**PROJECT REPORT - BINARY CLASSIFICATION OF BUS AND TRAM AUDIO SOUNDS**

Member 1: Nguyen The Long (151317891) – Member 2: Vu Dinh Thi (151394898)

## I. INTRODUCTION

Our report is used to document an overall process of the binary classification of bus and tram vehicles based on their audio sounds. The project focuses on extracting features from both audio sounds and choosing the suitable features for model building. From the features, a suitable model is selected, and the model is built using the data.

## II. DATA DESCRIPTION

Our dataset comprises audio recordings of both bus and tram sounds, captured from various urban environments available in Tampere. Additionally, online source sounds are also given to increase the variety of the model, marked to be also recorded in Tampere. The recordings are annotated and labeled accordingly, indicating whether the audio clip corresponds to a bus or a tram.

After going through all the data and choosing the most suitable data, the data set includes two classes: buses and trams. The “bus” class contains 50 different samples, while the “tram” class contains 69 different samples. All samples in both classes are recorded in multiple scenarios and places within Tampere using mobile phones. Locations include the bus/tram stop intersections, the bus only/tram only stops, on the bus and the tram. The records also include the crowd noise and specific sounds of each vehicle (such as tram announcements).

## III. FEATURE EXTRACTION

In this project, audio signal processing takes an important role in having features to classify the bus and the tram. Feature extraction involves transforming raw audio data into a set of representative features that capture essential information for subsequent analysis. The chosen features play a crucial role in defining the discriminative characteristics of the audio signals. From the audio signals, we define the most important features and consider them for extraction. The features include Mel-frequency cepstral coefficients (MFCC), Root Mean Square (RMS), Mel Spectrogram, Log Mel Spectrogram, Spectrogram, Constant-Q Transform (CQT), and energy of the signal.

MFCCs represent the short-term power spectrum of a sound and are derived from the mel-frequency cepstrum of the audio signal. Due to the evenly distributed Mel-frequency bands in MFCCs and their close resemblance to the human voice, MFCC proves to be an effective method for speaker characterization. Specifically, it can be employed to identify details about the speaker's cell phone model and provide additional insights into the speaker's characteristics. As a result, the feature is considered a potential feature for the model build.

RMS measures the average power of the signal by calculating the square root of the mean of the squared values of the signal. It provides information about the overall amplitude of the signal, giving insights into the signal's energy content.

Mel spectrograms are a visual representation of the short-term power spectrum of a signal, where the frequencies are converted to the mel scale. The spectrograms offer insights into the frequency distribution over time, emphasizing the perceptually relevant frequency bands.

Similarly, the logarithm mel spectrograms have the same characteristics but with the values transformed using the logarithm. Log transformation is applied to compress the dynamic range, making it suitable for machine learning tasks.

The spectrogram is a 2D representation of the spectrum of frequencies in a signal as they vary with time. It provides a detailed view of how the frequency content of a signal changes over time.

CQT is a frequency transform that uses a varying resolution across different frequency bands. Particularly useful for capturing both high and low-frequency components with varying resolution, which aligns with the nonlinear frequency resolution of human hearing.

Energy, or to be precise, the total energy present in the signal, represents the overall intensity of the signal, providing valuable information about its loudness and amplitude characteristics.

The choice of these features is considered because of their ability to capture diverse aspects of audio signals, such as spectral content, energy distribution, and perceptually relevant features. MFCCs, for instance, mimic human auditory perception, while spectrograms and their variants provide time-frequency representations essential for understanding temporal variations. The inclusion of features like RMS and energy further contributes to a comprehensive characterization of the audio signals.

Theoretically, all features can be included in the model for training. However, we would need to check the features to see if the features would be fit for the model, together with other features. We check the feature values visibly through plots of bus and tram audios.

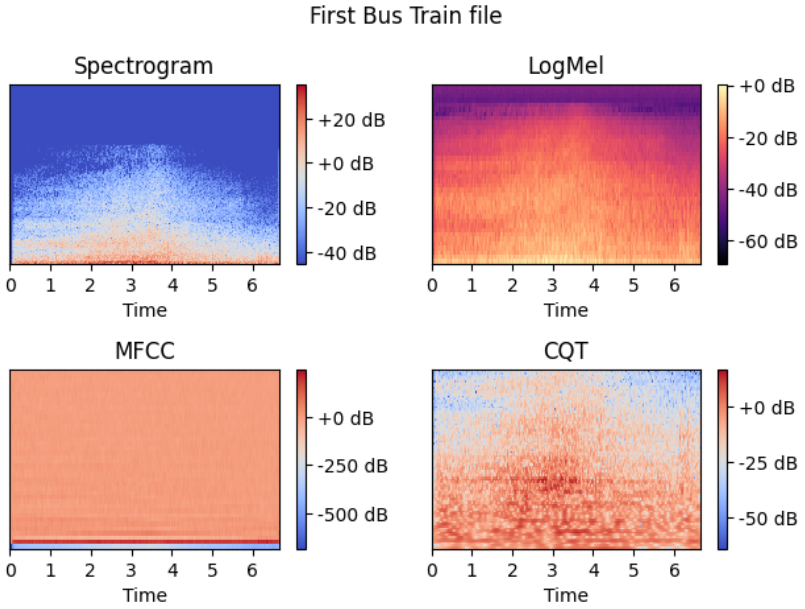


*Figure 1. Features values of an arbitrary bus audio file. The figure shows the features within the bus audio signal.*

For the bus audios, the features show clearly the changes and the effects within each feature. Again, the features can be seen as suitable for the model.

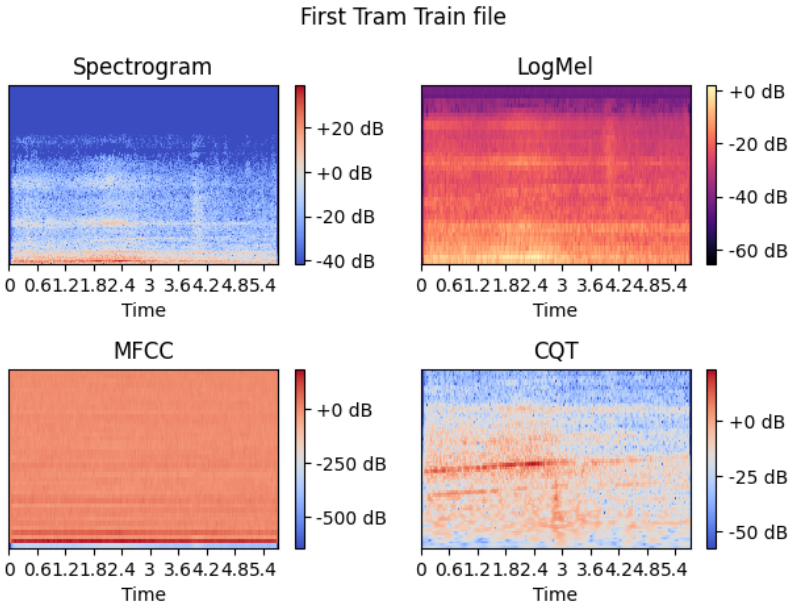


*Figure 2. Features values of an arbitrary bus audio file. The figure shows the features within the tram audio signal.*

From both figures, we can see that both figures have differences, and we can utilize that to distinguish audio signals between both classes. However, looking more carefully, we can see that the MFCC feature has the most significant distinction in most levels of frequencies. Therefore we will focus on having a higher percentage of inclusion for MFCC in the model. Mel

spectrogram is also having great potential, but it lacks diversity when the frequency becomes higher.

## IV. MODEL SELECTION

In this project, we decided to build two different models for the problem using two different types of models: k-Nearest Neighbour and Support Vector Machine (SVM), to consider for building the model. Both of these are useful machine learning algorithms.

In k-Nearest Neighbour, the prediction for a new data point is based on the majority class or average of its k-Nearest neighbors in the feature space. The choice of 'k' (the number of neighbors) is a crucial parameter that affects the model's performance. In our project model, we decided to choose k equal to 5 to see the result.

Support Vector Machine aims to find a hyperplane that best separates the data into different classes. It works well in high-dimensional spaces and is effective in cases where the data is not linearly separable by transforming it into a higher-dimensional space.

The model would be built based on those seven features above, with each of the features having a different percentage of inclusion. To decide the inclusion rate, we decided to build the models based on each feature and check its accuracy, precision, and recall.

For the data, we decided to divide the data into a ratio of 70:10:20 for training, validation, and testing, respectively. The validation dataset is used to evaluate the model during training. It helps in tuning hyperparameters and preventing overfitting. The model's performance on the validation set gives insights into how well it might generalize to new, unseen data. The data is divided randomly, but all of the sets must contain both self-recorded audio and online audio to ensure the diversity and integrity of the model.

## V. RESULTS

We build the models based on each feature and check their accuracy, precision, and recall. The results are shown in the table below.

| Feature | k-Nearest Neighbour | | | SVM | | |
|---|---|---|---|---|---|---|
| | Accuracy | Precision | Recall | Accuracy | Precision | Recall |
| MFCC | 91.30% | 91.67% | 92.31% | 86.96% | 88.46% | 88.46% |
| RMS | 69.56% | 73.33% | 66.15% | 69.56% | 73.33% | 66.15% |
| Mel Spectrogram | 69.56% | 73.33% | 66.15% | 65.21% | 64.58% | 63.46% |
| Log Mel Spectrogram | 39.13% | 35.83% | 36.92% | 34.78% | 22.22% | 30.77% |
| Spectrogram | 69.56% | 70.09% | 67.31% | 73.91% | 76.96% | 71.15% |
| CQT | 47.82% | 46.92% | 46.92% | 30.43% | 20.59% | 26.92% |
| Energy | 69.56% | 73.33% | 66.15% | 69.56% | 73.33% | 66.15% |

*Table 1. Results from the k-Nearest Neighbour and SVM models for each feature.*

From the table, we can see that MFCC features have a high rate of accuracy, precision, and recall. Thus it holds as an important feature in the model. The Log Mel Spectrogram and CQT show a poor result and, therefore hold a smaller percentage in the model. The k-Nearest Neighbour outperforms SVM slightly in accuracy, precision, and recall, yet those proved to have a similar result, yet two different kinds of models.

We also used the built models in k-Nearest Neighbour and SVM to check the test data to see the results. The models worked really great with the given data and no faults are found during runtime. The results returned are positive and shown in the figure below.

Both models show that the MFCC feature is highly recommended to be used for building the models. However, the k-Nearest Neighbour model seems to have more balanced results, with the logarithm mel and CQT having a decent rate of correct classification. However, the SVM model shows an impression of which features are more important, with major differences between them.

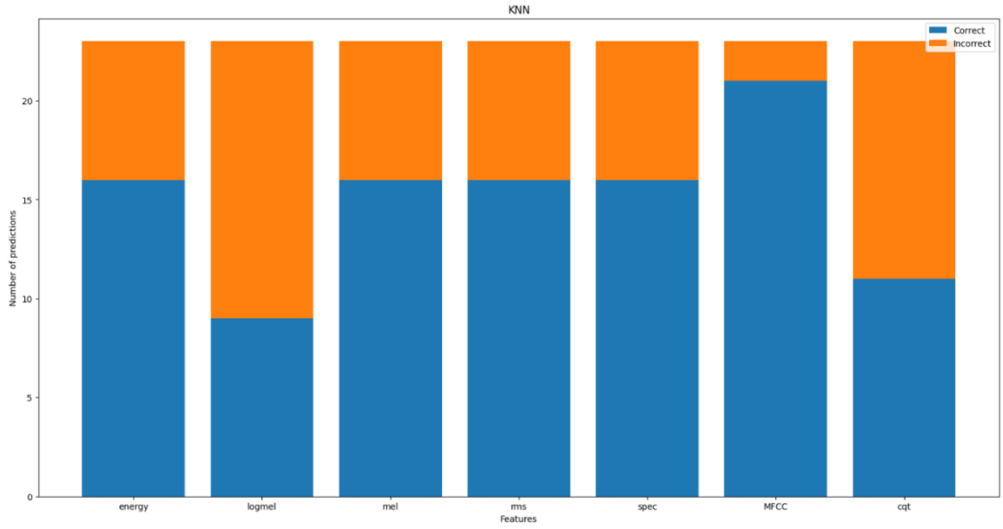


*Figure 3. Results from the test data for k-Nearest Neighbour model*

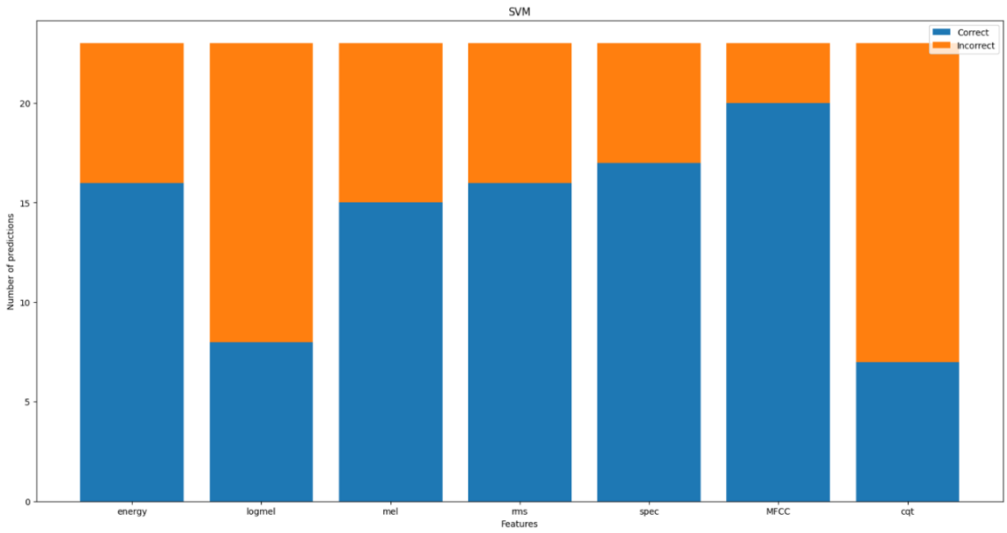


*Figure 4. Results from the test data SVM model*

## VI. CONCLUSION

Overall, the model has done a really great job and shows potential. The accuracy rate for both the models is good to high, especially on important features such as MFCCs or spectrograms. However, during the process, some issues appear and can be improved. Firstly, it's the imbalanced data. The dataset contains 50 samples for the "bus" class and 69 samples for the "tram" class, which can lead to biased models, as the model may lean towards the majority class. As shown in the result, some of the bus samples tend to have marked all as "tram" classes, which is considered a flaw. Techniques such as oversampling, undersampling, or using more advanced methods like SMOTE (Synthetic Minority Over-sampling Technique) could help balance the dataset and improve model performance.

Additionally, some features show poor results in training. Therefore, further analysis needs to be conducted using techniques like Recursive Feature Elimination (RFE) to prioritize informative features.