School of Computer Science and Electronic Engineering

"Software Requirements Specification Document"

Under the subject of CE903 Group Project

Project Topic: Generating Synthetic Data in the context of Health Data, and comparing it to Real World Data using Artificial Intelligence Systems

Supervised by: Dr Ana Matran-Fernandez

Submitted by:

Team 15 Group Members	
Samuel Jularbal	2101224
Ahmed Ali	2201163
Priteshkumar Thakkar	2201235
Isreal Ufumaka	2204569
Abdul Wahid	2202396
Shikhar Sharma	2205456
Yuan Zan	2200557

Dharsingan Bharathidasan	2205532
Adedeji Adetunji	2201823
Ashu Berwal	2204914

Table of Contents

1. INTRODUCTION

1.1 Purpose of this project

This is a System Requirements Specification document for the synthetic data generator, which is supposed to generate synthetic data from a medical dataset. Its purpose is to find whether synthetic data may be a practical solution to the issue of scarce and confidential data in the healthcare domain and to what extent it can supplement the existing data sets, by highlighting the comparison between the characteristics of the original dataset, the synthetic dataset, and an oversampled dataset, used in the domain of medical research and development. The oversampled dataset under consideration will be the original dataset, whose minority classes will be oversampled using the Synthetic Minority over-sampling Technique (SMOTE), in contrast to an entirely new dataset which will be generated by using Generative Adversarial Network (GAN). Agile development & CRISP-DM methodologies are used for this project.

We aim to check the feasibility and effectiveness of synthetic data generated from the medical observations, that have been recorded and stored as data, to be used as training data for modern-day Machine learning (ML) and Artificial Intelligence (AI) based applications. We will see how closely the synthetic dataset generated from a GAN follows the statistical characteristics of the original dataset.

Keeping in mind that the use of data including sensitive personal information (such as a medical dataset that holds patient records) raises many data privacy concerns such as identity theft. The data available for modern-day applications suffer from imbalance and have varying feature distributions that introduce bias [1]. Synthetic data helps us to mitigate such issues by over-sampling the minority classes present in the dataset, thus dealing with the underrepresented features by obtaining a balanced data distribution. This further results in better performance of trained models.

Synthetic data can help us by reducing the cost of data collection, improving the accuracy of labeled data, scaling the training and testing of the ML model, and adding a variety of rare edge cases thus helping in reducing bias and improving the generalization capabilities of the model.

1.2 Scope

With the use of synthetic data, our focus is to address the previously mentioned concerns by comparing and visualizing the statistical characteristics of the original data, oversampled data (using SMOTE), and data generated using GAN [2]**.** The motivation for using GAN in data synthesis is an introduction to more enriched data features and an introduction to variability. This makes it possible to produce a variety of representative data sets that can be used to train and validate AI models [3].

The original dataset will be analyzed to find out certain characteristics such as missing values, data types, distribution of variables, class imbalance, correlations, and trends. Univariate, bivariate, and multivariate analyses will be performed and visualized to understand the underlying structure of the original dataset. A similar analysis will be conducted on the synthetic datasets to check their conformity with the original data.

This development process will include a thorough use of statistical evaluation methods that will help us decide how close the synthetic dataset's characteristics are to the original dataset.

2. REQUIREMENTS ANALYSIS AND SPECIFICATION

The requirements have been elicited by considering medical use cases such as liver lesion classification and diabetes prediction - both using synthetic data. Further brainstorming is done to specify the aim of the project. The requirements have been analyzed and specified by creating a system model, using data-flow diagrams [4].

Owing to the agile method of project development, the requirements will be prioritized and developed in order. Small parts of the process will be modeled to better understand the concepts under consideration. Frequent review meetings will be conducted to check that the development is taking its right course and to improve the system design iteratively.

2.1 Key system functional requirements

Functional Requirements are expected necessities for the system to handle. Here are some of functional requirements of the system:

1. Data acquisition will be the first step in the direction of generating synthetic data. To select a suitable dataset, we will look at some of the published medical datasets and apply basic statistical methods to them to get an understanding of their underlying properties such as the size of the dataset, continuous and categorial features, class imbalance, and probability distribution. We would aim to find a dataset such that its characteristics give us enough scope to see the effects of synthetic data in contrast to the original dataset.

2. Data analysis and visualization of the earlier method outcomes for the selected dataset, will be done to specify the dataset's characteristics.

3. Data splitting will be performed on the selected dataset. The original data set will be randomized and split into training and testing sets. Randomization will be done to avoid biased selection due to unbalanced feature variables. The test set will be kept untouched till we compare the results. Further, we will create two copies of the training dataset.

4. First copy will be used for training a Generative Adversarial Network (GAN) to generate a synthetic dataset.

5. Over-sampling of the second dataset will be done using Synthetic Minority Over-sampling Technique (SMOTE)

6. Statistical evaluation will be performed on both datasets, as in the original data's case.

7. Visualization of statistical outcomes will be done to present the outcomes of the study.

2.2 Design Constraint

The system should preferably include a data pipeline that can fulfill the above key functional requirements sequentially and deliver the output for visualization.

A clean, concise, and modifiable code structure is preferred, following Object-Oriented Programming (OOP) practices.

GAN should be trained carefully to give consistent and reliable samples as outputs.

Most medical information is very private and cannot be shared to develop AI. Due to privacy restrictions, including those enacted under the General Data Protection Regulation (GDPR) in the

European Union, and ethical considerations about the management of sensitive patient data, this is the case. According to [1] Privacy should be an important concern in a big data era of data sharing, and the process involving it should be a top priority.

3. Testing Schedule

3.1 EVALUATION CRITERIA AND TESTING SCHEDULE

For evaluating data generated from GAN different metrices can be used, which can compare to the synthetic data to the original data. There are two kinds of evaluations [5]:

1. Qualitative evaluation: non-numeric forms of evaluations fall under this type. This can be done by visualizing dataset characteristics using scatter plots, histograms, and clustering.
2. Quantitative evaluation: This consists of numeric tests which evaluate factors such as:

- Statistical summary
  - Mean, median, mode
- Outlier / Anomaly detection
- Statistical test to evaluate distribution
  - T-test
  - Kolmogorov-Smirnov Test
  - Anderson-Darling Test
  - Chi-squared test
  - Two sampled t-test
  - Mann-Whitney U test
  - Willcoxon rank-sum test
  - Kullback-Leibler divergence test
- Model performance metrics
  - Accuracy
  - F1 score
  - ROC AUC

Specialised tests can be used to evaluate data generated using GANs, such as:

- Fréchet Inception Distance (FID) [6][7]
- Robust Inception distance (R-FID) [7]
- Kernel Inception Distance (KID) [5]
- Wasserstein distance metrics [6]

The evaluation metrics for the Synthetic data generator will be selected from these available methods of evaluation. To select the correct evaluation metrics, we would first need to find the right set of statistical tests, and consider the follow factors [13]:

1. Types of data
   - Categorical

- Continuous

2. Distribution of data
   - Normal *distribution:* use parametric test for e.g., *T-test* or *ANOVA*
   - Otherwise, use *non-parametric tests* like *Mann-Whitney U test* or the *Wilcoxon rank-sum test*
3. Nature of relationship
   - Test relationship between two continuous variables, use *Pearson-correlation* or *regression analysis*
   - Test relationship between categorical and continuous variables, use *ANOVA* or *chi-squared test*
4. Number of groups
   - If dataset has more than two groups, then use a test that can handle multiple groups, such as *ANOVA* or *Kruskal-Wallis*.
5. Independence of groups
   - For independent groups, use *t-test* or *Mann-Whitney U test*.
   - For dependent groups, use *Paired t-test* or *Wilcoxon signed-rank test*

3.2 GANs Evaluation

Generative Adversarial Networks (GANs) are a type of deep learning algorithm used for generating new, synthetic data.

GANs consist of two main components: a generator and a discriminator. The generator handles generating new data, while the discriminator handles evaluating the authenticity of the generated data. The generator and discriminator are trained together in an adversarial manner, with the generator trying to create data that is as close as possible to the real data and the discriminator trying to correctly identify whether the data is real or generated. The two components are trained iteratively until the generator can create data that is indistinguishable from the real data.

There are several ways to evaluate synthetic data generated by GANs:

3.2.1 Qualitative evaluation:

Visual inspection of the generated samples can give an idea of how well the GAN has learned to capture the training data distribution. For example, in the case of images, you can display a grid of generated samples and compare them to real samples from the training set.

3.2.2 Quantitative evaluation:

Several metrics can be used to quantify the quality of the generated data. Some of the commonly used metrics include:

3.2.3 Frechet Inception Distance (FID) or Inception Score (IS) [5]:

The FID measures the similarity between two datasets in a high-dimensional feature space, which is obtained using a pre-trained neural network. The idea is to compare the activations of the network for real data and generated data, and the FID score is calculated based on the distance between these activations in the feature space.

3.2.4 Kernel Inception Distance (KID):

The Kernel Inception Distance (KID) is a metric for evaluating the quality of the generated dataset, specifically for GANs. It measures the distance between two distributions of features generated by a pre-trained neural network (usually called an inception network) for real images and generated datasets.

The KID is calculated using a kernel mean embedding, which maps the feature distributions into a reproducing kernel Hilbert space. The distance between the two feature distributions is then calculated using the Maximum Mean Discrepancy (MMD) metric, which measures the difference between the means of the two distributions.

3.2.5 Precision and Recall:

These metrics compare the generated data to a validation set and report the proportion of generated samples that match real samples and the proportion of real samples that match generated samples, respectively.

3.3 Testing Schedule: Functional & Non-functional Requirements

In the context of the testing schedule, functional requirements consist of requirements that perform appropriate tests for the model to work successfully, while a non-functional requirement is defined by the functions that are needed to support the testing of the overarching project [13].

Functional requirements for a Machine Learning testing system can be defined as a series of tasks that the model is expected to handle, such as:

- Data Input: The model should handle and process different datasets and data inputs I.e., categorical and text data
- Choosing the model: The model should either be pre-determined (an already developed algorithm) or a customized model
- Training the model: The model should train on the dataset to have a given output
- Model testing: The model should be evaluated through a validation set and given testing matrices
- Model Output: The model should supply an output that will be put through an acceptance test, and decide whether the results are significant

On the other hand, Non-Functional Requirements are more difficult to define compared to their counterparts [14]. Reasons for this are due to the difficulty for measuring and defining the Non-Functional requirements for ML systems [14]. Some possible Non-Functional Requirements for this project may include:

- Performance: The model should be able to process a large dataset in good time
- Privacy: The model should protect sensitive data, and should not disclose personal information (which is especially important given that we are working with health-based data)
- Scalability: The model should be able to scale to larger amounts of data without having its usability or accuracy degraded.

3.4 Agile Testing

For the testing schedule, we will be following the agile methodology for testing. Agile Testing is a software process that follows the principles of agile, aligning with the iterative development methodology [15].

3.4.1 Agile Testing Principles:

- Testing is Continuous**:** An agile team tests continuously & tests are performed by the whole team
- Continuous feedback: Agile testing provides consistent feedback on an ongoing basis
- Less documentation: An agile team uses reusable checklist and focus on the tests
- Test Drive: In agile, testing methods is performed at the time of implementation (during sprints)

[15]

3.4.2 Advantages of Agile Testing

- Saves time
- Reduces documentation
- Flexible and highly adaptable to changes
- Provides a way for receiving regular feedback from end user
- Better determination through daily meetings

[15]

Although Agile Testing is a software focused methodology, it can also be applied to Machine Learning & AI in a comparable way. Such ways are:

1. Continuous testing: All stakeholders should work together on testing and collaborating towards the desired acceptance criteria.
2. Continuous feedback: Constant feedback should be given regarding the process of developing the model
3. Continuous delivery: new algorithms or data should be integrated, whilst coding needs to be tested and updated often
4. Flexibility: The model should be flexible and easily adaptable to changes or necessary changes to functional requirements

Practically, we will implement the testing schedule through iterations (Sprints) which are performed between 1-4 weeks, with each sprint iterating on the earlier sprint's work and improving the model. Examples of how a 4 Sprint Agile Testing system could look like:

Sprint 1:

- Defining the objectives of the project, explicitly outlining the functional and non-functional requirements of the model
- Setting up systems for data storage and selecting the applications that will be used
- Gather and prepare training data

Sprint 2:

- Train the AI model on the training data
- Evaluate the model's performance and accuracy on generating synthetic data
- Address any issues identified during the testing

Sprint 3:

- Refine the model by incorporating further training data, adjusting the model where necessary and continue to evaluate the model's performance
- Evaluate the accuracy of the model
- Address any issues identified during testing

Sprint 4:

- Test the AI model against real-world data by generating a large volume of synthetic data
- Compare the synthetic data to the real data
- Evaluate the quality and reliability of the synthetic data and make any improvements
- Address any issues identified during the testing

This process will continue if the project requires it or until close to the deadline for the final submission. Having a model that fulfills the acceptance criteria will also mean the end of the process.

4. Project Management

4.1 Project Management Structure and Methodologies

For this project's management structure, we will use a combination of two methods: a CRISP-DM Reference Model and Agile. CRISP-DM, (Cross-Industry Standard Process for Data Mining) is a common place project management methodology for Data Science and data driven projects [16]. Agile focuses on the understanding that software development is knowledge work performed by individuals working together with other individuals in teams [13].

CRISP-DM follows six distinct phases, each of high importance in achieving a complete data project.

- Phase One: Business Understanding
    - Defining the data problem
    - Determine the business objectives
- Phase Two: Data Understanding
    - Collecting the initial data
    - Describe the data and analyse the properties
    - Verify quality of data
- Phase Three:
    - Construct the data
    - Clean the data
    - Format the data
- Phase Four: Modelling
    - Select the modelling technique
    - Generate test and train sets
    - Build the model
- Phase Five: Evaluation
    - Evaluate Results
    - Review Process
- Phase Six: Deployment
    - Plan deployment
    - Plan monitoring
    - Produce final report

[14].

CRISP-DM is used widely across many industry-based data-driven projects, particularly in Data mining, AI and ML. The methodology suits the typical steps of an AI project, starting from proposing a question, to gathering data and pre-processing data – and eventually producing an output based on that said data. Given the usefulness of CRISP-DM in the context of this project’s scope, we will be able to follow each step for the given methodology. However, an

issue to using CRISP-DM is the ability to cope with scaling an ML project with huge complexity, especially when working with Big Data [16].

Using CRISP-DM to structure the project, we will be working in an Agile way to complete tasks for the overall project. Agile is an adaptive rather than a predictive approach to project management, allowing people to work flexibly and react to changes in developments quickly [16]. As a team, we will be following the Scrum method of Agile – working on product backlogs, having regular meetings (scrums) and working on a weekly sprint basis [17].

Using Agile will give us flexibility to change project direction when unforeseen problems occur. Examples of such problems could be the dataset may be unworkable, the topic question may be out of scope for this project, the results may not be desirable. However, since this is a data-centric project, it will be difficult to get customer or interview feedback regularly.

As such, we will be using both CRISP-DM and Agile to provide a set data-centric structure for our project, whilst integrating the flexibility of Agile and working in sprints to achieve our final AI model.

4.2 Tools for Project Planning and Management

Jira – Sprint planning

We will be using Jira as our project management platform which we used for issue/work tracking for our project [16]. Our team uses the software as our collaboration and communication platform, organizing our workflows, assigning tasks, and our team management.

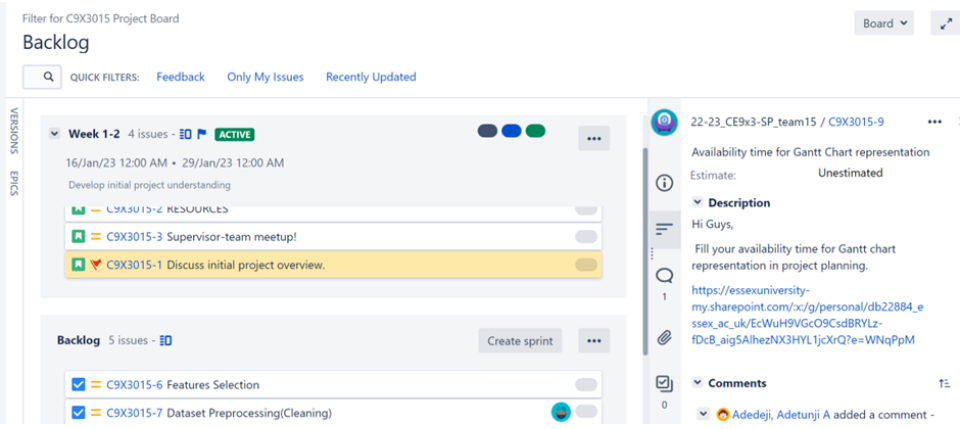

Fig. 1 showing our Jira Filter Interface for Backlog.

We will conduct our workflow on Jira in stages of project issues tracking from their creations to their completion respectively, the stages are basically - To do, In Progress and Done stage.

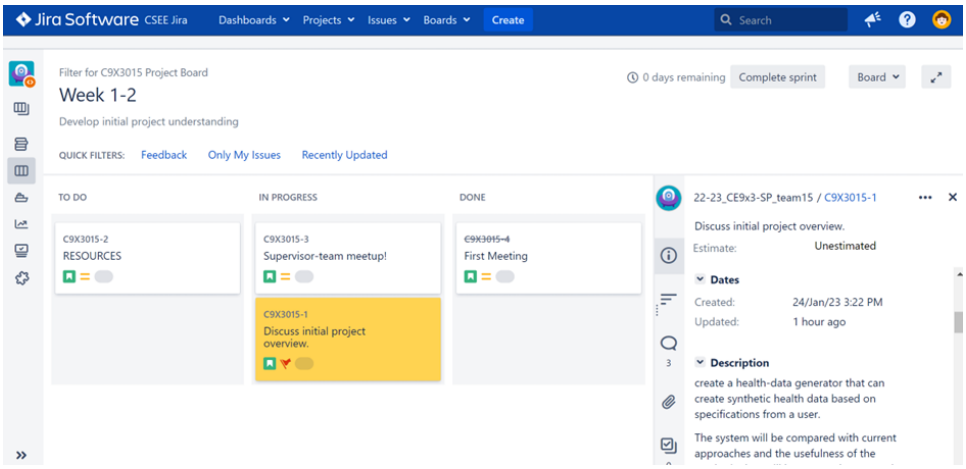

Fig. 2 showing our Jira Interface for Project Board.

The platform allows us to integrate other third-party applications, which is made accessible and easy to do in case we need further tools for our workflows.

Using Jira, we will be able to work agile-like by aligning with each other and providing support where necessary.

Other tools we will use to work in an agile-like manner:

- **Zoom:** this is an online video communication platform. We use this tool as our virtual meeting place to discuss our workflows allocations, progress made, and challenges encountered during the project. To keep the team updated with the information from our Project Supervisor. We also use this platform to have weekly scrums, any meetings in-between sprints and for overall communication between one another.

- **Communication Apps:** We will use tools like WhatsApp for quick updates on our project activities, to keep track of each other's work progress and notify each other when it is best to reach out to one another. This helps us work in a more agile manner, with clear flexibility and working human oriented.

4.3 Project Plan

For the designated project, we have divided the entire process into six different tasks such as Data Pre-Processing, Data Visualization, Modelling, Testing, Report Preparation and Compilation. These tasks will fall into Agile sprints which we will work weekly to implement.

The total number of days for completing the given project is **43 days (about 1 and a half months)** from 5th February 2023 to 22nd March 2023, however by using the Gantt Chart we have minimized the number of days into **38 days (about 1 month 1 week)** by allocating the

efficient time for the completion of each task with respect to the availability of the team members.

- Number of days allocated for Data pre-processing – 4 Days
- Number of days allocated for Data Visualization – 3 Days
- Number of days allocated for Modelling – 14 Days
- Number of days allocated for Testing – 5 Days
- Number of days allocated for Report Preparation – 5 Days
- Number of days allocated for Compilation– 4 Days

4.3.1 Gantt Chart Representation:

Gantt chart is the graphical visualization for scheduling, managing and monitoring the tasks in the given project. It is a type of bar chart which represents the project's progress and the tasks lists. This representation shows the project's timeline with each task's allocation with respect to the team member's availability, it also represents the start and end dates for each task and for the completion of the given project. [17]

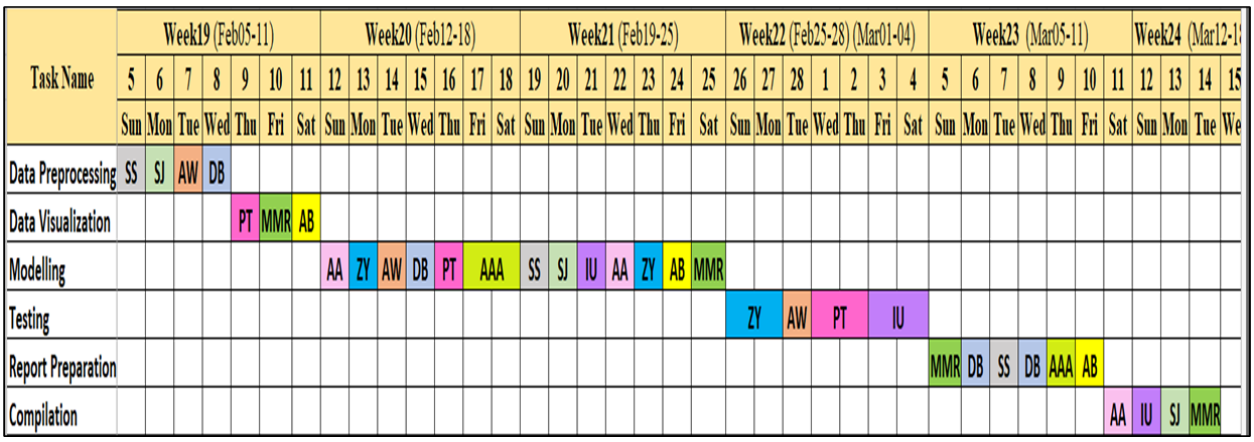

Figure: Gantt Chart Representation

Gantt Chart Explanation:

- The project is divided into six tasks which include Data Pre-Processing, Data Visualization, Modelling, Testing, Report Preparation and Compilation.
- We abbreviated the team members names as follows: Ahmed Ali – **AA**, Adetunji A Adedeji – **AAA**, Ashu Berwal – **AB**, Abdul Wahid – **AW**, Dharsigan Bharathidasan – **DB**, Isreal Ufumaka- **IU**, Mohammed M Rob – **MMR**, Priteshkumar H K Thakkar – **PT**, Samuel Julabal – **SJ**, Shikhar Sharma – **SS**, Zan Yuan – **ZY**.
- The number of days to complete the project using Gantt Chart representation is **38 days (about 1 month 1 week).**

5. Conclusion

Synthetic data can offer a mechanism to train and evaluate AI models that are precise, varied, and representational of real-world data by addressing the issue of limited and sensitive data. By developing a health-data generator that can produce synthetic health data based on user-specified parameters and assess its utility in comparison to real-world data, this project hopes to contribute to the field using GANs. When dealing with limited or biased real-world data, creating synthetic data with GANs can be an effective method for generating new, diverse, and representative data samples for a given problem. GANs, on the other hand, can be difficult to train and necessitate careful tuning to produce high-quality samples; additionally, the generated data may not precisely match the actual underlying distribution. Additionally, the GAN architecture and training procedure may have introduced biases or artifacts into the generated data. Through this research, we want to show how artificial intelligence (AI) can change the healthcare industry.

References

[1] Kuo, N.IH., Polizzotto, M.N., Finfer, S. et al. The health Gym: synthetic health-related datasets for the development of reinforcement learning algorithms. Sci Data 9, 693 (2022). https://doi.org/10.1038/s41597-022-01784-7

[2] Maayan F., Eyal K., Michal A., Jacob G., Hayit G. SYNTHETIC DATA AUGMENTATION USING GAN FOR IMPROVED LIVER LESION CLASSIFICATION. IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). https://www.researchgate.net/publication/325522195_Synthetic_data_augmentation_using_GAN_for_improved_liver_lesion_classification

[3] Noseong P., Mahmoud M., Kshitij G., Sushil J., Hongkyu P., and Youngmin K. Data Synthesis based on Generative Adversarial Networks. PVLDB, 11 (10): 1071-1083, 2018. DOI: https://doi.org/10.14778/3231751.3231757

[4] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements engineering and agile software development," WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003., Linz, Austria, 2003, pp. 308-313, Doi: 10.1109/ENABL.2003.1231428. https://ieeexplore.ieee.org/abstract/document/1231428

[5] Sandra Carrasco, Sylwia Majchrowska. On the evaluation of Generative Adversarial Networks, Artificial Intelligence in Healthcare Part III. https://towardsdatascience.com/on-the-evaluation-of-generative-adversarial-networks-b056ddcdfd3a

[6] Borji, Ali. Computer Vision and Pattern Recognition. Pros and Cons of GAN Evaluation Measures https://doi.org/10.48550/arxiv.1802.03446

[7] Alfarra, M., Pérez, J.C., Frühstück, A., Torr, P.H.S., Wonka, P., Ghanem, B. (2022). On the Robustness of Quality Measures for GANs. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds) Computer Vision – ECCV 2022. ECCV 2022. Lecture Notes in Computer Science, vol 13677. Springer, Cham. https://doi.org/10.1007/978-3-031-19790-1_2

[8] Kokosi, Theodora; Harron, Katie; (2022) Synthetic data in medical research. **BMJ Medicine,** 1 (1), Article e000167. 10.1136/bmjmed-2022-000167.

[9] K. El Emam, "Seven Ways to Evaluate the Utility of Synthetic Data," in IEEE Security & Privacy, vol. 18, no. 4, pp. 56-59, July-Aug. 2020, doi: 10.1109/MSEC.2020.2992821.

[10] K. M. Habibullah, G. Gay, and J. Horkoff, "Non-functional requirements for machine learning," *Proceedings of the 1st Workshop on Software Engineering for Responsible AI*, 2022.

[11] "Agile testing methodology - methods, Principles & Advantages," *ReQtest*, 07-Sep-2020. [Online]. Available: https://reqtest.com/testing-blog/agile-testing-principles-methods-advantages/. [Accessed: 08-Feb-2023].

[12] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying CRISP-DM process model," *Procedia Computer Science*, vol. 181, pp. 526–534, 2021.

[13] N. W. Grady, J. A. Payne, and H. Parker, "Agile Big Data Analytics: AnalyticsOps for Data Science," *2017 IEEE International Conference on Big Data (Big Data)*, 2017.

[14] Shearer, C. The CRISP-DM Model: The New Blueprint for Data Mining. Journal of Data Warehousing, 5, 13-22, 2000.

[15] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements Engineering and Agile Software Development," *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*

[16] Everything you need to know about Jira [online] https://jexo.io/blog/beginners-guide-to-jira/ [Accessed: 06 February 2023]

[17] Investopedia, 2022, "Gantt Charting: Definition, Benefits, and How They're Used" [online], https://www.investopedia.com/terms/g/gantt-chart.asp [Accessed: 04 February 2023]