GANBLR Project Interface Documentation

Prepared by: Rachel Moraa

# 1. Introduction

This project provides a web-based interface for the GANBLR pipeline to generate and verify synthetic credit card data. The goal is to ensure that the synthetic data closely resembles real data for statistical and machine learning purposes.

The interface allows users to:

* Load and preprocess datasets
* Train a GAN model to generate synthetic data
* Verify the quality of synthetic data
* Generate and visualize comparisons between real and synthetic data

# 2. Technologies Used

* Backend: Flask (Python Web Framework)
* Frontend: HTML, CSS, Flask Templating
* Data Visualization: Matplotlib, Seaborn
* Data Processing: Pandas, NumPy
* Model Training: Keras, TensorFlow
* Development Tools: Python 3.9+, IDE (e.g., VSCode, PyCharm)

# 3. Project Structure

The folder structure is organized as follows:

GANBLR-Interface/  
│  
├── app.py Flask main application  
├── templates/  
│ ├── index.html Main interface  
│ ├── visualizations.html Visualization results  
│  
├── static/  
│ └── visualizations/ Generated plots stored here  
│  
├── load\_credit\_dataset.py Loads the dataset  
├── preprocess\_credit\_data.py Preprocesses the data  
├── train\_ganblr\_credit.py Trains GAN model  
├── verify\_synthetic\_data.py Verifies synthetic data quality  
├── feature-distribution.py Generates feature comparison plots  
└── correlation-heat-maps.py Generates correlation heatmaps

# 4. System Workflow

Overview:

* Load Dataset: Load the credit card dataset (`creditcard.csv`) into the system.
* Preprocess Data: Normalize and split the dataset into training and testing sets.
* Train Model: Train the GANBLR model to generate synthetic data.
* Verify Synthetic Data: Compare synthetic data against real data.
* Generate Visualizations: Feature Distributions and Correlation Heatmaps.
* View Results: Results and plots are displayed on the web interface.

# 5. Installation and Setup

* Install Python 3.9+
* Install necessary libraries using pip
* Run the application using 'python app.py'

# 6. Application Interface Overview

* Load Dataset: Executes load\_credit\_dataset.py
* Preprocess Data: Executes preprocess\_credit\_data.py
* Train Model: Executes train\_ganblr\_credit.py
* Verify Data: Executes verify\_synthetic\_data.py
* Generate Visualizations: Generates feature and correlation plots

# 7. Scripts and Functionalities

* load\_credit\_dataset.py: Loads the dataset.
* preprocess\_credit\_data.py: Splits and normalizes the data.
* train\_ganblr\_credit.py: Trains GANBLR to generate synthetic data.
* verify\_synthetic\_data.py: Verifies synthetic data quality.
* feature-distribution.py: Generates feature comparison plots.
* correlation-heat-maps.py: Generates correlation heatmaps.

# 8. Visualizations and Explanations

* Feature Distributions: Compares selected feature distributions (V1, V2, Time, Amount).
* Correlation Heatmaps: Visualizes relationships between features in real and synthetic data.

# 9. Testing and Validation

* Functional Testing: Ensure all buttons trigger their respective scripts.
* Visualization Validation: Charts are saved to 'static/visualizations'.

# 10. Troubleshooting

* Dataset not loading: Ensure 'creditcard.csv' is in the root directory.
* Charts not displaying: Verify output directory paths are correct.

# 11. Future Improvements

* Add progress bars for model training.
* Enhance UI/UX with interactive visualizations.
* Include more metrics like precision and recall.

# 12. Conclusion

* This interface provides an efficient way to process, generate, and validate synthetic data. It ensures that GAN-generated data can closely resemble real data for use in ML pipelines.