**Report 2: Modelling and Design Implementation of Comparison**

**Implementation**

The models have been implemented in the Python language using libraries like scikit-learn for the traditional machine learning models and TensorFlow for neural networks.

**Decision Tree Classifier:**

The Decision Tree Classifier from scikit-learn was trained on SMOTE-balanced data using a random\_state of 42 to tackle class imbalances efficiently. This method guaranteed equal representation for all categories. The model was trained using balanced X\_train\_balanced data along with y\_train\_balanced\_encoded labels and then evaluated for accuracy on new data using the accuracy\_score function. A report on classification included metrics such as precision, recall, and f1-score, giving details on the performance of individual classes and the overall efficiency of the model.

**K-Means Clustering:**

The data is divided into seven clusters using the K-Means Clustering method in order to recognize inherent groupings. Trained on a balanced dataset with 10 centroid seeds starting points (n\_init=10) and a fixed random\_state=42 for consistency, the model ensures fair representation of all classes. The performance of clustering is assessed through the Adjusted Rand Index (ARI), where a higher ARI suggests better accuracy in clustering. The ARI that is produced, known as kmeans\_ari, acts as a crucial indicator of how well the model is clustering data.

**MLP Neural Network:**

The architecture consists of three layers, including a two-layer setup with 64 neurons each using ReLU activation, and a softmax output layer that corresponds to the number of classes. Utilized Adam optimizer and categorical crossentropy during training for 50 epochs with a batch size of 16.Results were assessed based on accuracy and visually examined using a confusion matrix to evaluate the performance of each class.

**CNN:**

The CNN model utilizes a Conv1D layer with 32 filters and a kernel size of 3 with ReLU activation, then a MaxPooling1D layer with a pool size of 2, a Flatten layer, and two Dense layers, with the last layer using softmax activation. It underwent training for 50 epochs with a batch size of 16 using the Adam optimizer, resulting in a test set accuracy of {cnn\_accuracy}. The performance across various categories is shown by the confusion matrix.

**Comparison**

Accuracy will be the main metric used for comparing the performance of the models after building them. In addition, confusion matrices give deeper meaning to the performance of classification: bar plot representations and confusion matrix visualizations can make the results transparent and draw the figure effectively.