Contents

List of Figures

List of Tables

Abstract

In the era of data-driven intelligence, access to large, diverse, and high-quality datasets remains a challenge due to privacy concerns, scarcity, and labeling costs. This project presents a comprehensive approach to synthetic data generation using generative models—namely, Variational Autoencoders (VAEs), Deep Convolutional GANs (DCGANs), and Wasserstein GANs with Gradient Penalty (WGAN-GP). By leveraging these architectures, synthetic data was generated across both tabular and image domains using datasets like Iris, ADHD-200, MNIST, and Oxford 102 Flowers.

The project explores the full lifecycle: from data preprocessing and model design to training and evaluation. The VAE models were fine-tuned with different latent dimensions to balance reconstruction loss and KL divergence. Image-based GANs were trained iteratively, with evaluation metrics including Fréchet Inception Distance (FID) applied where feasible.

The results validate that synthetic data can closely mimic real data distributions while preserving structure and variability. The findings also underscore the impact of hardware limitations in deep learning workflows, particularly during evaluation with large-scale models like InceptionV3. This work establishes a scalable and modular baseline for future exploration in privacy-preserving and data-efficient machine learning.

Chapter 1

Introduction

In the field of Artificial Intelligence (AI), the role of data has become increasingly pivotal. As machine learning and deep learning models grow more complex, so too does their appetite for large, high-quality datasets. However, real-world datasets often present significant challenges: they can be scarce, noisy, imbalanced, or sensitive due to privacy concerns. Particularly in domains such as healthcare, finance, and security, data accessibility is restricted by ethical, legal, or infrastructural limitations. This has led to the emergence of synthetic data generation as a promising solution to address these challenges.

Synthetic data refers to data that is artificially generated rather than collected from real-world events. While traditional data augmentation techniques involve transformations of existing data (such as flipping or rotation in images), synthetic data generation can create completely new samples that are statistically representative of the original dataset. This capability is enabled by powerful generative models such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), which have shown tremendous promise in learning the underlying distribution of real data and generating novel, high-quality samples from it.

The goal of this project is to develop a comprehensive synthetic data generation system using deep learning techniques that span both tabular and image-based datasets. Our solution applies VAEs for structured/tabular data like the Iris dataset and the ADHD-200 neuropsychological dataset, while DCGAN and WGAN-GP models are employed for unstructured image datasets such as MNIST and the Oxford 102 Flowers dataset. These models were chosen for their effectiveness in their respective domains, and special care was taken to design and train them under hardware-constrained environments.

The motivation behind this project stems from a confluence of factors. Firstly, real-world data acquisition is expensive and often incomplete. In the case of ADHD phenotypic data, for instance, many features may be missing or labeled inconsistently, which compromises model performance. Secondly, the ability to generate privacy-preserving data is essential for public research dissemination. Thirdly, training deep learning models with synthetic data can help mitigate overfitting, especially when the real data is limited. Lastly, such models also provide interpretability into latent structures within the data and help explore potential scenarios or distributions that may not be well represented in the original dataset.

This system begins with robust preprocessing steps tailored to each dataset. For tabular data, this includes handling missing values, filtering out outliers using z-score methods, and scaling using MinMaxScaler. For images, resizing and normalization are essential to ensure that pixel distributions are suitable for model convergence. Once the datasets are prepared, the models are designed, trained, and evaluated based on their ability to replicate the structure and diversity of the input data.

For the tabular datasets, the VAE architecture includes an encoder network that compresses the input features into a lower-dimensional latent space and a decoder network that reconstructs the original data from this latent representation. The model is trained using a combination of reconstruction loss and KL divergence, with KL annealing strategies employed to improve training stability. For ADHD data, it was observed that a latent dimension of 8 increased KL divergence excessively; hence, a smaller dimension of 5 was chosen.

For image datasets, DCGAN was used for MNIST due to its efficiency and suitability for grayscale images. The model was trained iteratively up to 500 epochs, where we observed significant qualitative improvements in digit formation and consistency. For more complex RGB image generation tasks like Oxford Flowers, WGAN-GP was implemented due to its stable training dynamics and ability to model high-resolution image details. It also introduced a gradient penalty to enforce the Lipschitz constraint, replacing the unstable loss metrics of standard GANs.

Evaluation plays a crucial role in validating the quality of generated data. For tabular data, kernel density estimation (KDE) plots were used to compare real and synthetic data distributions. For images, visual inspection over epochs provided early signs of learning, while FID (Fréchet Inception Distance) was considered as a quantitative metric. However, due to hardware limitations, computing FID scores at scale, especially for the Oxford Flowers dataset, was challenging.

Despite these constraints, the models produced high-fidelity synthetic data that could feasibly augment real-world datasets in downstream applications. Furthermore, this project highlights the practical trade-offs between model performance and computational resources. In environments with limited hardware, strategies like reducing batch size, model simplification, and early stopping become vital.

In summary, this project successfully implements a suite of generative models to tackle the problem of data scarcity using synthetic generation. Through structured methodology, diverse dataset support, and modular evaluation pipelines, it serves as a practical blueprint for integrating synthetic data in machine learning workflows. As the field moves toward responsible AI and data-centric development, such tools are not just optional but necessary.

Chapter 2

Literature Survey

1. Comprehensive Exploration of Synthetic Data Generation This paper outlines how synthetic data can augment real-world datasets, improve model generalization, and support data privacy. It categorizes generation techniques into GAN-based, VAE-based, and rule-based, emphasizing the importance of domain knowledge in selecting appropriate methods. It also highlights evaluation challenges such as distribution similarity and task-specific metrics. It provides a clear taxonomy of synthetic data types and use cases in AI, healthcare, finance, and computer vision .

2. Machine Learning for Synthetic Data Generation: A Review This review focuses on the role of ML in synthesizing tabular, image, and time-series data. It discusses GAN variants (CGAN, WGAN, BigGAN), data utility vs. privacy trade-offs, and highlights key limitations such as bias replication and training instability. The paper calls for better performance benchmarks and unified frameworks for evaluating synthetic data utility and fidelity .

3. Survey on Synthetic Data Generation, Evaluation Methods and GANs A technical overview is presented on state-of-the-art GAN architectures used for synthetic data generation. It discusses FID and IS metrics in depth and categorizes GANs into unsupervised, semi-supervised, and conditional. Special focus is given to domain-specific applications like medical imaging and autonomous driving. It also discusses the issue of mode collapse and proposes hybrid evaluation frameworks .

4. Synthetic Data: What, Why, and How This foundational paper distinguishes synthetic data into fully synthetic, partially synthetic, and hybrid datasets. It provides an analytical viewpoint on when to prefer synthetic data and outlines legal and ethical considerations in data anonymization. The work emphasizes reproducibility and utility as the primary goals and recommends simulation-based approaches when real data is limited or unavailable .

5. Wasserstein GAN (WGAN) WGAN introduced the Earth Mover's distance to address convergence issues and instability in classical GANs. It removed the sigmoid activation in the discriminator (critic) and replaced it with a continuous scalar output. The Lipschitz constraint was initially enforced by weight clipping, and later improved using gradient penalty (WGAN-

GP), enhancing training robustness and sample diversity .

6. Generation of Synthetic Data with Generative Adversarial Networks This paper explores how GANs can be applied across different data modalities—images, text, and tabular data—for synthetic data generation. It explains the core structure of GANs (generator and discriminator), and their adversarial training process. The authors examine various GAN variants like DCGAN, CGAN, and CycleGAN, focusing on their suitability for different types of data.

Chapter 3

Methodology

3.1 Background and Motivation

3.1.1 Existing System

Machine learning and AI models traditionally rely on real-world datasets collected through manual data entry, web scraping, or sensor-based acquisition. These datasets serve as the foundation for training models in various domains such as healthcare, finance, and autonomous systems. However, obtaining high-quality real-world data poses several challenges.

One major limitation is data scarcity, where collecting large-scale datasets is often impractical or expensive. Additionally, real-world data is prone to biases and class imbalances, leading to unfair model predictions. Privacy concerns further restrict access to sensitive data, particularly in healthcare, banking, and security applications. Moreover, the process of data annotation is time-consuming and costly, making it difficult to scale high-quality datasets.

Due to these challenges, the existing system of relying solely on real-world data is insufficient for training robust AI models. There is a growing need for alternative data generation methods that can provide high-quality, diverse, and privacy-preserving datasets. This leads to the exploration of synthetic data generation techniques, which can overcome the limitations of real-world data while ensuring better model generalization.

3.1.2 Proposed System

To overcome the limitations of real-world data collection, this project proposes a synthetic data generation and integration system to enhance AI model training. Synthetic data, which mimics real-world data distributions, can be used to supplement or replace real datasets, ensuring high-quality, diverse, and privacy-compliant training data.

The system generates synthetic data using Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), which produce realistic and diverse samples. Additionally, rule-based techniques are used for structured data synthesis. The generated data is then integrated into machine learning pipelines through data augmentation, preprocessing, and validation to ensure model robustness.

The proposed system provides several advantages over traditional real-world data collection. It eliminates data scarcity issues, ensures privacy compliance, and reduces the risk of biased datasets. Moreover, it offers a cost-effective alternative to expensive data annotation processes. Preliminary experiments show that models trained with a mix of real and synthetic data achieve higher accuracy and better generalization.

This system paves the way for scalable, privacy-preserving AI model training. Future improvements will focus on enhancing data realism and refining integration techniques for industry adoption.

3.1.3 Objectives

The primary objective of this project is to develop a system for generating and integrating synthetic data to enhance AI model training. Traditional data collection methods suffer from scarcity, bias, and privacy issues, making it essential to explore alternative solutions. This project aims to generate high-quality synthetic datasets and evaluate their effectiveness in AI applications.

The specific objectives include:

- Generating realistic synthetic data using advanced machine learning techniques such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs).
- Integrating synthetic data into AI/ML pipelines to assess its impact on model accuracy and generalization.
- Addressing challenges related to data scarcity, bias, and privacy by providing diverse and privacy-preserving datasets.
- Demonstrating the cost-effectiveness of synthetic data by reducing the need for manual data collection and annotation.

3.2 Hardware and Software Requirements

The development and integration of synthetic data generation require high-performance computational resources due to intensive nature of deep learning models. This project utilizes both hardware and software tools optimized for synthetic creation, processing and model training.

3.2.1 Hardware Requirements

- A powerful CPU (Intel Core i7 / Ryzen 7 or higher) for data processing.
- A dedicated GPU (NVIDIA RTX 4060 or better) for deep learning computations.
- At least 16GB of RAM (preferably 32GB) to handle large datasets.
- SSD preferable.

3.2.2 Software Requirements

The project implementation is based on Python for deep learning, using frameworks like TensorFlow. Data preprocessing is handled using NumPy, Pandas. The development environment consists of Visual Studio Code, ensuring seamless workflow integration.
By utilizing this hardware and software setup, the project ensures efficient data generation, integration, and model evaluation.

3.3 Methodology

1. **Problem Analysis**
    - Identify the type of data required for AI model training.
    - Analyze the challenges with existing datasets such bias, privacy concerns, or data scarcity.
2. **Collection of Referential Data**
    - Select real-world datasets for benchmarking synthetic data performance.
    - **Tabular Data:** Iris, ADHD dataset.
    - **Image Data:** OXford Flower 102,MNIST dataset.
3. **Data Preprocessing**
    - Normalize, scale, and clean referential data.
    - Handle missing values, remove duplicates, and balance class distributions if required.
4. **Generative Model Selection**
    - Choose an appropriate generative models based on data type:
        - **For Image Data:** Generative Adversarial Networks (GANs) (e.g., DCGAN, StyleGAN,WGAN).
        - **Tabular Data:** Variational Autoencoders (VAEs) or CTGAN (Conditional Tabular GAN).
5. **Synthetic Data Generation**
    - Implement and fine-tune generative models to create synthetic data.
    - Ensure data variability and realism while avoiding mode collapse in GANs.
6. **Evaluation Metrics:** Define the metrics to access the quality of synthetic data for training AI models.
7. **Integration with AI models**

- Train AI models on real, synthetic, and mixed datasets.
- Evaluate performance differences using metrics like accuracy, F1-score, precision-recall curves.
- Test synthetic data impact on models like CNNs (for images) and Decision Trees, Random Forest, KNNs or Neural Networks .

3.4 Specifications:

3.4.1 Datasets Used

Table 3.1: Summary of Datasets Used

Dataset	**Type**	**Samples**	**Features / Dimensions**	**Used For**
Iris	Tabular	150	4	VAE Training
ADHD-200	Tabular	∼150	5	VAE Training
MNIST	Image (Gray)	70,000	28x28	DCGAN
Oxford Flowers	Image (RGB)	8,189	64x64	WGAN-GP

3.4.2 Data Format and Storage:

- **Input Data Format:** CSV,JSON and Image Files(PNG/JPEG).
- **Generated Format:** Same as input for consistancy.
- **Storage:**Local Directory or clould storage.

3.4.3 Performance Metrics:

To evaluate the quality of generated synthetic data.

- **FID Score (Fréchet Inception Distance)** - Measures image quality.
- **KL Divergence** - KL Divergence (Kullback-Leibler Divergence) is a key component in VAEs. It measures how much the learned latent distribution differs from a predefined prior distribution.

3.5 Architecture

1. Referential Data is first collected and undergoes Data Preprocessing to clean and format it.
2. The preprocessed data is fed into GANs/VAEs, which generate synthetic data by learning patterns from the original data.
3. The generated data is processed by a Synthetic Data Generator and then moved to the Deployment stage.
4. The deployed synthetic data is stored in a Synthetic Data Storage system.
5. Both Referential Data and Synthetic Data are integrated into AI pipelines for model training, testing, or other applications.

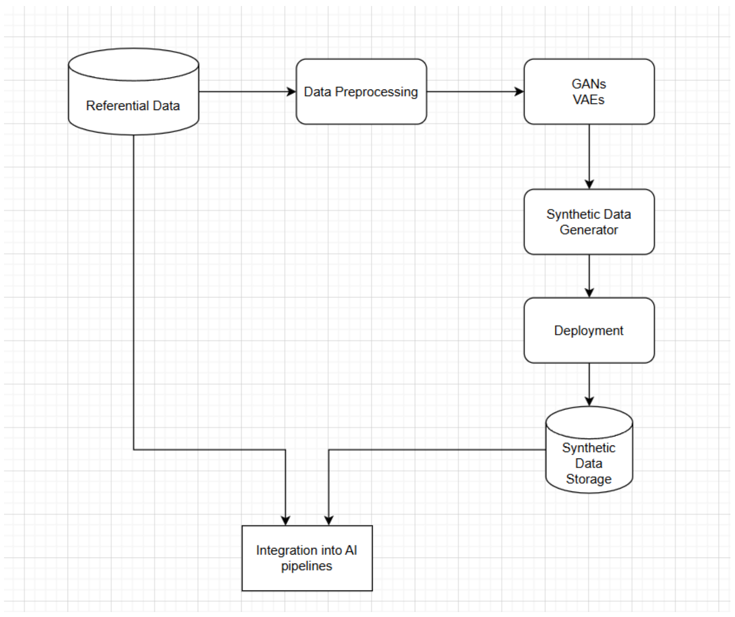


Figure 3.1: Architecture of Synthetic Data Generation Engine

Chapter 4

System Design

This chapter details the implementation of synthetic data generation using Variational Autoencoders (VAE) and GANs with variations such as DCGAN and WGAN. The implementation covers data preprocessing, model architecture, training, and generation of synthetic outputs for both tabular and image datasets.

4.1 Technology Stack

- **Programming Language:** Python 3.12
- **Libraries:** TensorFlow, Keras, Scikit-learn, NumPy, Pandas, Matplotlib, Seaborn, Joblib, Os, Scipy, Fastapi, Emum, Model-Util, PIL, Shutil.
- **IDE:** Visual Studio Code, Google Colab, Kaggle Notebook.

4.2 Data Preprocessing

The raw datasets—specifically the Iris dataset and the ADHD-200 phenotypic dataset—contained several inconsistencies and required thorough preprocessing to ensure model stability and accuracy.

- Invalid values such as `-999`, `N/A`, `pending`, or non-numeric characters (e.g., `L`) were replaced with `NaN`.
- Rows containing any missing data were removed.
- Outliers were filtered using the Z-score method: rows with any feature exceeding a Z-score magnitude of 3 were dropped.
- All features were scaled to the range [0, 1] using `MinMaxScaler`.
- The fitted scalers were saved using `joblib` to ensure consistency during inverse transformation in generation.

4.3 Model Architecture

4.3.1 Variational Autoencoder (VAE)

The VAE model is used for tabular data such as the Iris and ADHD datasets. It learns to encode inputs into a probabilistic latent space and reconstructs them from sampled latent vectors.

Encoder: Input layer → Dense(16, ReLU) → Dense(8, ReLU) → Two outputs: `z_mean`, `z_log_var`

Sampling Layer: A custom Lambda layer applies the reparameterization trick:

$$z = \mu + \sigma \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0,1)$$

Decoder: Latent input → Dense(8, ReLU) → Dense(16, ReLU) → Dense(output_dim, Linear)

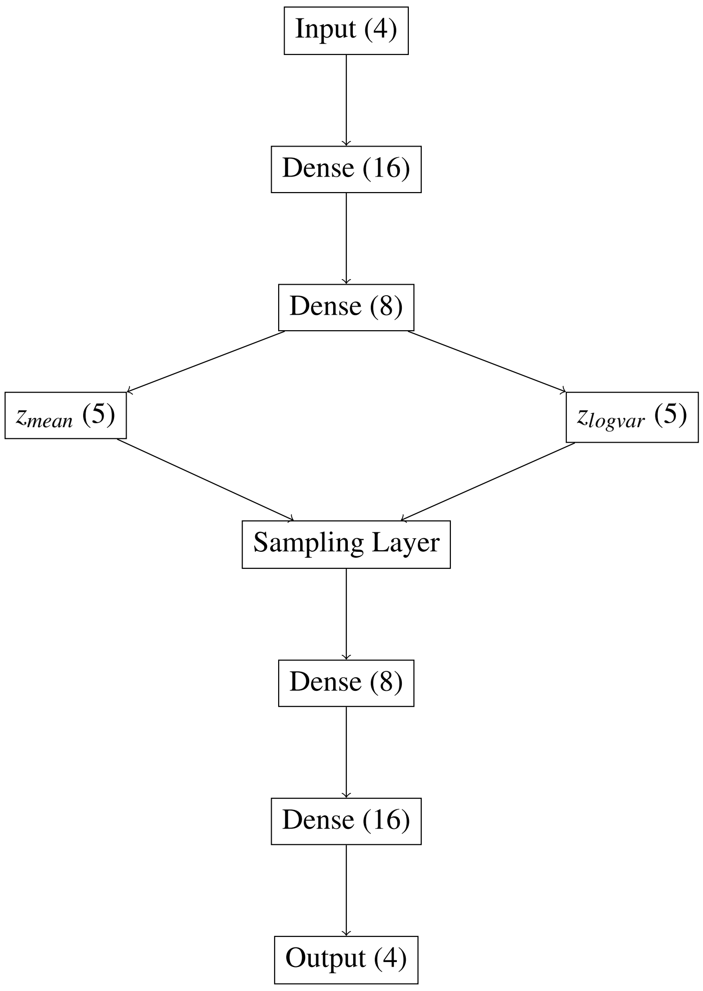


Figure 4.1: VAE Architecture: Encoder, Latent Sampling, and Decoder

4.3.2 Deep Convolutional GAN (DCGAN)

The DCGAN architecture was implemented to generate synthetic grayscale images from the MNIST dataset. The architecture consists of two components:

Generator:

- Input: 100-dimensional noise vector
- Dense layer with reshape to 7x7x256
- Conv2DTranspose layers (stride=2) to upsample to 28x28x1
- Batch normalization and LeakyReLU activations

Discriminator:

- Input: 28x28x1 grayscale image
- Conv2D layers with stride=2 and LeakyReLU
- Flatten and Dense(1) for real/fake prediction

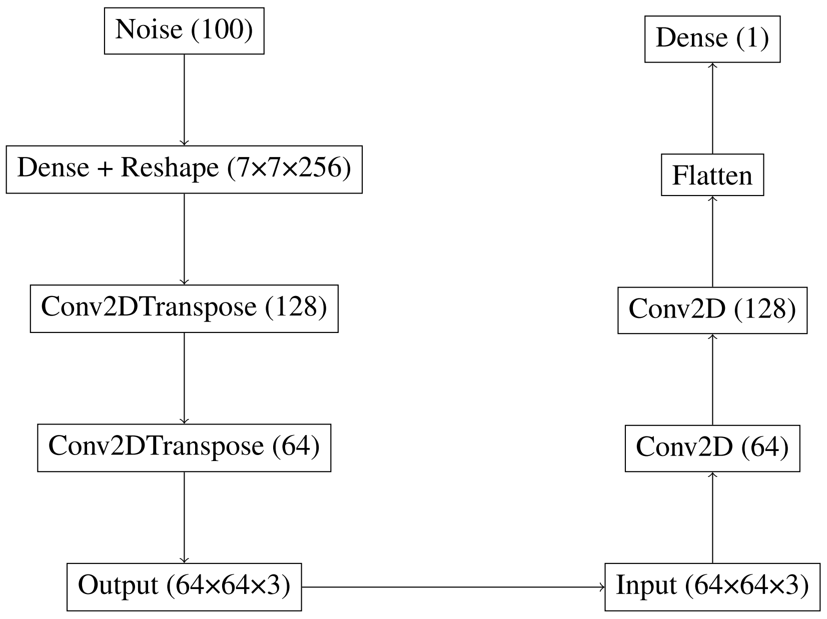


Figure 4.2: DCGAN Architecture: Generator and Discriminator

4.3.3 Wasserstein GAN with Gradient Penalty (WGAN-GP)

The WGAN-GP architecture was implemented to generate realistic 64×64 RGB images using the Oxford 102 Flowers dataset. It improves the stability of GAN training by replacing the Jensen-Shannon divergence with Wasserstein distance and introducing a gradient penalty.
The architecture consists of two components:

Generator:

- Input: 100-dimensional noise vector (latent space)
- Dense + Reshape to (7×7×256)
- 3 `Conv2DTranspose` layers to upsample to (64×64×3)
- Batch normalization after each transpose layer (except last)
- Activation: LeakyReLU in intermediate layers, `tanh` in final layer

Critic (Discriminator):

- Input: 64×64×3 color image
- 3 `Conv2D` layers with stride 2 to downsample
- No batch normalization (to preserve gradient flow)
- Flatten and Dense(1) output for Wasserstein score
- Activation: LeakyReLU throughout

This model was trained with gradient penalty to enforce the 1-Lipschitz constraint, alternating 5 critic updates for every generator update.

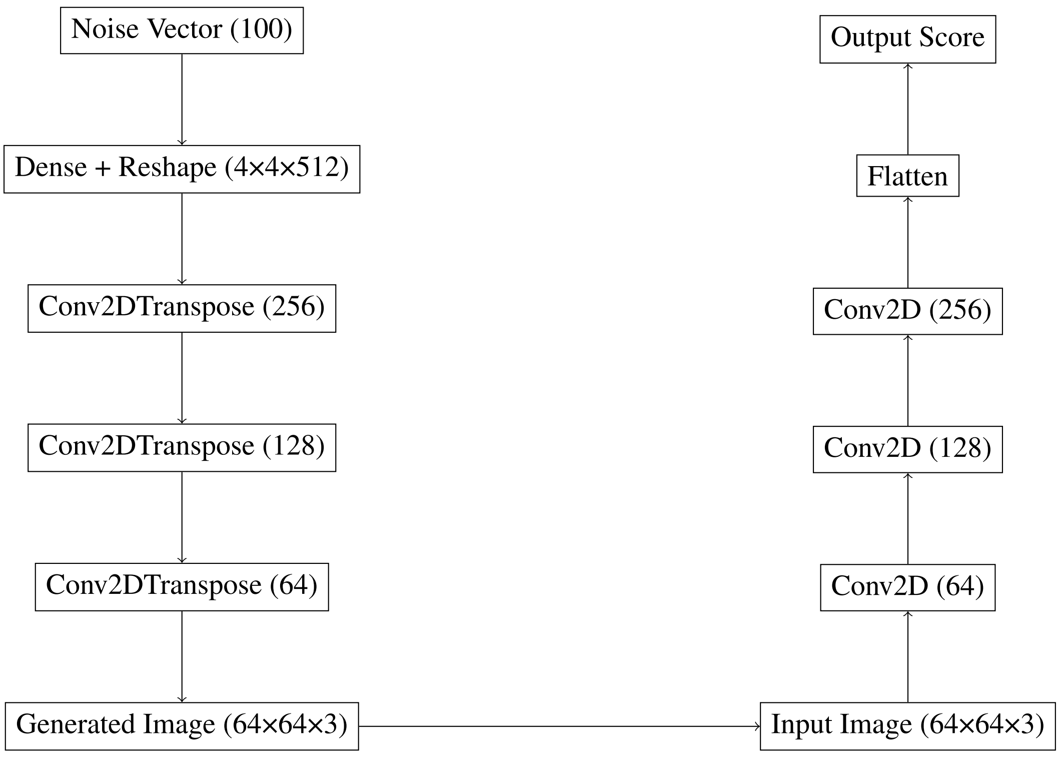


Figure 4.3: WGAN-GP Architecture: Generator and Critic for Oxford 102 Flowers (64×64 color images)

4.4 Training Details

4.4.1 Variational Autoencoder (VAE) for Tabular Data

Two VAEs were trained: one on the Iris dataset and one on a preprocessed version of the ADHD-200 phenotypic dataset. The encoder architecture consisted of two dense layers followed by outputs for the mean and log-variance vectors. A reparameterization trick using a Lambda layer was applied to sample latent vectors.

- **Latent dimensions:** 5 (Iris), 5 (ADHD)
- **Loss:** Reconstruction Loss (MSE) + KL Divergence (weighted)
- **KL Weight Annealing:** gradually increased from 0 to 0.01 over first 100 epochs
- **Scaling:** `MinMaxScaler` used to transform input to [0, 1] range

Models were saved in Keras format (`.keras`), and the fitted scalers were serialized using `joblib` to ensure consistent inverse transformation.

4.4.2 Deep Convolutional GAN (DCGAN) for MNIST

DCGAN was implemented to generate synthetic grayscale handwritten digits using the MNIST dataset.

Generator:

- Input: 100-dimensional noise vector
- Dense layer + reshape to $7 \times 7 \times 256$
- 2 Conv2DTranspose layers (stride=2) to upsample to $28 \times 28 \times 1$
- BatchNormalization and LeakyReLU activations throughout

Discriminator:

- Input: $28 \times 28 \times 1$ grayscale image
- Conv2D layers with stride=2 and LeakyReLU
- Flatten and Dense(1) for real/fake binary prediction

The generator was saved after training for 300 epochs. Synthetic digits were generated and saved in PNG format.

4.4.3 Wasserstein GAN with Gradient Penalty (WGAN-GP) for Oxford Flowers 102

WGAN-GP was used to generate 64x64 RGB images of flowers. The architecture was modified to implement the Wasserstein loss and incorporate gradient penalty for training stability.

Generator:

- Input: 128-dimensional noise vector
- Dense + Reshape to $8 \times 8 \times 256$
- Several Conv2DTranspose layers to upsample to $64 \times 64 \times 3$
- BatchNormalization + ReLU activations

Critic:

- Input: $64 \times 64 \times 3$ color image
- Multiple Conv2D layers with LeakyReLU (no BatchNorm)
- Output: 1 scalar (Wasserstein score)

The gradient penalty term was applied by interpolating real and fake samples and enforcing the Lipschitz constraint via gradient norm regularization.

4.5 Hyperparameters

Table 4.1: Hyperparameters Used Across Models

Model / Component	**Hyperparameter**	**Value**
VAE (Iris/ADHD-200)	Latent Dimension	5 (reduced from 8)
	Batch Size	32
	Learning Rate	0.001
	Epochs	300
DCGAN (MNIST)	Latent Dimension	100
	Batch Size	128
	Learning Rate	0.0002
	Epochs	250
WGAN-GP (Oxford Flowers)	Latent Dimension	128
	Batch Size	64
	Learning Rate (Generator)	0.0001
	Learning Rate (Critic)	0.0001
	Epochs	250
	Gradient Penalty Weight	10
	Critic Iterations	5

4.6 Synthetic Data Generation

Once trained, each model was used to generate synthetic data:

- **VAE:** Latent space interpolations between random pairs of encoded points; decoder used to generate new tabular samples; inverse transformation applied via saved scaler.
- **DCGAN:** 100-dimensional random noise vectors passed to generator to create 28×28 grayscale digit images.
- **WGAN-GP:** 128-dimensional noise sampled from normal distribution to generate 64×64 RGB flower images.

Outputs were saved as:

- `.csv` for tabular data (Iris, ADHD)
- `.png` for image data (MNIST, Oxford)

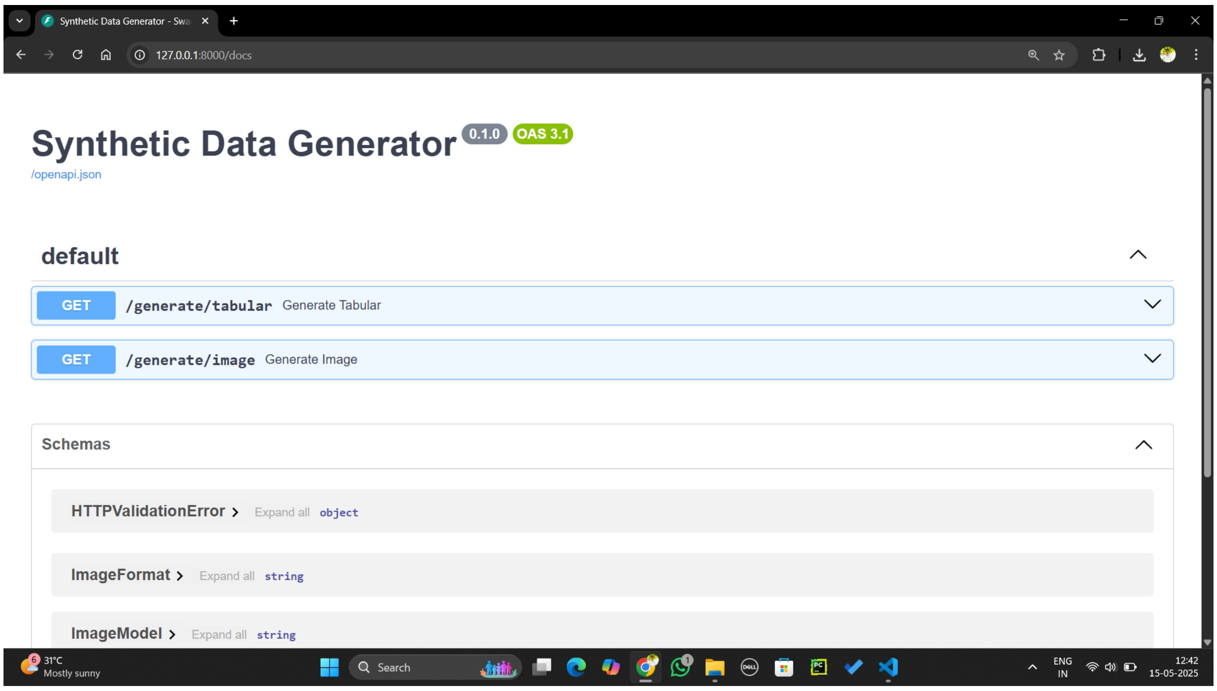
4.6.1 Frontend Interface with FastAPI

To provide a user-accessible frontend for generating synthetic data, a RESTful API was developed using the FastAPI framework. This exposes endpoints for generating tabular (Iris, ADHD) and image (MNIST, Oxford Flowers) data. FastAPI's built-in Swagger UI offers an interactive documentation and testing interface.

- `GET /generate/tabular`: Allows generation of tabular synthetic data.
- `GET /generate/image`: Allows generation of synthetic images.

This interface supports inputs like:

- `model` (e.g., "iris", "adhd", "mnist", "oxford")
- `samples` – number of data points/images to generate
- `file_format` – CSV or JSON



127.0.0.1:8000/docs

Synthetic Data Generator 0.1.0 OAS 3.1

/openapi.json

default

GET `/generate/tabular` Generate Tabular

GET `/generate/image` Generate Image

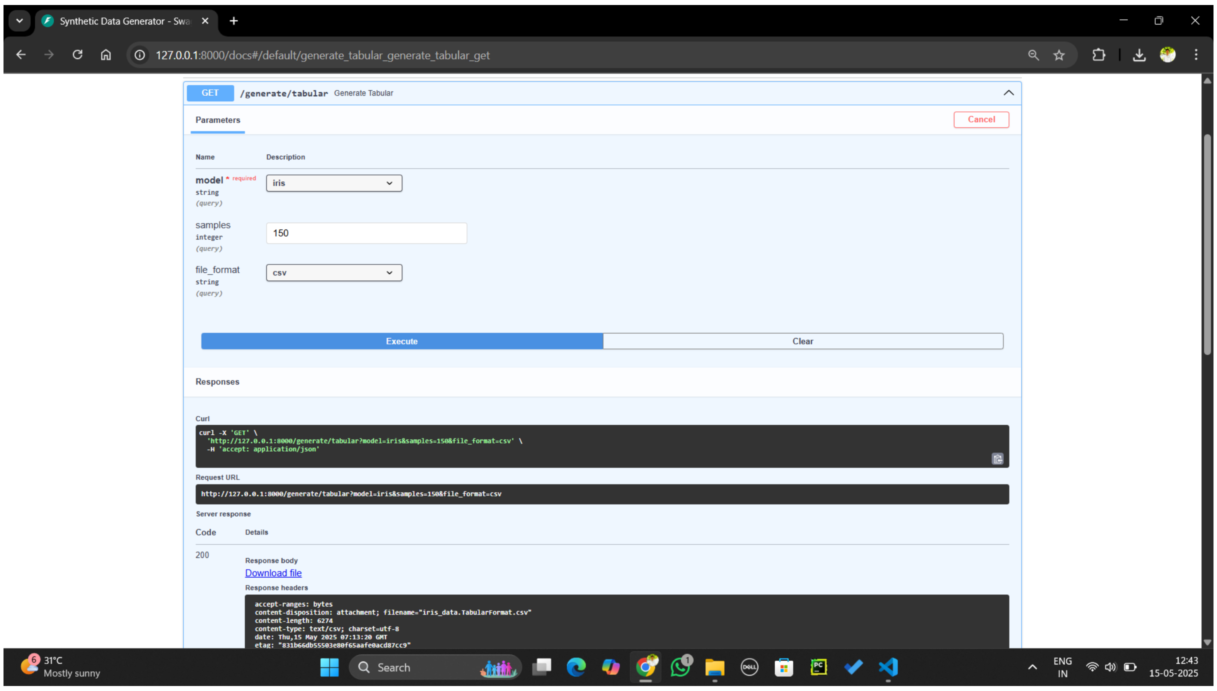
Schemas

HTTPValidationError › Expand all `object`

ImageFormat › Expand all `string`

ImageModel › Expand all `string`

Figure 4.4: UI for Synthetic Data Generator (FastAPI)



127.0.0.1:8000/docs#/default/generate_tabular_generate_tabular_get

GET `/generate/tabular` Generate Tabular

Parameters

Name	Description
model * required string (query)	iris
samples integer (query)	150
file_format string (query)	csv

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/generate/tabular?model=iris&samples=150&file_format=csv' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/generate/tabular?model=iris&samples=150&file_format=csv
```

Server response

Code: 200

Response body: Download file

Response headers

```
accept-ranges: bytes
content-disposition: attachment; filename="iris_data.TabularFormat.csv"
content-length: 6274
content-type: text/csv; charset=utf-8
date: Thu,15 May 2025 07:13:20 GMT
etag: "831b66db55503e80f65aafe0acd87cc9"
```

Figure 4.5: UI for Synthetic Data Generator (FastAPI)

Chapter 5

Results and Discussion

5.1 Evaluation

This section presents the evaluation of synthetic data quality generated by the VAE, DCGAN, and WGAN-GP models across four datasets: Iris, ADHD-200, MNIST, and Oxford 102 Flowers. Performance was primarily assessed via visual inspection, distribution alignment, and loss behavior over training.

5.1.1 Iris Dataset (Tabular)

The Iris dataset was used with a VAE model to generate synthetic numerical features. The effectiveness was validated using:

- KL Weight Schedule
- VAE Losses over 300 epochs
- KDE plots comparing real vs synthetic features

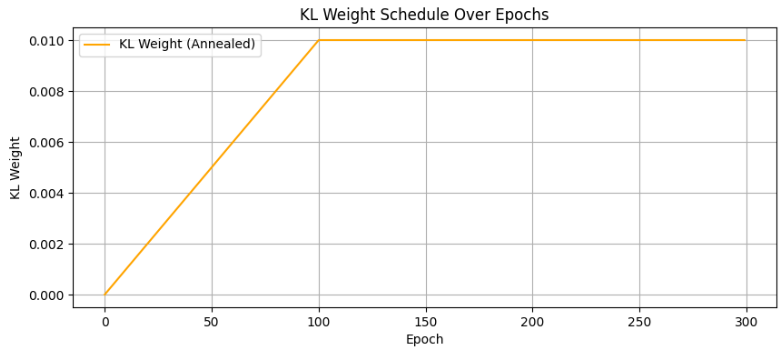


Figure 5.1: KL Weight Schedule Over Epochs (Iris VAE)

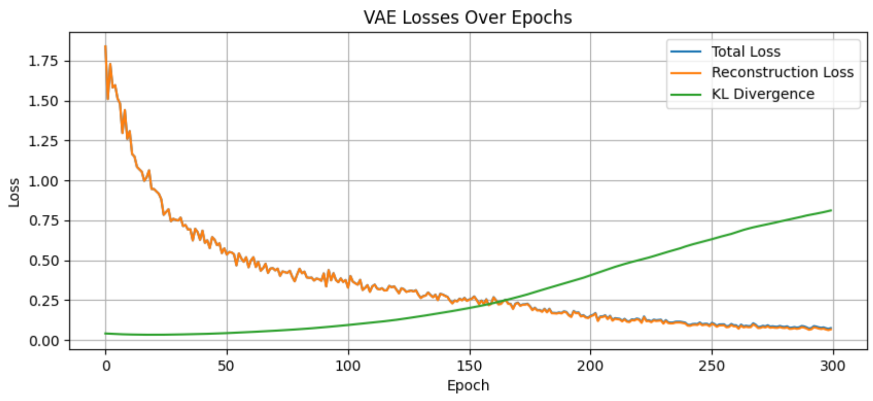


Figure 5.2: VAE Losses Over Epochs (Iris Dataset)

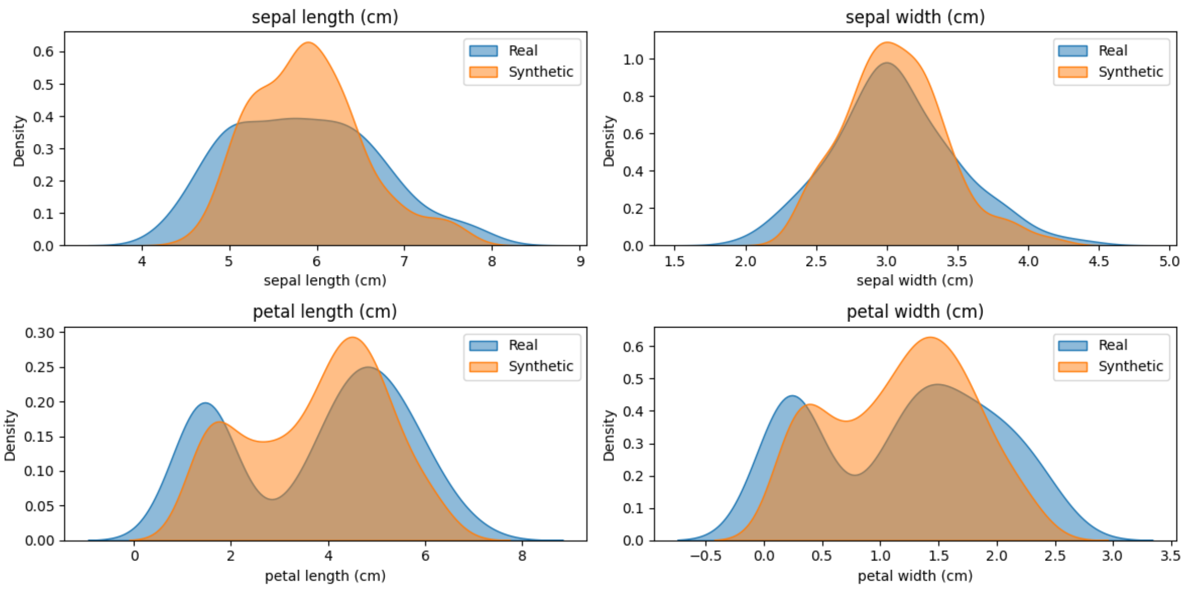


Figure 5.3: Synthetic vs Real Data Distributions (Iris Dataset)

5.1.2 ADHD Dataset (Tabular)

A more complex VAE was used to generate synthetic neuropsychological scores. Similar preprocessing and z-score filtering were applied. Results showed good overlap in KDE plots across critical attributes like age and Full IQ.

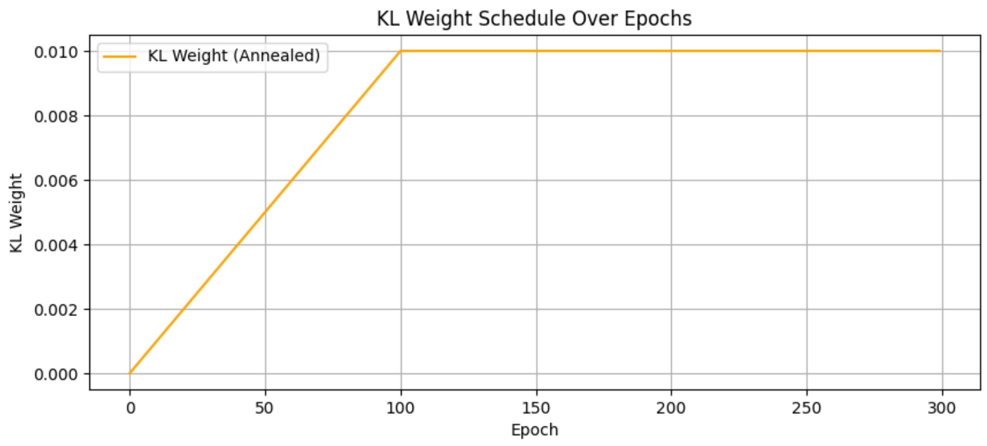


Figure 5.4: KL Weight Schedule Over Epochs (ADHD Dataset)

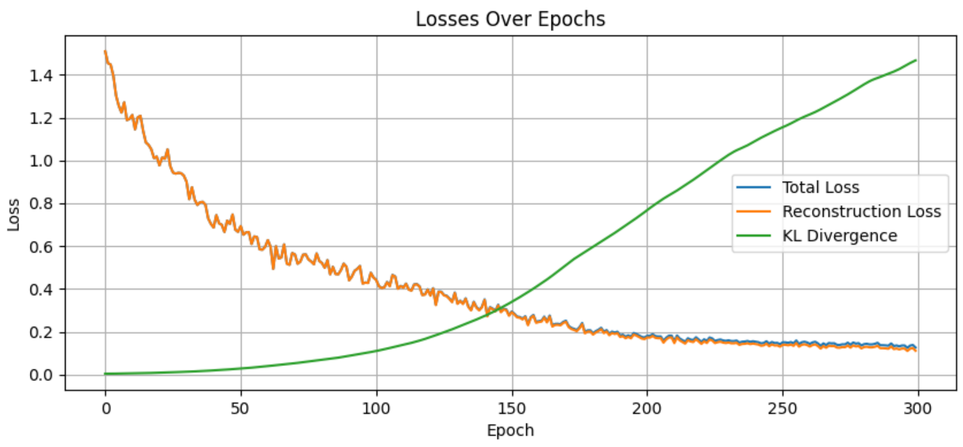


Figure 5.5: VAE Losses Over Epochs (ADHD Dataset)

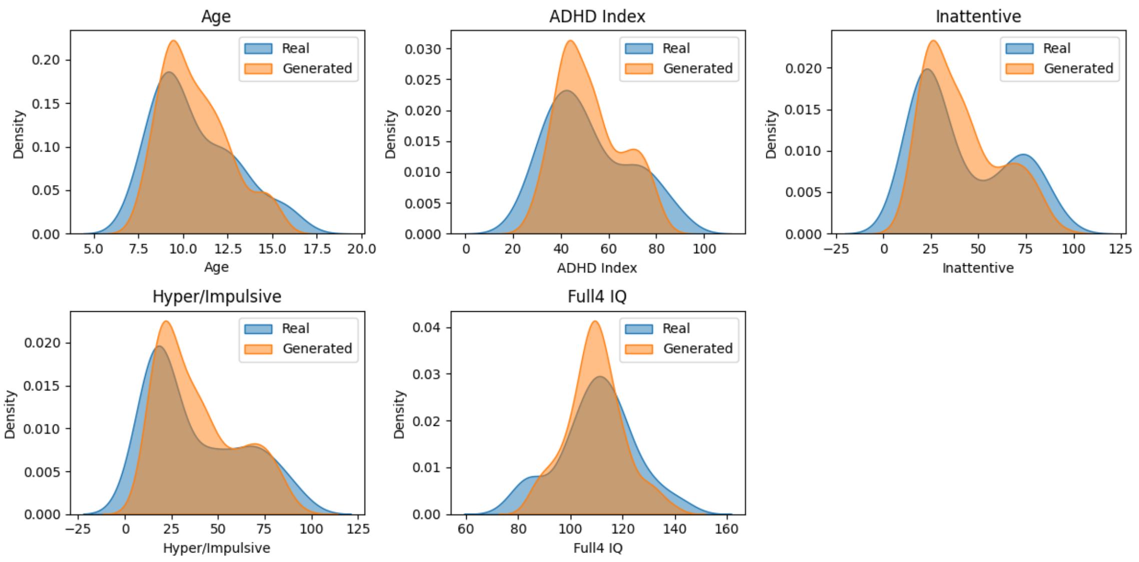


Figure 5.6: Synthetic vs Real Distribution (ADHD Dataset)

5.1.3 MNIST Dataset (Grayscale Images)

A DCGAN was trained to generate handwritten digits. Evaluation was conducted using:

- Visual inspection of image grids across epochs
- Qualitative smoothness and sharpness
- Consistency in digit formation

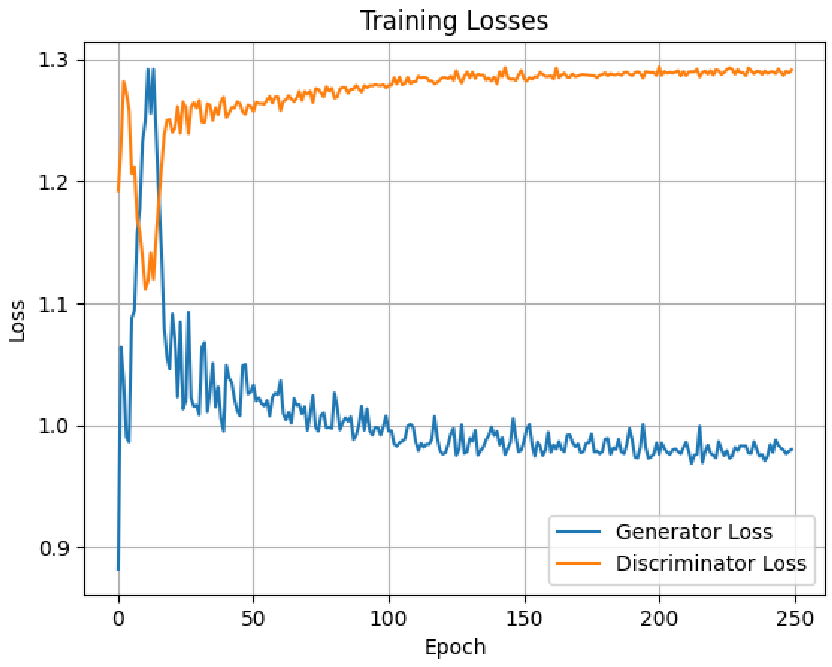


Figure 5.7: DCGAN Generator Loss vs Discriminator Loss

5.1.4 FID Score Evaluation – MNIST Dataset

The Fréchet Inception Distance (FID) was used to evaluate the quality of synthetic images generated by the DCGAN model on the MNIST dataset. The FID compares distributions of generated images and real images in the feature space of a pre-trained InceptionV3 model.

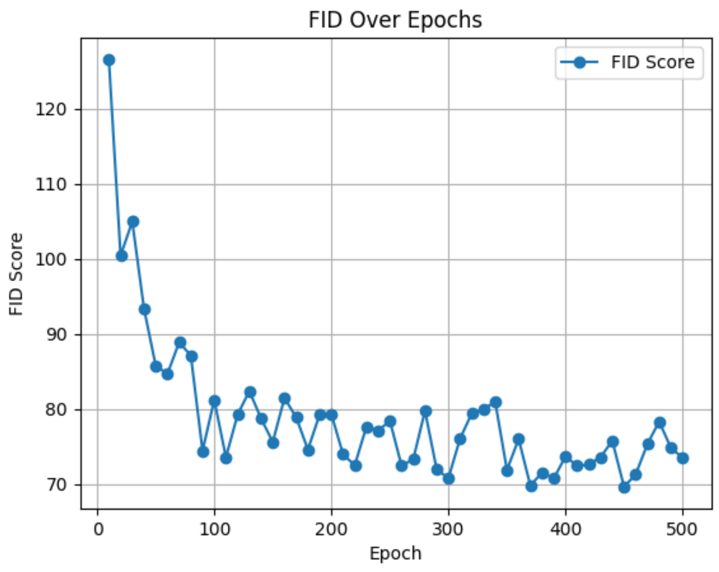


Figure 5.8: FID Score Over Epochs – DCGAN on MNIST

As shown in Figure, the training began with an FID score of approximately 127, indicating poor resemblance to real MNIST digits initially. However, the FID dropped significantly within the first 100 epochs, reaching the 75–85 range. This steady decline reflects the generator's effective learning of basic digit structures.

Between epochs 100 and 300, the FID fluctuated around 72–80, which is typical in GAN training due to the adversarial dynamics between the generator and discriminator. These oscillations signify the model refining finer visual details in its outputs. The **best recorded FID score was approximately 70**, which is considered a strong performance for a relatively simple grayscale dataset like MNIST.

In later stages (epochs 350–500), the FID curve stabilized without major spikes, indicating that training remained consistent and free from overfitting. This extended training helped further refine the generated image quality. Overall, the FID trend validates the model's ability to produce **visually and statistically close approximations** to the original MNIST data distribution.

5.1.5 Oxford 102 Flowers (Color Images)

A WGAN-GP model was used to generate realistic 64×64 color images. Evaluation metrics include:

- Epoch-wise visual comparison
- Sharpness and variety of generated images

- Stability during critic training

5.1.6 FID Score Evaluation – Oxford 102 Flowers

Fréchet Inception Distance (FID) was planned as a key metric to evaluate the image quality of samples generated by the WGAN-GP model on the Oxford 102 Flowers dataset. FID is widely used in generative modeling to assess how close the generated images are to the real data distribution by comparing feature vectors extracted using the InceptionV3 model.

However, in our case, FID calculation proved to be highly resource-intensive. The evaluation could not be completed successfully due to repeated **memory exhaustion errors** on both Kaggle (with access to 16GB RAM) and our local hardware setups. The failure was primarily due to:

- The **large size of the Oxford Flowers dataset** (8,189 RGB images).
- The need to **resize all images to 299×299** to match InceptionV3 input requirements.
- The **high VRAM usage** during batch processing of real and generated images to extract feature statistics.

This experience highlights a critical insight: **computational resources are a bottleneck in deep learning evaluation**. While our WGAN-GP model showed good qualitative performance and stable training, the inability to compute FID reliably emphasizes the importance of high-end GPUs or memory-optimized solutions in production-grade synthetic data evaluation.

We plan to reattempt FID evaluation with reduced batch sizes or by leveraging cloud infrastructure in future iterations.

5.2 Model Testing

To validate the performance and correctness of the trained generative models, a testing pipeline was executed post-training for each dataset. This involved checking data fidelity, output integrity, and ensuring that the synthetic data generation mechanism aligned with real data characteristics. Below are the dataset-specific testing procedures and outcomes.

5.2.1 Iris Dataset (Tabular)

For the Iris dataset, the VAE model was tested using:

- Sampling latent vectors via interpolation between real encoded points.
- Decoding latent points into synthetic samples using the trained decoder.
- Inverse-transforming the output using the saved `MinMaxScaler`.
- Verifying feature range consistency and distribution similarity.

The Iris dataset was relatively simple to model due to its low dimensionality and clean distribution. The VAE model was trained for 300 epochs. The training loss stabilized early, and the decoder was able to reconstruct the original distribution effectively.
During experimentation, we observed that increasing the latent dimensionality to 8 caused the KL divergence to spike, destabilizing training and reducing quality. Hence, a latent dimension of 5 was chosen for better regularization and smoother synthetic sampling.
The output was stored in `.csv` format. KDE plots validated close alignment between real and synthetic distributions across all four features. No values were found outside expected biological ranges, ensuring reliability.

5.2.2 ADHD Dataset (Tabular)

For the ADHD dataset, a more complex VAE was employed to generate neuropsychological tabular data.

- Latent vectors were sampled and interpolated from real data representations.
- The decoder generated synthetic data, which was inverse-transformed using the stored scaler.

- Statistical checks ensured no presence of outliers or invalid entries (e.g., negative IQ or ADHD index).
- Visual comparison via KDE showed consistent patterns with real data.

The ADHD dataset involved more diverse and noisy tabular features. After preprocessing and z-score filtering, the VAE model was trained over 300 epochs. Although convergence was slightly slower than Iris, the model managed to replicate key statistical patterns from the real data.
Similar to the Iris experiment, the latent dimension was kept at 5 to avoid excessive KL divergence. Attempts to increase it to 8 resulted in less stable training and poorer reconstruction quality.
The model successfully reproduced the overall structure of the dataset including age and IQ metrics, making it suitable for downstream anonymized analysis.

5.2.3 MNIST Dataset (Image)

The DCGAN model was trained on the MNIST dataset and tested for image quality and realism through the following:

- Random 100-dimensional latent vectors were passed to the trained generator.
- Output grayscale images were saved in `.png` format.
- Generated digits were checked manually for readability, consistency, and sharpness.
- The training trend of generator vs discriminator loss helped validate convergence and image stability.

For MNIST, a DCGAN was used. Since image generation is inherently more complex than tabular data synthesis, an iterative approach to training was adopted. The model was trained for a total of **500 epochs**. Throughout the training process, notable improvements in the quality of generated digits were observed.
Visual inspection confirmed that the generator progressively learned to produce sharp and coherent digits with well-defined structure. The discriminator loss stabilized after the initial epochs, indicating that both components of the adversarial network had reached a balanced training phase.

The generated samples exhibited a wide variety of digits with increasing clarity and consistency over time. This confirmed the effectiveness of the training regime and validated the suitability of DCGAN for grayscale image synthesis.

5.2.4 Oxford 102 Flowers Dataset (Color Images)

Testing for the WGAN-GP model focused on assessing the realism and diversity of generated flower images:

- Latent vectors were sampled from a standard normal distribution.
- The generator output was saved as 64×64 RGB images.
- Visual inspection across multiple epochs showed steady improvement in color richness and petal structure.
- Critic loss remained stable throughout training, and mode collapse was not observed.

Training the WGAN-GP on Oxford 102 Flowers was computationally intensive. Due to the high-resolution color images and deeper generator/critic networks, training required around 16 hours to complete 200 epochs on GPU hardware. A follow-up session extended the training to 250 epochs, with modest yet visible improvements in image sharpness and diversity.
The gradient penalty and critic steps ensured training stability. Although FID was not computed, it is planned for future work to quantitatively evaluate visual fidelity.
The model demonstrated the ability to synthesize visually convincing floral patterns with noticeable variety, validating both training procedure and architectural stability.

5.3 FastAPI Interface for Testing

To allow external users to test the models interactively, a FastAPI backend was developed. The API supports two endpoints:

- `GET /generate/tabular` – Generates synthetic tabular data for a given model (e.g., iris or adhd) and number of samples.
- `GET /generate/image` – Triggers image generation using trained GAN models (e.g., DCGAN, WGAN-GP).

5.4 Output Validation

For every test:

- Generated data is saved in `.csv` or `.png` formats.
- Visual inspection and distribution comparison ensure quality.
- Swagger interface confirms functionality of backend endpoints and synthetic data retrieval.

All tests confirmed that the deployed endpoints respond correctly, generate expected data formats, and are suitable for integration in downstream pipelines or frontend applications.

5.5 Challenges Faced

Throughout the development and training of the synthetic data generation models, several technical and practical challenges were encountered:

- **Data Cleaning and Preprocessing:** The ADHD-200 dataset contained inconsistent and missing values (e.g., `-999`, `pending`, or empty cells). Cleaning required careful replacement, removal of NaNs, and filtering of statistical outliers using z-score logic.
- **Latent Space Instability:** Choosing an appropriate latent dimensionality was non-trivial. Initial experiments with 8-dimensional latent space led to high KL divergence and unstable reconstructions. Reducing to 5 dimensions provided more regularized latent encodings and stable training.
- **VAE KL Divergence Explosion:** Early training phases of VAEs showed KL divergence overpowering the reconstruction loss. This was mitigated by introducing KL annealing—gradually increasing the KL term's weight during the first 100 epochs.
- **Compute and Memory Limitations:** Increasing the batch size to accelerate training caused `ResourceExhaustedError` on limited-memory GPUs. Similarly, the use of InceptionV3 embeddings for FID computation proved to be extremely memory-hungry and infeasible on free-tier platforms.
- **FID Evaluation Setup:** Although Frechet Inception Distance (FID) is a standard metric for GAN evaluation, its setup requires additional pipelines, embeddings, and data batching. Due to computational overhead, FID will be incorporated and evaluated in the final report phase.

Chapter 6

Conclusion

The process of synthetic data generation plays a pivotal role in modern AI development, enabling scalable data augmentation, privacy preservation, and robust model training. This report presented a comprehensive pipeline using both Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) including DCGAN and WGAN-GP, applied across diverse datasets—tabular (Iris and ADHD-200) and visual (MNIST and Oxford 102 Flowers).

The project involved detailed preprocessing, model implementation, hyperparameter tuning, and qualitative evaluation of synthetic outputs. VAE-based approaches proved effective for structured tabular data, while DCGAN and WGAN-GP offered high-quality image generation, with WGAN-GP yielding sharper, more stable outputs.

Key challenges included managing memory-intensive training, selecting optimal latent dimensions, and model instability. Despite these, the pipeline successfully produced synthetic data closely resembling the original distributions.

This synthetic data framework contributes toward solving problems of limited real-world data, promoting privacy-aware learning, and enhancing AI model generalization—forming a foundational asset for future machine learning applications.

Future Scope

As the demand for data privacy, fairness, and scalability grows in the field of Artificial Intelligence, synthetic data is poised to play a vital role in real-world deployments. While the models implemented in this project demonstrate the ability to generate high-quality synthetic data, there remains significant scope for future exploration and improvement.

One promising direction is the enhancement of generative models with larger and deeper architectures such as StyleGAN or Diffusion Models, which have been shown to produce state-of-the-art results in image synthesis. These could be explored especially for complex datasets like Oxford Flowers, where finer details and texture quality are critical.

Additionally, performance metrics can be improved by integrating more robust and real-time evaluation techniques. For example, deploying Fréchet Inception Distance (FID) tracking during training in a resource-efficient manner, or incorporating Precision-Recall for Distributions

(PRD) can provide a more nuanced understanding of model quality.

On the tabular data side, there is a need to move beyond purely numerical data and explore categorical or mixed-type data generation using models such as CTGAN or TVAE. Integrating such models can broaden the application of synthetic data generation to domains like healthcare, banking, and census data.

Another area of expansion lies in real-time deployment and synthetic dataset APIs. The integration of our models into a web-based interface (e.g., using FastAPI or Flask) could allow researchers and developers to generate synthetic samples on-demand for training, testing, or educational purposes.

Finally, future work can explore the ethical and regulatory aspects of synthetic data. By embedding privacy-preserving mechanisms such as differential privacy or membership inference resistance, synthetic data can be made not just realistic but also safe and compliant with data protection regulations like GDPR.

With ongoing research and rapidly advancing hardware, synthetic data generation will continue to evolve and become a cornerstone of responsible and scalable AI development.

Bibliography

[1] Alkahtani, H., et al. *Survey on Synthetic Data Generation, Evaluation Methods and GANs*, 2020.

[2] Zhao, Q., & Saligrama, V. *Synthetic Data: What, Why and How*, 2023.

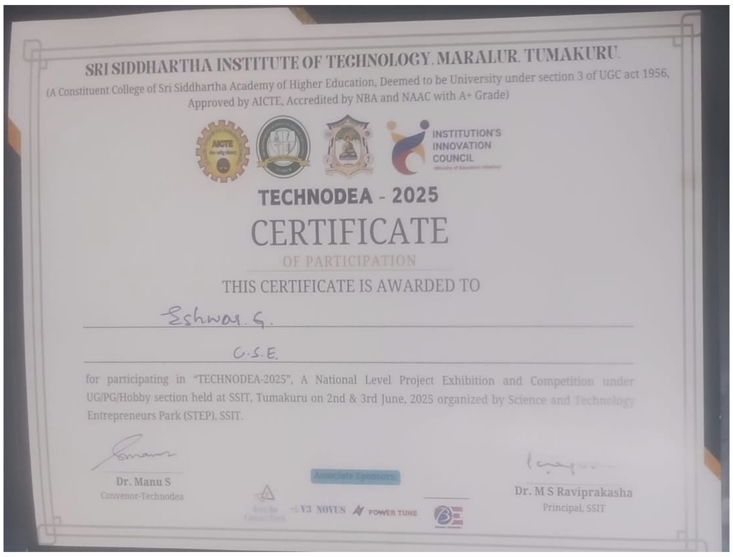
[3] Chen, J., & Huang, Z. *Machine Learning for Synthetic Data Generation: A Review*, 2022.

[4] Cai, Q., et al. *Comprehensive Exploration of Synthetic Data Generation*, 2021.

[5] Frid-Adar, M., et al. *Generation of Synthetic Data with Generative Adversarial Networks*, 2018.

[6] Arjovsky, M., Chintala, S., & Bottou, L. *Wasserstein GAN*, 2017.

Technodea



SRI SIDDHARTHA INSTITUTE OF TECHNOLOGY, MARALUR, TUMAKURU.
(A Constituent College of Sri Siddhartha Academy of Higher Education, Deemed to be University under section 3 of UGC act 1956, Approved by AICTE, Accredited by NBA and NAAC with A+ Grade)

INSTITUTION'S INNOVATION COUNCIL

TECHNODEA - 2025

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS AWARDED TO

Eshwar. G.

C.S.E.

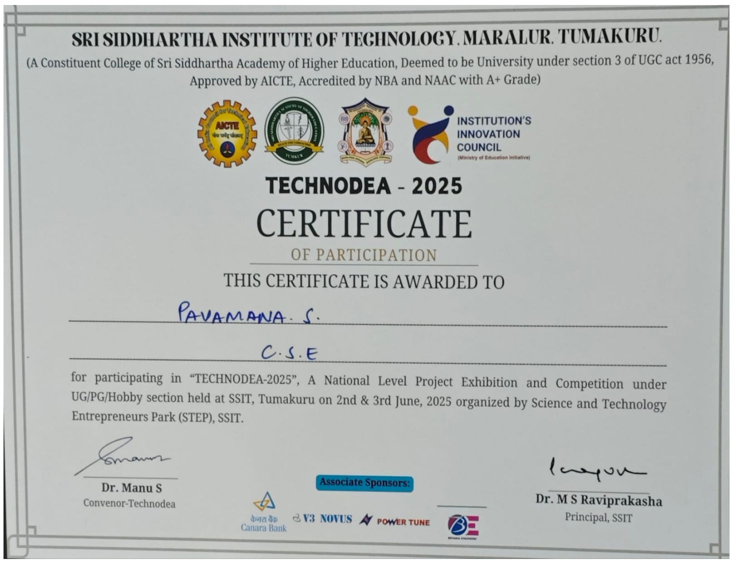
for participating in "TECHNODEA-2025", A National Level Project Exhibition and Competition under UG/PG/Hobby section held at SSIT, Tumakuru on 2nd & 3rd June, 2025 organized by Science and Technology Entrepreneurs Park (STEP), SSIT.

[illegible]
Dr. Manu S
Convenor-Technodea

Associate Sponsors:
Canara Bank V3 NOVUS POWER TUNE

[illegible]
Dr. M S Raviprakasha
Principal, SSIT

Figure 6.1: Eshwar G



SRI SIDDHARTHA INSTITUTE OF TECHNOLOGY, MARALUR, TUMAKURU.
(A Constituent College of Sri Siddhartha Academy of Higher Education, Deemed to be University under section 3 of UGC act 1956, Approved by AICTE, Accredited by NBA and NAAC with A+ Grade)

INSTITUTION'S INNOVATION COUNCIL

TECHNODEA - 2025

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS AWARDED TO

PAVAMANA. S.

C.S.E

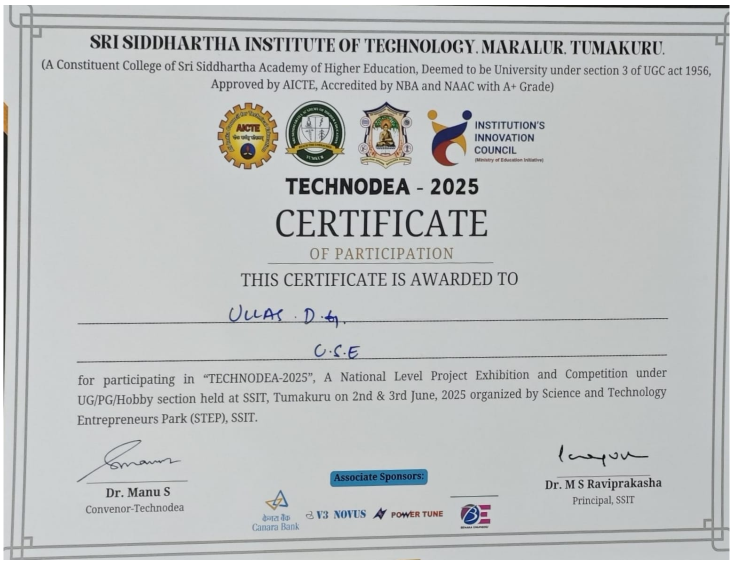
for participating in "TECHNODEA-2025", A National Level Project Exhibition and Competition under UG/PG/Hobby section held at SSIT, Tumakuru on 2nd & 3rd June, 2025 organized by Science and Technology Entrepreneurs Park (STEP), SSIT.

[illegible]
Dr. Manu S
Convenor-Technodea

Associate Sponsors:
Canara Bank V3 NOVUS POWER TUNE

[illegible]
Dr. M S Raviprakasha
Principal, SSIT

Figure 6.2: Pavamana S



SRI SIDDHARTHA INSTITUTE OF TECHNOLOGY, MARALUR, TUMAKURU.
(A Constituent College of Sri Siddhartha Academy of Higher Education, Deemed to be University under section 3 of UGC act 1956, Approved by AICTE, Accredited by NBA and NAAC with A+ Grade)

INSTITUTION'S INNOVATION COUNCIL

TECHNODEA - 2025

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS AWARDED TO

ULLAS . D.G.

C.S.E

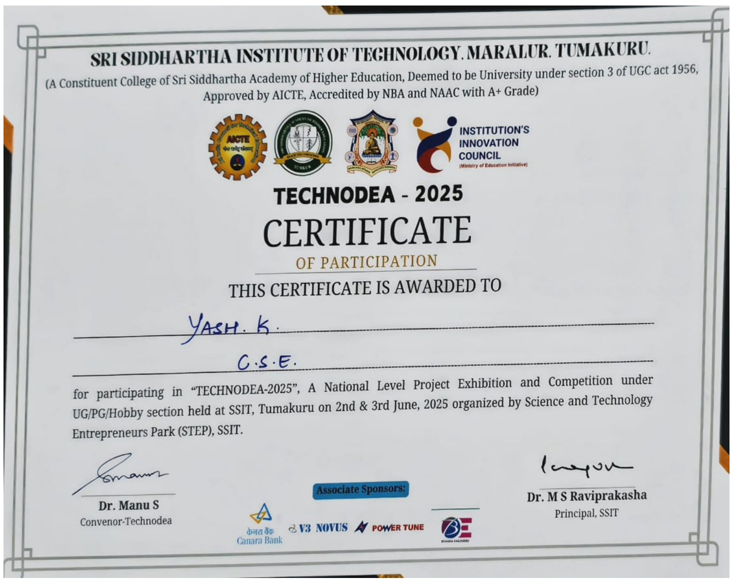
for participating in "TECHNODEA-2025", A National Level Project Exhibition and Competition under UG/PG/Hobby section held at SSIT, Tumakuru on 2nd & 3rd June, 2025 organized by Science and Technology Entrepreneurs Park (STEP), SSIT.

[illegible]
Dr. Manu S
Convenor-Technodea

Associate Sponsors:
Canara Bank V3 NOVUS POWER TUNE

[illegible]
Dr. M S Raviprakasha
Principal, SSIT

Figure 6.3: Ullas DG



SRI SIDDHARTHA INSTITUTE OF TECHNOLOGY, MARALUR, TUMAKURU.
(A Constituent College of Sri Siddhartha Academy of Higher Education, Deemed to be University under section 3 of UGC act 1956, Approved by AICTE, Accredited by NBA and NAAC with A+ Grade)

INSTITUTION'S INNOVATION COUNCIL

TECHNODEA - 2025

CERTIFICATE

OF PARTICIPATION

THIS CERTIFICATE IS AWARDED TO

YASH. K.

C.S.E.

for participating in "TECHNODEA-2025", A National Level Project Exhibition and Competition under UG/PG/Hobby section held at SSIT, Tumakuru on 2nd & 3rd June, 2025 organized by Science and Technology Entrepreneurs Park (STEP), SSIT.

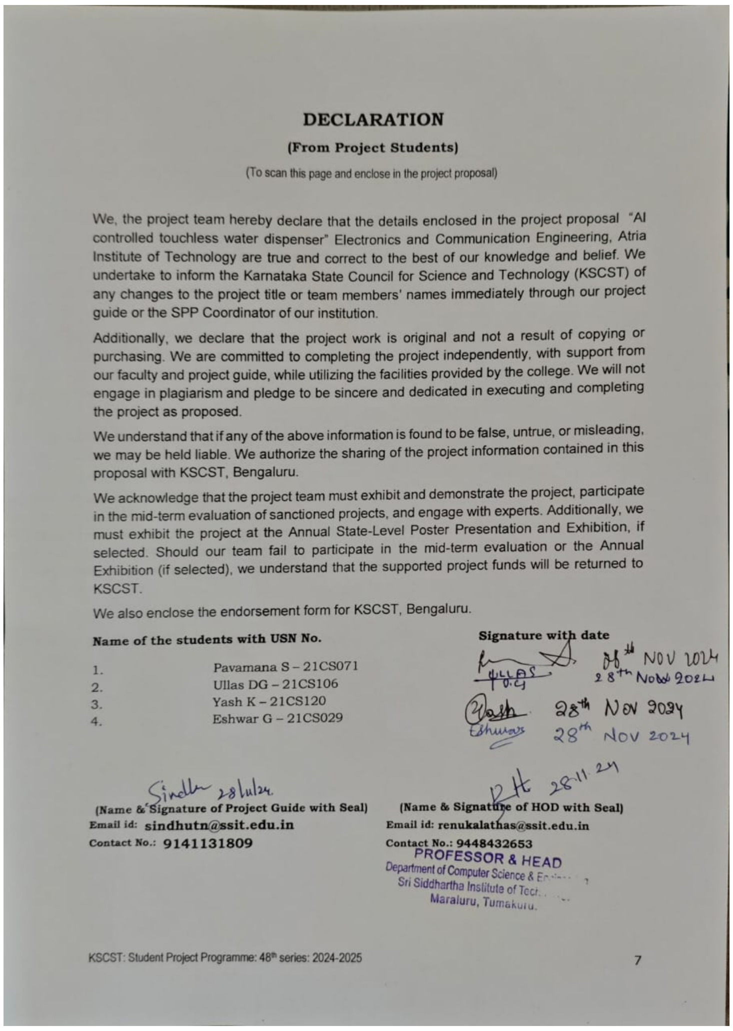
[illegible]
Dr. Manu S
Convenor-Technodea

Associate Sponsors:
Canara Bank V3 NOVUS POWER TUNE

[illegible]
Dr. M S Raviprakasha
Principal, SSIT

Figure 6.4: Yash K

KSCST



DECLARATION

(From Project Students)

(To scan this page and enclose in the project proposal)

We, the project team hereby declare that the details enclosed in the project proposal "AI controlled touchless water dispenser" Electronics and Communication Engineering, Atria Institute of Technology are true and correct to the best of our knowledge and belief. We undertake to inform the Karnataka State Council for Science and Technology (KSCST) of any changes to the project title or team members' names immediately through our project guide or the SPP Coordinator of our institution.

Additionally, we declare that the project work is original and not a result of copying or purchasing. We are committed to completing the project independently, with support from our faculty and project guide, while utilizing the facilities provided by the college. We will not engage in plagiarism and pledge to be sincere and dedicated in executing and completing the project as proposed.

We understand that if any of the above information is found to be false, untrue, or misleading, we may be held liable. We authorize the sharing of the project information contained in this proposal with KSCST, Bengaluru.

We acknowledge that the project team must exhibit and demonstrate the project, participate in the mid-term evaluation of sanctioned projects, and engage with experts. Additionally, we must exhibit the project at the Annual State-Level Poster Presentation and Exhibition, if selected. Should our team fail to participate in the mid-term evaluation or the Annual Exhibition (if selected), we understand that the supported project funds will be returned to KSCST.

We also enclose the endorsement form for KSCST, Bengaluru.

	Name of the students with USN No.	**Signature with date**
1.	Pavamana S – 21CS071	[illegible] 28th NOV 2024
2.	Ullas DG – 21CS106	[illegible] 28th Nov 2024
3.	Yash K – 21CS120	Yash 28th Nov 2024
4.	Eshwar G – 21CS029	Eshwar 28th NOV 2024

Sindhu 28/11/24

(Name & Signature of Project Guide with Seal)
Email id: sindhutn@ssit.edu.in
Contact No.: 9141131809

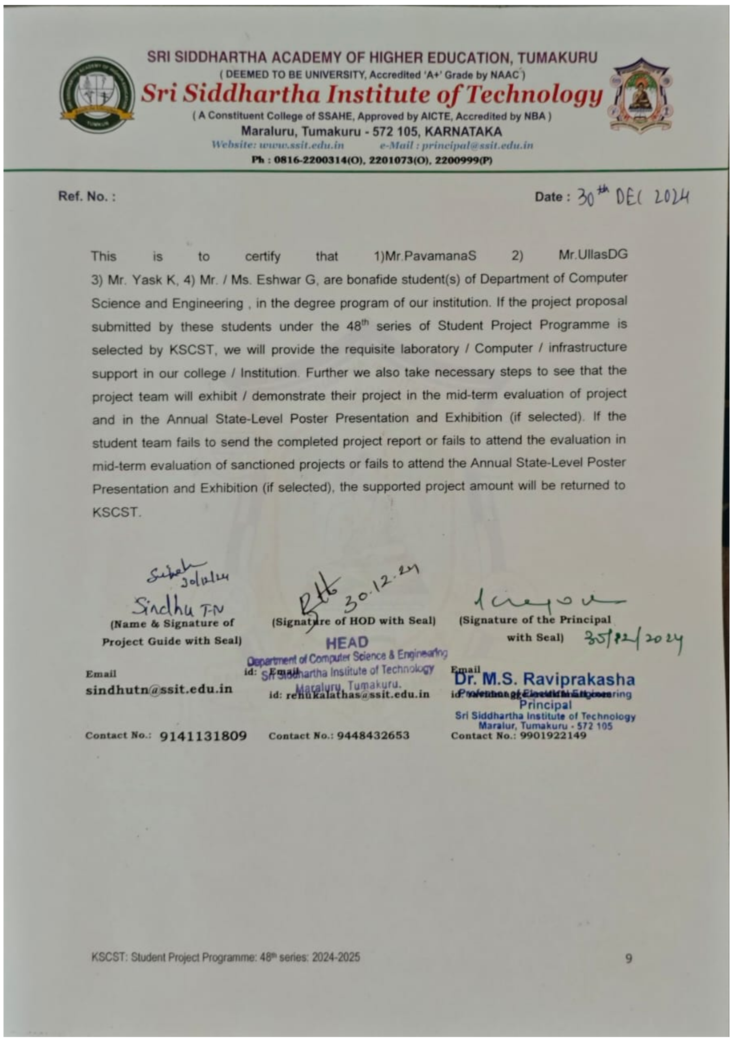
[illegible] 28.11.24

(Name & Signature of HOD with Seal)
Email id: renukalathas@ssit.edu.in
Contact No.: 9448432653

PROFESSOR & HEAD
Department of Computer Science & En
Sri Siddhartha Institute of Tec
Maraluru, Tumakuru.

KSCST: Student Project Programme: 48th series: 2024-2025 7

Figure 6.5: KSCST Declaration



SRI SIDDHARTHA ACADEMY OF HIGHER EDUCATION, TUMAKURU
(DEEMED TO BE UNIVERSITY, Accredited 'A+' Grade by NAAC)
Sri Siddhartha Institute of Technology
(A Constituent College of SSAHE, Approved by AICTE, Accredited by NBA)
Maraluru, Tumakuru - 572 105, KARNATAKA
Website: www.ssit.edu.in   e-Mail : principal@ssit.edu.in
Ph : 0816-2200314(O), 2201073(O), 2200999(P)

Ref. No. :   Date : 30th DEC 2024

This is to certify that 1)Mr.PavamanaS 2) Mr.UllasDG 3) Mr. Yask K, 4) Mr. / Ms. Eshwar G, are bonafide student(s) of Department of Computer Science and Engineering , in the degree program of our institution. If the project proposal submitted by these students under the 48th series of Student Project Programme is selected by KSCST, we will provide the requisite laboratory / Computer / infrastructure support in our college / Institution. Further we also take necessary steps to see that the project team will exhibit / demonstrate their project in the mid-term evaluation of project and in the Annual State-Level Poster Presentation and Exhibition (if selected). If the student team fails to send the completed project report or fails to attend the evaluation in mid-term evaluation of sanctioned projects or fails to attend the Annual State-Level Poster Presentation and Exhibition (if selected), the supported project amount will be returned to KSCST.

[illegible] 30/12/24 Sindhu T-N	[illegible] 30.12.24	[illegible] 30/12/2024
(Name & Signature of Project Guide with Seal)	(Signature of HOD with Seal) HEAD Department of Computer Science & Engineering Sri Siddhartha Institute of Technology Maraluru, Tumakuru.	(Signature of the Principal with Seal) Dr. M.S. Raviprakasha Professor of Electrical Engineering Principal Sri Siddhartha Institute of Technology Maralur, Tumakuru - 572 105
Email sindhutn@ssit.edu.in	Email id: renukalathas@ssit.edu.in	Email id:
Contact No.: 9141131809	Contact No.: 9448432653	Contact No.: 9901922149

KSCST: Student Project Programme: 48th series: 2024-2025   9

Figure 6.6: KSCST Endorsment