BRAIN. Broad Research in Artificial Intelligence and Neuroscience

e-ISSN: 2067-3957 | p-ISSN: 2068-0473

Covered in: Web of Science (ESCI); EBSCO; JERIH PLUS (hkdir.no); IndexCopernicus; Google Scholar; SHERPA/RoMEO; ArticleReach Direct; WorldCat; CrossRef; Peeref; Bridge of Knowledge (mostwiedzy.pl); abcdindex.com; Editage; Ingenta Connect Publication; OALib; scite.ai; Scholar9; Scientific and Technical Information Portal; FID Move; ADVANCED SCIENCES INDEX (European Science Evaluation Center, neredataltics.org); ivySCI; exaly.com; Journal Selector Tool (letpub.com); Citefactor.org; fatcat!; ZDB catalogue; Catalogue SUDOC (abes.fr); OpenAlex; Wikidata; The ISSN Portal; Socolar; KVK-Volltitel (kit.edu)

2024, Volume 15, Issue 3, pages: 208-222.

Submitted: August 18th, 2024| *Accepted for publication:* October 1st, 2024

CGAN Facilitated Data Augmentation of Voice and Speech Parameters for Detecting Parkinson's Disease in the Prodromal Phase

Sandhya Chandrabhanu [1]
Shanmugam Hemalatha [2]

[1] Department of Computer Science, Karpagam Academy of Higher Education,
Coimbatore, Tamil Nadu, India,
horlogue@gmail.com
[2] Department of Computer Science, Karpagam Academy of Higher Education,
Coimbatore, Tamil Nadu, India,
drhemashanmugham@gmail.com,
hemalatha.s@kahedu.edu.in

Abstract: *Parkinson's disease is a multi-faceted disease affecting the brain. The enormity of its recent rise is quite alarming. This calls for intense research to diagnose early to hasten the progress of diagnosis. Voice distortion is considered an early precursor for Parkinson's disease. Though several studies in Machine Learning using voice parameters have provided useful information, none of them have been successful in evolving an efficient and generalized model to detect it. Deep Learning techniques were applied to improve the performance of the model but its major limitation was the size of the dataset. Hence, a need arose to extend the dataset using an appropriate data augmentation method. At this juncture, the conditional generative adversarial network (CGAN) proved to be a useful technique because of its innate feature for generating synthetic data from input noise. The RNN-LSTM classifier could achieve a training accuracy of 87.32%, testing accuracy of 86.3%, training precision of 87.92 %, and testing precision of 89.94%. The results of the experimental study are compared with other state-of-the-art methods. This technique succeeded in reducing the problem of over-fitting and could elevate the performance of the RNN-LSTM classifier in the prediction of Parkinson's disease.*

Keywords: *Parkinson's disease; machine learning; deep learning; data augmentation; conditional generative adversarial network; recurrent neural networks; long short-term memory*

How to cite: Chandrabhanu, S., & Hemalatha, S. (2024). CGAN facilitated data augmentation of voice and speech parameters for detecting Parkinson's disease in the prodromal phase. *BRAIN: Broad Research in Artificial Intelligence and Neuroscience*, 15(3), 208-222. https://doi.org/10.70594/brain/15.3/16

©2024 Published by EduSoft Publishing. This is an open access article under the CC BY-NC-ND license

1. Introduction

Parkinson's disease is a multifaceted neurodegenerative disease affecting more than ten million population worldwide. Despite extensive research, a proper cure for this disease has not been identified (Mughal et al., 2022). The pathophysiology of this disease is a reduction in the neurotransmitter dopamine in a specific area called substantia nigra in the midbrain. The impairment of dopaminergic systems is responsible for the PD symptoms, with the depletion of dopaminergic neurons that causes a wide range of motor and non-motor symptoms.

Motor symptoms (Bhowmick et al., 2020) include tremors, stiffness in the extremities of the body, postural imbalances, movement disorders, etc. which show that the disease is progressing. The non-motor symptoms (Todorova et al., 2014) include speech and communication problems, olfactory disturbances, sleep disorders, cognitive impairments, and dementia. However, diagnosing PD based solely on qualitative criteria can be challenging since other diseases share similar symptoms.

Diagnosis of PD by clinicians depends on a combination of clinical symptoms and diagnostic tests. The diagnosis of PD is confirmed by a significant persistent effect of dopaminergic therapy. Transcranial Sonography (TCS), Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and Single-Photon Emission Computerized Tomography (SPECT) (Trifonova et al., 2020; Mortezazadeh et al., 2021), etc., are the widely used non-invasive, diagnostic imaging techniques.

Even though the current treatment helps diagnose the symptoms, they do not reduce or curb the progression of the disease. The motor symptoms manifest as the disease progresses. If the non-motor symptoms can be identified earlier further progression of the disease can be prevented. Therefore, the focus is made on identifying the disease at an early stage where the non-motor symptoms may prove to be useful. Hence, the researchers are focusing their attention on both ways to spot the non-motor symptoms that manifest at an early stage and have the potential to delay the progression of the disease (Ehgoetz et al., 2018). The methods currently in vogue for diagnosing PD are a bit cumbersome. It involves invasive techniques like Deep Brain Stimulation (DBS) that are very expensive. Studies have revealed that PD can precede the development of non-motor symptoms and about 90% of PD patients experience voice disorders (Sakar, et al., 2010). Voice recordings provide an effective non-invasive diagnostic tool because PD patients exhibit distinct vocal features.

The methods that are in practice rely on machine learning models using voice data due to their simplicity and non-invasive methods of acquiring data. The objective metrics for the detection of speech changes in PD occur before the overt motor symptoms. This presents a promising avenue of research in the detection of PD at a prodromal stage (Iyer, et. al., 2023; Postuma, et al., 2016). Even though they are useful in predicting the disease their effectiveness is questionable. This calls for further innovative techniques using deep learning technologies to be introduced which can deal with multi-dimensional and semi-structured data that cannot be analyzed effectively with machine learning algorithms (Gupta, et al., 2023).

Several deep neural networks and hybrid models were proposed and tried in an attempt to improve performance and scalability. Even though the deep learning models outperformed the traditional machine learning models they suffered from a major problem of over-fitting. This normally occurs due to a lack of clean data for training, models with high variance, limited training data, and the complexity of the deep neural networks. The experimental studies conducted using a simple Artificial Neural Network, Recurrent Neural Network with Long-Short Term Memory, and Gated Recurrent Unit Networks with the Parkinson's dataset revealed that the dataset used for training the model was limited (Rehman, et al., 2023). Hence, it necessitated the need to augment data to yield better and more accurate results. Many data augmentation techniques can be employed but none of them produce new data samples. This paved the way to choose a special type of framework for approaching generative artificial intelligence namely GAN (Generative Adversarial

Network) to be tested for this study.

Generative Adversarial Networks (Li et al., 2021) offer a novel method for data augmentation that produces new data samples. There has been a renewed interest in GANs owing to their versatility and ease with which they can be applied in various domains. GAN has been extensively studied in a multitude of domains such as music, computer vision, and arts. It takes random noise from a latent space and produces data that mimics the feature distribution of the original dataset. GAN consists of two networks, a Generator G(x), and a Discriminator D(x).

They both play an adversarial role where the generator tries to deceive the discriminator by generating data that resembles the training set, while the discriminator avoids being deceived by the generator by identifying the fake or synthetic data from the actual data. They work in tandem and get trained with high-dimensional data like video, images, and audio.

Since the study was based on binary classification, we need to create generated data samples belonging to two different classes. Hence, a special type of GAN model called Conditional GAN (CGAN) was proposed for this study. After applying sampling using uniform or normal distribution, the random noise is passed to the generator along with a conditional vector that represents the class. The generator creates synthetic or fake data which mimics the original dataset. The output of the generator is passed to the discriminator along with the real data from the original dataset and the conditional vector. The discriminator learns how to classify real and fake data. The output D(x) is the probability that the input is real or fake. If the input is real, the D(x) would give output 1 and if it is generated, the D(x) would give output 0.

Our proposed model is designed to achieve the following objectives:

1. This study adopted a resampling technique to balance the highly imbalanced Parkinson's disease dataset. Moreover, with these techniques, the problem of model over-fitting can be solved and its overall performance improves.
2. Data augmentation is incorporated by utilizing CGAN, which to a large extent can reduce the problem of over-fitting in deep neural networks. This is made possible by extending the small dataset which when applied to deep neural networks limits the performance of the model in predicting the results.
3. The comparison of our model with other state-of-the-art deep learning models are carried out.

Further, the paper is structured as follows: II. Related work, III. Materials and Methods, IV. Working Methodology, V. Results and Discussion, VI. Conclusions and VII. Limitations of the study and future perspectives.

2. Related work

Generative Adversarial Networks (GANs) are a novel architecture that can produce authentic data that closely correlates with the training data they are exposed to. GAN is composed of two neural networks, namely a generator and discriminator which engage in a competitive gaming environment (Pradhyumna, & Mohana., 2022).

Even though its utilization in the health care sector is rising steadily it is not keeping pace with the growth in other sectors. It encompasses a wide range of applications in this area according to the study (Karras, et al., 2020). One of the key benefits of using GAN in the medical field is its capacity to produce artificial healthcare data that closely emulates the original data. This trait of GAN carries paramount importance in safeguarding data privacy and maintaining confidentiality in the healthcare sector (Ghosheh, et al., 2022).

The generation of synthetic medical data without losing confidentiality, preserving data privacy, and its capability to extend the availability of data make it more appealing in the healthcare industry. Issues like patient consent, data privacy, and the risk of using synthetic data in decision-making processes are not pertinent at this juncture of this study, since it has been retrieved from a secondary source(Sakar, et al., 2018). The utilization of simulated environments helps researchers in designing authentic and regulated settings for the evaluation of algorithms and

models. This obviates the need for obtaining genuine patient data.

Data augmentation is a widely employed technique in the field of data science. It helps in expanding the size of available data and increasing the diversity of a given dataset through the application of different techniques in the existing data (Garcea, et al., 2022).

Revelations were made in the study (Kamran, et al., 2021) for the early detection of Parkinson's Disease involving deep generative adversarial networks. In another work (Karras, et al., 2020) an attempt was made to train GAN using accelerometer data collected from wearable sensors. The study showed that this network can accurately record movement patterns that serve as indicators of Parkinson's disease. This results in timely identification and intervention.

In the study (Kaur, et al., 2021) analysis of voice samples was used to detect Parkinson's Disease in an early stage. The approach used in this work was based on GAN combined with a deep neural network (DNN). In this study, GAN was proposed to create a synthetic data package for the numerical dataset and to produce a classifier using the hybrid dataset. Experimental tests have shown more acceptable results than conventional approaches. Using GAN-based augmentation an increase of 11.68% could be achieved compared to the traditional methods.

In the study (Peppes et al., 2023) the FoGGANs architecture proved to be a very useful tool for data augmentation in the context of PD by generating realistic Parkinson's Disease freezing of gait dataset. This could address the problem of data shortage in many studies related to neurodegenerative diseases.

The results obtained from various studies emphasize the potential of Generative Adversarial Networks (GANs) in the prediction and identification of Parkinson's disease. This is made possible by exploiting the innate property of data augmentation of GANs. This in turn helps in extending the dataset by adding synthetic or generated data to the original one. By applying this newly accrued dataset to deep learning models we can exhilarate the performance of these models. Thus, it paves the way for the creation of a comprehensive predictive model, which acts as an impetus for carrying out this study.

3. Materials and Methods

3.1. Data Collection

The dataset is obtained from the Department of Neurology in Cerrahpasa, Faculty of Medicine of Istanbul University which was made available from the UCI Learning repository (Sakar, et al., 2018). It contains a total of 252 cohorts out of which 188 are PD patients and 64 are healthy individuals. There are about 754 total attributes. The subjects of the study were directed to repeat the vowel /a/ for three consecutive periods. A microphone with 44.1KHZ frequency was used to record this sustained phonation.

The physicians validated this data by applying various signal processing algorithms such as Time-Frequency Features, Mel Frequency Cepstral Coefficients (MFCCs), Wavelet Transform Features, Vocal Fold Features, and TWQT features have been applied to the speech recordings of Parkinson's Disease (PD) patients to extract clinically useful information for PD assessment. The signal processing algorithms converted the voice signals into different classes of features or attributes which when subjected to machine learning, deep learning algorithms can help researchers in finding useful insights. The dataset thus created contained different classes of features which were termed as baseline and advanced. Some of the advanced classes of features included Vocal Fold, Mel Frequency Cepstral Coefficients (MFCCs), Tunable Q- factor Wavelet Transform (TWQT), etc. To extract the baseline features, a special acoustic analysis software named Praat was used (Hoq, et al., 2021).

3.2. Data Preparation and Preprocessing

The preparation and pre-processing of data is a very important phase that needs to be implemented carefully before we create the model and train it. Here, the data were processed by checking the missing values and null values. For normalizing the values in the dataset, the Standard

Scalar function was employed. Correlation was applied to reduce the dimensionality problem. This is achieved by creating a correlation matrix that identifies the parameters that have a high degree of correlation that need to be eliminated by reducing the features in the dataset.

When features in a dataset are highly correlated (with a correlation coefficient greater than 0.6 in this case), it indicates that these features convey similar information. Keeping both features doesn't add significant new information but can increase computational complexity and the risk of overfitting. Identifying and removing one feature from each highly correlated pair reduces redundancy, diminishes overfitting, and enhances efficiency by decreasing computational workload by speeding up the model training and prediction process. Overfitting in neural network means that the model memorizes the training data instead of learning general patterns resulting in poor performance on new unseen data. The concept of overfitting in neural networks was first postulated by Minsky, M. & Papert, S. A. in 1969 in their paper titled "Perceptrons: An introduction to Computational Geometry" (Minsky & Papert, 1969). In essence, by dropping highly correlated features, we can streamline our dataset to contain the most relevant and distinctive information, optimizing model performance and interpretability.

A heat map was generated with the help of the correlation coefficients with values ranging from -1 to +1. A correlation heat map is a graphical representation that displays the correlation between multiple variables as a color-coded matrix. Positive correlations are represented by brighter colors indicating that when one variable increases, the other tends to increase as well. Negative correlations are shown with darker colors, suggesting that when one variable increases, the other tends to decrease. This visualization is particularly useful for identifying highly correlated or inversely correlated variables.

Resampling is done to the resultant matrix obtained after applying correlation. Since the majority of the dataset consisted of PD patients compared to healthy individuals a uniform distribution method was applied to balance it. A resampling technique was applied for feature reduction. This could reduce the features to 384 records and 197 attributes. This reduction aimed to enhance model efficiency and interpretability by focusing on the most informative features while discarding redundant or less significant ones. The dataset used for the study is an imbalanced one with two classes – class 1 (with PD) and class 2 (without PD). Out of which class 1 had more matching records. To overcome this problem of imbalanced datasets, a resampling technique called stratified cross-fold validation (Ron et al.,1995) is applied. It ensures that each fold has the same proportion of classes as the original dataset. This leads to more reliable and consistent performance metrics across folds. By maintaining the class distribution in each fold, the model is exposed to a similar class balance in training and validation, making it more likely to generalize well to unseen data. This helps to avoid overfitting to a particular class distribution and ensures that the model's performance metrics (e.g., accuracy, precision) are more representative of its performance on the actual distribution of the target variable. This consistency is crucial for reliable model evaluation.

4. Working methodology

4.1. CGAN Architecture

The architecture of CGAN is composed of two neural networks, the generator, and the discriminator as shown in (see Figure 1). They are always competing with each other in a game-like pattern. The goal of the generator is to create synthetic data which mimics real data whereas the discriminator assesses the veracity of the data which were fed into it.

Essentially, generative models create their training data. When training begins, the generator produces fake data by taking noise as input from latent space along with the class or category as a conditional vector. Artificial intelligent systems process input data, identify patterns and relationships, and then organize this information in latent space for easy access. This helps these systems in making better predictions, generate new data, or classify information efficiently. The output of the generator is connected directly to the discriminator's input. The discriminator also

receives instances from the original dataset and the conditional vector during its training period. As training progresses, the generator gets closer in producing output that can deceive the discriminator. The discriminator learns not to be fooled by the generator's synthetic data from real data. The generator and discriminator loss has to be reduced to build an efficient CGAN model. The discriminator after classification sends a signal to the generator through backpropagation to update its weights. As the feedback loop between the adversarial networks continues, the generator will begin producing high-quality and more realistic output and the discriminator will become better in classifying the real and fake ones.

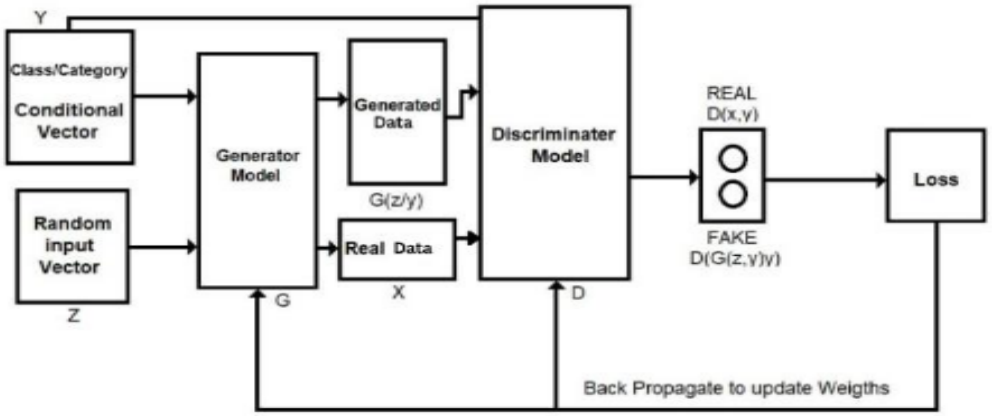


Figure 1. Architecture of CGAN Model

4.2. Implementation of CGAN

The CGAN model architecture consists of two sub-models: a generator and a discriminator. The generator model is used to create synthetic data from the problem domain whereas the discriminator model is used to classify real data from fake (generated) ones. The CGAN training algorithm trains both generator and discriminator models separately.

The dataset involved in the study is obtained from the UCI machine learning repository (Sakar, et al., 2018) which has voice attributes of patients suffering from Parkinson's disease and healthy cohorts. The initial speech samples were divided into training and test sets. To build a CGAN model we created separate generative and discriminative models and combined it to form the basic CGAN architecture. The trained CGAN generator produces synthetic samples using noise from latent space and conditional vectors that represent the class or category. The real samples from the training set, conditional vectors along with the synthetic samples from the generator were fed to the discriminator. It has to properly filter out these samples. Both models were combined to build the CGAN that produces the synthetic data that closely resembles the original data, thus helping in achieving data augmentation. The output of CGAN is used for classifying PD with the help of a classifier. In this model, we have used RNN-LSTM as the classifier.

4.2.1. Generator network

The generator network can be defined as a function G: (Z/Y) → $\bar{x}$, which has as input data (random noise) z e Z and the conditional vector Y that produces an output $\bar{x}$ e $\bar{X}$. The generator network G is an artificial neural network, which takes random noise as input (input dimension or l-dimension) along with the conditional vector to generate synthetic data samples as output. Leaky ReLU was used as an activation function and batch normalization was applied at each layer except the last one which uses tanh for activation.

Algorithm: The Generator network

1. Input: - Random noise (Z), conditional vector(Y)
   Output : Synthetic data similar to actual dataset $\bar{x}$
2. define_generator(l_dim)
3. for i in the range do
4. model.add(dense(nodes,l_dim=100)
5. model.add(LeakyReLU)
6. model.add(BatchNormalization)
7. model.add(dense(1024))
8. model.add(LeakyReLU)
9. model.add(dense(nodes, tanh))
10. return generator model

4.2.2. Discriminator Network

The discriminative network implements a function:

D: (x1,x,y) → [0,1]. This network takes input as the generated data x1 e $\bar{X}$ (from the generator), x e X. (real data set), y e Y (conditional vector) and gives output as a binary value [0,1] deciding whether the data is real or generated. The D(x,y) shows the real data as output and D(G(z,y),y)) shows the fake data as output. Leaky ReLU was used as an activation function in all layers except the output layer which uses sigmoid as an activation function. The loss function used in the model is binary cross-entropy which was used as output of the model and has a probability of 0 or 1. Binary cross entropy is a loss function used in machine learning, particularly for binary classification problems. It measures the difference between predicted probabilities and actual labels. The concept of cross-entropy originated in information theory introduced by Claude Shannon in his 1948 paper titled "A Mathematical Theory of Communication" (Shannon, et al., 1948). This method was popularized in the 90's for neural networks, particularly in the context of logistic regression. Some notable researchers who contributed to its development and applications in machine learning include David Rumelhart, Geoffrey Hinton, and Ronald Williams in the year 1986 (Rumelhart, et al., 1986).

Algorithm: The Discriminator network

1. Input: Synthetic data created by the generator network($\bar{x}$),real data (x) and conditional vector (y)
   Output: Class label 1 for original data and 0 for generated data
2. define discriminator(X)
3. for i in the range do
4. model.add(dense(1024,X))
5. model.add(LeakyReLU)
6. model.add(dense(512))
7. model.add(LeakyReLU)
8. model.add(dense(1, sigmoid))
9. return discriminator model

4.2.3. Building CGAN

In this, a CGAN is created by stacking the generator and discriminator networks. First, we set the trainable parameter of the discriminator network to false. This helps in freezing the weights in the discriminator network while the generator network is trained. This prevents the discriminator from being updated while the generator creates new samples using noise and the conditional vector. The input dimension or shape passed to the CGAN network is the shape of noise, which is passed to the generator. The generator's output is fed to the discriminator, which classifies the data as original or fake. Finally, the CGAN produces synthetic samples which mimic the original data.

Algorithm: Building CGAN model

1. Input : generator, discriminator
   Output : generated data mimicking original data
2. define build_cgan(generator, discriminator)
3. set discriminator.trainable =false
   cgan_inp =Input(input_dim=100)
4. x= generator(cgan_inp)
5. cgan_out = discriminator(X)
6. gan model= call build_cgan(input=cgan_inp, outputs=cgan_out)
7. return cgan model

4.2.4. Classifier for Prediction

The output of CGAN consisting of generated data that mimics the real data is combined with the original dataset to create a new hybrid dataset which is then fed to the RNN -LSTM classifier to distinguish PD patients from healthy cohorts.

Algorithm: Deep Learning Classifier using RNN (LSTM)

1. Input: Combined dataset created using synthetic data
   generated by the CGAN model and real dataset is
   passed as input_shape
   Output: Class label 1 for PD patients and 0 for healthy cohorts
2. model = sequential () // deep learning classifier initialization
3. model.add(LSTM(64, input_shape))
4. model.add(SpatialDropoutID(0.2))
5. model.add(LSTM(32,dropout=0.2,recurrent-dropout=0.2))
6. model.add(dense(1, sigmoid))
7. final_model = model.fit(input_shape)
8. predicted_result = final_model. predict(test_sample)
9. return predicted_result

4.2.5. Training CGAN model

The discriminator is designed to correctly classify the actual and synthetic data. This is made possible by maximizing the log of the inverted probability of fake data and the log of the predicted probability of real data. These are averaged over each mini-batch of sample data. The loss function of the discriminator searches for probabilities that are close to 1.0 for actual data and probabilities close to 0.0 for synthetic data which then invert and become larger numbers. The addition of these values indicates that a lower average value of this loss function can lead to better performance of the discriminator. The generator is trained with the intention of stimulating the discriminator to predict the generated data to be real with high probability. This is done by adjusting the generator's weights through the discriminator by setting class labels of 1 for the generated data. No changes are made to the discriminator throughout this process. The gradient information required to adjust the generator's weights is passed as input. For example, if the discriminator predicts a low average probability for the batch of synthetic data, this results in large error signal due to high generator loss being propagated backward into the generator as the expected probability value is 1.0 for the real samples. These large error signals indicate the generator to improve its prediction rates by adjusting the weights and enhancing its ability to generate fake samples that closely resemble real ones in the subsequent batches. The training of the generator and discriminator continues until it reaches a point of saturation, where no further improvement in the performance of either neural network is expected. At this point, the loss of generator and discriminator would have been reduced to a bottom level which is not subject to changes. This marks the end of CGAN's training. The results obtained conform to the real data. This generated data is combined with the original data and can be fed to any classifier to predict the outcome. In

this study RNN -LSTM is implemented for the classification of PD patients from healthy cohorts.

4.2.6. CGAN Loss Function

In the paper by Ian Goodfellow (Goodfellow et al., 2014) the standard GAN loss function was described initially which can be represented as follows

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))] \quad (1)$$

The above equation represents the standard loss function of GAN. For our model CGAN the objective function can be denoted as shown below

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))] \quad (2)$$

Here E is used to indicate the expected value of a random variable.

$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}$ - denotes the expected value with respect to actual data distribution $p_{\text{data}}(\boldsymbol{x})$

$\mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}$ - denotes the expected value with reference to the prior noise distribution $p_z(z)$

The aim is to simultaneously minimize the generator's ability to deceive the discriminator and maximize the discriminator's ability to accurately classify real and synthetic samples.

The term [log D(x|y)] encourages the discriminator to precisely classify real samples whereas the term [log(1-D(G(z|y))] encourages the generator to create samples that are accurately classified as real by the iscriminator. This creates a balance in which the generator can refine its ability to generate realistic samples, and the discriminator becomes more adept at classifying between real and generated samples. It can be further classified into two parts – the discriminator loss and the generator loss.

4.2.7. Discriminator Loss Function

The discriminator is trained to distinguish both the actual data from the dataset and the synthetic data from the generator. The discriminator's loss is a measure of how well it can be differentiated between the two. The most popular loss function for the discriminator in a CGAN is the binary cross-entropy loss. It compares the discriminator's predictions with the true labels (0 for real samples and 1 for generated samples) and calculates the loss accordingly. The formula for binary cross-entropy loss is as follows:

loss = -sum(y_true * log(y_pred) + (1 – y_true) * log(1 - y_pred)) (3)

Here, y_true represents the true labels and y_pred represents the discriminator's predictions. The aim is to minimize this loss, which indicates that the discriminator is becoming more accurate in distinguishing
between real and generated samples.

4.2.8. Generator Loss Function

The generator aims to create realistic samples that can deceive the discriminator. Its loss is inversely related to the discriminator's loss. In other
words, the generator tries to minimize the discriminator's ability to distinguish between actual and generated samples. The generator's loss in a CGAN is calculated using binary cross-entropy. However, the labels for the generated samples are inverted compared to the discriminator's loss. The

formula for the generator's loss is as follows:

$$loss = -sum((1 - y_true) * log(y_pred)) \quad (4)$$

Here, y_true represents the inverted labels for the generated samples, and y_pred represents the discriminator's predictions for those samples. By minimizing the loss, the generator learns to generate samples that have more probability to be categorized as real by the discriminator. Based on the the ability of the discriminator to classify the real and fake data, the probability score of either 0 (fake) or 1(real) is determined. This probability of prediction by discriminator is used to calculate the generator loss. If the generator can successfully deceive the discriminator, it will be rewarded, if not it will be penalized. The generator gets trained by adjusting the parameters to maximize the log of the discriminator probabilities. Eventually, the generator tries to maximize the probability of the data being real rather than minimizing the probability of the data being fake. The generator tries to minimize the GAN loss function meanwhile the discriminator tries to maximize it.

5. Results and Discussions

5.1. Performance Analysis

The adoption of a Generative Adversarial Network (CGAN) for Parkinson's disease prediction represents an innovative approach, integrating generative and discriminative modelling within a unified framework. The CGAN architecture harnesses deep learning capabilities to generate synthetic data, enhancing overall model performance. This work addresses the limited size of the Parkinson's disease dataset by using Conditional Generative Adversarial Networks (CGAN) to generate fake data. The data generated is combined with the original dataset to create a large and more diverse dataset. Subsequently, a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) units is implemented for the predictive analysis of Parkinson's disease. The trained model is evaluated for its accuracy in predicting the disease, potentially contributing to improved understanding and management of Parkinson's disease.

Accuracy and precision were used as the performance metrics to measure the model's ability to correctly classify PD cases. The dataset obtained on applying CGAN is used for augmenting the original dataset and the combined dataset was used to train the RNN-LSTM classifier to yield a good precision score of 0.8792 for training and 0.8994 for testing. The following plot (see Figure 2) shows the precision of the model obtained during the training and testing phases. The plot (see Figure 3) shows the accuracy scores of the model on applying CGAN.

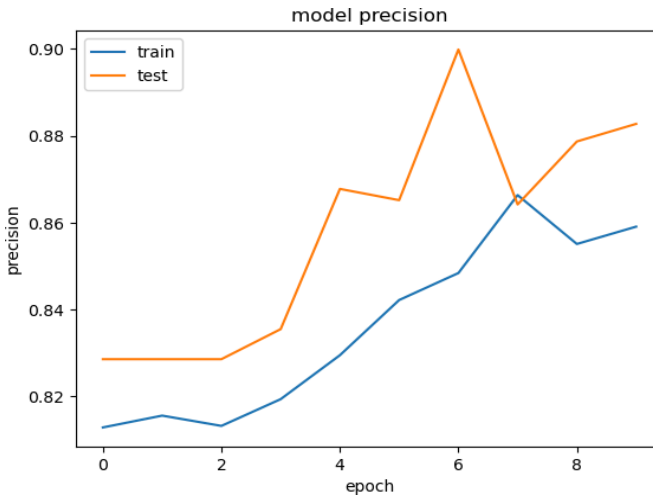


Figure 2. Performance of the model based on precision with CGAN

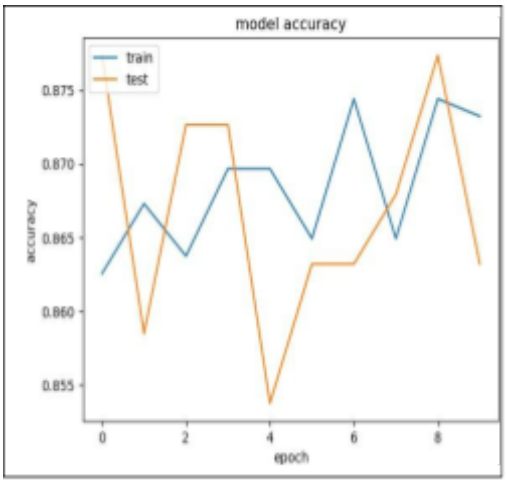


Figure 3. Performance of the model based on accuracy with CGAN

The binary cross entropy was applied as the loss function to calculate the loss during the testing and training phases. The following plot (see Figure 4) depicts the loss during the training and testing phases. The final loss obtained during training is 0.2974 and testing is 0.2884 which is considerably small. This is a good indication when we take into account the performance of the model.

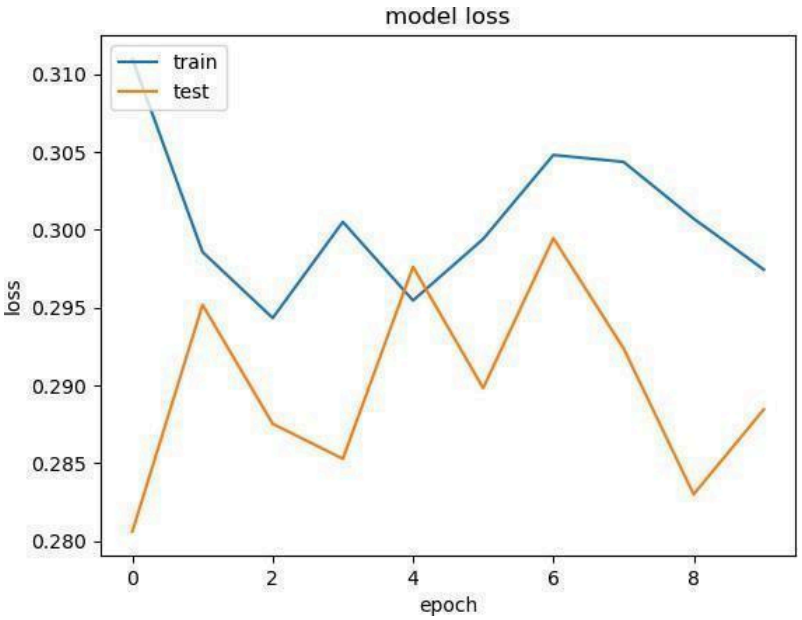


Figure 4. Training and testing loss

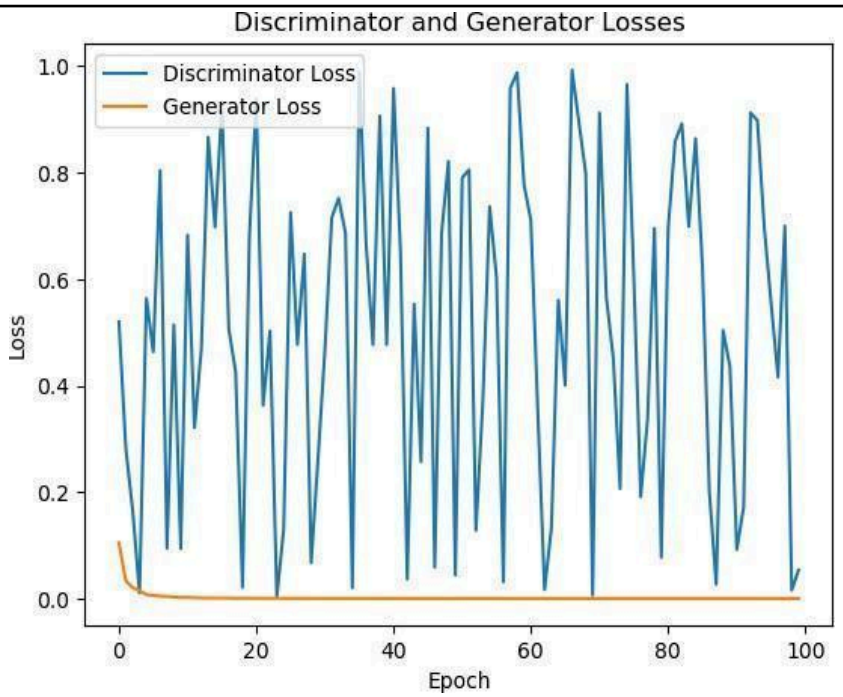


Figure 5. Discriminator and Generator loss

The discriminator and generator loss on applying CGAN is shown in (see Figure 5). The generator (0.01) and discriminator (0.9) loss indicates that the generator was successful in generating data that closely resembles the original dataset. This augmented data was combined with the original dataset to produce a diverse dataset which was used for training the RNN-LSTM classifier.

Table 1 summarizes the accuracy score obtained for the RNN-LSTM model with and without data augmentation. With data augmentation, the model's performance could be improved considerably.

Table 1.Performance of the RNN-LSTM with and without data augmentation

Model	Training Accuracy	Testing Accuracy
RNN-LSTM (without data augmentation)	80	76
RNN-LSTM (with data augmentation using CGAN)	87	86

5.2. Comparison with other deep neural networks

Table 2 summarizes the performance of different deep-learning models implemented with the Parkinson's speech dataset. On analyzing the results obtained as shown in Table 2 we conclude that even though the ANN model showed a high train accuracy compared to other models it showed overfitting as the difference between training and testing, accuracies were more compared to other models. GRU models could eliminate the problem of overfitting and could achieve a reasonably good performance but it suffered from a high training loss. RNN- LSTM classifier showed the best performance among these three models with fairly good accuracy and precision without much loss. Implementing data augmentation using CGAN the model could significantly improve the results in terms of accuracy and precision. The model's earlier issue of overfitting could be solved using the standard method for data augmentation through the application of CGAN.

Table 2 . Comparison of performance with other deep learning models

Models	cy	Testing Accuracy	Training Precision	n	Training Loss	Testing Loss
ANN	91	83	92	87	0.2623	0.403
RNN- LSTM	80	76	81	81	0.4498	0.5014
GRU	75	75	75	75	0.20651	0.5949
CGAN with RNN- LSTM		86	87.92	89.94	0.2974	0.2884

6. Conclusion

Parkinson's Disease is a progressive degenerative disease that affects the brain, the pathogenicity of which is still unresolved. Many unexplained attributes need to be investigated. The researchers despite their enormous influx of efforts have not been able to clinch the prime attributes that are responsible for this disease. Parkinson's Disease shares similar symptoms with many other neurological diseases. Therefore, it is difficult to identify the disease in its early stages. Various studies were conducted for diagnosing PD at an early stage using voice and speech attributes. This has proved worthwhile owing to its non-invasive, inexpensive, and simple methods for acquiring datasets.

This study is intended to focus on the problem of overfitting which results from training the deep learning models using a relatively limited dataset. The solution for this was a generalized method for data augmentation. The conditional generative adversarial networks proved to be a conducive and efficient method for generating synthetic data that mimics the original data. This technique could eliminate the problem of overfitting and the results showed significant improvement in the performance of the model (RNN-LSTM classifier) in the prediction of Parkinson's Disease.

7. Limitations and Future Perspectives

Even though the study could enhance the performance of the model by eliminating the problem of overfitting it could not succeed in extricating the attributes that have a high propensity for Parkinson's Disease. This study has opened new vistas in the application of data augmentation in identifying Parkinson's Disease. This could prove useful in creating a generalized model while dealing with diverse datasets for Parkinson's Disease. An architecture that can enhance these models by selectively focusing on pertinent elements and helps in improving the prediction accuracy and computational efficiency needs to be implemented. This should act as an impetus for further advancement in deep-learning algorithms.

The present study is based on a limited dataset which is not composed of real-time values. Therefore, efforts to identify a large dataset with real-time data were made but it failed to deliver useful results. Hence, we resorted to augmenting the present dataset. CGAN was chosen for the study. CGAN is an innovative approach for data augmentation that has a definite advantage over the problem of overfitting and dimensionality reduction. It could generate more controlled and diverse outputs compared to traditional GANs. Even though the advantages mentioned above opened up new avenues for future research, they have not succeeded in identifying features that have significant contributions to predicting the output. Data complexity, limited interpretability and scalability, training and evaluation challenges, etc., are the other drawbacks of CGAN. To overcome these issues a new approach is suggested, which incorporates attention networks into the CGAN architecture.

These networks, when used with CGAN for tabular data, have the benefits of improved feature selection and representation, enhanced handling of missing values, increased robustness to noisy data, and better preservation of data patterns and relationships. It also improves the quality of synthetically generated data and enhances interpretability through feature importance by assigning weighted scores to all attributes in the dataset. It also enables compact data representation through dimensionality reduction.

Hence, the drawbacks of CGAN can be reduced by the fusion of CGAN with attention

networks that can revolutionize future applications using Generative AI.

References

Bhowmick, S. S., & Lang, A. E. (2020). Movement disorders and renal diseases. *Movement Disorders Clinical Practice*, 7(7), 763–779. https://doi.org/10.1002/mdc3.13005

Ehgoetz Martens, K. A., & Shine, J. M. (2018). The interactions between non-motor symptoms of Parkinson's disease. *Expert Review of Neurotherapeutics*, *18*(6), 457–460. https://doi.org/10.1080/14737175.2018.1472578

Garcea, F., Serra, A., Lamberti, F., & Morra, L. (2023). Data augmentation for medical imaging: A systematic literature review. *Computers in biology and medicine*, *152*, 106391. https://doi.org/10.1016/j.compbiomed.2022.106391

Ghosheh, G.O., Li, J., & Zhu, T. (2022). A Survey of Generative Adversarial Networks for Synthesizing Structured Electronic Health Records. *ACM Computing Surveys, 56*, 1 - 34. https://doi.org/10.1145/3636424

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144. https://doi.org/10.1145/3422622

Gupta, R., Kumari, S., Senapati, A., Ambasta, R.K., & Kumar, P. (2023). New era of artificial intelligence and machine learning-based detection, diagnosis, and therapeutics in Parkinson's disease. *Ageing Research Reviews, 90*. https://doi.org/10.1016/j.arr.2023.102013

Hoq, M., Uddin, M. N., & Park, S. B. (2021). Vocal Feature Extraction-Based Artificial Intelligent Model for Parkinson's Disease Detection. *Diagnostics (Basel, Switzerland)*, *11*(6), 1076. https://doi.org/10.3390/diagnostics11061076.

Iyer, A., Kemp, A., Rahmatallah, Y., Pillai, L., Glover, A., Prior, F., Larson-Prior, L., & Virmani, T. (2023). A machine learning method to process voice samples for identification of Parkinson's disease. *Scientific reports*, *13*(1), 20615. https://doi.org/10.1038/s41598-023-47568-w.

Kamran, I., Naz, S., Razzak, I., & Imran, M.A. (2021). Handwriting dynamics assessment using deep neural network for early identification of Parkinson's disease. *Future Generation Computer Systems, 117*, 234-244. https://doi.org/10.1016/j.future.2020.11.020

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., & Aila, T. (2020). Training generative adversarial networks with limited data. *arXiv preprint* arXiv:2006.06676. https://doi.org/10.48550/arXiv.2006.06676

Kaur, S., Aggarwal, H., & Rani, R. (2021). Data augmentation using GAN for Parkinson's disease prediction. *Recent Innovations in Computing (ICRIC 2020)*, *Lecture Notes in Electrical Engineering*, 701, 589–597. https://doi.org/10.1007/978-981-15-8297-4_47

Li, Y., Wang, Q., Zhang, J., Hu, L., & Ouyang, W. (2021). The theoretical research of generative adversarial networks: an overview. *Neurocomputing*, 435, 26-41. https://doi.org/10.1016/j.neucom.2020.12.114

Minsky, M., & Papert, S. A., (1969). *Perceptrons: An Introduction to Computational Geometry*. The MIT Press.

Mortezazadeh, T., Seyedarabi, H., Mahmoudian, B., & Islamian, J.P. (2021). Imaging modalities in differential diagnosis of Parkinson's disease: opportunities and challenges. *Egyptian Journal of Radiology and Nuclear Medicine, 52*, 1-12. https://doi.org/10.1186/s43055-021-00454-9

Mughal, H., Javed, A.R., Rizwan, M., Almadhor, A.S., & Kryvinska, N. (2022). Parkinson's Disease Management via Wearable Sensors: A Systematic Review. *IEEE Access*, PP, 1-1. https://doi.org/10.1109/ACCESS.2022.3162844

Peppes, N., Tsakanikas, P., Daskalakis, E., Alexakis, T., Adamopoulou, E., & Demestichas, K.

(2023). FoGGAN: Generating Realistic Parkinson's Disease Freezing of Gait Data Using GANs. *Sensors (Basel, Switzerland)*, *23*(19), 8158. https://doi.org/10.3390/s23198158

Postuma R. B. (2016). Voice changes in prodromal Parkinson's disease. *Sleep medicine*, *19*, 148–149. https://doi.org/10.1016/j.sleep.2015.08.019

Pradhyumna, P., & Mohana, P. (2022). A survey of modern deep learning-based generative adversarial networks (GANs). In *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE. https://doi.org/10.1109/ICCMC53470.2022.9753782

Rehman, A., Saba, T., Mujahid, M., Alamri, F.S., & ElHakim, N. (2023). Parkinson's Disease Detection Using Hybrid LSTM-GRU Deep Learning Model. *Electronics*.

Ron Kohavi. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *14th international joint conference on Artificial intelligence* - Volume 2 (IJCAI'95),1137–1143.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature, 323*, 533-536. https://doi.org/10.1038/323533a0

Sakar, C.O., & Kursun, O. (2010). Telediagnosis of Parkinson's Disease Using Measurements of Dysphonia. *Journal of Medical Systems,* 34, 591-59. https://doi.org/10.1007/s10916-009-9272-y

Sakar, C., Serbes, G., Gunduz, A., Nizam, H., & Sakar, B., (2018). Parkinson's Disease Classification. *UCI Machine Learning Repository.* https://doi.org/10.24432/C5MS4X

Shannon, C.E. (1948). A mathematical theory of communication. *Bell Syst. Tech. J., 27, 379- 429.*

Todorova, A., Jenner, P. G., Ray Chaudhuri, K., Chaudhuri, R., Ray, J. P., & K, C. (2014). Non-motor Parkinson's: Integral to motor Parkinson's, yet often neglected. *Practical Neurology*, 14, 310–322. https://doi.org/10.1136/practneurol-2013-000741

Trifonova, O. P., Maslov, D. L., Balashova, E. E., Urazgildeeva, G. R., Abaimov, D. A., Fedotova, E. Y., Poleschuk, V., Illarioshkin, S. N., & Lokhov, P. G. (2020). Parkinson's disease: Available clinical and promising omics tests for diagnostics, disease risk assessment, and pharmacotherapy personalization. *Diagnostics*, 10(5), 339. https://doi.org/10.3390/diagnostics10050339