National College of Ireland

A Generative AI Framework for Data Augmentation Employing Generative Adversarial Networks to Predict Parkinson's Disease

MSc Research Project
MSc in Data Analytics

Irfan Pervez
Student ID: 23174854

School of Computing
National College of Ireland

Supervisor: Hamilton Niculescu

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	IRFAN PERVEZ
Student ID:	23174854
Programme:	MSc in Data Analytics
Year:	Jan-2024
Module:	MSc Research Project
Supervisor:	Hamilton Niculescu
Submission Due Date:	12/12/2024
Project Title:	A Generative AI Framework for Data Augmentation Employing Generative Adversarial Networks to Predict Parkinson's Disease
Word Count:	5950
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

Signature:	Irfan Pervez
Date:	28th January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	■
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	■
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	■

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Generative AI Framework for Data Augmentation Employing Generative Adversarial Networks to Predict Parkinson's Disease

Irfan Pervez
23174854

Abstract

Early detection of Parkinson's disease (PD), a degenerative neurological ailment, can lead to more effective treatment. Traditional diagnostic testing methods often rely on clinical observations analysis, which might delay diagnosis. Voice sample analysis has recently been found to be a possible early sign of Parkinson's disease (PD) because vocal problems are linked to motor weakness. Deep learning (DL) algorithms and other advanced AI based prediction models are not as useful as they could be because there are not enough big datasets that have been labeled. This study looks into whether synthetic data generated with a Generative Adversarial Network (GAN) can increase the accuracy of Parkinson's disease prediction.We conducted experiments and generated multiple different versions based on the number of voice recordings of PD patients and evaluated the quality of synthetic data by predicting the PD using several machine and deep learning classifiers, including random forest, XGBoost, artificial neural networks (ANN), convolutional neural networks (CNN), fully connected neural networks (FCNN). The results show that using GAN-generated synthetic data improved diagnostic performance across many models, specifically deep learning models, where ANN and FCNN achieved the 99% of accuracy rate in predicting PD for all the synthetic data versions compared to 89% for original data, representing a significant 10% increase in this study. For machine learning models the precision, recall, and f1-score values were all around 98% in all the versions of synthetic data. These results also underline the need of generative artificial intelligence in improving medical diagnosis and point to possible uses in healthcare industry.

Keywords – Parkinson's Disease (PD), Generative Adversarial Networks (GAN), Synthetic Data Generation, Machine Learning (ML), Deep Learning (DL) Algorithms, precision, recall, f1-Score

1 Introduction

Parkinson's disease (PD) is a progressive neurological disorder in the neuroscience field that affects an estimated 1 million people, with approximately 90,000 new cases diagnosed each year in United states Dorsey et al. (2018). Early detection of Parkinson's disease remains a key challenge in the healthcare domain, where an accurate and prompt diagnosis can have a big impact on the line of treatment and the outcome. Voice sample analysis has recently been investigated as a potential method for early PD identification,

as vocal alterations caused by motor weakness are common Little et al. (2009). However, the availability of big, labeled datasets limits the performance of advanced approaches specifically related to Deep learning (DL) in diagnosing PD.DL algorithms often require large volumes of data to train the models properly, which makes it difficult to work with small data samples Goodfellow et al. (2014).

In this study, we trained the GAN models on an original dataset of PD voice recording samples, generated new synthetic data, and conducted a total of three experimental case studies. The new AI application will facilitate the execution of a larger volume of experiments for the intuitive generation of synthetic data.

The artificially produced data by GAN models improves the performance of the machine and deep learning models, which lets it generalize better. GANs are also flexible, so we can change how the data is generated to work in different areas. This makes them a useful tool for many independent learning tasks in the healthcare industry.In more detail, here are the things we bring to the table with our research.

Expansion of Dataset Size — GAN models used in this study successfully increases the size of the original dataset, which solves the problem of not having enough data for training the complex deep learning models.

Better Model Performance — Utilizing the synthetic data makes DL models work better by giving them more examples and different types of data patterns. This helps the ML and DL models to understand more situations and scenarios compared to the real data and deal with outliers and rare events better.

Cost Effective Application — Creating synthetic data is a very cheap option compare to collecting, cleaning, and handling new datasets of Parkinson's disease patients, which can be hard to perform and cost a lot of effort and funds.

Balanced Feature Distributions — Uneven distributions in columns that don't go in a straight line in the original dataset might be fixed by our method. By creating artificial data, it evens out these uneven feature distributions, which makes it more useful for machine learning jobs. Therefore, based on the previous discussion, the following research question will be addressed:

Can synthetic data generated by a generative AI application based on a GAN model improve the prediction of Parkinson's disease diagnosis compared to state of the art technology?

We have created a new generative AI web application called SyntheticDataQ.ai (see Figure 1). We used this app to create different versions of synthetic datasets for a series of experiments, and this tool provided an important assistant in answering the research questions, with the aim of improving the accuracy of PD prediction. GAN models are part of generative AI and are known for their capacity to generate real world synthetic data samples, making them excellent alternatives to real datasets in a variety of domains Creswell et al. (2018).GANs provide a means to augment dataset size and diversity through the generation of synthetic data, hence facilitating more robust training for machine and deep learning models.

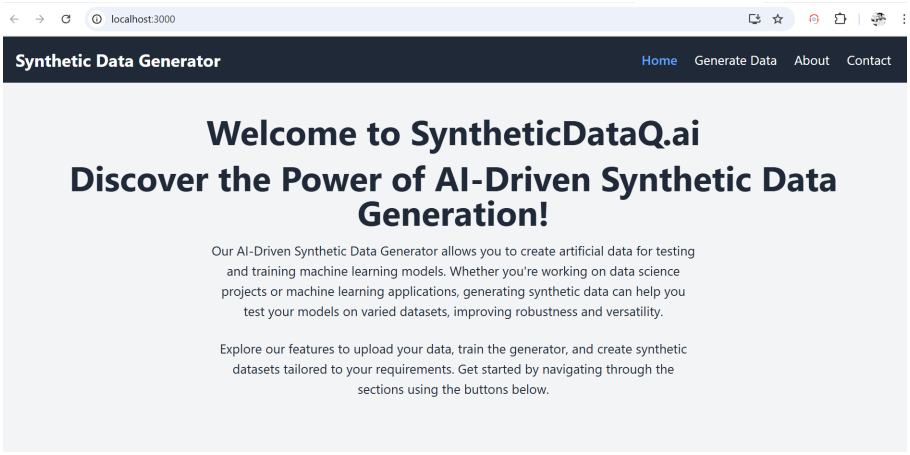


Figure 1: A novel Generative AI app (SyntheticDataQ.ai) (Source: Pervez, Irfan (2024))

This research looks into and talks about the latest developments in making synthetic health data with GANs in the related work section, as well as time-series data, medical diagnosis, and education. The newly developed AI app uses the front end library for the user interface, with GAN deployed on a python based flask server as the backend. The specifics of the application development, including the tech stack and multi tier architecture, are thoroughly explained in the project specification and implementation sections.

In evaluation section, we explained the three experimental case studies conducted in this research and completed the quality and performance evaluation of the new synthetic data by predicting PD using DL and ML models against the real data. In discussion section we suggested modifications and areas for improvement. A critical analysis of the experiments is next performed for determining whether the initial design for this research is sufficient. The study continues with a discussion of future work and talks about research looking at more advanced GAN designs to handle more complex and imbalanced datasets to encourage a broad range of use cases.

2 Related Work

This study outlines the production of synthetic datasets in the healthcare sector using advanced versions of GAN models. Addressing data shortage challenges, GAN-based models are effective in generating artificial datasets that replicate the real-time data generated by sensors of biomedical instruments. GAN have been effectively applied previously to various different data types, including tables, pictures, video, audio, and time series-based data.

GAN models used for disease diagnosis classification and generate synthetic health data based on tabular datasets helped to classify the diabetic patients introduced by researchers Hargreaves & Heng (2020) and proved that a machine learning model trained with both original and synthetic data showed an 8.3% improvement in classification accuracy to predict the number of people with diabetes. Later, the MedGAN model,

which is introduced for generating synthetic electronic health records (EHRs), particularly for medical datasets created by Choi et al. (2017).

A good time series dataset generating model can show how features are spread out and how they change over time. To make realistic time series data quickly across various domains and mixed-data settings, TimeGAN combines adversarial training with supervised learning for stepwise transitions by embedding latent spaces developed by Yoon et al. (2019).

The synthetic data for the time series dataset of ICU patient monitoring proposed by Esteban et al. (2017) has been generated using the recurrent conditional GAN (RCGAN) model by means of recurrent neural networks (RNN), hence producing realistic sequential data. Similarly, Kiyasseh et al. (2021) introduced a 29% improvement in disease classification performance by use of synthetic health-based monitoring signals created by the PlethAugment model.Moreover, Hazra & Byun (2021) created SynSigGAN models to examine medical diagnosis and education, so producing the realistic data with medical signals such as electrocardiogram (ECG) and electroencephalogram (EEG) for model training needs.

GAN models are being used in the field of genetics to create synthetic gene expression datasets for addressing problems such as high dimensionality and sparse data information in gene and these synthetic datasets replicate real-world biological patterns generated by Lee (2023), hence improving machine and deep learning models.By use of high-quality synthetic data, this technology is supporting disciplines including toxicology, drug development, and illness diagnostics. For instance, GANs have been demonstrated to lower biases and errors in the conventional data augmentation techniques while improving information dependability by Lee (2023).GAN models shown in Figure 2 help to enrich genetics expression data. The GAN generator G is shown while inducing random noise to create the synthetic gene expression and the further discriminator D used to identify if the data are real or fake (A) training process.

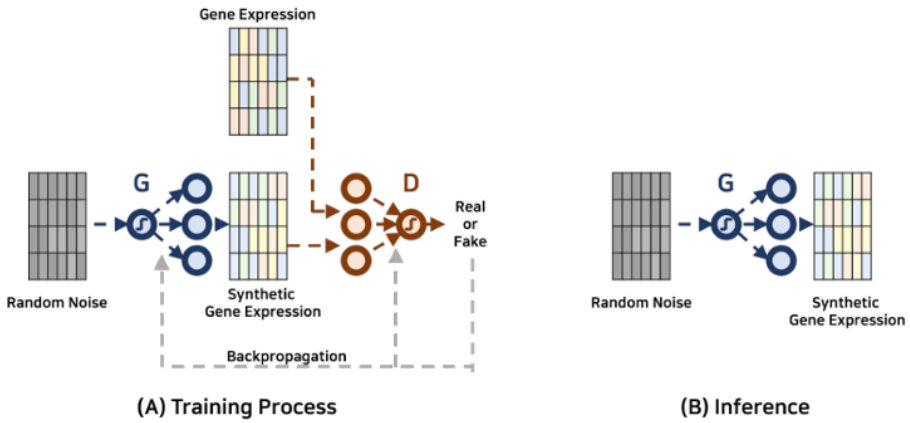


Figure 2: Gene expression data augmentation with GANs. (Reproduced from Lee (2023), p. 5)

Xu et al. (2020) proposes a new methodology for identifying the patients affected by Parkinson's disease by utilizing the Spectrogram Deep Convolutional Generative Ad-

versarial Network (S-DCGAN) integrated with a CNN ResNet50 model.Their research study leveraged the Sakar dataset collected by Sakar et al. (2013) used to perform a comparative analysis across multiple classification models, with the main objective of improving the accuracy of PD diagnosis among patients. RestNet 50 CNN model is used to extract important features in voice print or audio data and later used to predict the presence of PD.The results demonstrated good performance improvements as the S-DCGAN model achieved a voiceprint recognition accuracy of 91.25%, with a recall of 92.5%. These findings indicate the effectiveness of the approach in identifying PD patients from healthy patients. This study highlights the potential of advanced deep learning techniques in leveraging voiceprint data for accurate and early diagnosis of Parkinson's disease.

The research paper published by Xu et al. (2020) worked as a motivation to conduct this research because this study is utilizing a different set of labeled PD voice recording data that is freely available for research and development on UCI Machine Learning Repository (2008) collected by, Little et al. (2008) and this labeled data offers a strong foundation for generating synthetic data using generative adversarial networks by enhancing the robustness of the GAN model. The aim of this study is to investigate, how effectively the new synthetic data produced by GAN models help to predict the PD in early stages and mitigate the challenge in estimating Parkinson's illness resulting from the lack of sufficient labeled data needed by sophisticated and complex deep learning models.This study highlights the research gap in improving the classification accuracy of machine and deep learning models for predicting Parkinson's Disease (PD) compared to the previous study conducted on the Sakar dataset.

3 Methodology

This research follows the knowledge discovery in databases (KDD) approach that was first introduced and formalized by Gregory Piatetsky-Shapiro in 1989 (KDD-1989) as shown in Figure 3. It includes the steps as per KDD methodology to achieve the research objectives.

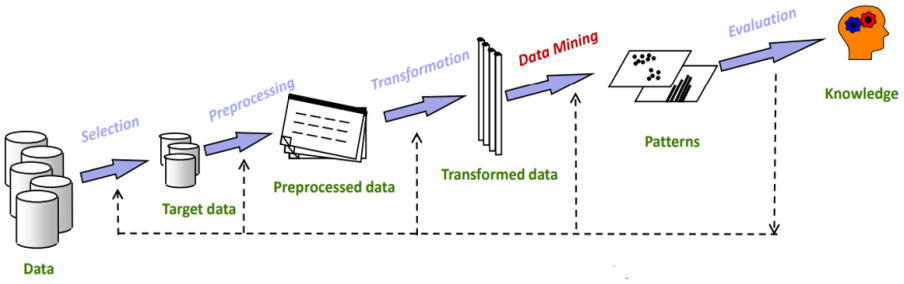


Figure 3: Knowledge Discovery in Databases (KDD) workflow

3.1 Data Selection

This is the starting phase of the KDD approach.We have developed the SyntheticDataQ.ai application to produce synthetic data for conducting experiments. The AI app uses the original data as input for training the GAN model to create a new synthetic dataset. All versions of the synthetic data we generated for examination have two classes: healthy and unhealthy patients with PD. The original dataset has 195 voice recordings; version 1 comprises 5,000; version 2 includes 10,000; and version 3 features 15,000 voice recordings of synthetic data generated by the AI program, with each recording represented by a row in the dataset. Table 1 outlines the data type associated with each of the 22 features included in the dataset.

Variable Name	Role	Type	Description
name	ID	Categorical	ASCII subject name and recording number
MDVP:Fo(Hz)	Feature	Continuous	Average vocal fundamental frequency
MDVP:Fhi(Hz)	Feature	Continuous	Maximum vocal fundamental frequency
MDVP:Flo(Hz)	Feature	Continuous	Minimum vocal fundamental frequency
MDVP:Jitter(%),MDVP:Jitter(Abs),MDVP:RAP, MDVP:PPQ,Jitter:DDP	Feature	Continuous	Several measures of variation in fundamental frequency
MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer: APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:D DA	Feature	Continuous	Several measures of variation in amplitude
NHR,HNR	Feature	Continuous	Two measures of ratio of noise to tonal components in the voice
status	Target	Integer	Health status of the subject (one) - Parkinson's, (zero) - healthy
RPDE,D2	Feature	Continuous	Two nonlinear dynamical complexity measures
DFA	Feature	Continuous	Signal fractal scaling exponent
spread1,spread2,PPE	Feature	Continuous	Three nonlinear measures of fundamental frequency variation

Table 1: Overview of Parkinson's Disease Dataset(Source: UCI Machine Learning Repository (2008))

3.2 Data Preprocessing

The primary objective of our research is to accurately categorize individuals as either healthy or affected by PD in a synthetic dataset, as denoted by the "status" column in all versions of the synthetic and original datasets.A score of 0 indicates healthy persons, whereas a value of 1 represents those diagnosed with PD. A generative adversarial network (GAN) failed to accurately generate the target variable "status" column with binary values (0 and 1) when generating synthetic data. We implemented post-processing logic to convert the Status column to binary, applying a rule top of output synthetic data generated by GAN that converts the generated value to 1 if it exceeds 0.5 value, and to 0 otherwise. We set the consistent ratio of healthy to unhealthy patients in all three versions of the synthetic data to 25:75, which is the same as the real data as shown in

the Figure 4.

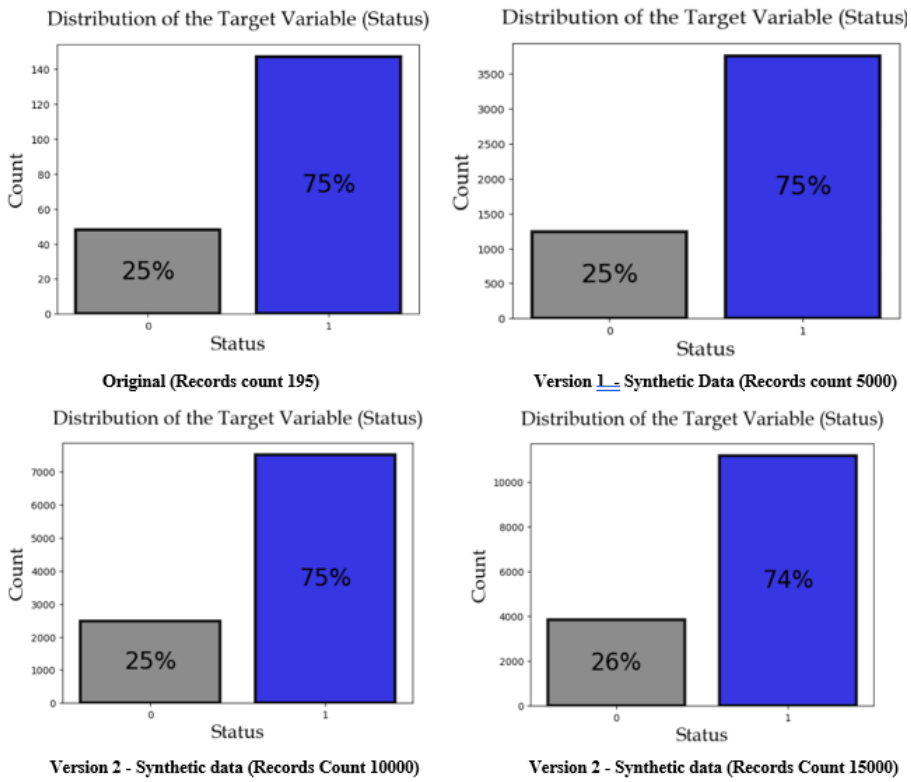


Figure 4: Distribution of the Target Variable (Status) in (Original Dataset (195 voice recordings), Version 1(5000 voice recordings), Version 2(10000 voice recordings), and Version 3(15000 voice recordings) (Source: Pervez, Irfan (2024))

The assumption for the variable column "name" is that it contains unique identifiers of subject names for each voice recording in the original dataset. We will not include the "name" variable in synthetic data because its unique nature does not help predict PD. The original dataset is required as input to train the GAN model to generate synthetic data, as the "name" column is categorical in nature; we cannot pass that "name" column as input to the GAN model.

The correlation matrices illustrate in Figure 5 show the variations in feature relationships between the original and synthetic datasets. The original dataset reveals that several features have significant connections, highlighting distinct linkages specific to Parkinson's symptoms and measures. In Version 1, these patterns are more spread out, with certain original linkages appearing diminished, indicating a failure to accurately capture some relationships.

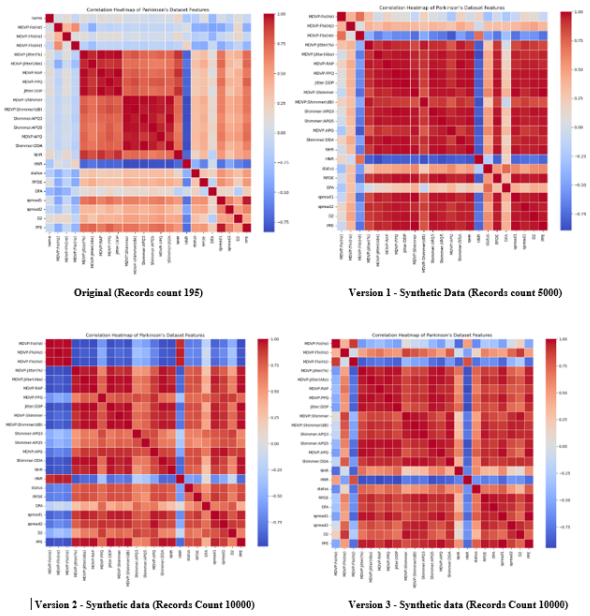


Figure 5: Correlation Matrix Comparison for Original Dataset (195 recordings) and Expanded Versions (5,000, 10,000, and 15,000 recordings) (Source: Pervez, Irfan (2024))

In Versions 2 and 3, the patterns show greater homogeneity, suggesting that the synthetic models are effectively generalizing the correlations. Version 3 in particular has persistently high correlations between characteristics, which suggests that the generator may have made associations between features that are stronger than they really are (See Appendix 1 and 2 for individual feature distributions in all three versions).

We evaluate both the original and the synthetic data generated by 3 different iterations using machine and DL models.In data preprocessing, we utilize the Standard Scaler, a class from the scikit-learn module in Python, to normalize all numeric variables. The employed machine learning algorithm relies on distance computations, such as linear and non-linear models, which are sensitive to feature scaling. For algorithms that lack scale sensitivity, such as decision trees or random forests, standardizing the characteristics may provide very minimal advantages.

Table 2: Train Test Split for Different Data Versions (Source: Pervez, Irfan (2024))

Train Test Split	**Original Data**	**Version 1 (5000 records)**	**Version 2 (10000 records)**	**Version 3 (15000 records)**
Training Data	(156, 22)	(4000, 22)	(8000, 22)	(12000, 22)
Testing Data	(39, 22)	(1000, 22)	(2000, 22)	(3000, 22)

Table 2 shows the allocation of training and testing datasets for an original dataset and three synthetic data versions. We performed three case studies using synthetic datasets of increasing sizes: 5,000, 10,000, and 15,000 records, respectively. Every synthetic dataset consistently consists of 22 features. We used an 80/20 split for training and testing in each experiment because using this split ensures the training sufficiency and testing reliability, which help the models to generalize effectively by minimizing overfitting and underfitting. Version 1 divided the dataset of 5,000 records into 4,000 training records and 1,000 testing

records. In Versions 2 and 3, datasets with 10,000 and 15,000 records were used, giving training and testing splits of 8,000 to 2,000 and 12,000 to 3,000, respectively.

3.3 Data Modelling

The SyntheticDataQ.ai app created using a Generative Adversarial Network model to produce synthetic data from a GAN trained on pre-existing dataset. The GAN was constructed with pytorch, keras, and scikit learn python libraries ensuring a streamlined implementation. The generator model, structured with nine hidden layers and relu activations, transforms any randomness into synthetic data samples. Meanwhile, the discriminator distinguishes real from synthetic data using dense layers with leaky relu activations, finalizing with a sigmoid output for classification. Both models are fully trainable and enabling the efficient adversarial learning. While running various iterations for generating the appropriate sets of synthetic dataset we adjusted the parameters like learning rate(LR), Latent space dimensions, and batch size to produce high quality synthetic data. Later migrated the optimized GAN model python code with tuned hyperparameters to VS code enviroment and used the react native library to run the user interface and backend code of the GAN and generated the 3 different sets of synthetic data.

For validating the quality and performance of the synthetic data, we leveraged python based jupyter lab environment and added the code for machine and deep learning models for testing all 3 versions of sythetic data. Machine learning models like random forest and xgboost applied to predict the PD and hypermeter tuning added by adjusting number of trees, maximum depth of trees and minimum split to get the best classification accuracy, precision, and recall. similarly deep learning models like artificial neural networks and fully connected neural networks tuned by adjusting the number of layers and number of neurons per layer to achieve the best classification accuracy.

4 Design Specification

The II-tier project design architecture is shown in Figure 6. The advanced generative AI methodologies applied for this research analysis and visualization of new synthetic data. The presentation tier utilizes react Native web development framework to create an interactive user interface (UI) that allows users to upload the original data in CSV format to train the GAN model and define the new synthetic data creation parameters (for example, the number of voice recordings of PD patients that need to be generated). Power BI dashboards displaying predictive outcomes using graphical representations. The business tier employs a REST API for efficient communication between the UI and the backend server. A python based Flask server operates the GAN model, which produces synthetic datasets derived from original submitted data.

Deep learning architectures like Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Fully Connected Neural Networks (FCNN) combine with machine learning models like Random Forest, XGBoost, Logistic Regression, and De-

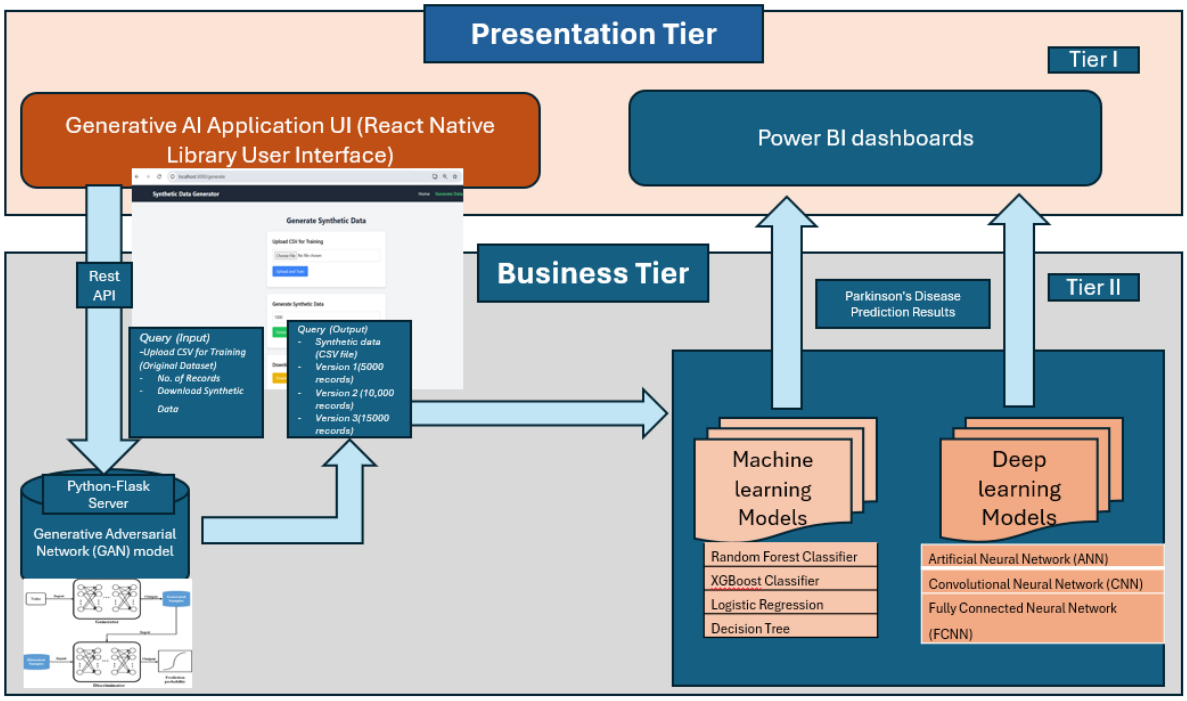


Figure 6: Project (II-Tier) Design Architechture (Source: Pervez, Irfan (2024))

cision Tree to analyze the newly generated artificial datasets. This mixed machine and deep learning pipeline guarantees more reliable predictive outcomes Brownlee (2020).

User input fields, like original data to train the model and the number of records that need to be generated are transmitting from the web-based user interface to the backend server, where the GAN model synthesizes information and generates the artificial data. Machine and deep learning models subsequently evaluate artificial data, yielding the predictions that Power BI effectively visualizes results.

SyntheticDataQ.ai app has a built-in GAN model that generates synthetic data with a privacy preserving technique, rigorous validation, and ethics and compliance associated with GDPR. Training the GAN model used prevents data memorization to ensure the privacy and identifying again the anonymity in the data. Ethical safeguarding include data minimization and explicit consent before using it for any other research purpose. All the workflows and application design are auditable. A transparent two tier architecture with user defined parameterization and Power BI visualizations (As shown in Appendix 3) enhances the interpretability of the results.

5 Implementation

This research presents the development of a web-based generative AI app SyntheticDataQ.ai, designed to generate new synthetic data using the generative adversarial network model derived from the original dataset. The Microsoft Visual Studio Code tool is used as an integrated development environment (IDE) for the development of the project. The react library structure was used to build the front end of the web app. This makes the user

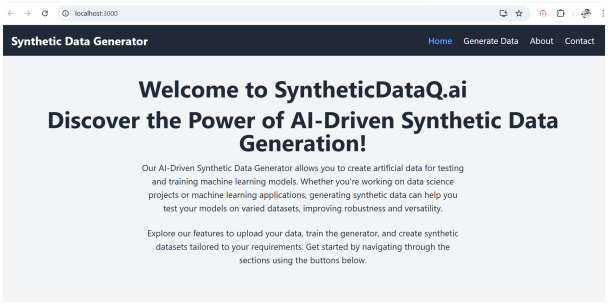


Figure 7: Home Screen(SyntheticDataQ.ai)(Source: Pervez, Irfan (2024))

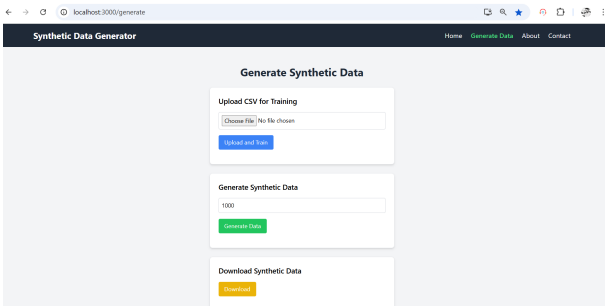


Figure 8: Data Generator Screen (SyntheticDataQ.ai)(Source: Pervez, Irfan (2024))

experience interactive. The GAN model was built on the back end using the advanced pytorch library. The integration of these latest technologies enabled the efficient design of the tool. Figures 7, 8, and 9 show the Home, Data generator, and Contact screens of the app's user interface.

The application was deployed on the python based flask server, which provides seamless integration between the servers. This setup made it possible for the generative AI model to function well in a web based environment by facilitating an efficient workflow. React for the front end and flask for the deployment were combined to create a user friendly interface and a reliable backend that could manage complex calculations.

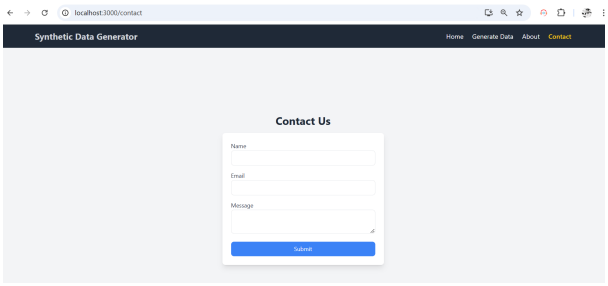


Figure 9: Contact Screen (SyntheticDataQ.ai)(Source: Pervez, Irfan (2024))

5.1 GAN Implementation

The proposed GAN model defines the design architecture to generate high quality synthetic data. Figure 10 shows that a normal GAN is made up of two main parts, generative network the generator, G, utilizes randomness or input as its initial point from the latent space and this noisy input is produced from random distribution, such as Gaussian distribution, and functions as the initial signal used by the generator to produce new synthetic data. Whereas, while using the GAN in this study, we are using Descriminator (D) to test the loss by combining both the real and synthetic data, but the Generator (G) is used only for testing the loss in the synthetic data. The generator's goal is to learn to transform randomness into the data samples that closely mimic real data, with the intention of misleading the discriminator within the system.Simultaneously, the discriminator, D, distinguishes between real and artificial data points produced by G utilizing noise from the latent space.

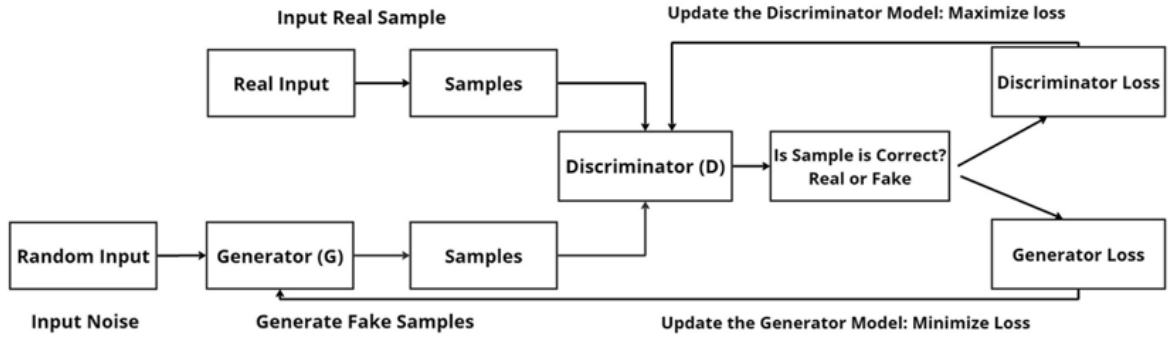


Figure 10: The architecture of the Generative Adversarial Network (source: Ramzan et al. (2024))

Table 3: Neural Network Summary - Network 1 (Source: Pervez, Irfan (2024))

Layer (Type)	**Output Shape**	**Param #**
Linear-1	[-1, 64]	704
ReLU-2	[-1, 64]	0
Linear-3	[-1, 128]	8,320
ReLU-4	[-1, 128]	0
Linear-5	[-1, 23]	2,967
Total Params		11,991
Trainable Params		11,991
Non-Trainable Params		0

Table 3 details the architecture of the generator model, which uses the Sequential API for the progressive stacking of its layers. The generator begins with an input layer that accepts scaled random noise, followed by nine hidden layers triggered by relu and concludes with an output layer using linear activation to align with the datasets dimension.

Table 4 outlines the construction of the discriminator model as a sequential model with eleven thick layers. The initial ten layers using relu activation, but the output layer

Table 4: Neural Network Summary - Network 2(Source: Pervez, Irfan (2024))

Layer (Type)	**Output Shape**	**Param #**
Linear-1	[-1, 128]	3,072
LeakyReLU-2	[-1, 128]	0
Linear-3	[-1, 64]	8,256
LeakyReLU-4	[-1, 64]	0
Linear-5	[-1, 1]	65
Sigmoid-6	[-1, 1]	0
Total Params		11,393
Trainable Params		11,393
Non-Trainable Params		0

utilizes sigmoid activation to categorize samples as authentic or artificial. We applied a 20% dropout rate on the input and two hidden layers to mitigate overfitting. We trained the GANs for 500 epochs with a batch size of 50.

Table 5: Configuration for GAN classification training (Source: Pervez, Irfan (2024))

Components	**Values**
Epochs	5000
Batchsize	64
Initial learning rate	0.0001
Learning rate decay Exponential	0.1
Learning rate decay gamma	0.9
Optimizer	Adam
Optimizer weight decay	0.0005
Loss Cross Entropy	0.0002

In Table 5, you can see the list of the important parts and values that are used in the training configuration of a DL model. The model trained for 5,000 epochs using a batch size of 64. We start with a learning rate of 0.0001 as explained by Tsourtis et al. (2022) and use an exponential decay method with a rate of 0.1 and a decay factor (gamma) of 0.9 to lower the learning rate over time. We utilize the Adam optimizer known for its adaptive learning features with a weight decay of 0.0005 to prevent overfitting. During training, we employ the cross entropy loss, set at 0.0002, to evaluate the loss in model prediction.

There are fully linked layers and relu activations in Table 3, which show how the generator changes a noise vector into new data samples. The architecture has dense levels with 64 and 128 units, concluding with a final output layer of 23 units yielding a total of 11,991 trainable parameters.

This study compares the performance of ML and DL models using new synthetic datasets with 5,000, 10,000, and 15,000 records, comparing them to models that were trained on real data. We have found that synthetic data improves the accuracy of both machine learning and deep learning models. The performance gains level off at around 10,000 to 15,000 records, which suggests a saturation point.

Overall, all models performed almost the same on synthetic data as they did on real data. This suggests that GANs can create synthetic data with useful and accurate features for classification purposes. Different classifiers and different sizes of synthetic data showed different levels of performance, showing both the advantages and disadvantages of using GAN generated data.

5.2 Implementation of ML and DL models

Machine and deep learning models were developed using the widely recognized machine learning libraries in Python, including scikit-learn, XGBoost, Keras, and TensorFlow. The dataset was prepared with pandas and numpy, and matplotlib and seaborn made it easier to see how well the model worked. The train test split function from scikit-learn divided the data into training and testing sets at an 80/20 ratio.

A fixed random state parameter was used while data splitting and training to ensure reproducibility of results. Added hyperparameter tuning in the machine learning models to optimize parameters, such as the number of estimators and maximum depth and Log loss was employed as a performance metric for probabilistic outputs, ensuring a robust evaluation of model predictions for xgboost , random forest and decision tree classifiers.

Deep learning models like ANN, CNN and FCNN were fine tuned using keras tuner, optimizing parameters such as learning rate, number of layers, number of epochs and drop out rate. Metrics such as accuracy and log loss were tracked, and performance trend were visualized to ensure proper convergence in the results.

6 Evaluation

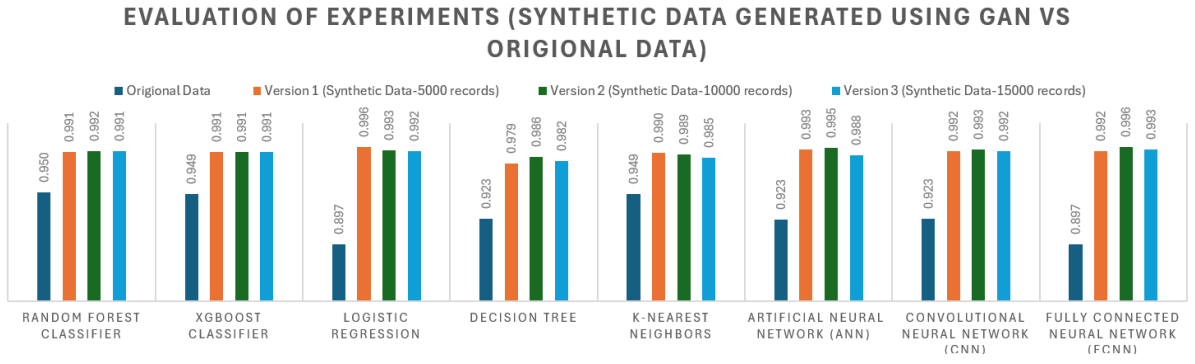


Figure 11: Classification Accuracy of Machine and Deep Learning Models on Original and Synthetic Datasets (5,000, 10,000, and 15,000 records) (Source: Pervez, Irfan (2024))

The research assesses ML and DL models utilizing new versions of synthetic datasets including 5,000, 10,000, and 15,000 records contrasting their efficacy with models trained on original data shown in Figure 11. We have found that synthetic data improves the

accuracy of both ML and DL models. The performance gains stop growing after 10,000 to 15,000 records, which is called a saturation threshold.

Critical findings indicate that accuracy markedly improves using synthetic data. The accuracy of logistic regression increased from 0.8974 with the original data to 0.9960 with 5000 records. DL models, notably Fully Connected Neural Networks (FCNN) and Artificial Neural Networks (ANN), attained the best accuracy, especially with 10,000 records. Conventional machine learning models, such as Decision Trees and K-Nearest Neighbors, showed enhancement but remained inferior to deep learning models, which adeptly utilized bigger synthetic datasets. Even though convolutional neural networks(CNNs) were successful, but they couldn't beat fully connected neural networks(FCNN) and artificial neural networks (ANN). This is probably because CNNs are usually used with picture data instead of tabular data.

This is in line with previous research that has shown synthetic data can help when data is scarce Frid-Adar et al. (2018). Synthetic data provides practitioners with a cost efficient means to enhance model performance in domains with restricted data availability, such as life sciences. Data augmentation in excess can show poor results, it's required to perform this in producing an adequate amount of data along with a suitable technique.

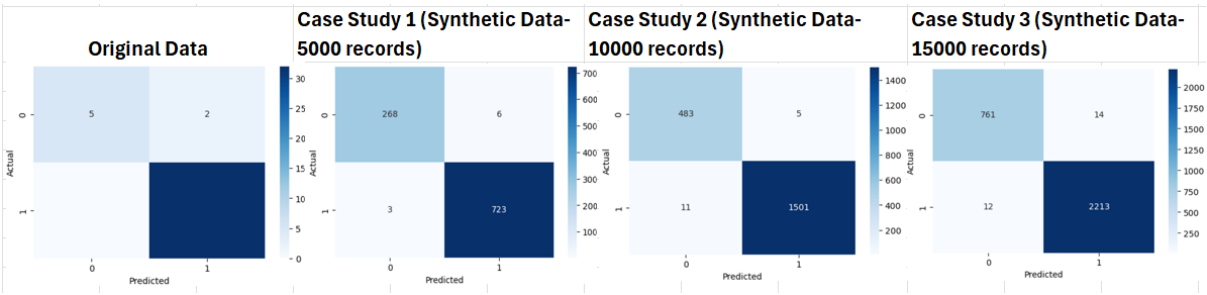


Figure 12: Confusion matrix for random forest (Source: Pervez, Irfan (2024))

Figure 12 displays total four confusion matrices: one for the actual dataset and three for new synthetic datasets of differing sizes (5,000, 10,000, and 15,000 records). Each matrix illustrates random forest model performance by contrasting actual and expected results. The rows denote actual labels (0 or 1), while the columns indicating expected labels.The synthetic data with 5000 records matrix plot indicates that the model accurately identified 5 true negatives (top-left) and 723 true positives (bottom-right), while error in classifying 2 false positives (top-right) and 3 false negatives (bottom-left).

As synthetic data increases in number of records, the true positives and true negatives expand correspondingly, indicating model consistency across varying dataset sizes. Misclassification rates shows minor fluctuations yet consistently remain low.In the ANN, CNN, and FCNN models, we trained the models for 50 epochs, during which we noted an enhancement in training accuracy over the epochs.

Looking at performance indicators like accuracy rate, precision, recall, and f1 score, shown in Table 6, that new synthetic data versions improves ML model performance.

The original data has good recall scores, especially in ensemble models like xg boost

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	0.888889	1.000000	0.941176	0.897436
XG Boost	0.941176	1.000000	0.969697	0.948718
Decision Tree	0.939394	0.968750	0.953846	0.923077
Random Forest	0.941176	1.000000	0.969697	0.948718
K-Nearest Neighbor	0.941176	1.000000	0.969697	0.948718

Original Data

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	0.998619	0.995868	0.997241	0.996000
XG Boost	0.994483	0.993113	0.993797	0.991000
Decision Tree	0.984869	0.986226	0.985547	0.979000
Random Forest	0.991770	0.995868	0.993814	0.991000
K-Nearest Neighbor	0.989071	0.997245	0.993141	0.990000

Version 1 (Synthetic Data-5000 records)

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	0.996024	0.994048	0.995035	0.992500
XG Boost	0.996676	0.991402	0.994032	0.991000
Decision Tree	0.990086	0.990741	0.990413	0.985500
Random Forest	0.996680	0.992725	0.994698	0.992000
K-Nearest Neighbor	0.996003	0.988757	0.992366	0.988500

Version 2 (Synthetic Data-10000 records)

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	0.993274	0.995506	0.994388	0.991667
XG Boost	0.994599	0.993258	0.993928	0.991000
Decision Tree	0.987427	0.988315	0.987871	0.982000
Random Forest	0.993714	0.994607	0.994160	0.991333
K-Nearest Neighbor	0.990112	0.990112	0.990112	0.985333

Version 3 (Synthetic Data-15000 records)

Table 6: Precision,Recall,F1-Score and Accuracy (Source: Pervez, Irfan (2024))

and random forest, demonstrating accurate identification of true and false positive cases. However, models such as logistic regression had lower precision (0.8889), indicating difficulty in eliminating false positives. Ensemble models (random forest, xg boost) achieved the highest accuracy, with values around 0.9487, showing reliable predictive capacity but room for improvement.

Version 1 (5,000 records) showed significant results. It got better until it hit 0.996 for accuracy and precision. It went from 0.8889 to 0.9986. The models like Random Forest and XGBoost are based on ensemble techniques of machine learning, showed high scores for accuracy and recall, which shows the importance of synthetic data for making machine and deep learning models more robust.

The precision, recall, and f1 score values were all around 0.99 in Version 2 (10,000 records of PD voice recordings), which meant that all ML models were performing at the same level. In Version 3 (15,000 records of PD voice recordings), the gains were almost unnoticeable. This shows when the synthetic datasets get bigger in size, the benefits decrease.

Figures 13, 14, and 15 are graphs made with version 2 (10,000 records) because it performed better than other versions of synthetic data. The graphs also show how the training and validation loss for a Convolutional Neural Network (CNN) model changed over 50, 100, and 500 epochs. For measuring the modelling error by analysing its loss which shows lower values mean better results in synthetic data. By comparing different numbers of epochs, we can observe how the model generalizes and determine the chances of overfitting in DL models.

We also found that the training loss for the 50 epoch model is showing relatively low and stable performance, whereas the validation loss fluctuates but remains close to the training loss. This shows that the trade off between overfitting the training data and

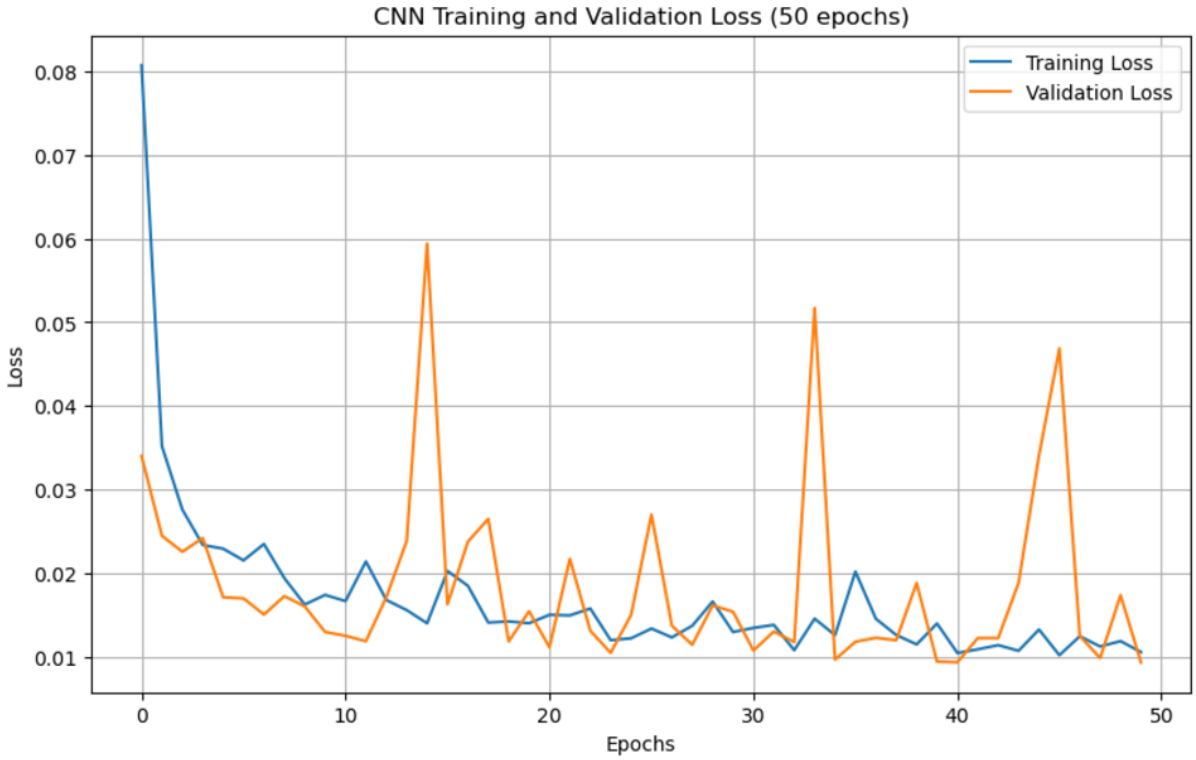


Figure 13: CNN Training and Validation loss (50 epochs) (Source: Pervez, Irfan (2024))

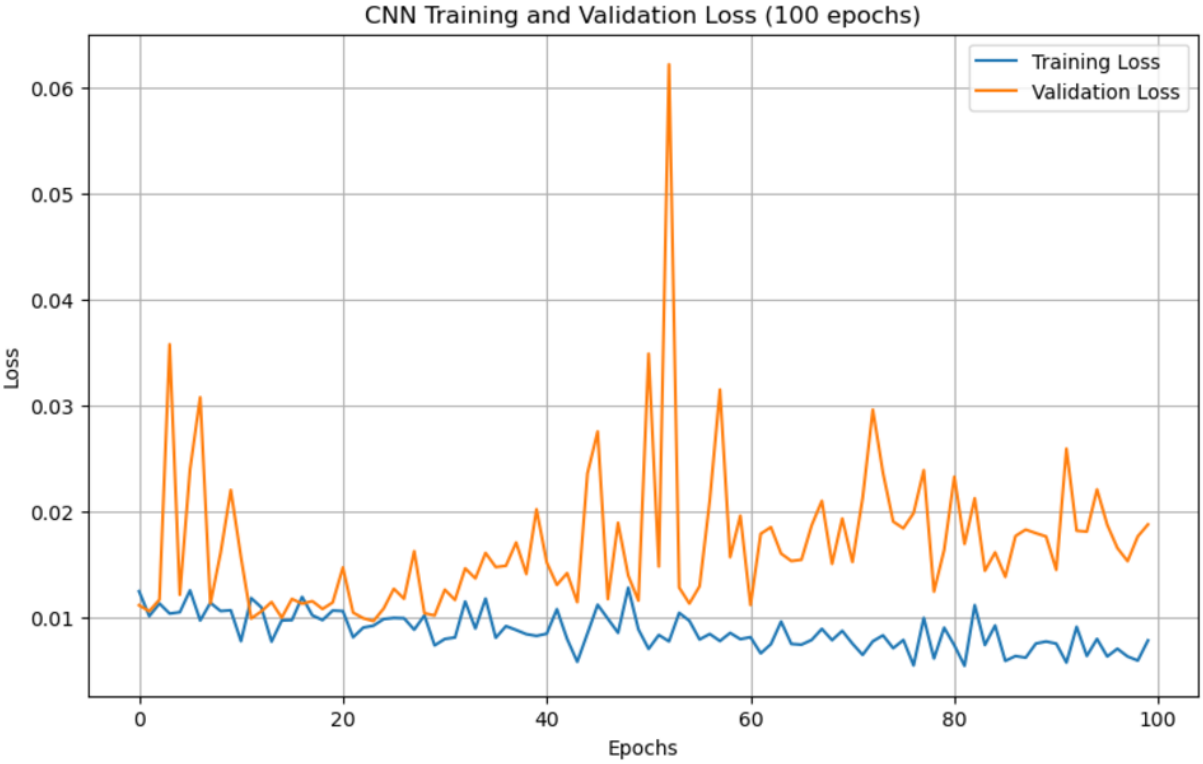


Figure 14: CNN Training and Validation loss (100 epochs) (Source: Pervez, Irfan (2024))

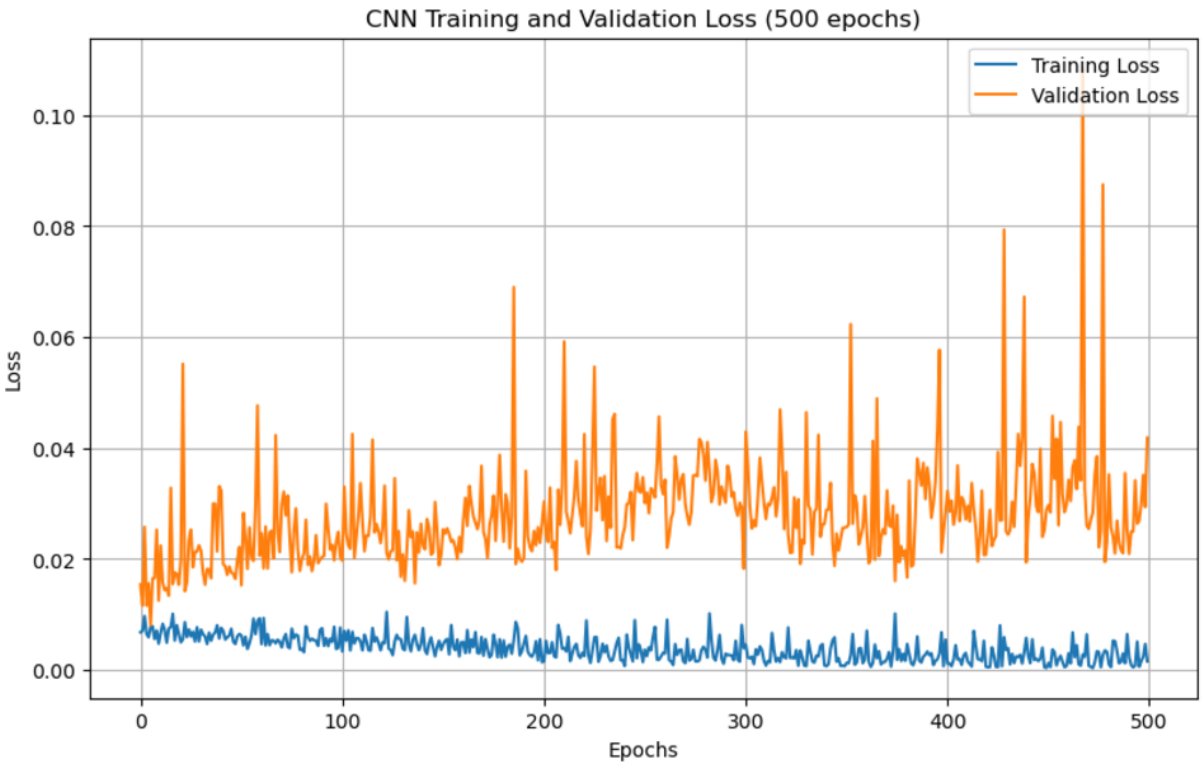


Figure 15: CNN Training and Validation loss (500 epochs) (Source: Pervez, Irfan (2024))

being able to use it with new data. Even though we observed that it's not fully optimized, the model has a good mix between bias and variance.

The 100 epoch model shows a low and stable training loss, where as the validation loss begins to rise and shows greater fluctuations. This behavior proves the emergence of overfitting, where in the model starts to memorize the training data instead of generalizing to new data. The disparity between training and validation loss is more significant than in the case of 100 epoch.

In the 500 epoch model, the training loss remains consistently low however, the validation loss exhibits instability and increases significantly over time. The pronounced overfitting indicates that the model has excessively adapted to the training data, resulting in diminished performance on new data.

The model trained for 50 epochs has the best performance, as shown by the fact that the training and validation losses are close to each other. This suggests that the model is generalizing this well. In contrast to that, extended training in the 100 and 500 epochs model results in overfitting. Implementing techniques like early stopping and regularization improves performance in extended training scenarios.

7 Discussion

The point of this study is to find out how well GAN generated synthetic data working for training ML and DL models for classification. There were eight different classifiers tested. You can see them all in Figure 13. The study looked at how well each classifier did on artificial datasets that had 5,000, 10,000, and 15,000 recordings of PD voice recordings.

According to the results, models trained on synthetic data performed well compared to the models trained on real data. This suggests that GANs can create synthetic datasets with useful and typical features for classification tasks. The effectiveness of classifiers changed depending on the amount of synthetic input. This shows the pros and cons of using GAN generated data for ML and DL model evaluation.

Increasing the number of records in the artificial datasets from 5,000 to 10,000 led to small improvements in performance across most of ML and DL models. This trend was especially clear in simple models like decision trees and logistic regression, where more data probably made it easier to tell features apart and made generalization better. DL models like ANN and FCNN, on the other hand, are less affected by the size of the dataset. This means they can find relevant trends in smaller synthetic datasets. This fits with what is Esteban et al. (2017) found, which is that neural networks are very good at finding trends in GAN generated data.

Random Forest and XGBoost, which are both ensemble classifiers, always did a great job with all types and amounts of data. On the other hand, logistic regression and decision trees gave less reliable results. In this case, it shows that ensemble models can handle changes in the quality of synthetic data well. This finding backs up a study conducted by Zhang et al. (2020) that talked about how robust ensemble methods are when dealing with changing data quality aspects.

There were different amounts of synthetic datasets that DL classifiers like ANN, CNN, and FCNN did well with. This backs up the idea that neural networks can learn complicated patterns from data created by GAN models. This finding fits with what is Goodfellow et al. (2014) found neural networks are great at modeling the complex data distributions generated by GAN models for all three versions of the synthetic dataset (As shown in the appendix 1,2 and 3). It was clear that CNN and FCNN could do well even with small datasets, which shows that they can generalize based on synthetic data.

All models had similar results when trained on both synthetic and real data. This suggests that GANs may be able to make synthetic datasets that are very similar to how real-world data is distributed. This observation backs up what Esteban et al. (2017) found, which suggests that GANs may be a good option when there isn't a lot of real data available.

8 Conclusion and Future Work

This study demonstrates that GAN-generated synthetic data improves Parkinson's disease diagnosis accuracy rate by addressing the research question: "Can synthetic data generated by a generative AI application based on a GAN model improve the prediction of Parkinson's disease diagnosis compared to state-of-the-art technology?".

This study concluded that synthetic data significantly enhanced the accuracy of both machine and deep learning models. However, we observed the diminishing effect in the performance improvement after 10,000-15,000 records, indicating a saturation threshold.This study also emphasizes that synthetic data can be a valuable resource when real data is scarce, offering cost-effective solutions for improving model performance, particularly in fields with limited data, such as life sciences. However, excessive data augmentation can also lead to the poor results, showing the need for appropriate techniques and dataset sizes.

DL models, especially FCNN and ANN, showed marked improvements when they trained on synthetic data, particularly with around 10,000 records compared to the original dataset. In contrast to that, conventional ML models like decision trees and KNN showed improvement but were still outperformed by DL models. Models trained on synthetic data performed similarly to those trained on real data, suggesting that synthetic data can closely mimic real-world data distributions, making it a valid alternative when real data is scarce.

The future work related to this field should validate these findings across other diversified datasets and explore the ethical considerations in using synthetic data for medical diagnostics. Generative adversarial networks can produce much better simulated data if their design is improved. We have better models like Wasserstein GANs and conditional GANs which might be able to improve data variety, which could help easier classifiers like decision trees and logistic regression Arjovsky et al. (2017). Adding a local interpretable model-agnostic explanations (LIME) framework on top of results generated by machine and deep learning models classifiers may produce results that are strong and easy to understand Ribeiro et al. (2016).

We can also look for other newly introduced data augmentation techniques like Variational Autoencoders (VAEs) or synthetic minority oversampling technique (SMOTE), to make classifiers even more useful, especially for datasets that aren't balanced Chawla et al. (2002). New research proves that GAN generated data can be used to train machine learning models well, especially when there isn't enough real data. This supports earlier research on how GAN can be used to improve data Goodfellow et al. (2014).

If we are adding more comprehansive metrics and validation methods to the GAN architecture and making them better could make GAN generated data more reliable in many ML based applications. One important thing that this study shows is that GANs can make good synthetic data, especially for deep learning use cases.

References

Arjovsky, M., Chintala, S. & Bottou, L. (2017), 'Wasserstein gan', *arXiv preprint arXiv:1701.07875* . Available at: https://arxiv.org/abs/1701.07875 (Accessed: 2 December 2024).

Brownlee, J. (2020), *Mastering Machine Learning Algorithms*, Packt Publishing. Available at: https://www.packtpub.com/product/mastering-machine-learning-algorithms/.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002), 'Smote: Synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research* **16**, 321–357.

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W. F. & Sun, J. (2017), Generating multi-label discrete patient records using generative adversarial networks, *in* 'Proceedings of the 2nd Machine Learning for Healthcare Conference'. Available at: https://proceedings.mlr.press/v68/choi17a.html (Accessed: 1 December 2024).

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B. & Bharath, A. A. (2018), 'Generative adversarial networks: An overview', *IEEE Signal Processing Magazine* **35**(1), 53–65.

Dorsey, E. R., Elbaz, A., Nichols, E., Abbasi, N., Abd-Allah, F., Abdelalim, A. & Benseñor, I. M., e. a. (2018), 'Global, regional, and national burden of parkinson's disease, 1990–2016: a systematic analysis for the global burden of disease study 2016', *The Lancet Neurology* **17**(11), 939–953.

Esteban, C., Hyland, S. & Rätsch, G. (2017), 'Real-valued (medical) time series generation with recurrent conditional gans', *arXiv preprint arXiv:1706.02633* . Available at: https://arxiv.org/abs/1706.02633.

Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J. & Greenspan, H. (2018), 'Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification', *Neurocomputing* **321**, 321–331.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), 'Generative adversarial networks', *arXiv preprint arXiv:1406.2661* . Available at: https://arxiv.org/abs/1406.2661.

Hargreaves, C. & Heng, S. (2020), 'Synthetic diabetic patient datasets using generative adversarial networks (gans): Improving machine learning classification accuracy', *Artificial Intelligence in Medicine* **107**, 101871.

Hazra, A. & Byun, Y.-C. (2021), 'Synsiggan: A generative adversarial network framework for synthetic medical signals generation', *Biomedical Signal Processing and Control* **68**, 102683.

Kiyasseh, D., Zhu, T. & Clifton, D. A. (2021), 'Plethaugment: Gan-based data augmentation for improved classification of physiological waveforms', *arXiv preprint arXiv:2101.03002* . Available at: https://arxiv.org/abs/2101.03002.

Lee, M. (2023), 'Recent advances in generative adversarial networks for gene expression data', *Mathematics* **11**(14), 3055.
URL: *https://doi.org/10.3390/math11143055*

Little, M. A., McSharry, P. E., Hunter, E. J. & Ramig, L. O. (2008), 'Suitability of dysphonia measurements for telemonitoring of parkinson's disease', *IEEE Transactions on Biomedical Engineering* **56**(4), 1015–1022.

Little, M. A., McSharry, P. E., Hunter, E. J., Spielman, J. & Ramig, L. O. (2009), 'Suitability of dysphonia measurements for telemonitoring of parkinson's disease', *IEEE Transactions on Biomedical Engineering* **56**(4), 1015–1022.

Ramzan, F., Sartori, C., Consoli, S. & Reforgiato Recupero, D. (2024), 'Generative adversarial networks for synthetic data generation in finance: Evaluating statistical similarities and quality assessment', *AI* **5**(2), 667–685.
URL: *https://doi.org/10.3390/ai5020035*

Ribeiro, M. T., Singh, S. & Guestrin, C. (2016), '"why should i trust you?": Explaining the predictions of any classifier', *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* pp. 1135–1144.

Sakar, B. E., Isenkul, M. E., Sakar, C. O., Sertbas, A., Gurgen, F., Delil, S., Apaydin, H. & Kursun, O. (2013), 'Collection and analysis of a parkinson speech dataset with multiple types of sound recordings', *IEEE Journal of Biomedical and Health Informatics* **17**(4), 828–834.

Tsourtis, A., Papoutsoglou, G. & Pantazis, Y. (2022), 'Gan-based training of semi-interpretable generators for biological data interpolation and augmentation', *Applied Sciences* **12**(11), 5434.
URL: *https://www.mdpi.com/2076-3417/12/11/5434*

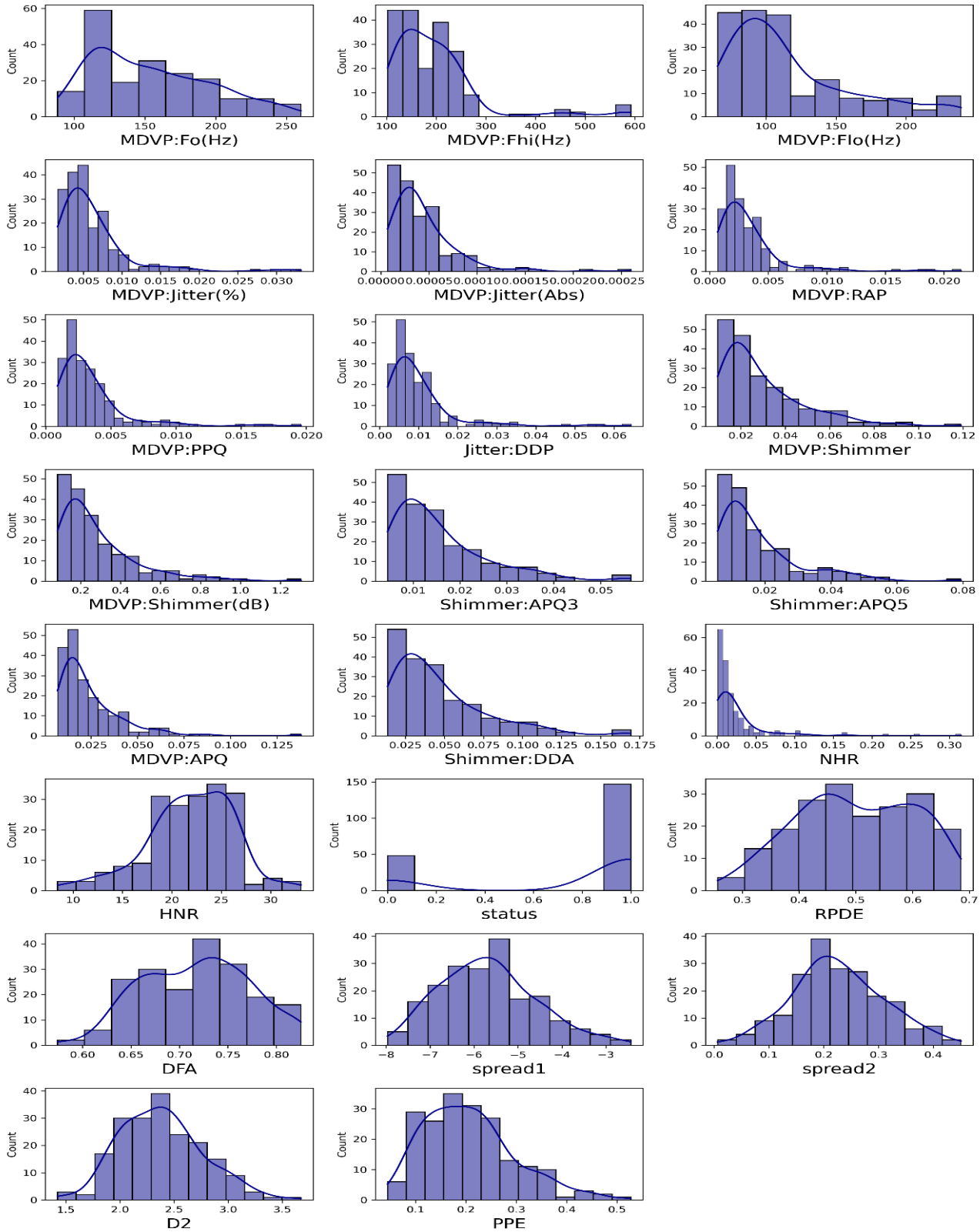
UCI Machine Learning Repository (2008), 'Parkinson's disease data set'. Available at: `https://archive.ics.uci.edu/ml/datasets/parkinsons`.

Xu, Z.-J. et al. (2020), 'Parkinson's disease detection based on spectrogram-deep convolutional generative adversarial network sample augmentation', *Journal of Biomedical and Health Informatics* **8**(12), 206889–206890.

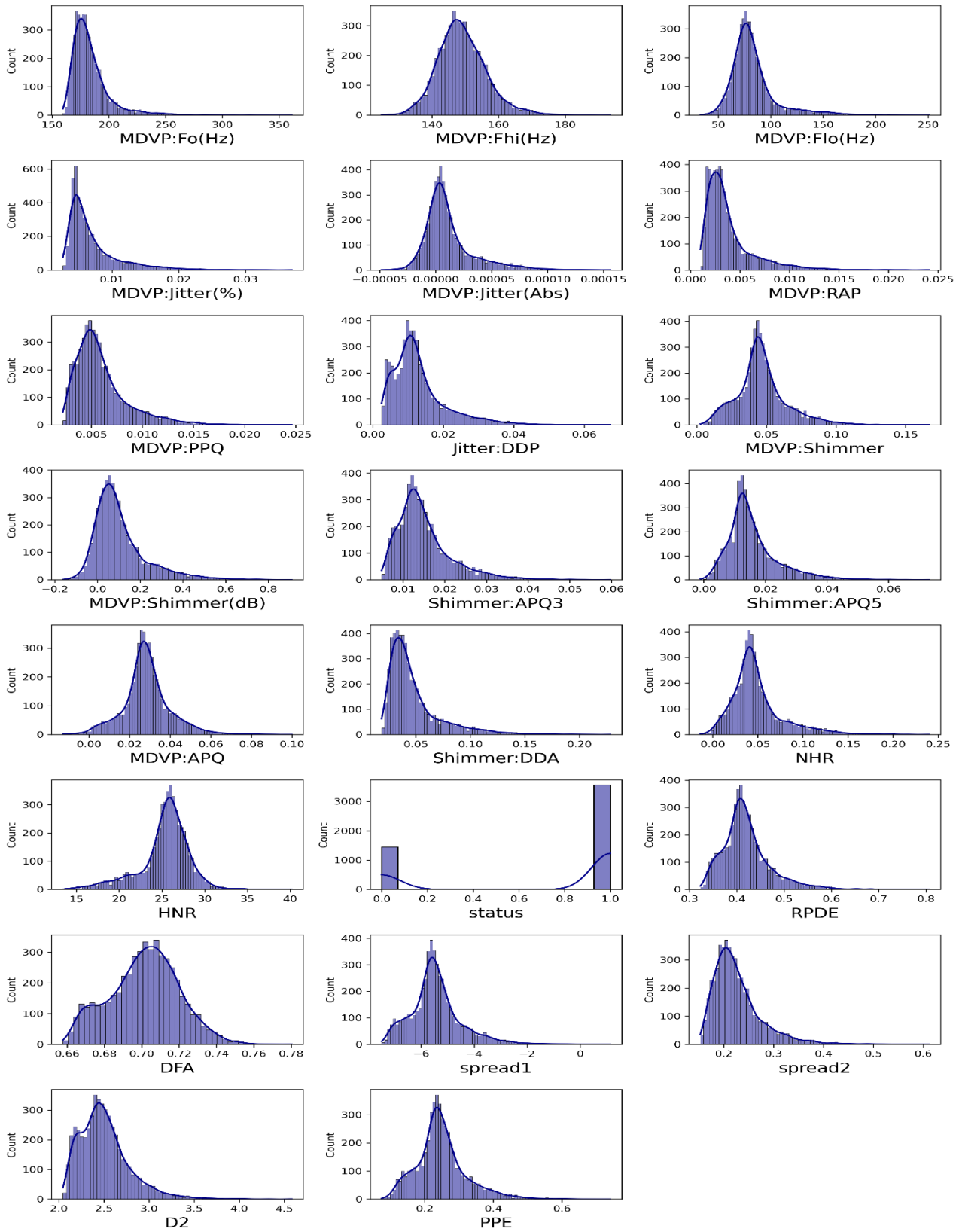
Yoon, J., Jarrett, D. & van der Schaar, M. (2019), 'Time-series gans for data augmentation in clinical studies', *arXiv preprint arXiv:1904.00503* . Available at: `https://arxiv.org/abs/1904.00503`.

Zhang, H., Li, Y. & Zhang, X. (2020), 'A survey of data augmentation techniques for deep learning', *Journal of Computer Science and Technology* **35**(4), 859–874.

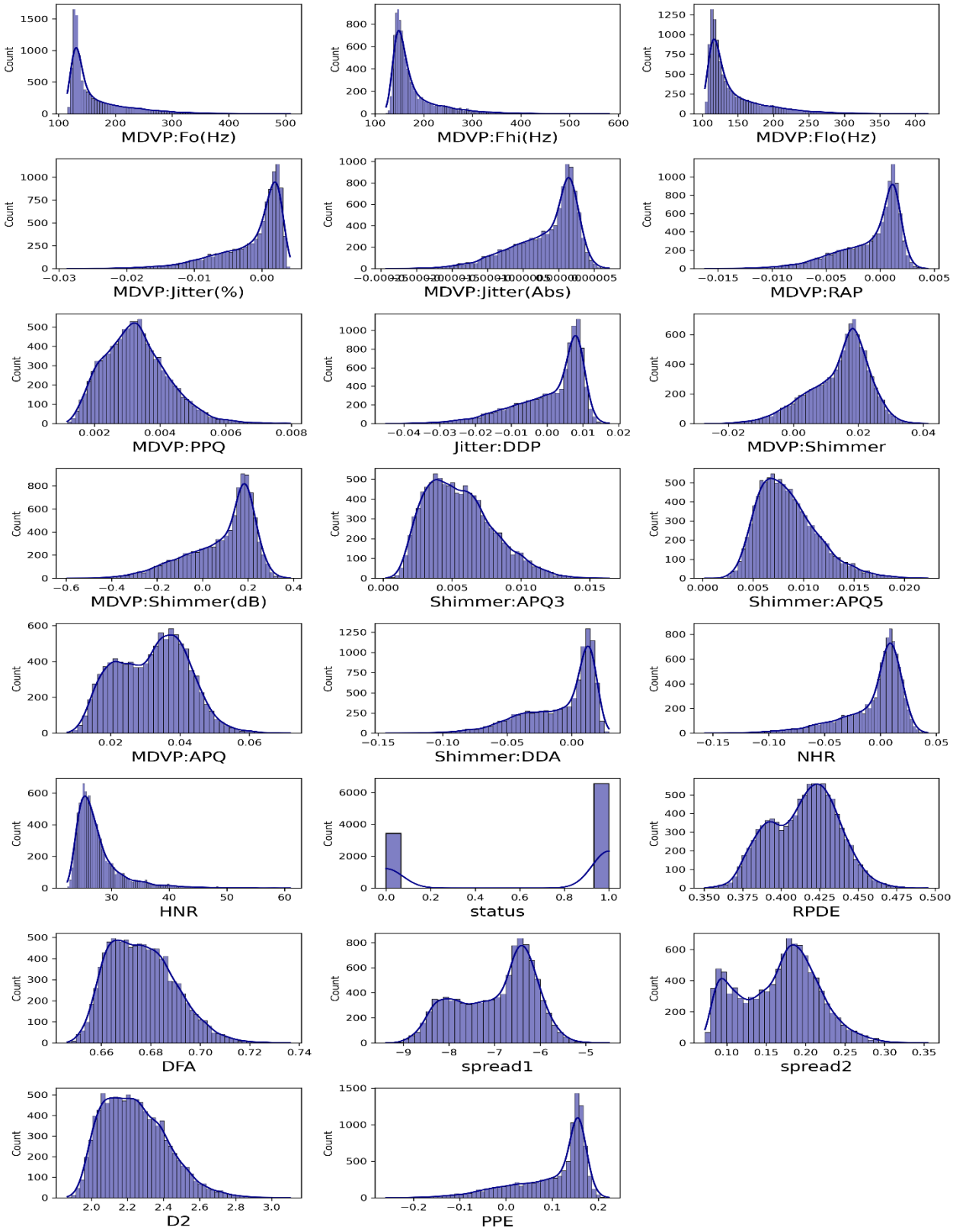
Appendix 1 : Data Visualization



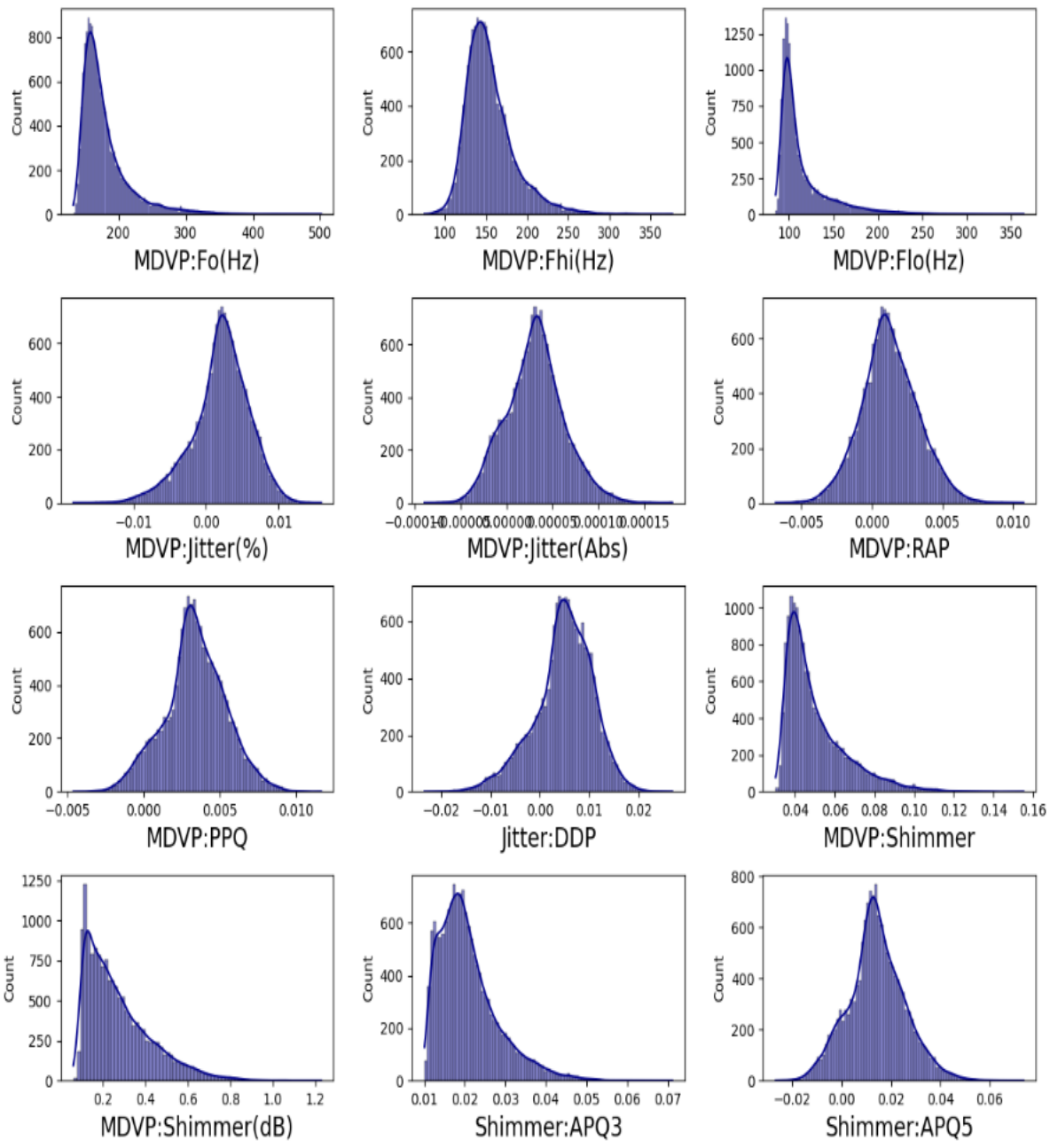
Original (Records count 195) (Source: Pervez, Irfan (2024))



Version 1 - Synthetic Data (Records count 5000) (Source: Pervez, Irfan (2024))

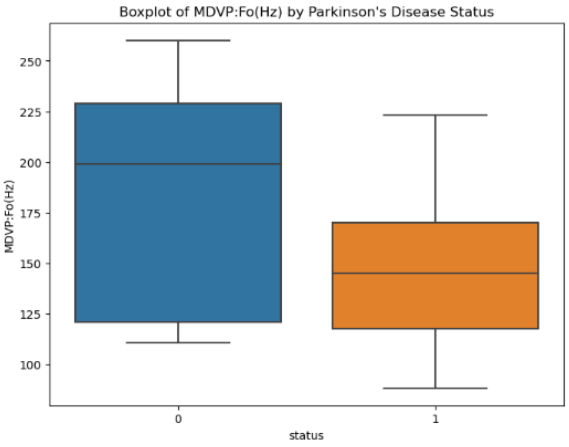


Version 2 - Synthetic data (Records Count 10000) (Source: Pervez, Irfan (2024))

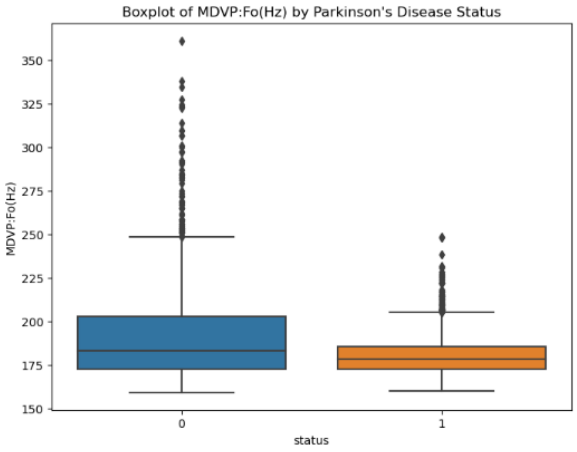


Version 3 - Synthetic data (Records Count 15000) (Source: Pervez, Irfan (2024))

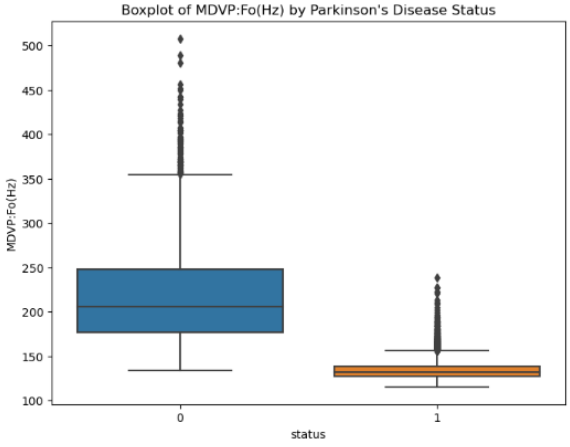
Appendix 2 :



Original (Records count 195)



Iteration 1 - Synthetic Data (Records count 5000)

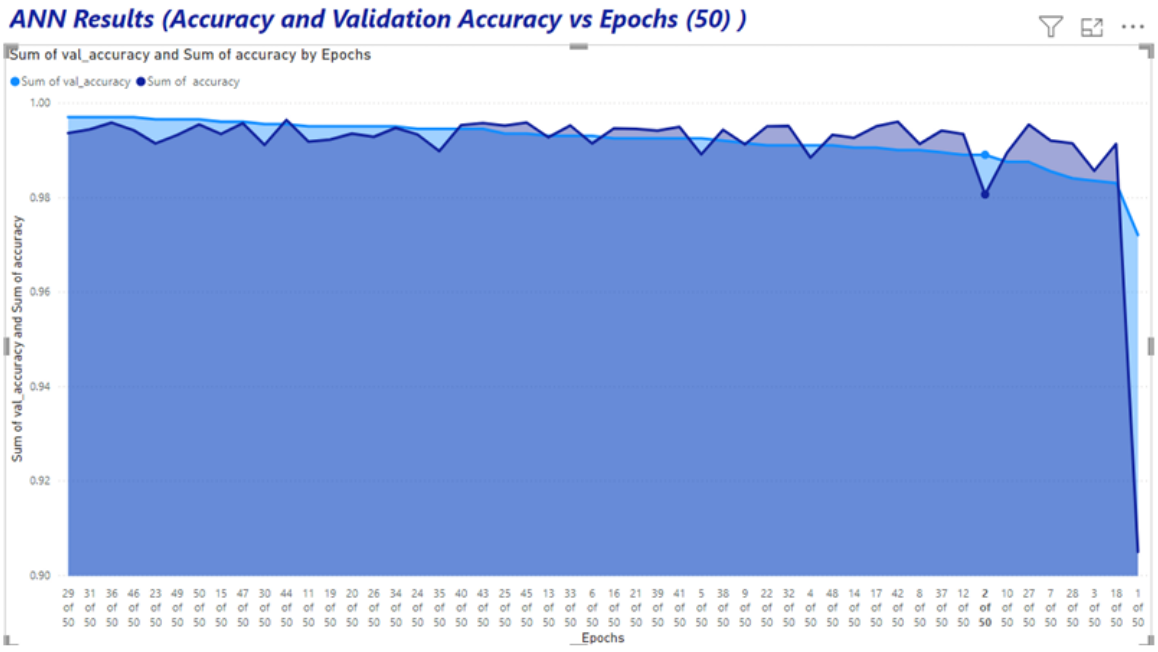


Iteration 2 - Synthetic data (Records Count 10000)

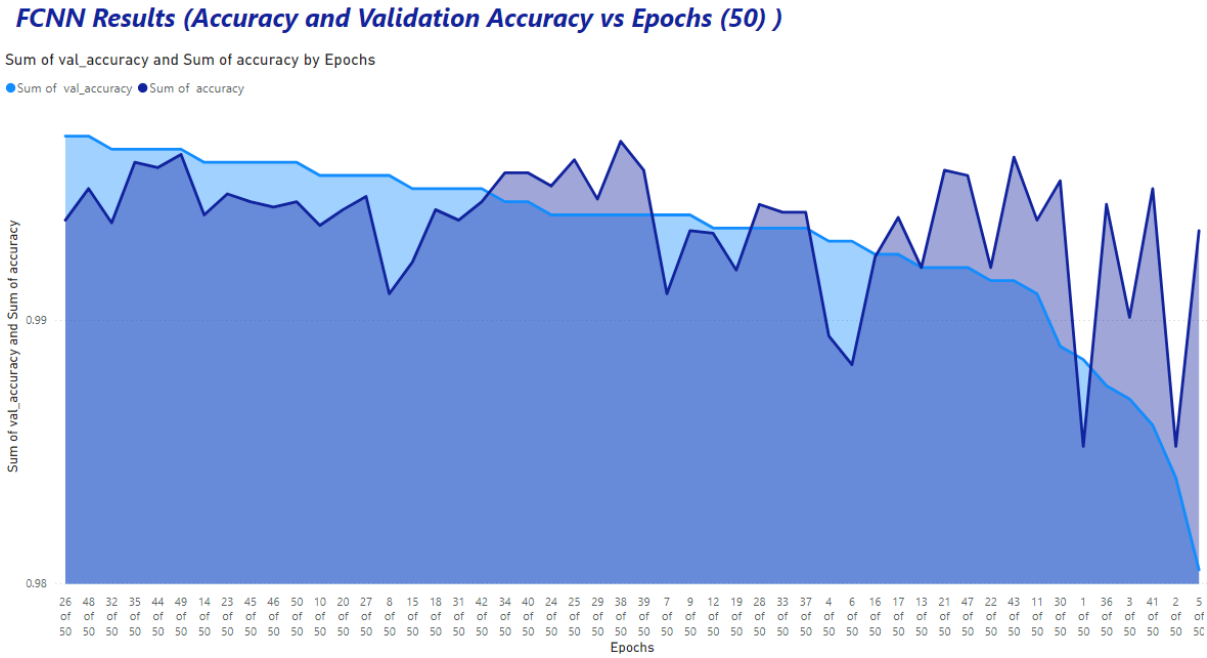
Iteration 2 - Synthetic data (Records Count 15000)

Boxplot to Show Feature Distributions MDVP:Fo(Hz) (Source: Pervez, Irfan (2024))

Appendix 3 : Power BI visualization for Deep learning and Machine learning classifier achieved best results on Version 2 - Synthetic data (10000 Records of PD voice recordings)



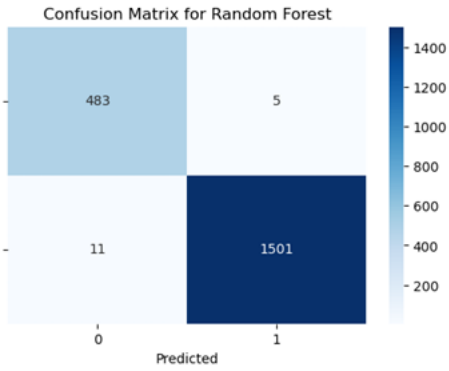
(Source: Pervez, Irfan (2024))

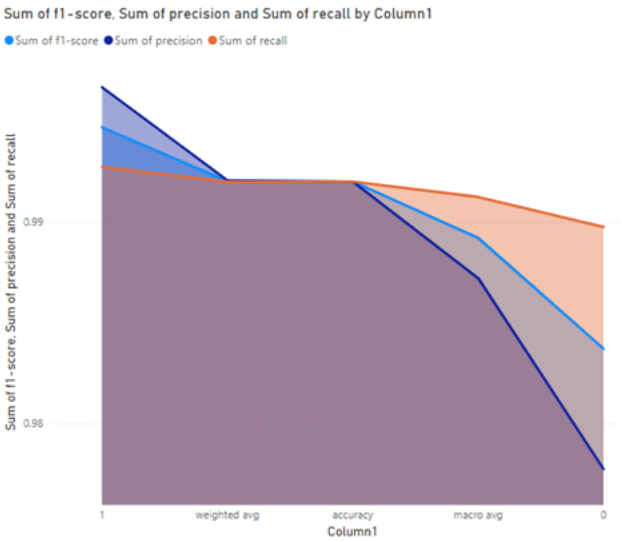


(Source: Pervez, Irfan (2024))

Random Forest Classifier Results

Column1	Sum of f1-score	Sum of precision	Sum of recall	Sum of support
0	0.98	0.98	0.99	488.00
1	0.99	1.00	0.99	1,512.00
accuracy	0.99	0.99	0.99	0.99
macro avg	0.99	0.99	0.99	2,000.00
weighted avg	0.99	0.99	0.99	2,000.00
Total	**4.95**	**4.95**	**4.96**	**6,000.99**

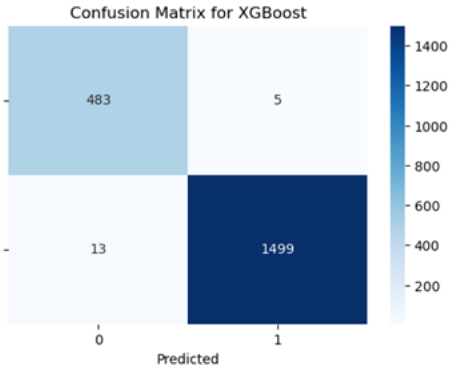


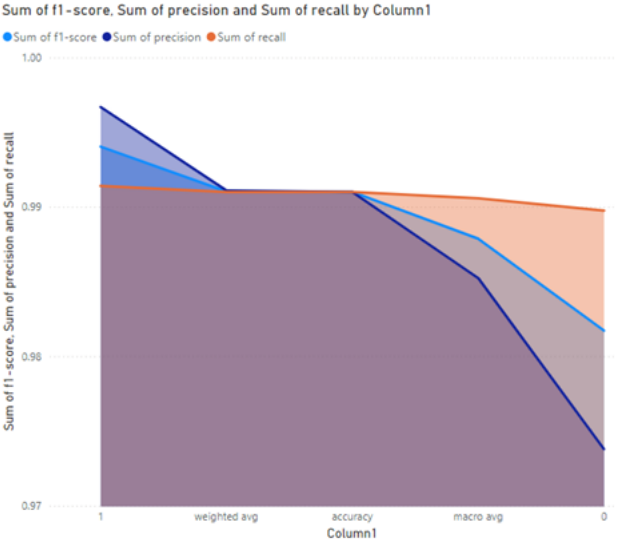


(Source: Pervez, Irfan (2024))

XG Boost Classifier Results

Column1	Sum of f1-score	Sum of precision	Sum of recall	Sum of support
0	0.98	0.97	0.99	488.00
1	0.99	1.00	0.99	1,512.00
accuracy	0.99	0.99	0.99	0.99
macro avg	0.99	0.99	0.99	2,000.00
weighted avg	0.99	0.99	0.99	2,000.00
Total	**4.95**	**4.94**	**4.95**	**6,000.99**

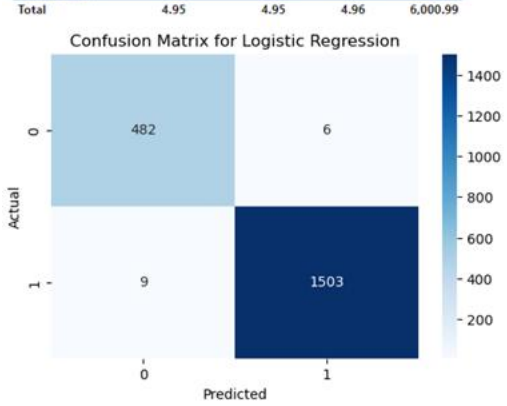


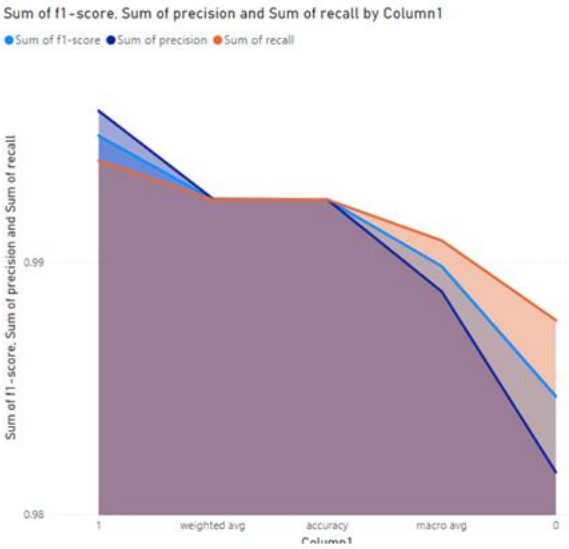


(Source: Pervez, Irfan (2024))

Logistic Regression Results

Column1	Sum of f1-score	Sum of precision	Sum of recall	Sum of support
0	0.98	0.98	0.99	488.00
1	1.00	1.00	0.99	1,512.00
accuracy	0.99	0.99	0.99	0.99
macro avg	0.99	0.99	0.99	2,000.00
weighted avg	0.99	0.99	0.99	2,000.00
Total	**4.95**	**4.95**	**4.96**	**6,000.99**

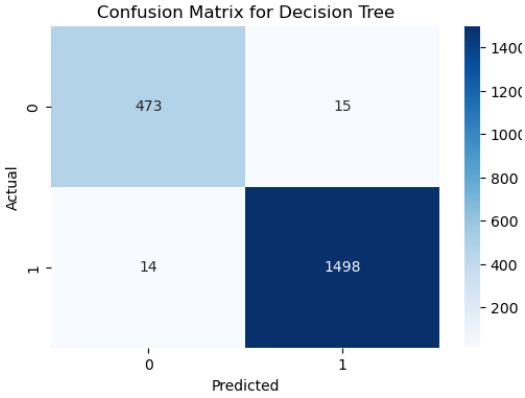


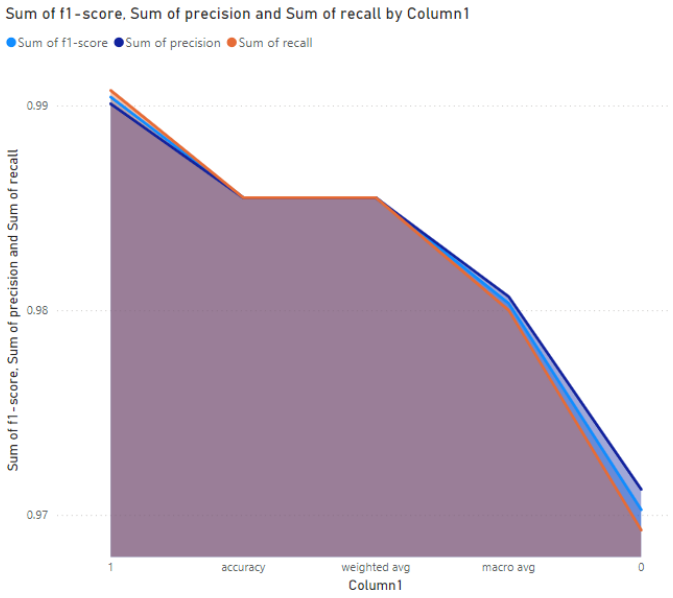


(Source: Pervez, Irfan (2024))

Decision Tree Results

Column1	Sum of f1-score	Sum of precision	Sum of recall	Sum of support
weighted avg	0.99	0.99	0.99	2,000.00
macro avg	0.98	0.98	0.98	2,000.00
accuracy	0.99	0.99	0.99	0.99
1	0.99	0.99	0.99	1,512.00
0	0.97	0.97	0.97	488.00
Total	**4.91**	**4.91**	**4.91**	**6,000.99**





(Source: Pervez, Irfan (2024))