Software Requirements Specification

for

adVAE – Alzheimer's Data Variational Autoencoder

Version 1.0

Prepared by

Name: Varnika Umashankar

UMID: 04150536

Email: varnika@umich.edu

Instructor: Cristina Mitrea

Course: BIOINF 576 001 WN 2025

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Varnika Umashankar	Original issue	

1. Introduction

This Software Requirements Specification (SRS) document defines the functions and requirements of the software, adVAE - Alzheimer's Data Variational Autoencoder.

1.1 Purpose

The purpose of this project is to develop a Variational Autoencoder model to generate synthetic data specifically to aid Alzheimer's research. The goal is to address the data scarcity issue in Alzheimer's research by augmenting and creating high-quality multimodal data that is biologically and statistically relevant. adVAE 1.0 will have multiple modules performing data augmentation for different kinds of data involved in Alzheimer's research, such as gene expression data, EEG data, and MRI data.

1.2 Project Scope

Deep learning (DL) models are becoming increasingly popular in the analysis of transcriptomic and other data aiding in research. However, a key challenge in training such models is their requirement for large datasets. High-quality clinical data for Alzheimer's patients is scarce, and hence, data augmentation is needed to address the challenges posed by small sample sizes. adVAE aims to tackle this challenge and aid Alzheimer's research by:

- Developing a VAE-based generative model for augmenting Alzheimer's datasets.
- Supporting multiple data modalities (gene expression, EEG, MRI).
- Implementing statistical and biological evaluation metrics to assess synthetic data quality.
- Providing an easy-to-use command-line interface for training and generating synthetic data.

1.3 Intended Audience and Document Overview

This document is intended to be used by bioinformaticians, researchers and neuroscientists who are building deep learning models for Alzheimer's research and are looking to create synthetic data to train their respective models. This document will provide the general outline of the project, its hardware and software requirements, and product functionality.

1.4 References and Acknowledgments

adVAE 1.0 was developed for the course "Tool Development for Bioinformatics" (course code: BIOINF576) (affiliated to University of Michigan) under the supervision of instructor Dr. Cristina Mitrea. The gene expression data used to develop this model was obtained from [source/database]. The EEG data was sourced from [source/database], and MRI data was retrieved from [source/database]. I also acknowledge the developers of conda, Snakemake, PyTorch, scikit, matplotlib, NumPy, pandas, and PyYAML which were used in the development of this project.

2. Overall Description

2.1 Product Overview

adVAE is a self-contained software package built on Python that aims to provide a pipeline for easy multimodal data augmentation to aid Alzheimer's research. It was developed to be easily integrable with other existing bioinformatics pipelines and provide biologically and statistically relevant synthetic data to be used for subsequent ML model training, validation, and testing.

2.2 Product Functionality

- Preprocessing of input data such as gene expression, EEG, and MRI data used to train the VAE.
- Latent space representation of various data modalities using the VAE.
- Generation of synthetic data from the latent space variables modelled while training.
- Easy integration with existing bioinformatics workflows.

2.3 Design and Implementation Constraints

This package was developed using Python 3.9.18 primarily making use of the library PyTorch 2.2.0. It is recommended to train the model on a GPU such as Nvidia's CUDA if available.

2.4 Assumptions and Dependencies

The package makes use of conda environment manager as well as Snakemake workflow manager to run the pipeline smoothly. Public Python libraries required for the smooth operation of adVAE would be installed into the conda environment that would be created during the initialization step and hence needn't be installed to the user's local machine.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The adVAE Python package needs to be cloned from its online public GitHub repository link] [to user's local machine. Once cloned, the dataset of user's choice would need to be downloaded to the user's local machine to the specified directory. The user would also need to follow the initialization steps as specified in README.md before training the model, evaluating its performance, and then generating the synthetic data of choice. The user would be making use of the conda environment's command prompt for carrying out adVAE's functional modalities, and the generated synthetic data would be stored in appropriate formats in the directories created during initialization.

3.1.2 Hardware Interfaces

It is recommended that the user has access to a GPU to train the model and perform the subsequent synthetic data generation or perform any other functional modality.

3.1.3 Software Interfaces

adVAE requires the user to have conda installed on their local machine as it makes use of multiple publicly available Python libraries such as PyTorch, matplotlib, scikit, NumPy, pandas and PyYAML. These libraries needn't be imported to the user's local machine but rather would be installed in the conda environment that the user would be required to create as part of the initialization step.

3.2 Functional Requirements

3.2.1 Data Preprocessing

- After the project initialization has been carried out, the user would be required to download their dataset of choice to the respective folders.
- This dataset could be sourced on their own. Alernatively, the user could use the recommended datasets available on adVAE's GitHub repository [link] to train the model.
- adVAE would have in-built functions to load and read the user's datasets.
- The dataset (gene expression, EEG or MRI) would then need to be preprocessed (data normalization, feature scaling, missing value handling etc.) through adVAE's in-built functions. Alternatively, the user could use already processed datasets and skip this step.

3.2.2 adVAE Model Implementation

adVAE defines classes that could be used by the user to create their VAE model architecture which would consist of an encoder, the latent space, and a decoder. It would also have in-built functions to initialize weights, calculate losses, and perform hyperparameter tuning for the user.

3.2.3 Data Generation and Evaluation

adVAE provides functional modalities to perform latent space modelling and subsequent latent space sampling to perform synthetic data generation. It would also have in-built functions to validate the generated data through various statistical similarity metrices and image similarity metrices.

3.3 Performance Requirements

adVAE aims to create a robust tool which can reduce the training time of the VAE model to less than 2 hours with the help of GPUs. The goal is to reduce synthetic data generation time by generating small batches of data at a time and validating it, rather than generating large batches of data. adVAE would also support multiple data modalities relevant to Alzheimer's research (gene expression, EEG, MRI). It would be designed in such a way that it would be robust against model overfitting, exploding and vanishing gradients.

3.4 Data Requirements

3.4.1 Input Data Format

Input data format depends on the datatype used to train the model.

- Gene expression data would be in a .csv format and will be sourced from NCBI's GEO and EMBL-EBI.
- EEG time-series data would be in a .edf format and will be sourced from OpenNeuro and TUH EEG Corpus.
- MRI data would be either in the NIfTI or .nii format and will be sourced from OpenNeuro and WashU's OASIS.

3.4.2 Output Data Format

Format of the output data generated depends on the type of data generated. Synthetic gene expression data would be in the .csv format, EEG data would be in the .edf format, and MRI data would be in the .nii format.

3.4.3 Data Security Requirements

In order to be compliant to HIPAA/GDPR rules and regulations, adVAE's recommended datasets would be HIPAA/GDPR compliant, with no patient identifiable information being used to train, test or validate the model.

4. Use Cases

4.1 Use Case 1: Data Preprocessing

User: Data Scientist/Bioinformatician/Alzheimer's Researcher
Aim: Prepare raw Alzheimer's data (gene expression/EEG/MRI) for VAE input.
Preconditions: Raw data is available and has been downloaded into the specified directory.
Main Flow:

- Load raw data.
- Clean (remove noise/outliers) and normalize data.
- Format the data into the required structure.

Result: Data is cleaned, structured and validated in required format.

4.2 Use Case 2: Latent Space Modelling

User: Data Scientist/Bioinformatician/Alzheimer's Researcher
Aim: Train a VAE on the pre-processed data and optimize the latent space variables.
Preconditions: Raw data has been preprocessed into the required format.
Main Flow:

- Load the pre-processed data.
- Initialize the VAE with chosen hyperparameters.
- Train the model while monitoring progress.
- Optimize the hyperparameters.
- Output a trained VAE model.

Result: Trained model is created for high-quality synthetic data generation.

4.3 Use Case 3: Synthetic Data Generation

User: Data Scientist/Bioinformatician/Alzheimer's Researcher
Aim: Generate synthetic Alzheimer's data by sampling latent space variables.
Preconditions: VAE has been trained on the dataset of interest and has been optimized.
Main Flow:

1.1 Sample latent space variables.
1.2 Generate synthetic data from the model.
1.3 Store generated data in required format.

Result: Trained model is created for high-quality synthetic data generation.

Appendix A – Abbreviations

AD	Alzheimer’s disease
VAE	Variational Autoencoder
EEG	Electroencephalogram
MRI	Magnetic Resonance Imaging
SRS	Software Requirements Specification
ML	Machine Learning
DL	Deep Learning
GPU	Graphics Processing Unit