**Part 1 : research and selection**

**Introduction:**

The process of audio deepfake detection (**ADD**) lies mainly on two steps: feature extraction and classification implementation. Feature could be mel spectrogram, Linear Frequency Cepstral Coefficients or can be extracted by encoder decoder model (**autoencoder**) or even by ResBlock. The classification model could be based on wide range of architectures as deep neural networks (**DNN**) and its types

**Selection :**

At first I was confused between 3 models : the first option is traditional machine learning (**ML**) algorithms like XGBoost and SVM and features got extracted as Gaussian Mixture Model (**GMM**). The second alternative is generative adversarial network (**GAN**). Third option is convolutional neural network (**CNN**) some recurrent neural network (**RNN**) in which its inputs is the vocals time step by step and output one cell whether the audio is fake or real (**sequence to one RNN**) or a mix between the 2 to get the benefits from both with avoiding the downsides of both models.

First is the classical ML methods by using **XGBoost**, it provides fast training and interference time due to its relatively low complexity that is suitable to our task to of real time performance. But on the other hand it does not generalize well on other data than training data and requires complex audio preprocessing steps that must be done manually, unfortunately this reduces the speed of the interfece a little bit and makes it hard for scalable applications

Also polynomial support vector machine (**SVM**) have the same problems but in research papers it performed better with F1 score of 99% (astonishing results of course but it can’t generalize data, this is overfitting that behaves poorly during interface)

The idea of GAN was also unapplicable due to GAN unstability during training stage and each of the generator and discremenator reaches to a state of equilibrium . furthermore, during my research stage seldom have I found a research paper of GAN that outperforms other model without high computational power

The last suggestion of DNN I was about to use pretrained models like EfficientNet or Xception architecture but after research I found **SpecRNet** research paper and its use of LFCC for feature extraction. I was fond of this idea but see to optimize it by introducing mel –spectrogram for better frequency analyzing of both low and high frequencies. I made some hyperparameter tuning for the model of no. of blocks and no. of neurons of each layer

After research I knew that SpecRNet was released in 2022 so I further researched on papers in 2024 and found CLAD model for audio manipulation and the utilizing of 2 encoders to extract features and getting similarity between them. Of course the model is too heavy for real time application because of the encoder step during interference time but I get use of the augmentation and applied some augmentation functions to the data

**Part 2 : Implementation**

I downloaded small part of ASVspoof5 as the data is too large and the model accuracy score is not a part of the evaluation of the assessment

I extracted the file and the labels file, applied preprocessing and augmentation and build the model

**Part 3 : documentation**

The most significant challenge I encountered was to understand the process of feature extraction and advanced math of Fourier transformation for audio processing. I solved this problem by YouTube videos specially 3Blue1Brown channel and Medium website to get accustomed with MFCC. These besides tens of research papers from 2020 to 2024 to get grasp of the advancements in audio deepfake detection models and new technologies and models

Assumptions I made was that the model must be light without high complexity to be suitable for the real time requirement specially if it is a conversation of 2 or more people the model should be fast as possible with the least processing during interface time without affecting the performance

I selected this model due to its speed during interface with no. of learned parameters less than 400 000. Other models like transformers may perform better than SpecRNet but with higher computational power. The model preprocess the audio into mel spectrogram to get high and low frequencies on mel scale. Features are then fed into the SpecRNet model of convolutional layers of stride 3 and SeLU activation that does not need batchnormalization after every layers (self normalizing layers) with doubling the layer neurons every step to extract deeper informations. At the end a dense layer and dropout layer for regularization and the classifier layer with one cell of sigmoid activation for classification fake or real

Accuracy score of higher than 85 for this model in short time of training is satisfactory. Observed strengths of model that t generalize well due to the augmentation by common manipulation in deepfake generators like noise and shifting. Also the interference time makes the model applicable for real time applications. Its downsides is that powerful deepfake audio generator may trick it by some maipulations. Also the achiticture is relatively old end of 2022. But these downsides could be avoided by introducing light transformers before dense layers. Also better augmentation may increase model ability of generalization and lastly longer training period will achieve our goal. At the end on implementing on tensorflow-lite will make the model easily deployed for mobile phones and simple devices like anti-spoofing locks

**3A:** data preprocessing of augmentation and implementing advanced mel spectrogram with its advanced math and changing it into code

**B:** in research datasets it will perform less efficiently than some models as other research models like CLAD and CRNN pay attention to only one point of the research not the whole performance of the model. But in constrast our model I think performs better in real environment due to its light weight and speed as our target is to achieve real time or near real time performance

**C:** of course more data will make the model better as I only applied the training process to only small portion of the ASVspoof5 dataset (less than 6 giga of sound audio)

**D:** deploying it to mobile apps and the web by tensorflow-lite after better training