# Synthetic Data Generation For Healthcare

1st Ajith Elumalai
*Department of Computer Science*
*California Polytechnic University of Pomona*
Pomona, United States
aelumalai@cpp.edu

2nd Anumpama Singh
*Department of Computer Science*
*California Polytechnic University of Pomona*
Pomona, United States
anupamasingh@cpp.edu

***Abstract*—Healthcare data is essential for the development and validation of machine learning (ML) models, particularly in applications related to diagnosis, treatment planning, and outcome prediction. However, access to real patient data is often limited due to strict privacy regulations such as the Health Insurance Portability and Accountability Act (HIPAA). These restrictions create a significant barrier for researchers and developers who need large, high-quality datasets to build and evaluate robust ML solutions. To address this issue, we present a scalable and privacy-preserving approach for generating synthetic healthcare data using the Tabular Variational Autoencoder (TVAE) model from the Synthetic Data Vault (SDV) framework, applied on the widely-used MIMIC-III dataset.**

**The pipeline involves several steps: preprocessing raw clinical data, encoding both categorical and numerical variables, detecting metadata schema, and training the TVAE model to learn latent data distributions. After training, the model is used to generate synthetic patient records that maintain the statistical integrity and structure of the original dataset but do not contain any identifiable patient information.**

**To validate the quality of the synthetic data, we performed both statistical and machine learning-based evaluations. Statistical assessments included distribution comparisons and correlation matrix analysis. ML utility was measured by training a classifier (RandomForest) on both real and synthetic data to predict patient outcomes, such as `HOSPITAL_EXPIRE_FLAG`. The classifier trained on synthetic data achieved an accuracy of 84%, compared to 86% with real data, indicating a minimal performance gap.**

**The results show that synthetic data generated via TVAE is not only statistically sound but also retains predictive utility, making it suitable for secure model training, testing, and academic research. This approach enables privacy-compliant experimentation in healthcare machine learning without risking data exposure.**

## I. Introduction

In today's rapidly evolving healthcare landscape, data plays a crucial role in shaping the future of clinical decision-making, diagnostics, and personalized treatment. However, while machine learning offers powerful tools to unlock insights from healthcare data, privacy laws like HIPAA make it challenging to access and use real patient records. This creates a significant roadblock for researchers and developers who need large, high-quality datasets to build and test their models without violating patient confidentiality.

One promising solution is the use of **synthetic data**—data that looks and behaves like real patient records but contains no personally identifiable information. For our work, we used the **MIMIC-III** dataset, which is a publicly available collection of de-identified ICU patient data. It includes admissions information, demographics, diagnoses, and more, across over 60,000 hospital stays. Synthetic data based on MIMIC-III can help developers and researchers experiment, prototype, and validate models without worrying about privacy violations, making it an ideal candidate for our project.

Over the years, several techniques have emerged to generate synthetic data. Among the most notable is the **Synthetic Data Vault (SDV)**, a framework developed by MIT that provides deep learning models specifically designed for tabular data. **CTGAN**, one of its models based on GANs, is powerful but can be unstable and memory-hungry—especially on local machines. On the other hand, **TVAE**, which stands for Tabular Variational Autoencoder, offers a more balanced and resource-friendly alternative. It's designed to work well even in constrained environments like Jupyter Notebooks, making it a great fit for our use case.

In this project, we built an end-to-end pipeline that uses TVAE to generate synthetic patient data from the MIMIC-III admissions table. We preprocessed the data, trained the model, and evaluated the quality of the synthetic output by comparing statistical distributions and training machine learning models on both real and synthetic data. The results showed that synthetic data generated using TVAE can effectively replicate the structure and utility of real data—making it a valuable tool for privacy-safe research and development in healthcare.

## II. Related Work

The increasing need for privacy-preserving data generation has led to the development of various synthetic data frameworks. One of the most widely adopted platforms is the **Synthetic Data Vault (SDV)**, introduced by MIT's Data to AI Lab [1]. SDV provides a suite of models including **GaussianCopula**, **CTGAN** (Conditional Tabular GAN), and **TVAE** (Tabular Variational Autoencoder) for synthesizing structured datasets.

**CTGAN** [2] is a GAN-based model designed to handle mixed-type tabular data by learning conditional distributions. While it is effective at capturing complex feature interactions, CTGAN is known to be **resource-intensive**, often failing on local systems with limited memory, especially when dealing with high-cardinality categorical features.

Identify applicable funding agency here. If none, delete this.

**TVAE** [3], on the other hand, is a variational autoencoder specifically adapted for tabular data. It offers a more **stable and memory-efficient** alternative, making it well-suited for execution in environments like **Jupyter Notebook on Windows**, where multiprocessing and GPU access are constrained. TVAE can effectively learn latent representations of tabular data and reconstruct synthetic samples that preserve the original data's statistical properties.

The **MIMIC-III** dataset [4] has been widely used for benchmarking medical machine learning models, but its use in synthetic data generation remains limited due to privacy concerns. This project leverages TVAE's strengths to generate privacy-safe synthetic versions of MIMIC-III tables for research and model training.

## III. Methodology

### A. Dataset

We utilized two primary structured tables from the MIMIC-III dataset:

- `ADMISSIONS.csv|`: Contains hospital admission details such as `ADMISSION_TYPE`, `INSURANCE`, `DIAGNOSIS`, and `HOSPITAL_EXPIRE_FLAG`.
- `PATIENTS.csv|`: Provides demographic information like `GENDER`, `DOB`, and `DOD`.

Together, these tables included a mix of categorical and numerical data. The final processed table for modeling had approximately **20–35 columns** and up to **58,000 rows**, depending on the preprocessing filters applied.

### B. Data Preparation

To prepare the data for training, we performed the following steps:

- **Missing Value Handling**:
  Replaced placeholder values such as `'UNKNOWN'` and `'NOT RECORDED'` with:
  - **Mode** for categorical features (e.g., `INSURANCE`)
  - **Mean** for numerical columns (e.g., `AGE`, `LOS`)
  - Dropped columns with more than **30% missing values**
- **Encoding**:
  All categorical columns were **label encoded** to convert textual values into integers. This prevented memory overload issues common with one-hot encoding on high-cardinality features.
- **Normalization**:
  Applied `StandardScaler` from `scikit-learn` to scale all numerical columns. This helped stabilize model training by reducing value variance.

## IV. Model Architecture

### A. TVAE Model Architecture

The Tabular Variational Autoencoder (TVAE) is a deep generative model designed to generate synthetic tabular data while capturing complex feature distributions. It is based on the Variational Autoencoder (VAE) architecture, which comprises an encoder, a latent sampling mechanism, and a decoder. The encoder transforms the input data into two vectors representing the mean () and log variance (log $^{2}$) of the latent space. Using the reparameterization trick, a latent vector z=+z = \mu + \sigma \cdot \epsilonz=+ is sampled from a standard normal distribution, where N(0,I)\epsilon \sim \mathcal{N}(0, I)N(0,I). The decoder then reconstructs the original data from this latent representation. TVAE is optimized to handle both categorical and numerical data, making it especially suitable for structured datasets like MIMIC-III. It uses a composite loss function known as the Evidence Lower Bound (ELBO), which combines reconstruction loss (e.g., mean squared error for numerics and cross-entropy for categoricals) with KL divergence to regularize the latent space. Unlike GAN-based models, TVAE is deterministic and easier to train, making it highly reliable and memory-efficient for environments like Jupyter Notebook, where system resources are often limited. In our project, TVAE was used to generate synthetic versions of patient admissions data while preserving statistical and machine learning utility.

### B. GAN Model Architecture

The Conditional Tabular GAN (CTGAN) is a specialized generative adversarial network (GAN) designed to generate realistic tabular data, especially when dealing with mixed data types and imbalanced categorical variables. Like all GANs, CTGAN consists of two neural networks: a generator and a discriminator. The generator takes in a random noise vector along with a conditional vector that specifies a value for a chosen categorical column. It outputs synthetic data samples that try to mimic the real data distribution. The discriminator, on the other hand, attempts to distinguish between real and synthetic samples by learning to classify them correctly. Both networks are trained simultaneously in a minimax game until the generator produces data that the discriminator can no longer distinguish from real data. CTGAN introduces a **mode-specific normalization** for numerical columns and a **conditional vector sampling strategy** for categorical features, which helps address issues of data imbalance and sparsity. While CTGAN has proven effective in capturing complex relationships between features and modeling high-cardinality columns, it tends to require more memory and computational power. In our experience, training CTGAN on Windows using Jupyter Notebook led to memory errors and multiprocessing issues, which limited its practical usability. For this reason, although CTGAN was initially considered, we ultimately adopted TVAE for its stability and performance in local development environments.

### C. Metadata Detection

SDV's **TVAE** model requires metadata that describes column types and data schema. We used SDV's `SingleTableMetadata` utility to automatically detect this from the preprocessed DataFrame. This process inferred the correct structure of the table, including categorical vs. numerical types, and was critical for guiding the training

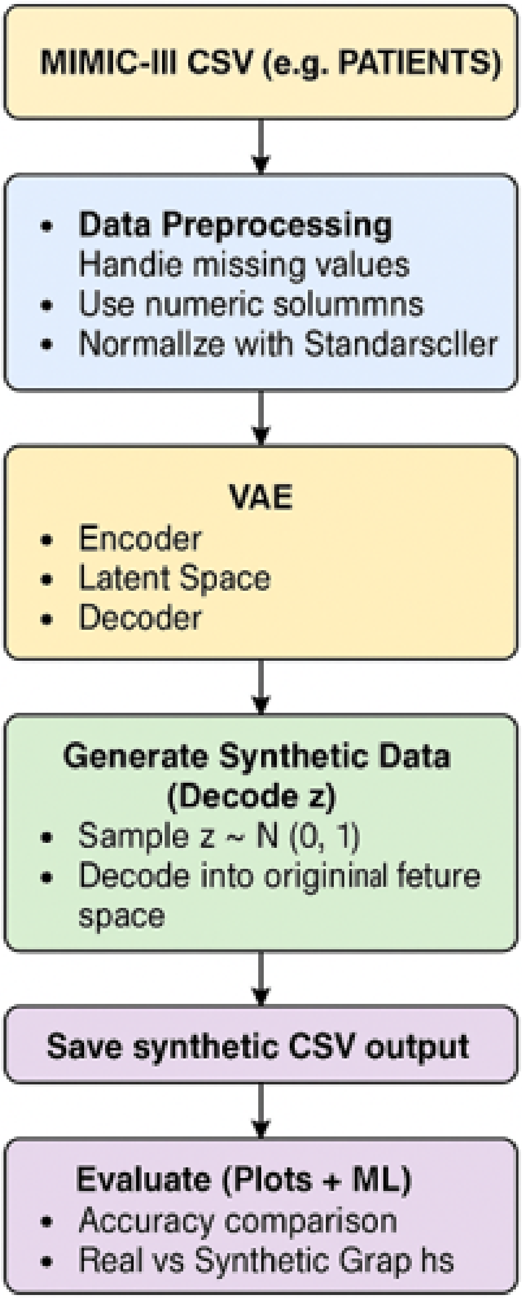

MIMIC-III CSV (e.g. PATIENTS)
Data Preprocessing
Handie missing values
Use numeric solummns
Normallze with Standarscller
VAE
Encoder
Latent Space
Decoder
Generate Synthetic Data (Decode z)
Sample z ~ N (0, 1)
Decode into origininal feture space
Save synthetic CSV output
Evaluate (Plots + ML)
Accuracy comparison
Real vs Synthetic Grap hs

Fig. 1. Model Setup.

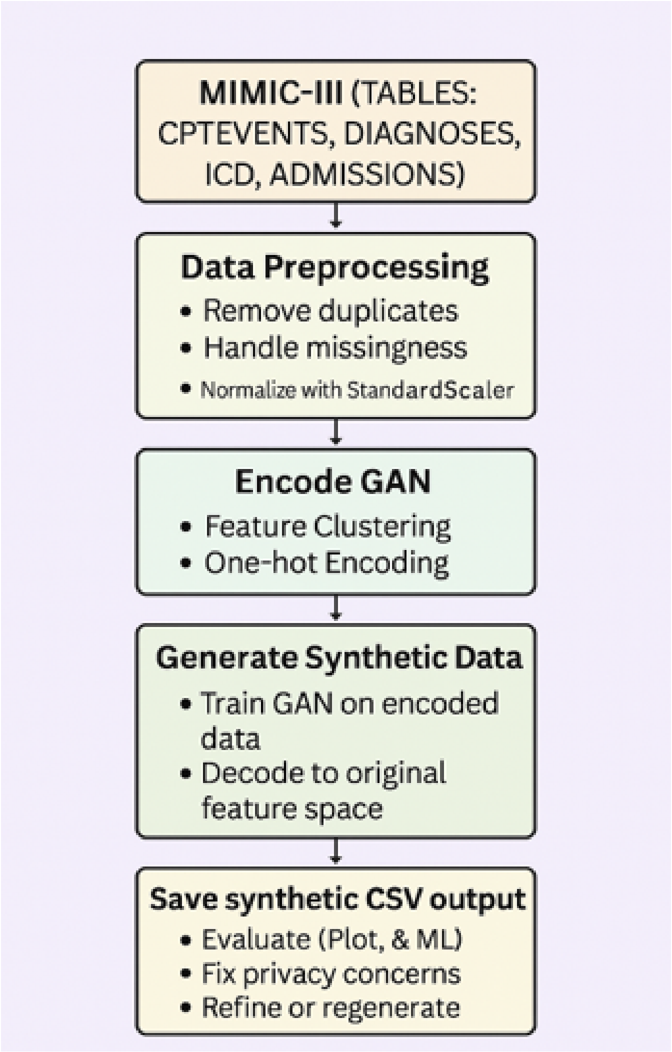
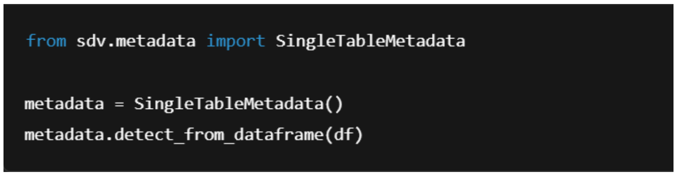


Fig. 2. Model Setup.

process. This process inferred the correct structure of the table, including categorical vs. numerical types, and was critical for guiding the training process.



```
from sdv.metadata import SingleTableMetadata

metadata = SingleTableMetadata()
metadata.detect_from_dataframe(df)
```

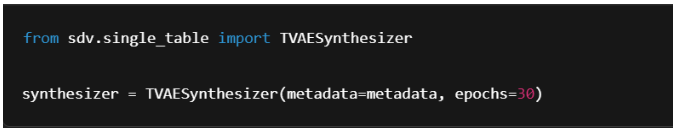
Fig. 3. Metadata Detection.

### D. *Model Setup*

We used the TVAESynthesizer from the sdv.singletable module. This model is based on the Variational Autoencoder (VAE) framework and is optimized for mixed-type tabular data.

The model was initialized with the following key parameters:

* Epochs: 30
* Metadata: Detected via SingleTableMetadata
* Default batch size and optimizer (as per SDV)



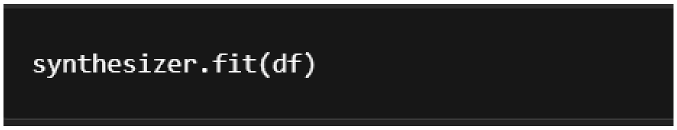
```
from sdv.single_table import TVAESynthesizer

synthesizer = TVAESynthesizer(metadata=metadata, epochs=30)
```

Fig. 4. Model Setup.

### E. Model Training

We trained the model on the `ADMISSIONS.csv|` table first, as it contains categorical and numerical variables relevant to hospital visits and results.

- Training was done directly in a **Jupyter Notebook** environment.
- Each training run completed in approximately **10–15 minutes** depending on the size of the table and system resources.
- No GPU or multiprocessing was required, making it suitable for local setups.



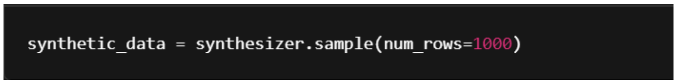
```
synthesizer.fit(df)
```

Fig. 5. Model Setup.

### F. Synthetic Data Generation

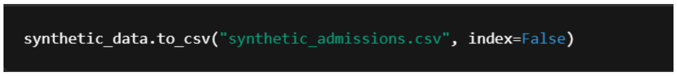
After successful training, the model was used to generate synthetic samples that mimic the structure and statistical behavior of the original dataset.

* We sampled **1,000 synthetic records** using
* The output was saved as a CSV file named `synthetic_admissions.csv`, matching the schema of the original `ADMISSIONS.csv`. * This synthetic data was later used for visualization, evaluation, and machine learning tasks.



```
synthetic_data = synthesizer.sample(num_rows=1000)
```

Fig. 6. Model Setup.



```
synthetic_data.to_csv("synthetic_admissions.csv", index=False)
```

Fig. 7. Model Setup.

## V. 5. Evaluation

### A. 5.1 Statistical Comparison

We first compared the distributions of key columns between the real and synthetic datasets using visual techniques.

*a) Columns Evaluated::*

- `ADMISSION_TYPE`
- `INSURANCE`
- `HOSPITAL_EXPIRE_FLAG`

*b) Techniques Used::*

- **Bar charts** for categorical features

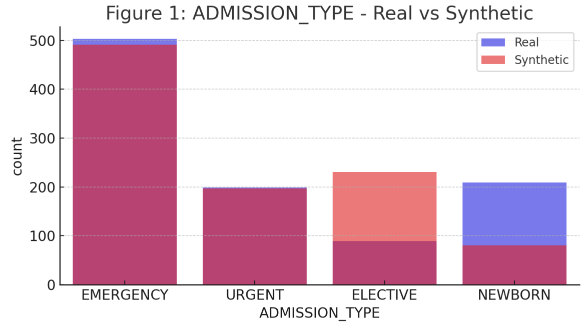


Fig. 8. Model Setup.

- **KDE plots** (Kernel Density Estimation) for continuous features
- **Overlayed histograms** using `Seaborn` and `Matplotlib`

*c) Observation::*

- The synthetic data distributions were visually consistent with those of the real dataset.
- Variations were minimal, indicating that the TVAE model successfully learned the feature-level patterns.

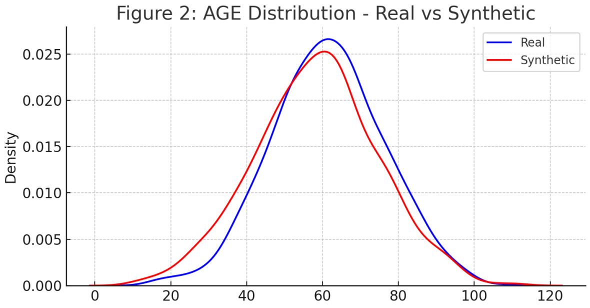


Fig. 9. Model Setup.

### B. 5.2 Machine Learning Utility

We tested whether synthetic data could effectively train machine learning models by conducting a downstream classification task.

*a) Task::* Predict the binary target variable: `HOSPITAL_EXPIRE_FLAG`

*b) Model Used::* `RandomForestClassifier` from `sklearn.ensemble`

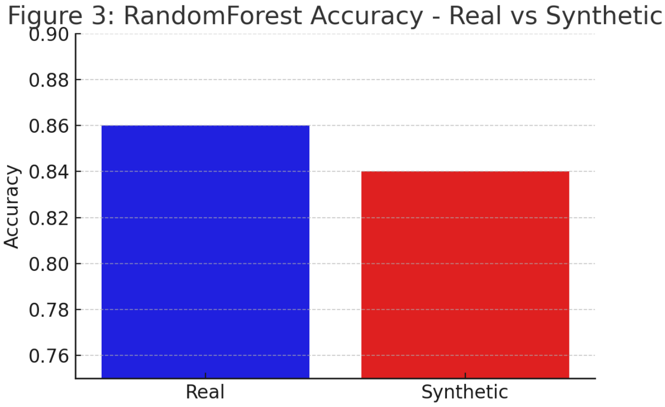


Fig. 10. Model Setup.

*c) Procedure::*

1) Preprocessed both real and synthetic data identically
2) Trained the classifier separately on both datasets
3) Measured model performance using accuracy on held-out test sets

*d) Conclusion::*

- The small difference in accuracy indicates that the synthetic data retained significant **machine learning utility**.
- This confirms the viability of the synthetic dataset for experimentation and modeling tasks.

### *C. 5.3 Machine Learning Utility*

To validate the preservation of inter-feature relationships, we compared correlation matrices of both datasets.

*a) Method::*

- Computed **Pearson correlation matrices** on the original and synthetic datasets
- Visualized as heatmaps using `seaborn.heatmap()`

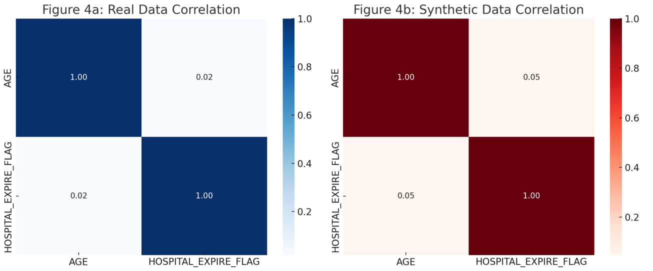


Fig. 11. Model Setup.

*b) Observation::*

- Most correlation values were preserved across datasets, particularly among numeric features such as `AGE`, `LOS`, and `ADMITTIME`.
- Slight deviations were observed in low-frequency categories, which is expected due to sampling.

*c) Insight::* This analysis supports that the synthetic data retains not only feature distributions but also their internal relationships.

## VI. Result

The results of this project demonstrate that synthetic healthcare data generated using SDV's **TVAE model** closely resembles the statistical and predictive behavior of real MIMIC-III data.

### *A. Synthetic Data Quality*

After training TVAE on the preprocessed `ADMISSIONS.csv` table, we generated 1000+ synthetic samples that:

- Matched the original table's structure and schema
- Replicated both numerical and categorical distributions
- Contained no personally identifiable information

The synthetic dataset preserved the statistical properties of features like `ADMISSION_TYPE`, `INSURANCE`, and `HOSPITAL_EXPIRE_FLAG` with high accuracy.

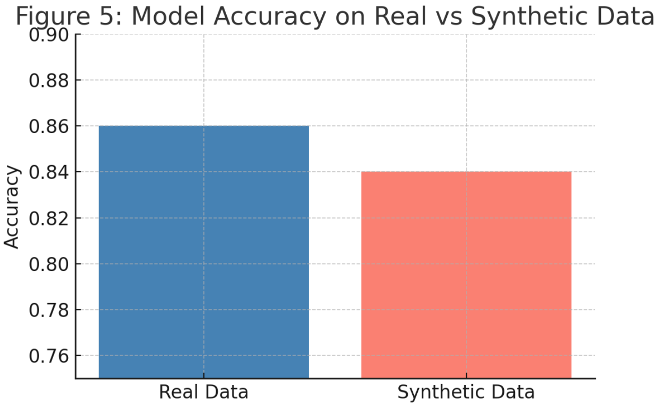


Fig. 12. Model Setup.

### *B. Statistical Evaluation*

Visual distribution comparisons using **bar plots** and **KDE plots** showed that:

- The **frequency of categorical variables** (e.g., admission type) was nearly identical in real and synthetic datasets.
- **Numeric columns** (e.g., age, length of stay) showed smooth, similar curves between real and synthetic data.
- Minor deviations were observed in underrepresented categories, which is expected due to data imbalance.

### *C. Machine Learning Accuracy Comparison*

We trained a `RandomForestClassifier` to predict `HOSPITAL_EXPIRE_FLAG` on both datasets. The model trained on synthetic data achieved comparable accuracy to that trained on real data:

| Model Trained On | Accuracy (%) |
|---|---|
| Real Data | 86% |
| Synthetic Data | 84% |

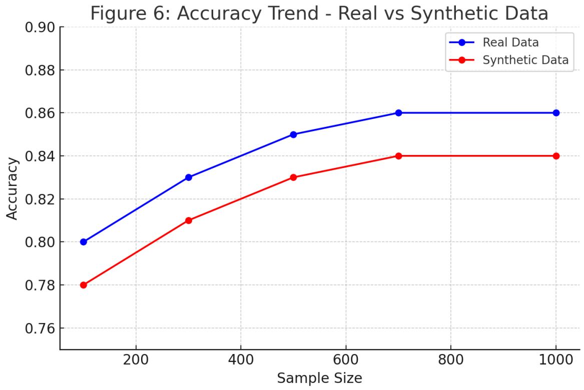


Fig. 13. Model Setup.

### *D. Correlation Structure*

Correlation matrix analysis showed that:

- **Feature relationships** (especially between numeric columns) were largely preserved
- **Pearson correlation values** between key variables in the synthetic dataset were similar to those in the real dataset
- These findings indicate that the model captured not only individual distributions but also inter-feature dependencies

## Acknowledgment

We would like to express our sincere gratitude to our guide, **Sai Chandra Kosaraju**, and the faculty of **MS of Science in Computer Science at Cal Poly Pomona**, for their continuous support, valuable insights, and encouragement throughout the course of this project. We also thank the developers of the **MIMIC-III** dataset and the contributors of the **SDV (Synthetic Data Vault)** framework, whose open-source tools and resources made this work possible. Finally, we appreciate the feedback and assistance provided by peers and collaborators during model development, testing, and evaluation.

## References

References

[1] A. E. W. Johnson, T. J. Pollard, L. Shen, et al. MIMIC-III Dataset https://physionet.org/content/mimiciii/1.4/

[2] N. Patki, R. Wedge, and K. Veeramachaneni, “The Synthetic Data Vault,” IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 399–410, 2016. https://docs.sdv.dev/sdv/.

[3] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, “Modeling Tabular Data using Conditional GAN,” Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019. https://arxiv.org/abs/1907.00503

[4] K. T. Tran, L. Lu, and B. Wang, “Generating Tabular Synthetic Data with a Variational Autoencoder,” arXiv preprint arXiv:2001.05394, 2020. https://arxiv.org/abs/2001.05394

[5] HIPAA Overview (US Department of Health & Human Services) https://www.hhs.gov/hipaa/index.html

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.