# Semantically Meaningful Clustering of Cultural Heritage Sites

Leah Gluckman
University of Cape Town
Cape Town, South Africa
glclea001@myuct.ac.za

Jared May
University of Cape Town
Cape Town, South Africa
myxjar002@myuct.ac.za

Jemma Sundelson
University of Cape Town
Cape Town, South Africa
sndjem001@myuct.ac.za

## CCS Concepts

• **Computing methodologies → Machine learning approaches**; **Feature selection**; **Cluster analysis**; **Machine learning approaches**; **Classification and regression trees**.

## Keywords

Point Clouds, Machine Learning, Classification, Clustering, Semantic Segmentation, Feature Extraction

## 1 Project Description

The preservation of Cultural Heritage (CH) sites is a vital means of transmitting a testimony of past human activity to future generations. The Visual Computing Group (VCG) at CNR-ISTI [4] and the Zamani Project at UCT [18] explore digital technologies to construct 3D models from laser scanners. These models can be used for CH preservation since preservationists can decipher where the site is deteriorating and where it needs improvement. However, the process of acquiring these models is not simple. Terrestrial Laser Scanning (TLS) scans CH sites, resulting in scans that can produce 3D point clouds. These scans contain both wanted and unwanted data, such as foliage, scaffolding, people, and animals that do not belong to the CH site and which one must remove. This removal process is called point cloud cleaning – a mainly manual, labour-intensive, and time-consuming process. There is a need to automate this process and make it more efficient.

Classification, also known as Semantic Segmentation, involves assigning points in the dataset to classes with meaningful labels that can be understood and interpreted by humans. A class label is a category such as ground plane, tree or wall. Classification is a supplementary interest of our project and will only occur after we have clustered the dataset into semantically meaningful *clusters*.

The project aims to investigate the decomposition of point clouds into semantically meaningful *clusters* of 3D points through point feature clustering. Ideally, such clusters will correspond to point collections that make sense to a human, such as all the points sampling a tree in the 3D scene. Accomplishing this step could significantly simplify and accelerate the arduous task of point cloud cleaning. In addition, it aims to assess whether or not clustering can augment the classification process. The resulting clusters will be combined, using the majority ground truth label of points per cluster, to form groupings that can be classified using binary classification labels of 'keep' and 'discard'.

3D point clouds are a datatype comprised of a set of points, each defined by three spatial coordinates $(X, Y, Z)$. Points may include additional features — basic descriptors — such as intensity and RGB colour, which describe useful characteristics of points or point neighbourhoods. Some datasets include extra features, but these differ among datasets. Thus, one can only reliably use spatial coordinates (raw 3D coordinates) when devising a general means to analyse 3D point clouds. However, basic features provide a poor input to algorithms that try to find patterns within the data (including clustering) and therefore yield unsatisfactory results. The extraction of relevant, descriptive features at single or multiple-scales [26] should minimise computational complexity and improve the clustering of our dataset.

Feature extraction is the extraction of rich descriptors — features — from a point cloud. We will implement feature extraction and use two methods to select two rich feature sets: a set of human-interpretable features and a set of non-interpretable features derived from PointNet++. CloudCompare, a 3D point cloud processing software, will be used to add up to 22 human-interpretable features such as covariance, geometric and height features. PointNet++ is a neural network architecture for processing point clouds and is effective in learning hierarchical features [16]. We will use it to extract over 1000 human-uninterpretable features from our dataset and add these features to the vector describing each point. The data sets containing a higher dimensional feature space will be fed into our clustering algorithms, forming more clearly distinct points. The additional features will increase the chance of finding 'similar' points to cluster into semantically meaningful classes.

This project will investigate the application of Machine Learning Clustering methods to assist the semantic segmentation of 3D point clouds. It aims to produce several human-interpretable clusters of points and determine whether those clusters or combinations thereof can assist with more complex structure classification. This process will materialise through three complementary approaches. Each team member will implement and evaluate three different clustering algorithms. We will then use the formed clusters to implement binary classification such as 'keep' and 'discard'. We plan to implement unsupervised clustering approaches to accurately and efficiently partition point clouds of CH sites into semantically meaningful classes, whilst discarding noise. This pipeline includes stages of data acquisition, feature extraction, classification, visualisation, and analysis. The final analysis phase will comprise a substantial component of this project and is where we will evaluate the quality and performance of our implemented clustering algorithms.

## 2 Problem Statement and Aims

### 2.1 Research Problem

The process of constructing 3D models from laser scans of CH sites is a critical but challenging area of research for the preservation of CH. Such models are required to record and visualise the three-dimensional structure of historical sites. TLS laser scans of CH sites are imperfect scans that contain noise, unwanted objects and varying point densities. Point cloud cleaning involves removing these imperfections to enable the construction of accurate 3D models with limited to no extraneous information. There is a need to automate this mainly manual and laborious classification process and make it more efficient. Point Cloud Classification aims to classify 3D points in laser scans. It has been applied as one method to automatically classify points as relevant or not in the context of point cloud cleaning by Marais et al. [12]. However, point clouds, particularly those of CH sites, have limited labelled training data, a requirement of supervised learning. In addition, point clouds of CH scans often comprise large datasets that contain a significant amount of 3D point data. Datasets also vary, each comprising a different set of objects, ruins, and buildings. CH site scans can range from well-preserved structures to eroding and highly irregular ruins. The variance in CH data makes a general classification model hard to construct. It is the main reason a single model cannot be used across all sites naively, as is possible in urban environments [5, 11]. It also makes the labelling process very complex and the classification process difficult.

### 2.2 Aims

This project explores the application of several clustering algorithms to 3D point clouds of CH sites. Clustering can create numerous context-based and variance-resistant groupings from the data. The primary aim is to implement, analyse and compare the performance of several clustering algorithms that can accurately and efficiently produce semantically meaningful clusters for a point cloud. A secondary aim is to use the created clusters as a means to implement binary classification labelling that labels each or groups of clusters as being part of the CH site ('keep') or erroneous ('discard').

### 2.3 Research Questions

The overarching research question of the project is: *do clustering methods produce clusters on point clouds of CH sites that are more semantically meaningful than those produced by a baseline k-means algorithm, using cluster validity metrics of Silhouette Coefficient and DB Index?* Each sub-project explores this question through several different classes of clustering algorithms. Namely, partitional-based, density-based and hierarchical-based clustering algorithms. An additional research questions is: *"what is the upper bound accuracy when using the resulting clusters to perform binary classification using 'keep' and 'discard' labels on a point cloud?"*

## 3 Related Work

Researchers explore Point Cloud Classification through various approaches that aim to classify 3D points in laser scans as either relevant or not. Marais et al. [12] implemented a Random Forest classifier to perform point cloud classification using an intuitive binary point labelling system of 'keep' and 'discard' labels. Their classifier achieved a 98% average accuracy and successfully predicted the class labels of points in the point cloud dataset. Whilst this is a supervised classification approach, we aim to incorporate their binary labelling approach to classify our resulting clusters. In addition, we strive to achieve a similar accuracy using an unsupervised clustering approach.

Shan and Sampath [20] implement a k-means algorithm to find points belonging to a specific roof segment. The algorithm successfully clusters the point clouds, despite the presence of outliers. Each team member will implement k-means clustering as a baseline algorithm. In addition, Sotoodeh [21] investigates the detection of outliers on TLS point clouds using a hierarchical algorithm. They yield satisfactory results in detecting single and even clustered outliers, but user editing is still required. The algorithm provides an easier editing procedure due to the predominantly successful clustering of points. This shows that hierarchical clustering can speed up and simplify point cloud cleaning. We will use it as one team member's clustering approach. Aljumaily et al. [2] apply the DBSCAN clustering algorithm to point clouds and achieve meaningful and relevant clusters, despite the high-dimensional nature of and substantial density discrepancies within the dataset. This suggests that density-based clustering algorithms should succeed in partitioning the CH point clouds into semantically meaningful classes. Similarly, Tonini and Abellan [25] effectively classify feature points within a dataset with DBSCAN and identify arbitrarily shaped clusters.

Weinmann et al. [26] summarises approaches to selecting meaningful features for classifying point clouds. The selection and extraction of *relevant* features should improve the quality of clusters, as relevant features should be better for training than random or oversized feature sets. We will extract a select set of human-interpretable features and cluster our dataset using them.

## 4 Procedures and Methods

### 4.1 Development platform

Our development platform is described by the table below:

| | |
|---|---|
| **Programming Language** | Python 3 |
| **Integrated Development Environment (IDE)** | Visual Studio Code (VS Code) |
| **Version Control (VCS)** | Gitlab (UCT) and Github for Redundancy |
| **3rd Party Libraries** | Python libraries, including Scikit-learn, PyTorch, Open3D and others that are listed later. |

**Table 1: Development Platform Summary**

### 4.2 Implementation Strategy

Our implementation strategy follows a 5 stage pipeline:
Data Acquisition, Feature Extraction, Clustering, and Classification.

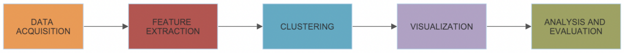
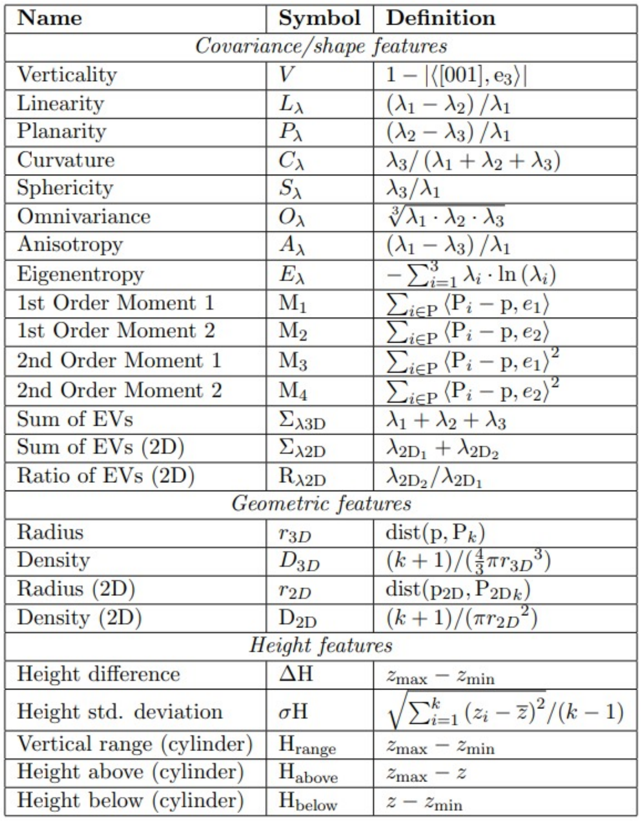


**Figure 1: Graphical Depiction of the the project's Implementation Pipeline for Point Cloud Classification using Clustering**

*4.2.1* ***Data Acquisition*** Our dataset will be 3D point clouds from TLS campaigns conducted by the VCG at CNR-ISTI [4] and the Zamani Project at UCT [18]. The recordings from TLS scanning - 3D point clouds - are used as input in our processing pipeline and at a later stage to generate 3D models. The procedure is as follows:

(1) Retrieve provided RAW or Registered or Ground Truth Point Clouds from the VCG at CNR-ISTI and the Zamani Project at UCT.
(2) Load the file in Python3 using the Python libraries asPy and NumPy.
(3) Store the file in a NumPy binary format.

*4.2.2* ***Feature Extraction*** Features are a vital input for the clustering stage. A baseline and two rich feature sets form the input of our clustering algorithms.



| Name | Symbol | Definition |
|---|---|---|
| *Covariance/shape features* | | |
| Verticality | $V$ | $1 - \lvert\langle[001], e_3\rangle\rvert$ |
| Linearity | $L_\lambda$ | $(\lambda_1 - \lambda_2)/\lambda_1$ |
| Planarity | $P_\lambda$ | $(\lambda_2 - \lambda_3)/\lambda_1$ |
| Curvature | $C_\lambda$ | $\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$ |
| Sphericity | $S_\lambda$ | $\lambda_3/\lambda_1$ |
| Omnivariance | $O_\lambda$ | $\sqrt[3]{\lambda_1 \cdot \lambda_2 \cdot \lambda_3}$ |
| Anisotropy | $A_\lambda$ | $(\lambda_1 - \lambda_3)/\lambda_1$ |
| Eigenentropy | $E_\lambda$ | $-\sum_{i=1}^{3} \lambda_i \cdot \ln(\lambda_i)$ |
| 1st Order Moment 1 | $\mathrm{M}_1$ | $\sum_{i\in \mathrm{P}} \langle \mathrm{P}_i - \mathrm{p}, e_1\rangle$ |
| 1st Order Moment 2 | $\mathrm{M}_2$ | $\sum_{i\in \mathrm{P}} \langle \mathrm{P}_i - \mathrm{p}, e_2\rangle$ |
| 2nd Order Moment 1 | $\mathrm{M}_3$ | $\sum_{i\in \mathrm{P}} \langle \mathrm{P}_i - \mathrm{p}, e_1\rangle^2$ |
| 2nd Order Moment 2 | $\mathrm{M}_4$ | $\sum_{i\in \mathrm{P}} \langle \mathrm{P}_i - \mathrm{p}, e_2\rangle^2$ |
| Sum of EVs | $\Sigma_{\lambda 3\mathrm{D}}$ | $\lambda_1 + \lambda_2 + \lambda_3$ |
| Sum of EVs (2D) | $\Sigma_{\lambda 2\mathrm{D}}$ | $\lambda_{2\mathrm{D}_1} + \lambda_{2\mathrm{D}_2}$ |
| Ratio of EVs (2D) | $\mathrm{R}_{\lambda 2\mathrm{D}}$ | $\lambda_{2\mathrm{D}_2}/\lambda_{2\mathrm{D}_1}$ |
| *Geometric features* | | |
| Radius | $r_{3D}$ | $\mathrm{dist}(\mathrm{p}, \mathrm{P}_k)$ |
| Density | $D_{3D}$ | $(k+1)/(\frac{4}{3}\pi {r_{3D}}^3)$ |
| Radius (2D) | $r_{2D}$ | $\mathrm{dist}(\mathrm{p}_{2\mathrm{D}}, \mathrm{P}_{2\mathrm{D}k})$ |
| Density (2D) | $\mathrm{D}_{2\mathrm{D}}$ | $(k+1)/(\pi {r_{2D}}^2)$ |
| *Height features* | | |
| Height difference | $\Delta\mathrm{H}$ | $z_{\mathrm{max}} - z_{\mathrm{min}}$ |
| Height std. deviation | $\sigma\mathrm{H}$ | $\sqrt{\sum_{i=1}^{k} (z_i - \overline{z})^2/(k-1)}$ |
| Vertical range (cylinder) | $\mathrm{H}_{\mathrm{range}}$ | $z_{\mathrm{max}} - z_{\mathrm{min}}$ |
| Height above (cylinder) | $\mathrm{H}_{\mathrm{above}}$ | $z_{\mathrm{max}} - z$ |
| Height below (cylinder) | $\mathrm{H}_{\mathrm{below}}$ | $z - z_{\mathrm{min}}$ |

**Figure 2: Table of Human-Interpretable Point Features [15]**

We will explore the application of three different feature sets in our project:

(1) Point coordinates: We will use basic point feature $(x, y, z)$ coordinates as the only feature in our feature set. This requires no feature extraction, only raw point clouds.
(2) Selected features: We will use CloudCompare to extract several useful human-interpretable features. We will decide on the number and set of useful features by following the point cloud classification strategy of Marais et al. [12].
(3) PointNet++ features: We will use PointNet++ [16] to generate a dataset with many human-incomprehensible features of the order 1000 (or another number through experimentation). PointNet++ utilises PointNet in sampled local regions and aggregates features using a hierarchical deep learning algorithm. It builds dynamic connections among points in their feature level and updates point features based on their neighbouring points in the feature space.

A table of features can be seen here.

*4.2.3* ***Clustering*** Clustering partitions a dataset into homogenous subgroups called clusters, such that each comprises similar points, thereby maximising inter-cluster similarity. In addition, distinct clusters contain unrelated points such that intra-cluster similarity is minimised [8].

*4.2.3.1* Clustering Implementation: We will use various clustering methods to partition the point clouds into semantically meaningful classes. There are several classes of clustering algorithms, and each team member will implement 3 methods from each class. This stage is where our work will begin to diverge. These methods include partitional-based clustering algorithms of k-means, k-medians, k-medoids and CLARANS, density-based clustering algorithms of DBSCAN, OPTICS and mean-shift, and hierarchical-based clustering algorithms of BIRCH, CURE and ROCK. The team member implementing partitional-based clustering algorithms will also implement the Gaussian Mixture Model method to ensure an even workload split. We will implement these algorithms using the Python library Scikit-learn.

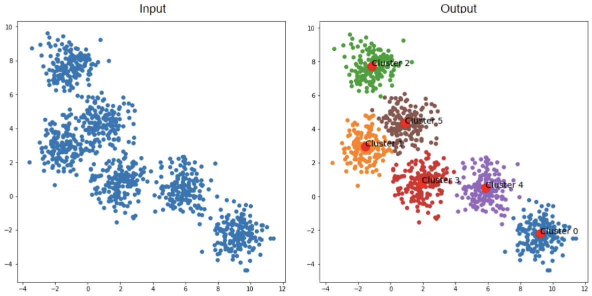


**Figure 3: Graphical Depiction of Clustering, illustrating how a set of points are clustered into distinct subgroups. Each cluster displays high intra-cluster similarity, and low inter-cluster similarity with other clusters [23]**

We will perform the different clustering methods on the three separate datasets generated in the feature extraction stage. The results of clustering these three datasets will be compared and critiqued in the pipeline's evaluation phase.

*4.2.3.2* Classification: The classification process is supplementary to the clustering task in our project. After clustering the data, we will combine the many resulting clusters into different groups to classify the data and detect outliers. The goal is to classify irrelevant clusters as noise or 'discard' and relevant clusters belonging to the CH site as 'keep'.

#### 4.2.4 *Visualisation*

The fourth step of the pipeline is visualisation, which includes visualising point clouds and their attributes [17]. We will use Open3D to perform visualisation. Open3D supports the rapid development of software that deals with 3D data. It renders point clouds together, allowing us to interpret the classification results [27].

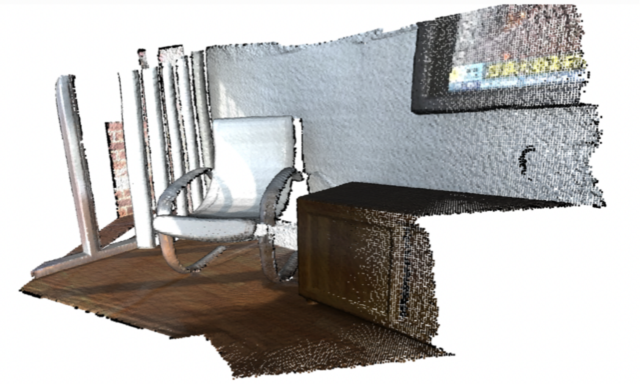


**Figure 4: Image of a point cloud rendered as surfels, depicting the output of visualising point cloud data in Open3D [22]**

#### 4.2.5 *Analysis and Evaluation*

Evaluation will form a large part of this project. The team members will work together to implement several algorithms that evaluate the results. Each team member will then analyse and compare their results independently. These metrics will quantify whether the implemented algorithms produced semantically meaningful clusters and determine the quality of the different clustering algorithms. There are two broad approaches to evaluating clustering algorithms: internal, unsupervised validation approaches and external, supervised validation approaches [13]. The following section describes both in detail.

## 5 Evaluation Metrics

We will use multiple evaluation metrics to determine the success or failure of our clustering algorithms. We will first test the clustering results using cluster validity metrics and then evaluate those clusters using classification performance metrics. We will use Sklearn.metrics in Python to report precision, accuracy, recall and F1 score evaluations, the Davies-Bouldin Index and Silhouette Coefficient.

### 5.1 Cluster Validity Metrics

A clustering algorithm's primary goal is to maximise intra-cluster similarity whilst minimising inter-clustering similarity. It is vital to test the validity of the created clusters to determine whether that goal holds.

#### 5.1.1 *Silhouette Coefficient*

Good clustering algorithms produce a high cohesion within and high separation between clusters [13]. The Silhouette Coefficient (SC) combines these two metrics to provide a simple qualification framework and computes a similarity metric using the distance between points or clusters [3]. One calculates it by combining the average distance between a point and every other point in the:

(1) same cluster
(2) nearest cluster

The SC ranges between -1 and 1. A positive value signifies highly dense, correct clusters and high separation between clusters [13]. A negative value suggests overlapping clusters, implying that several data points are assigned the incorrect class label. A zero value means data points are uniformly distributed throughout the space. The absolute values of this metric describe the quality of the clustering algorithms [1]. Thus, the SC can determine the algorithms' success rate, and whether they produced semantically meaningful clusters. This metric is also useful for visually portraying the similarities and differences within and between clusters.

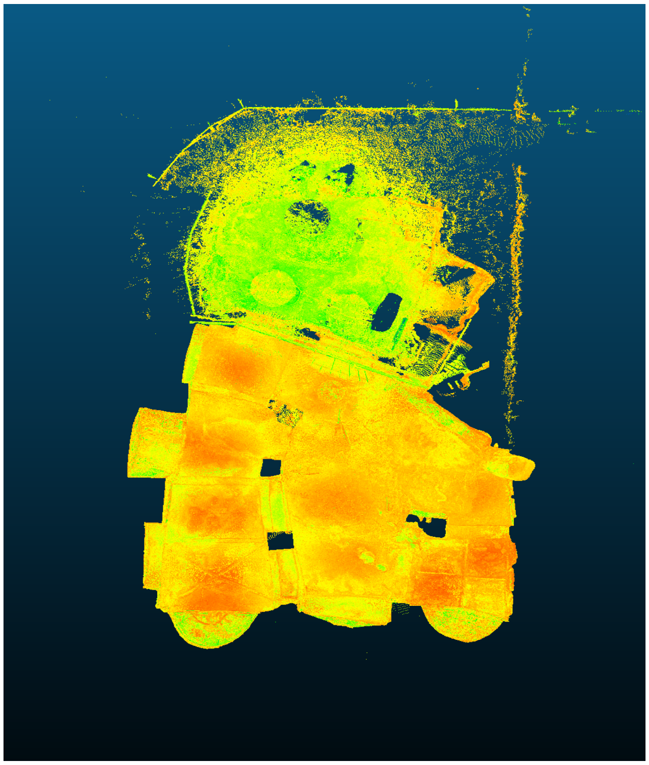


**Figure 5: Visualisation of a Church Point Cloud Dataset using CloudCompare (with keep/discard labelling)**

$$s = \frac{b - a}{max(a, b)}$$

*Silhouette Coefficient Formula*

#### 5.1.2 *Davies–Bouldin Index*

The David-Bouldin (DB) Index evaluates the density of data points in clusters and the degree of separation between different clusters. Very dense clusters that are well-spaced from others imply good clustering. The smaller the DB index, the better the clustering. This result is because small DB indexes indicate that clusters bear little to no similarity to one another, meaning that they are well-separated and compact [19].

**Table 2: Table depicting the truth table used for evaluating the classification performance of clustering algorithms. It illustrates an algorithm's classification result, showing that the predicted label does not always equate to the actual class.**

| | | **Actual Class** | |
|---|---|---|---|
| | | **Positive (P)** | **Negative (N)** |
| **Predicted Class** | **True (T)** | True Positive (TP) | False Negative (FN) |
| | **False (F)** | False Positive (FP) | True Negative (TN) |

## 5.2 Classification Performance Metrics

As mentioned in the pipeline explanation, classification will occur after completing the clustering task. Clustering point clouds of CH sites into semantically meaningful clusters is the main goal and classifying these groupings into 'keep' and 'discard' is a supplementary interest. We will estimate an upper bound on classification accuracy given the produced clusters by using the additional ground truth label data contained in the point cloud scans from VCG. We will classify each cluster with the majority ground truth class of points within. This will allow us to combine or separate the produced clusters to create the final classifications that we can evaluate with the following metrics. In addition, it would achieve the best possible accuracy, precision and recall that one can get from the clusters. Performance metrics determine how useful the produced clusters are for classifying points into binary classes. They describe the algorithm's ability to label points in a point cloud as part of the CH site or irrelevant and noise.

A truth table (see Table 2) illustrates the four possible classification performance outcomes of clustering algorithms that describe their precision, accuracy and recall. These outcomes comprise 'True Positive' (TP), 'True Negative' (TN), 'False Positive' (FP), and 'False Negative' (FN). TP is when the sample is positive and correctly classified as positive. FP is when the sample is negative, but the algorithm classifies it as positive. TN and FN behave similarly [24].

*5.2.1 Precision* The precision metric determines the number of TPs within the cluster over the total positive records [6]. It reflects the clustering algorithm's accuracy in classifying a point as positive. This metric is useful because it proves how reliably an algorithm can predict TPs. The more TP predictions an algorithm makes, the higher its precision percentage.

$$Precision = \frac{TP}{TP + FP}$$

*Precision Formula*

*5.2.2 Accuracy* The accuracy of a clustering algorithm is the number of correct predictions over the sum of all evaluated samples [6].

$$Accuracy = \frac{TP + TN}{TP + FN + FP + FN}$$

*Accuracy Formula*

Accuracy is beneficial because it determines how well an algorithm performs and how much it needs to improve. In addition, it is useful when classes are equally important, as is with point cloud classification of CH sites. A low accuracy score indicates that the clustering algorithm fails to cluster the point cloud into semantically meaningful classes. A high accuracy score signifies that a clustering algorithm successfully answers the project's research question.

*5.2.3 Recall* Recall determines the ratio of correctly classified points [24]. It reveals how well the algorithm can detect TP data points.

$$Recall = \frac{TP}{TP + FN}$$

*Recall Formula*

*5.2.4 F1 Score* The F1 score averages a model's performance by calculating the weighted average of precision and recall as a single metric. It is a number between 0 and 1, with scores closer to 1 being more desirable [10].

*5.2.5 Relative Speed* Relative speed determines which clustering algorithm completes in the shortest time. We will only consider if the results of the clustering algorithms are similar in more critical metrics like precision, recall, and accuracy.

*5.2.6 Intersection Over Union* The Intersection Over Union (IOU) score determines how well the model perfectly separates points belonging to CH sites and those that are noise. It may be worse (lower) when evaluating noisy data since clustering only the relevant data points becomes more difficult [7].

*5.2.7 Error Rate* The error rate is the ratio of incorrectly classified points over the total number of data samples. It determines a clustering algorithm's percentage of incorrect predictions.

## 5.3 Case Studies

The VCG at CNR-ISTI and the Zamani Project at UCT have several different campaign scans of CH sites that we can acquire, including a Church and an underground ruin. Each scan comprises a different set of objects with varying features. We can evaluate the clustering algorithms' results on more than one campaign scan and compare the results of each. This comparison would allow us to understand what the algorithm succeeds at clustering and where it needs improvement.

## 5.4 Visual Inspection

The human eye is adept at detecting disparities in visual data. Thus, we will use visual inspection of the resulting clusters as a heuristic for the quality of clustering approaches.

# 6 Ethical, Professional and Legal Issues

This project has no ethical issues since it does not involve human or animal subjects. The VCG at CNR-ISTI and the Zamani Project at UCT have granted us permission to use their point cloud data, provided it is not made public. This project has no legal concerns since PointNet++, and all the libraries we will use are open-source software. We will cite referenced works where necessary. Team members will abide by the Open Source Software guidelines [9] to guarantee that no professional issues arise and that the open-source libraries have ethical and professional use in the project.

# 7 Anticipated Outcomes

## 7.1 Expected Impact

This project is mostly experimental but could have useful practical applications in CH preservation, and interesting results in the general field of ML/DL and point cloud analysis. In terms of human impact, this can reduce the need for manual and labour-intensive cleaning.

## 7.2 Key Success Factors

The project's success will be developing a point cloud analysis system that evaluates the various clustering methods. Comparable findings and evaluation metrics will judge whether or not we successfully classify the point clouds into semantically meaningful clusters. Specifically, the SC and DB Index will determine the quality of the produced clusters. The goal is to have clusters with high-intracluster similarity and low inter-cluster similarity. A positive-valued SC discerns that the clusters are correct, well-separated, and highly dense and that the similarity measures succeed. This SC and a small DB index convey that we produced semantically meaningful clusters. A high precision, accuracy and recall percentage and an F1 score close to or equal to 1 determine the success of the binary classification labelling. In addition, consistency among clusters is a crucial success factor. This consistency is when clustering results are unaffected by increases or decreases in the distances between or within clusters, respectively [14]. Visual inspection can determine other success factors. The goal is to achieve clusters that appear convincing to the human eye. For instance, clusters that do not cross object boundaries are desirable, whereas those that merge with others are not.

# 8 Project Plan

Applying unsupervised clustering approaches to the problem of point cloud classification (in CH) removes the labelled training data requirement that limits supervised approaches to the problem. We expect some of the clustering algorithms we implement to produce semantically meaningful clusters on CH point clouds. The intention is to compare methods to determine which has the most success in assigning human-interpretable labels. We will perform a robust evaluation of the algorithms' effectiveness across several datasets.

## 8.1 Expected Challenges

*8.1.1 Lack of Knowledge and Experience:* This is the most foreseeable challenge, with the various algorithms and software required for this project and the consequential steep learning curve. This includes our understanding of the different clustering algorithms, evaluation metrics, and pipeline implementation. Another challenge is learning to analyse and evaluate the data throughout the pipeline, including difficulty interpreting the produced clusters and the metrics.

*8.1.2 Data Acquisition and Utilisation:* Challenges include the acquisition of very noisy data, failure to extract useful features, and clustering algorithms' inability to appropriately cluster the data.

## 8.2 Risks and Risk Management Strategies

We identified several project risks and risk management strategies these are described in the Appendix.

## 8.3 Timeline, Milestones, and Deliverables

A Gantt Chart in the Appendix includes the project timeline, milestones and deliverables. The due dates of each deliverable are stated in Table 2. Notable milestones include:

- Set up all the software
- Completed Feature Extraction using human-interpretable features generated with CloudCompare
- Completed Feature Extraction using human-uninterpretable features generated with PointNet++
- Completed Implementation of Clustering Methods
- Completed Visualisation Stage
- Implemented Clustering Validity Metrics
- Implemented Classification Performance Evaluation Metrics
- Completed Analysis and Evaluation of Results
- Completed Final Paper Draft
- Completed Final Paper

**Table 3: Project Deliverables and Due Dates**

| Deliverable | Due Date |
|---|---|
| Literature Review | 4 May |
| Proposal Presentations | 25 May |
| Final Project Proposal | 15 July |
| Initial Prototype | 25–29 July |
| Final Paper Draft | 23 August |
| Final Paper | 2 September |
| Final Code | 5 September |
| Final Demonstration | 19– 23 September |
| Project Poster | 3 October |
| Project Website | 10 October |
| Showcase | TBC |

## 8.4 Resources Required

- Scikit-learn [1] (Clustering Algorithms)
- Point Cloud Data from the Visual Computing Group at CNR-ISTI
- Point Cloud Data from the Zamani Project at UCT
- PointNet++ (Feature Extractions)
- Open3D [2] (Visualisation)
- PyTorch [3] to implement PointNet++
- NumPy to store point cloud data in Python
- Python3 to implement project
- Sklearn.metrics (Evaluation Metrics) [4]

[1] Scikit-learn is a free software machine learning library for the Python programming language.

[2] Open3D is an open-source library that supports rapid development of software that deals with 3D data

[3] PyTorch is an open source machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing

[4] The sklearn.metrics module implements several loss, score, and utility functions to measure classification performance

## 8.5 Software Setup

We need to ensure the data pre-processing step (feature extraction) is accurate to enable us to feed that data into the various clustering algorithms. In addition, we must implement each clustering algorithm correctly for a fair, robust evaluation. We will acquire equivalent datasets to ensure consistency amongst clustering methods.

## 8.6 Work Allocation to Each Team Member

Team members will work collaboratively on developing the front-end and back-end of the project. This includes data acquisition, feature extraction implementation, visualisation, and implementing of the metric evaluation algorithms. To maximise group productivity, we will split these sections such that we all can work in parallel. For instance, each team member can implement a subset of the evaluation metrics and share the implementations with the other team members. We will also divide the feature extraction component so various feature extraction approaches can complete simultaneously.

The work diverges in the pipeline's clustering stage. Each team member will implement the k-means partitional-based clustering algorithm and use the results as a baseline for comparison. In addition, each team member will implement three methods belonging to a class of clustering algorithms and aim to cluster the dataset into semantically meaningful clusters.

**Jared:** Will implement three density-based clustering algorithms (DBSCAN, OPTICS and Mean-Shift).

**Leah:** Will implement three hierarchical-based clustering algorithms (BIRCH, CURE, ROCK).

**Jemma:** Will implement a combination of partitional-based clustering algorithms (CLARANS, K-medians, k-medoids) and an algorithm from a different clustering class (Gaussian Mixture Model).

Each team member will then analyse their results and write their final paper individually.

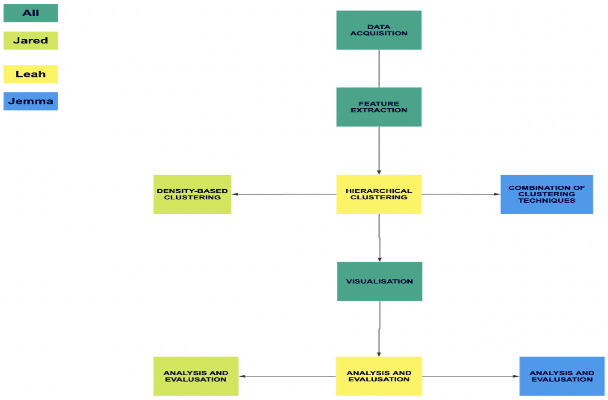


**Figure 6: Flow Diagram depicting the work allocation to each team member, including individual and collaborative sections**

## References

[1] Charu C. Aggarwal. 2015. (1 ed.). Springer, New York. https://doi.org/10.1007/978-3-319-14142-8

[2] Harith Aljumaily, Debra F Laefer, and Dolores Cuadra. 2017. Urban Point Cloud Mining Based on Density Clustering and MapReduce. *Journal of Computing in Civil Engineering* 31, 5 (2017), 04017021. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000674

[3] Hélio Almeida, Dorgival Guedes, Wagner Meira, and Mohammed J. Zaki. 2011. Is There a Best Quality Metric for Graph Clusters?. In *Machine Learning and Knowledge Discovery in Databases*, Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 44–59.

[4] Paolo Brivio. 2022. *Visual Computing Lab*. Visual Computing Lab. Retrieved May 22,2022 from http://vcg.isti.cnr.it/

[5] Nesrine Chehata, Li Guo, and Clément Mallet. 2009. Airborne Lidar Feature Selection for Urban Classification Using Random Forests. In *Laserscanning*. Hal, Paris, France, 207–212. https://hal.archives-ouvertes.fr/hal-02384719

[6] César Ferri, José Hernández-Orallo, and R Modroiu. 2009. An experimental comparison of performance measures for classification. *Pattern recognition letters* 30, 1 (2009), 27–38.

[7] Ahmad Gamal, Ari Wibisono, Satrio Bagus Wicaksono, Muhammad Alvin Abyan, Nur Hamid, Hanif Arif Wisesa, Wisnu Jatmiko, and Ronny Ardhianto. 2020. Automatic LIDAR building segmentation based on DGCNN and euclidean clustering. *Journal of Big Data* 7, 1 (2020), 1–18.

[8] Eleonora Grilli and Fabio Remondino. 2019. Classification of 3D Digital Heritage. *Remote Sensing* 11, 7 (2019), 1–23. https://www.mdpi.com/2072-4292/11/7/847

[9] Frances S Grodzinsky, Keith Miller, and Marty J Wolf. 2003. Ethical issues in Open Source Software. *Journal of Information, Communication and Ethics in Society* 1 (2003), 193–205. Issue 4. https://doi.org/10.1108/14779960380000235

[10] Yasen Jiao and Pufeng Du. 2016. Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quantitative Biology* 4, 4 (2016), 320–330.

[11] Clément Mallet, Frédéric Bretar, Michel Roux, Uwe Soergel, and Christian Heipke. 2011. Relevance assessment of full-waveform lidar data for urban area classification. *ISPRS Journal of Photogrammetry and Remote Sensing* 66, 6, Supplement (2011), S71–S84. https://doi.org/10.1016/j.isprsjprs.2011.09.008 Advances in LIDAR Data Processing and Applications.

[12] Patrick Marais, Matteo Dellepiane, Paolo Cignoni, and Roberto Scopigno. 2019. Semi-automated Cleaning of Laser Scanning Campaigns with Machine Learning. *ACM Journal on Computing and Cultural Heritage* 12, 3 (2019), 1–29. https://doi.org/10.1145/3292027

[13] Julio-Omar Palacio-Niño and Fernando Berzal. 2019. Evaluation Metrics for Unsupervised Learning Algorithms. *CoRR* abs/1905.05667 (2019), 1–9. arXiv:1905.05667 http://arxiv.org/abs/1905.05667

[14] Julio-Omar Palacio-Niño and Fernando Berzal. 2019. Evaluation Metrics for Unsupervised Learning Algorithms. https://doi.org/10.48550/ARXIV.1905.05667

[15] Christopher Pocock. 2019. *3D Scan Campaign Classification with Representative Training Scan Selection*. Master's thesis. Faculty of Science, Department of Computer Science.

[16] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* 30 (2017), 1–10.

[17] Rico Richter and Jürgen Döllner. 2014. Concepts and techniques for integration, analysis and visualization of massive 3D point clouds. *Computers, Environment and Urban Systems* 45 (2014), 114–124. https://doi.org/10.1016/j.compenvurbsys.2013.07.004

[18] Heinz Rüther, Christoph Held, Roshan Bhurtha, Ralph Schroeder, and Stephen Wessels. 2012. From Point Cloud to Textured Model, the Zamani Laser Scanning Pipeline in Heritage Documentation. *South African Journal of Geomatics* 1, 1 (Jan. 2012), 44–59.

[19] Sandro Saitta, Benny Raphael, and Ian F. C. Smith. 2007. A Bounded Index for Cluster Validity. In *Machine Learning and Data Mining in Pattern Recognition*, Petra Perner (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 174–187.

[20] Jie Shan and Aparajithan Sampath. 2008. Building extraction from LiDAR point clouds based on clustering techniques. In *Topographic laser ranging and scanning: principles and processing*. CRC press, Boca Raton, FL, 421–444.

[21] Soheil Sotoodeh. 2007. Hierarchical clustered outlier detection in laser scanner point clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 36, 3/W52 (2007), 383–388.

[22] Editorial Team. 2018–2021. *Point Cloud*. Open3D. http://www.open3d.org/docs/release/tutorial/geometry/pointcloud.html

[23] Editorial Team. 2019–2022. *Centroid Neural Network: An Efficient and Stable Clustering Algorithm*. Towards AI. https://towardsai.net/p/l/centroid-neural-network-an-efficient-and-stable-clustering-algorithm

[24] Alaa Tharwat. 2021. Classification assessment methods. *Applied Computing and Informatics* 17 (2021), 168–192. Issue 1.

[25] Marj Tonini and Antonio Abellan. 2014. Rockfall detection from terrestrial LiDAR point clouds: A clustering approach using R. *Journal of Spatial Information Science* 1, 8 (2014), 95–110.

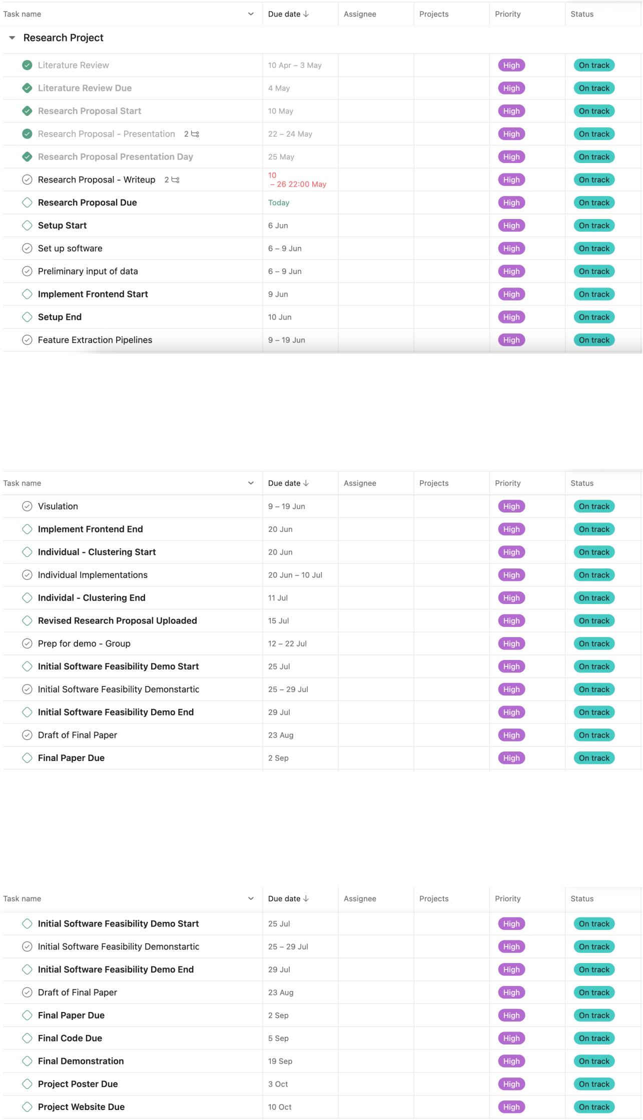
[26] Martin Weinmann, Boris Jutzi, Stefan Hinz, and Clément Mallet. 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing* 105 (2015), 286–304.

[27] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. 2018. Open3D: A Modern Library for 3D Data Processing. *CoRR* abs/1801.09847 (2018), 1–6. arXiv:1801.09847 http://arxiv.org/abs/1801.09847

# A Appendix

## A.1 Project Plan

Our Project Plan (Gantt) and Work Breakdown Schedule (WBS) are outlined below:

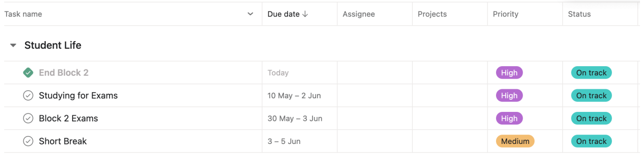


| Task name | Due date ↓ | Assignee | Projects | Priority | Status |
|---|---|---|---|---|---|
| Research Project | | | | | |
| Literature Review | 10 Apr – 3 May | | | High | On track |
| Literature Review Due | 4 May | | | High | On track |
| Research Proposal Start | 10 May | | | High | On track |
| Research Proposal - Presentation 2 | 22 – 24 May | | | High | On track |
| Research Proposal Presentation Day | 25 May | | | High | On track |
| Research Proposal - Writeup 2 | 10 – 26 22:00 May | | | High | On track |
| Research Proposal Due | Today | | | High | On track |
| Setup Start | 6 Jun | | | High | On track |
| Set up software | 6 – 9 Jun | | | High | On track |
| Preliminary input of data | 6 – 9 Jun | | | High | On track |
| Implement Frontend Start | 9 Jun | | | High | On track |
| Setup End | 10 Jun | | | High | On track |
| Feature Extraction Pipelines | 9 – 19 Jun | | | High | On track |

| Task name | Due date ↓ | Assignee | Projects | Priority | Status |
|---|---|---|---|---|---|
| Visulation | 9 – 19 Jun | | | High | On track |
| Implement Frontend End | 20 Jun | | | High | On track |
| Individual - Clustering Start | 20 Jun | | | High | On track |
| Individual Implementations | 20 Jun – 10 Jul | | | High | On track |
| Individal - Clustering End | 11 Jul | | | High | On track |
| Revised Research Proposal Uploaded | 15 Jul | | | High | On track |
| Prep for demo - Group | 12 – 22 Jul | | | High | On track |
| Initial Software Feasibility Demo Start | 25 Jul | | | High | On track |
| Initial Software Feasibility Demonstartic | 25 – 29 Jul | | | High | On track |
| Initial Software Feasibility Demo End | 29 Jul | | | High | On track |
| Draft of Final Paper | 23 Aug | | | High | On track |
| Final Paper Due | 2 Sep | | | High | On track |

| Task name | Due date ↓ | Assignee | Projects | Priority | Status |
|---|---|---|---|---|---|
| Initial Software Feasibility Demo Start | 25 Jul | | | High | On track |
| Initial Software Feasibility Demonstartic | 25 – 29 Jul | | | High | On track |
| Initial Software Feasibility Demo End | 29 Jul | | | High | On track |
| Draft of Final Paper | 23 Aug | | | High | On track |
| Final Paper Due | 2 Sep | | | High | On track |
| Final Code Due | 5 Sep | | | High | On track |
| Final Demonstration | 19 Sep | | | High | On track |
| Project Poster Due | 3 Oct | | | High | On track |
| Project Website Due | 10 Oct | | | High | On track |

Figure 7: Project - Work Breakdown Schedule



| Task name | Due date ↓ | Assignee | Projects | Priority | Status |
|---|---|---|---|---|---|
| Student Life | | | | | |
| End Block 2 | Today | | | High | On track |
| Studying for Exams | 10 May – 2 Jun | | | High | On track |
| Block 2 Exams | 30 May – 3 Jun | | | High | On track |
| Short Break | 3 – 5 Jun | | | Medium | On track |

Figure 8: Personal - Work Breakdown Schedule

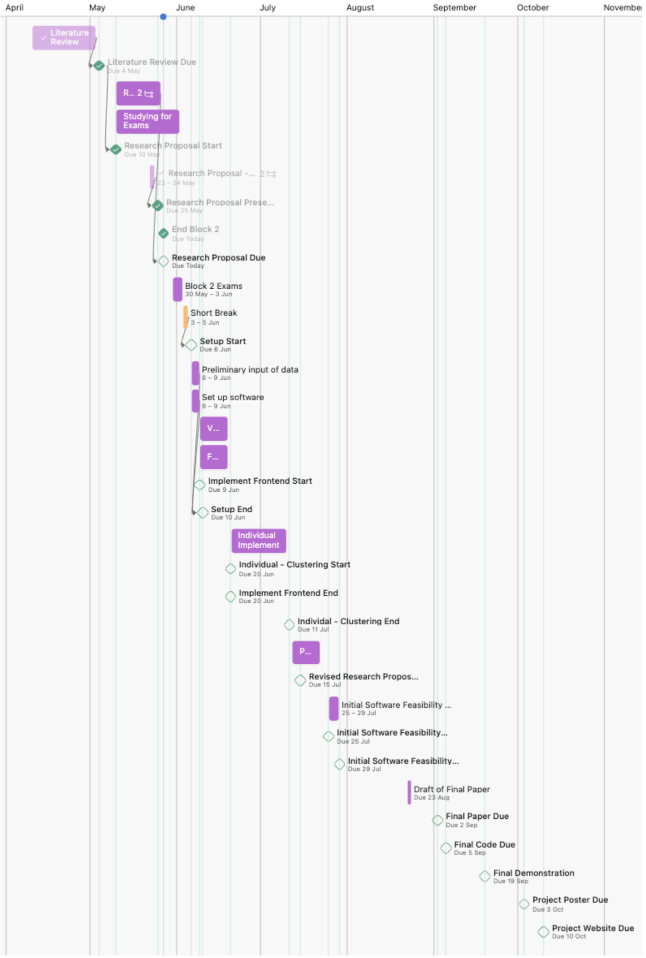


Figure 9: Gantt Chart of Years Work

## A.2 Risk Management

Our Risk Management strategy is laid out in the tables below. The first table is a key, that describes the Impact, Probability and Categories used for the Risk Matrix below.

**Table 4: Key: Risks and Risk Management**

| Risks and Risk Management Key | | |
|---|---|---|
| **Impact** | **Probability** | **Category** |
| Low | Low | Technical (Hardware/Software) |
| Medium | Medium | External (Extra-UCT) |
| High | High | Organisational (Intra-UCT) |
| Catastrophic | | |

**Table 5: Risk Matrix of the Project, stating the various risks and ways of mitigating, monitoring and managing those risks.**

| Risk No. | Risk Condition | Mitigation | Monitoring | Management |
|---|---|---|---|---|
| 1 | Protests at UCT shutdown campus operations | There is not much we can do. This is a factor outside the team's control | Check UCT announcements and follow the news | Have the ability to work remotely and not rely on campus amenities. Ensure we can communicate with the team and supervisor through MS Teams/Whatsapp. Use remote VCS like git or cloud backup like OneDrive |
| 2 | Deadlines are not met | Use shared team scheduling and project management tools like Trello, ClickUp or Miro, to track progress and goals. Create a shared Google Calendar, so all team deadlines sync to teams workstations. Follow an agile approach and have constant communication or stand-ups and check-ins so the team is aware of issues and deadlines. Include wiggle room to deadlines so there is an ability to reschedule and adjust goals if need be. | Regular effective communication between team members, including adherence to agile methodology like weekly/daily standup and consistent communication. Shared goal tracking and calendars. | Adjustment of deadlines/ goals if possible. Prioritisation if possible of late work. Alternatively, if future work is not dependent on missed/late outcome it can be left incomplete and consequence of that in terms of marking /results will be accepted. If possible, overtime work can be scheduled to catch up. |
| 3 | Lack of or misscommunication from supervisor and/or co-supervisor with the project team | Ensure constant, comfortable and clear communication. Spend time getting to know one another. Set up ground rules for effective communication. Relationship with ability to ask. Invite supervisors to standups/project management tools, etc. In meetings, minute and after note and discuss ambiguities. | Have supervisor and co-supervisor review work and plans. Include them in drafts and give them access to project management tools. Listen and note supervisor meetings. | Adjust goals based on advice from meetings and comments on drafts and plans. |
| 4 | Team member contracts COVID-19 and is unable to contribute as expected to collaborative aspects of the project whilst they are sick | Be able to adjust division of collaborative work. Divide work into independent sections that can be taken on by any member of the team. Ensure all team members understand each other's roles and work. | Regularly check-in with the health of all team members | Adjust the workload division of the sick team member between the rest of the team members. See what the sick team member can do. |
| 5 | Loss or corruption of source code and/or write-up | Use multiple backups and version control systems like Git and Gitlab. Use online tools for writeup like OverLeaf, but also backup copies to OneDrive and local machine. Always have multiple copies of work. | When a failed backup or corruption occurs, or when a source goes down, this will be noticed. | Switch to one of the many available sources. Asses losses, if any. Continue work. |

**Table 6: Risk Matrix Continued**

| Risk No. | Risk Condition | Mitigation | Monitoring | Management |
|---|---|---|---|---|
| 6 | Conflict between team members | Promote clear clear communication in team. Help each other and know when to ask for help. Let team members know when issues arise, and not when it is too late. If there is an issue that arises, deal with it then. | Have non-work check-ins with the team. Have supervisors conduct check-ins if possible. If animosity or resentment is brewing, it needs to be dealt with. | Mediation with other team members or supervisors. Readjustment or reassignment of work. Assessment of root cause and work on resolution of that. |
| 7 | Failure to understand some of the related content | Ask questions. Ask for help. Read more. Do related tutorials on Medium/YouTube/etc. Focus on other work and come back to the problem the second reading may shed more light on the misconception. | Mention when misconceptions arise. Take note of team members' struggles in meetings and in general and offer help. Supervisor may notice misunderstandings in drafts. | Get assistance from the team, supervisors or reading. Note impact of misconception. Correct for this. Adjust schedule and work accordingly. |
| 8 | Failure to implement or understand 3rd party libraries | Meet regularly to share difficulties and knowledge about the technology and consider preparing alternatives. Share code base and work on Git. Have a whiteboard or knowledge share sessions in the team or with supervisors. Do online tutorials. | At each meetings, check in with team members to see how well they are managing to work with the technology. Note struggles. | Get assistance from the team, supervisors or reading. Note impact of misconception. Correct for this. Adjust schedule and work accordingly. Switch library if need be. |

**Table 7: Table depicting the Consequence, Category, Probability and Impact associated with each risk in the Risk Matrix (see table 5 and 6). The risk number associates each risk in the Risk Matrix with the details in this table.**

| Risk Number | Consequence | Category | Probability | Impact |
|---|---|---|---|---|
| 1 | Unable to access UCT campus and amenities physically | Organisational | Medium | Low |
| 2 | Deadlines are missed and the project is put behind schedule. Additionally, there may be consequences like marking penalties or the scrapping of planned activities | Project Management/ Organisational | High | High |
| 3 | The team plans of executes aspects of the project incorrectly | Project Management/ Organisational | High | High |
| 4 | Loss of a team member/productivity for a period of time. Increased pressure on the rest of the team | Project Management/ External | Low | Low |
| 5 | Loss of work or corruption of source code and/or write-up | Project Management/ External | Low | Major |
| 6 | Issues within team and potential inability to complete project as expected | Project Management | Medium | Catastrophic |
| 7 | Inability to properly conduct or explain parts of the project | Internal/ Technical | High | Medium |
| 8 | Inability to implement or conduct experiments, benchmarks and analysis for project | Technical | Low | Medium |