAN

The generator uses the Keras Sequential model along with dense and batch normalization layers. The activation function used is leaky ReLU. The generator model is divided into several iterations, each of which is structured as Dense Layer → Activation → Batch Normalization. Nodes in each dense layer increase as the model progresses. The discriminator takes in input data and converts it into a one-dimensional array and then passes it through two blocks of Dense → Activation.

The MG-GAN approach uses mean square error loss function. The feed forward network aims at reducing the error, which means that the data generated by the generator resemble original data to a large extent, which leads the discriminator to accept it as original data.

Figure 4 shows that the loss value decreases from 0.6978 at the start to 0.0082 by the end of 20,000 epochs. Thus, the augmented data generated through our approach are extremely close to the original data ensuring the sensitivity of our data. This helps us deduce that our proposed MG-GAN approach is suitable for critical and sensitive application areas like medical data.

5.5 Results for precision and recall values

Precision and recall values are used to estimate the rightness of classification. The performance of a classifier hugely depends on the data being input to it. We have compared precision and recall value of deep neural

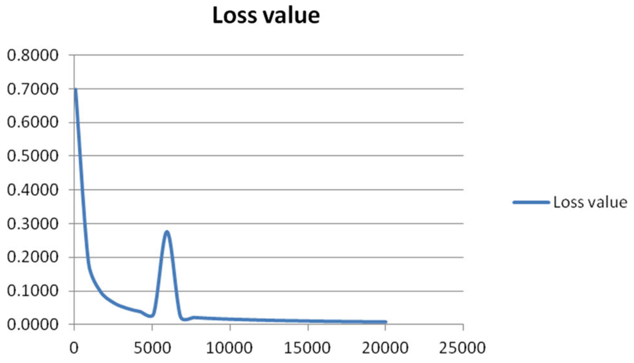


Fig. 4 Graphical representation of loss function across epochs

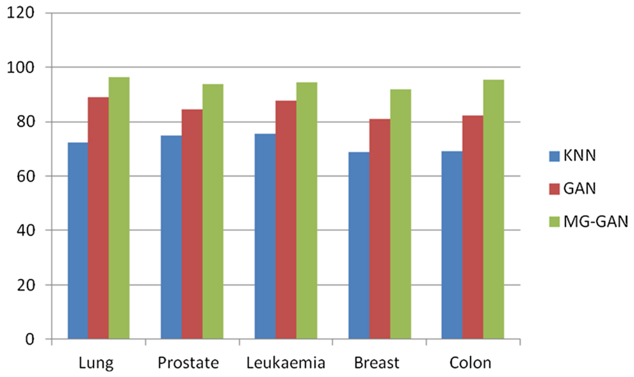


Fig. 5 Comparative recall values on dataset

Table 3 Comparative precision values on dataset (in %)

Dataset/ augmentation techniques	GDS	KNN–DNN classifier	GAN–DNN classifier	MG-GAN–DNN classifier
Lung	GDS2771	70.37	81.2	92.1
Prostate	GDS2545	71.25	78.48	91
Leukaemia	GDS2118	72.45	80.56	90.47
Breast	GDS27567	65.9	77.82	89
Colon	GDS3268	67.12	79.8	91.5

network classifier with data generated by three augmentation techniques, viz. *K*-nearest neighbourhood rule, Basic GAN and modified generator GAN. Deep learning classifier has been modelled for various applications (Khémiri et al. 2019). The DNN classifier model was generated using Keras library (https://keras.io/), tensor flow (https://www.tensorflow.org/) and scikit-learn library (https://scikit-learn.org/stable/). A four-layer DNN was constructed with ReLU activation function and ADAM optimization. The classifier configurations are the same throughout the experiments. The precision and recall value show the direct impact of augmentation techniques.

Table 4 Comparative classification accuracy (in %)

Cancer dataset	GDS	Original data–DNN classifier	KNN–DNN classifier	GAN–DNN classifier	MG-GAN–DNN classifier
Lung	GDS2771	68.21	72.3	80.2	88.1
Prostate	GDS2545	71.43	77.8	85.4	93.6
Leukaemia	GDS2118	62.34	69.8	78.8	88.4
Breast	GDS27567	69.8	73.3	81.8	90.3
Colon	GDS3268	67.59	72.9	79.8	91.7

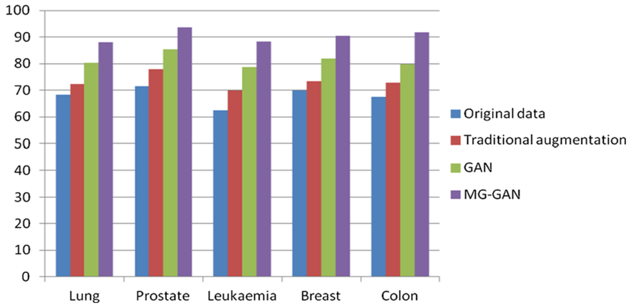


Fig. 6 Graphical representation of comparative classification accuracy

Recall value states the sensitivity of the synthetic data generated. As we deal with gene expression data for cancer classification, the synthetic data need to lie in close proximity to the original data. Figure 5 below shows the recall value on the datasets using the three augmentation techniques. We can clearly conclude that MG-GAN has a better recall value for all the datasets.

The precision values also state improvement with MG-GAN. Table 3 shows the precision value of the classifier, with the augmentation techniques.

5.6 Results for classification accuracy

We ran multiple tests for classification using different datasets for input:

- Original data,
- Data augmented using a traditional augmentation technique (k-nearest neighbourhood rule),
- Data augmented using Basic GAN,
- Data augmented using MG-GAN

An encouraging observation was that results achieved using our MG-GAN technique show significant improvement in accuracy. Table 4 depicts the comparative results for classification accuracy.

The results after our contribution (MG-GAN) showed improvements in accuracy, as seen above Fig. 6:

- Between 15.8% and 18.6% as compared to traditional augmentation techniques
- Between 7.8% and 11.9% as compared to traditional GAN

6 Conclusions and future directions

Sensitive real-life applications like medical data and spatial temporal data consist of data which is disproportionate with respect to the number of samples versus the number of features. Obtaining this data is strenuous and expensive. Analysing gene expression data using limited samples could yield inappropriate results; thus, data need to be augmented to generate synthetic samples and increase the size of the dataset. However, to ensure effective performance, these generated samples need to be very close in behaviour to the original samples. Three different approaches for augmentation were applied viz. *k*-nearest neighbourhood rule (used traditionally for data augmentation), Basic GAN and MG-GAN. The loss function was examined, and it showed significant reduction from 0.6978 to 0.0082. Lower loss value indicates the proximity of the synthetic samples and original ones. The precision and recall values are also high, which conclude that the generated data are sensitive as well as specific to the original data. Classification accuracy improved to 93.6% with MG-GAN. We therefore conclude that GAN and its variants, specifically MG-GAN, can be successfully used for augmenting data and generating synthetic samples which are as good as the original samples. Our approach MG-GAN is suited for critical applications which deal with insightful and sensitive data.

Our vision for the future includes working on selection of the most relevant feature subset to further improve prediction accuracy and reduce execution time.

Funding There are no funding agencies involved in this research.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent The authors declare that they have no consent.

References

Antipov G, Baccouche M, Dugelay JL (2017) Face aging with conditional generative adversarial networks. In: 2017 IEEE international conference on image Processing (ICIP), Beijing, China, pp 2089–2093

Antoniou A, Storkey A, Edwards H (2017) Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340

Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: International conference on machine learning, pp 214–223

Chan S, Elsheikh AH (2017) Parametrization and generation of geological models with generative adversarial networks. arXiv preprint arXiv:1708.01810

Chaudhari P, Agarwal H (2018) Improving feature selection using elite breeding QPSO on gene data set for cancer classification. In: Intelligent engineering informatics, advances in intelligent systems and computing book series, vol. 695, pp. 209–219

Chaudhari P, Agarwal H (2019) Data augmentation for cancer classification in oncogenomics: an improved KNN based approach. Evol Intell. https://doi.org/10.1007/s12065-019-00283-w

Chen X, Yu J, Kong S, Wu Z, Fang X, Wen L (2017) Towards quality advancement of underwater machine vision with generative adversarial networks. arXiv preprint arXiv:1712.00736

Collins F (2002) Oncogenomics: cancer and technology. Nat Genet 31:117–119

Creswell A, Bharath AA (2018) Inverting the generator of a generative adversarial network. IEEE Trans Neural Netw Learn Syst 30(7):1967–1974

Deng X, Zhu Y, Newsam S (2018) What is it like down there?: generating dense ground-level views and image features from overhead imagery using conditional generative adversarial networks. In: Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems, Seattle, Washington, pp 43–52

Deverall J, Lee J, Ayala M (2017) Using generative adversarial networks to design shoes: the preliminary steps. CS231n in Stanford. http://cs231n.stanford.edu/reports/2017/pdfs/119.pdf

Dutt RK, Premchand P (2017) Generative adversarial networks (GAN) review. CVR J Sci Technol 13:1–5

Eghbal-zadeh H, Widmer G (2017) Likelihood estimation for generative adversarial networks. arXiv preprint arXiv:1707.07530

Frid-Adar M, Klang E, Amitai M, Goldberger J, Greenspan H (2018) Synthetic data augmentation using GAN for improved liver lesion classification. In: 2018 IEEE 15th international symposium on biomedical Imaging (ISBI 2018), Washington, DC, USA, pp 289–293

Gharakhanian A (2017) Generative adversarial networks—hot topic in machine learning. http://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html

Ghasedi DK, Wang X, Huang H (2018) Semi-supervised generative adversarial network for gene expression inference. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining, London, UK, pp 1435–1444

Gong M, Niu X, Zhang P, Li Z (2017) Generative adversarial networks for change detection in multispectral imagery. IEEE Geosci Remote Sens Lett 14(12):2310–2314

Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Bengio Y (2014) Generative adversarial networks. Adv Neural Inf Process Syst 3:2672–2680

Gurumurthy S, Kiran Sarvadevabhatla R, Venkatesh Babu R (2017) Deligan: generative adversarial networks for diverse and limited data. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 166–174

Huang X, Li Y, Poursaeed O, Hopcroft J, Belongie S (2017) Stacked generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, vol 1, pp 5077–5086

Hui J (2018) GAN—whats generative adversary networks GAN? https://medium.com/@jonathan_hui/gan-whats-generative-adversarial-networks-and-its-application-f39ed278ef09

Huszár, F (2015) How (not) to train your generative model: scheduled sampling, likelihood, adversary?. arXiv preprint arXiv:1511.05101

Khémiri A, Echi AK, Elloumi M (2019) Bayesian versus convolutional networks for arabic handwriting recognition. Arab J Sci Eng 44(11):9301–9319

Konidaris F, Tagaris T, Sdraka M, Stafylopatis A (2018) Generative Adversarial Networks as an Advanced Data Augmentation Technique for MRI Data. IEEE Trans Med Imaging 37(3):673–679

Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Shi W (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, pp 4681–4690

Li J, Madry A, Peebles J, Schmidt L (2017) On the limitations of first-order approximation in GAN dynamics. arXiv preprint arXiv:1706.09884

Li D, Chen D, Goh J, Ng SK (2018) Anomaly detection with generative adversarial networks for multivariate time series. arXiv preprint arXiv:1809.04758

Li Y, Xiao N, Ouyang W (2018b) Improved boundary equilibrium generative adversarial networks. IEEE Access 6:11342–11348

Li J, He H, Li L, Chen G (2019) A novel generative model with bounded-gan for reliability classification of gear safety. IEEE Trans Industr Electron 66(11):8772–8781

Liu F, Jiao L, Tang X (2019a) Task-oriented GAN for PolSAR image classification and clustering. IEEE Trans Neural Netw Learn Syst 30(9):2707–2719

Liu Y, Zhou Y, Liu X, Dong F, Wang C, Wang Z (2019b) Wasserstein GAN-based small-sample augmentation for new-generation artificial intelligence: a case study of cancer-staging data in biology. Engineering 5(1):156–163

Lu Y, Kakillioglu B, Velipasalar S (2018) Autonomously and simultaneously refining deep neural network parameters by a bi-generative adversarial network aided genetic algorithm. arXiv preprint arXiv:1809.10244

Luc P, Couprie C, Chintala S, Verbeek J (2016) Semantic segmentation using adversarial networks. arXiv preprint arXiv:1611.08408

Lucas A, Lopez-Tapiad S, Molinae R, Katsaggelos AK (2019) Generative adversarial networks and perceptual losses for video super-resolution. IEEE Trans Image Process 28(7):3312–3327

Matlab Documentation Classification using Nearest neighbours (2019). https://ch.mathworks.com/help/stats/classification-using-nearest-neighbors.html

Marchesi M (2017) Megapixel size image creation using generative adversarial networks. arXiv preprint arXiv:1706.00082

Marouf M, Machart P, Magruder DSS, Bansal V, Kilian C, Krebs CF, Bonn S (2018) Realistic in silico generation and augmentation of single cell RNA-seq data using Generative Adversarial Neural Networks. bioRxiv 390153

Metz L, Poole B, Pfau D, Sohl-Dickstein J (2016) Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163

Mustafa M, Bard D, Bhimji W, Lukić Z, Al-Rfou R, Kratochvil JM (2019) CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks. Comput Astrophys Cosmol 6(1):1

Namozov A, Im Cho Y (2018) An efficient deep learning algorithm for fire and smoke detection with limited data. Adv Electr Comput Eng 18(4):121–129

Oliehoek FA, Savani R, Gallego J, van der Pol E, Groß R (2018) Beyond local nash equilibria for adversarial networks. arXiv preprint arXiv:1806.07268

Pan Z, Yu W, Yi X, Khan A, Yuan F, Zheng Y (2019) Recent progress on generative adversarial networks (GANs): a survey. IEEE Access 7:36322–36333

Quan TM, Nguyen-Duc T, Jeong WK (2018) Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss. IEEE Trans Med Imaging 37(6):1488–1497

Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434

Shang C, Palmer A, Sun J, Chen KS, Lu J, Bi J (2017) VIGAN: Missing view imputation with generative adversarial networks. In: 2017 IEEE international conference on big data (big data), Boston, MA, USA, pp 766–775

Tembine H (2019) Deep learning meets game theory: Bregman-based algorithms for interactive deep generative adversarial networks. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2018.2886238

Vertolli MO, Davies J (2017) Image quality assessment techniques show improved training and evaluation of autoencoder generative adversarial networks. arXiv preprint arXiv:1708.02237

Wan G et al (2018) Spatiotemporal regulation of liquid-like condensates in epigenetic inheritance. Nature 557:679–683. https://doi.org/10.1038/s41586-018-0132-0

Wang X, Ghasedi Dizaji K, Huang H (2018) Conditional generative adversarial network for gene expression inference. Bioinformatics 34(17):i603–i611

Wang C, Xu C, Yao X, Tao D (2019) Evolutionary generative adversarial networks. IEEE Trans Evol Comput 23(6):921–934

Weng L (2017) From GAN to WGAN. https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html

Wu D, Rice CM, Wang X (2012) Cancer bioinformatics: a new approach to systems clinical medicine. BMC Bioinf 13(1):71

Xuan Q, Chen Z, Liu Y, Huang H, Bao G, Zhang D (2018) Multiview generative adversarial network and its application in pearl classification. IEEE Trans Industr Electron 66(10):8244–8252

Yu B, Zhou L, Wang L, Shi Y, Fripp J, Bourgeat P (2019) Ea-GANs: edge-aware generative adversarial networks for cross-modality MR image synthesis. IEEE Trans Med Imaging 38(7):1750–1762

Zhu L, Chen Y, Ghamisi P, Benediktsson JA (2018) Generative adversarial networks for hyperspectral image classification. IEEE Trans Geosci Remote Sens 56(9):5046–5063