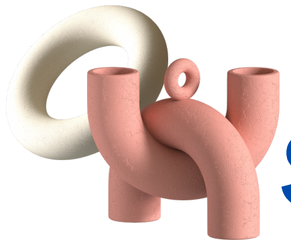
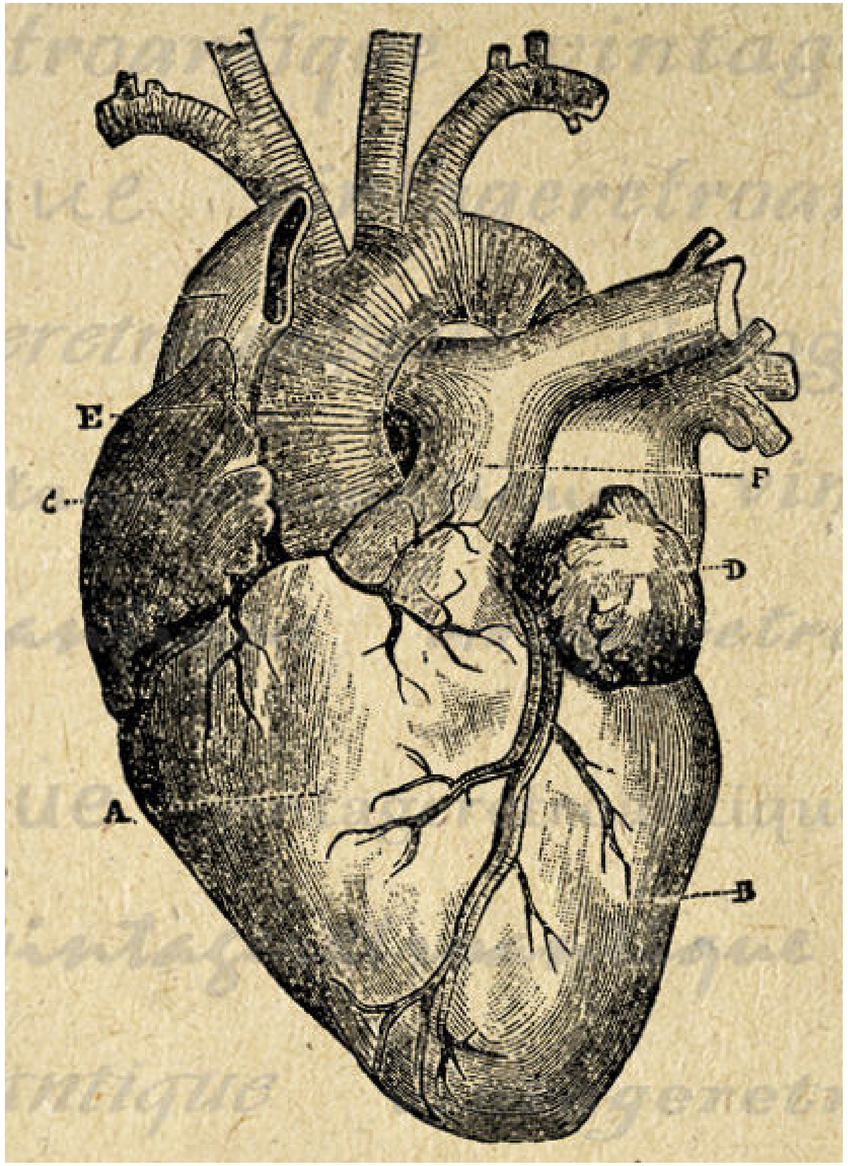
Department of Applied Mathematics and Statistics

Heart Sound

Classification Using Deep learning

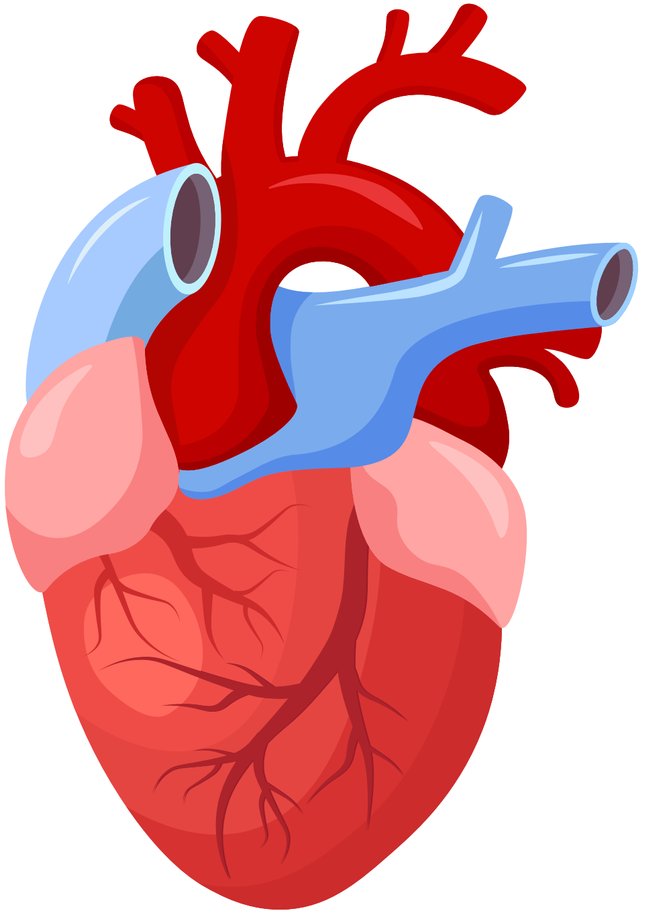


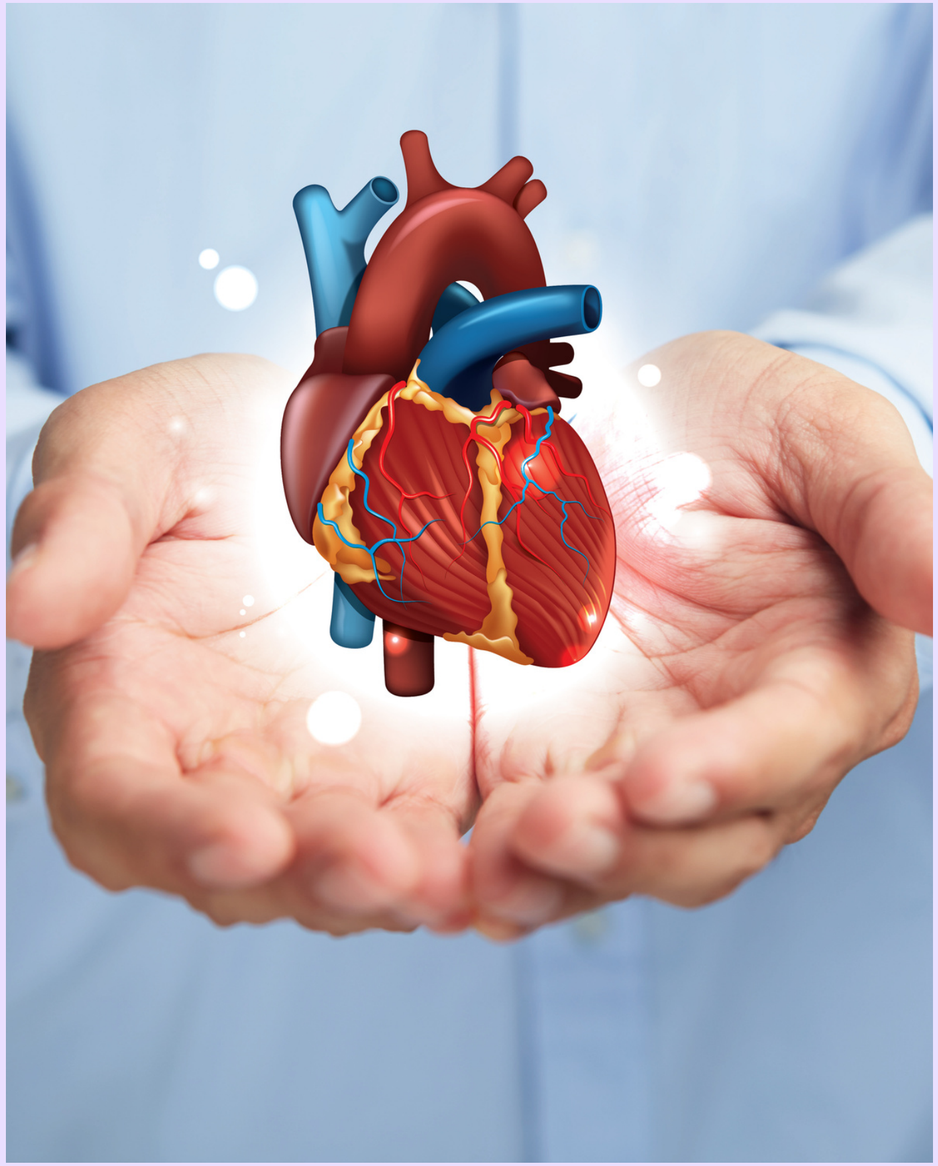
SEAN VENGNGY



Agenda

1. Introduction
2. Heart Sounds
3. Dataset and Preprocessing
4. Machine Learning Overview
5. Deep Learning Overview
6. Future Work
7. Results
8. Conclusion





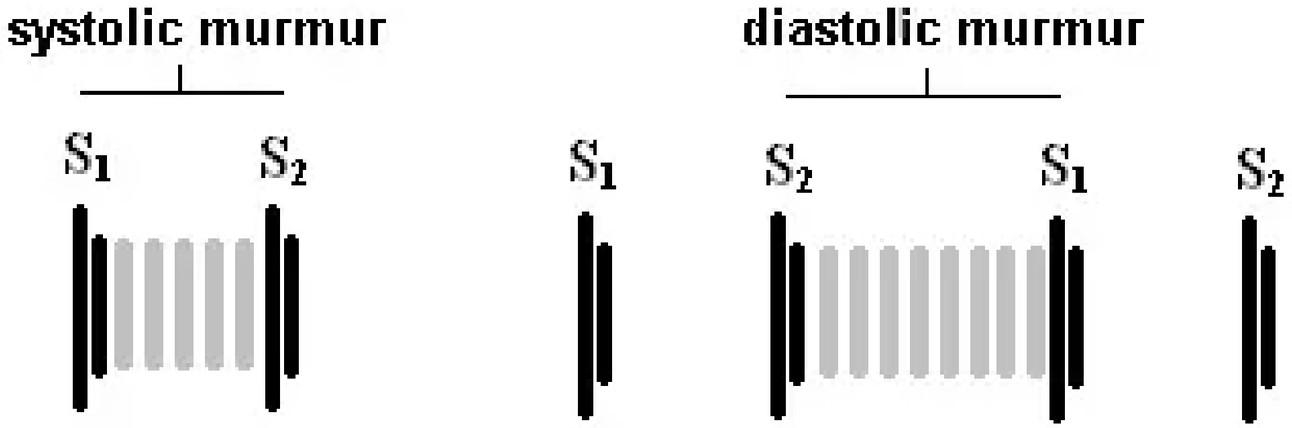
INTRODUCTION

Brief of the importance of heart sound classification

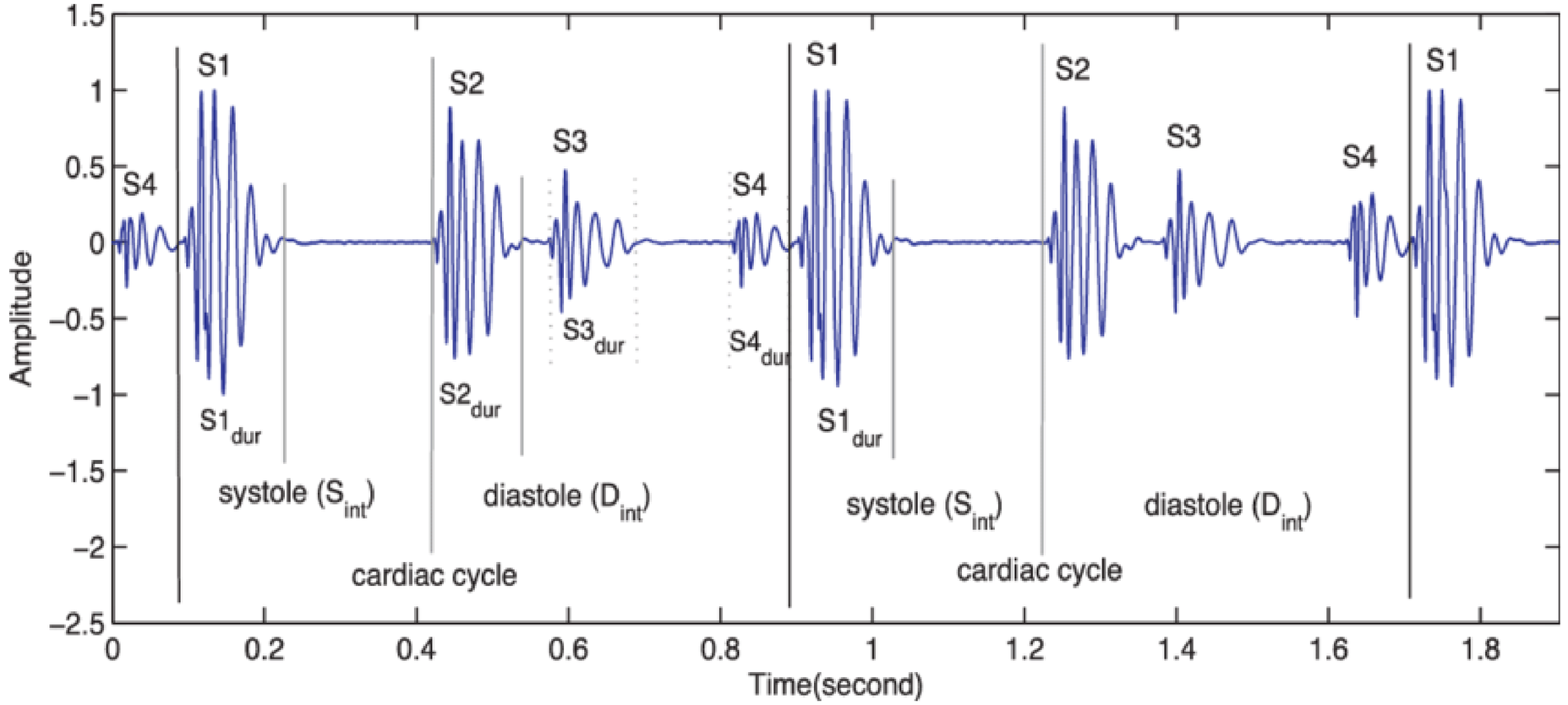
As per the World Health Organization (WHO) report published in September 2016, around 17 900 000 people die every year because of heart-related disease. In medical terminology, heart disease is called cardiovascular disease (CVD) which occurs either due to narrowed or blocked blood vessels or due to problems in heart rhythm. Sometimes it also occurs due to some inborn problem. Heart sounds carry crucial information about the functioning of the cardiovascular system. Accurate classification of these sounds is vital for diagnosing various heart conditions. We'll begin by understanding heart sounds and their pathological implications. Then, we'll delve into the dataset and preprocessing steps. Afterward, we'll explore the basics of deep learning and recurrent Neural Networks (RNNs). We'll discuss our proposed model architecture, training and validation process, and evaluation metrics.

Heart Sound

- The human heart produces distinct sounds during each cardiac cycle.
- S1 (Lub): The first heart sound occurs when the mitral and tricuspid valves close, marking the beginning of systole.
- S2 (Dub): The second heart sound occurs when the aortic and pulmonary valves close, marking the beginning of diastole.
- Additional Heart Sounds and Murmurs: Beyond S1 and S2, there are additional heart sounds (S3 and S4) that can indicate abnormal blood flow. Heart murmurs, characterized by turbulent blood flow, are often indicative of underlying cardiovascular issues.



Types of heart sounds.



while in previous studies, the data were classified into two modes of normal and abnormal sounds, in the current research, the heart sounds are classified into three classes of normal, abnormal due to the third heart sound, and abnormal due to the fourth heart sound.

Systolic murmurs	Diastolic murmurs
Aortic stenosis (AS) Pulmonic stenosis (PS) Mitral regurgitation (MR) Tricuspid regurgitation (TR) Mitral valve prolapse (MVP) Atrial septal defect (ASD) Ventricular septal defect (VSD) Hypertrophic Cardiomyopathy	Aortic regurgitation (AR) Pulmonic regurgitation (PR) Mitral stenosis (MS) Tricuspid stenosis (TS) Austin-Flint murmur Continuous murmurs Patent ductus arteriosus (PDA) Combination murmurs

Dataset and preprocessing

- **Dataset Overview:** Our study utilized a curated dataset of recorded heart sounds. The dataset was divided into 3labeled segments (normal, ms, mr) for training and validation.
- **Importance of Preprocessing:** Raw heart sound recordings often contain noise and artifacts that can interfere with accurate analysis.
- **Noise Reduction:** Filtering out ambient noise and artifacts to isolate heart sounds.
- **Feature Extraction:** Extracting relevant features from segmented heart cycles to feed into the deep learning model.

ROADMAP

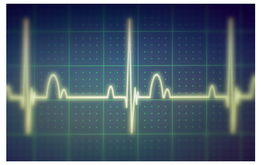
01

Heart Sound

.wav

02

Pre-Processing



.wav

03

Sound Feature Extraction

04

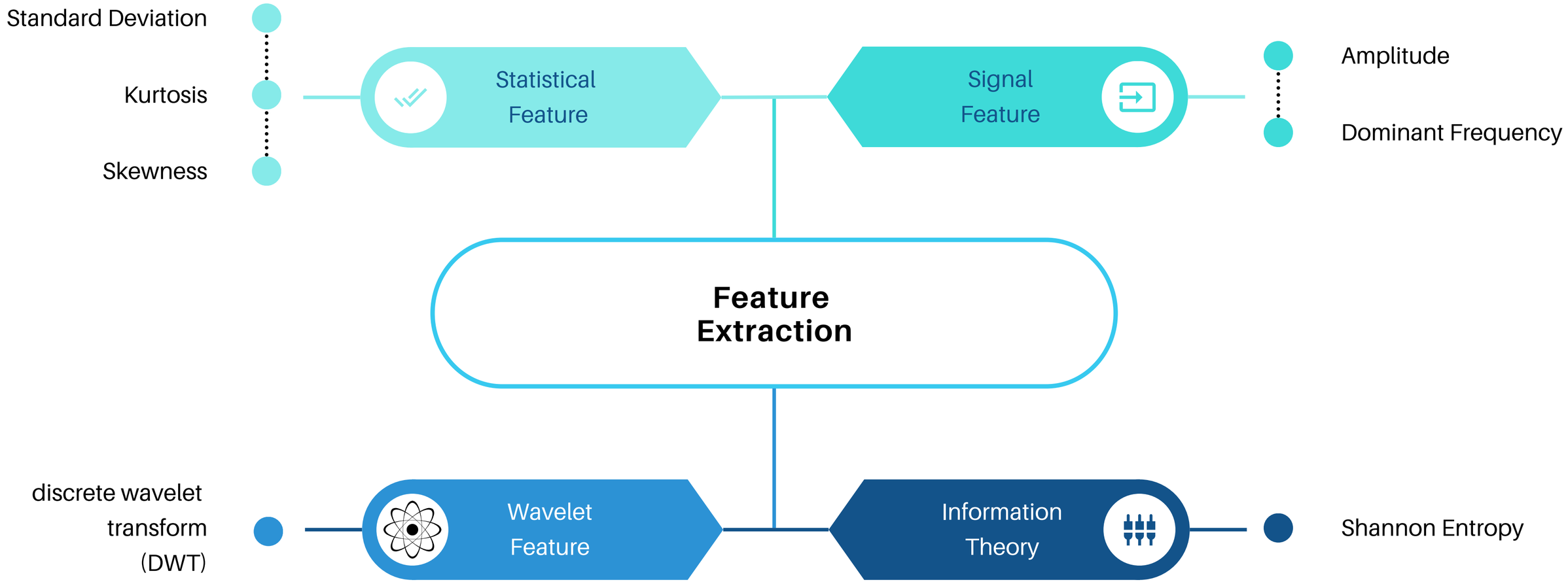
Structure Feature Engineering

05

ML And DL

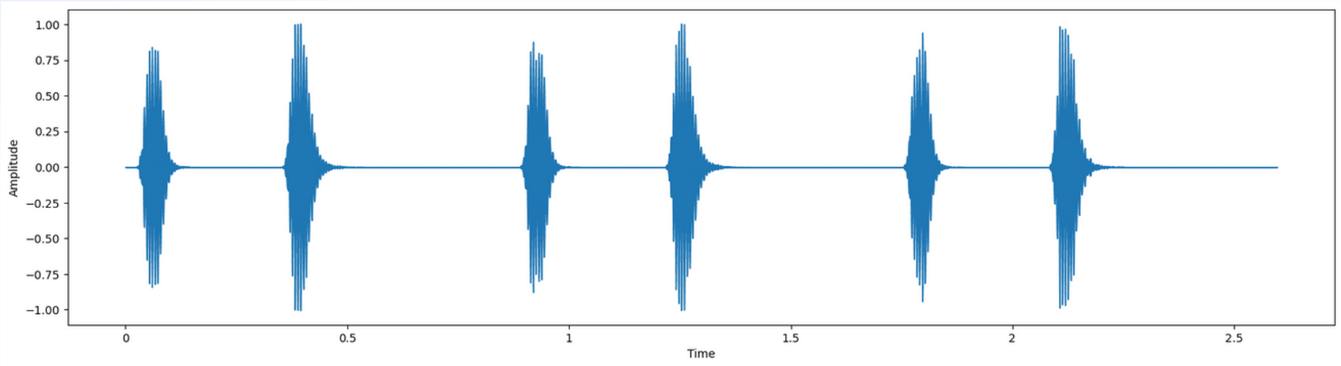
06

Result

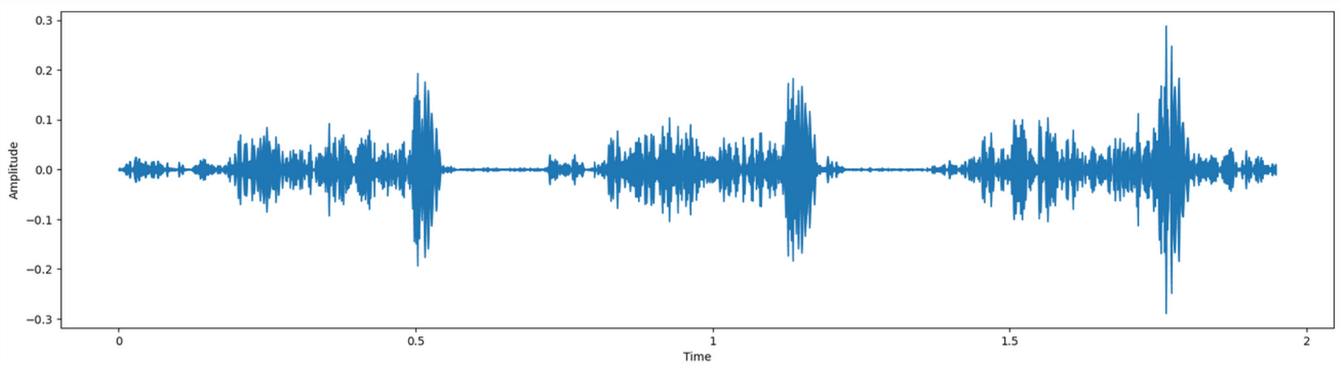

Standard Deviation
Kurtosis
Skewness
Statistical Feature
Signal Feature
Amplitude
Dominant Frequency
Feature Extraction
discrete wavelet transform (DWT)
Wavelet Feature
Information Theory
Shannon Entropy

S	S	W	I
Statistical Feature	**Signal Feature**	**Wavelet Feature**	**Information theory**
The statistical features include standard deviation, skewness, and kurtosis. They are Used to examine how the data is distributed.	Amplitude is the maximum displacement or distance made by a point on a wave measured from its equilibrium position. Besides, as the lowest frequency component is known as the fundamental frequency, the dominant frequency is the fundamental frequency with the highest amplitude	The discrete wavelet transform (DWT) is a wavelet transform for which the wavelets are discretely sampled. It's used for tasks like image compression, where you want to represent images efficiently while preserving important details. DWT breaks down a signal into high and low-frequency components, which can reveal details about the data at various levels of granularity.	Information theory is the mathematical treatment of the concepts, parameters, and rules governing message transmission through communication systems. The entropy of a discrete random variable X with the mass probability function p(x) is denoted by $H(x) = E(I(x)) = -log_b(p(x))$

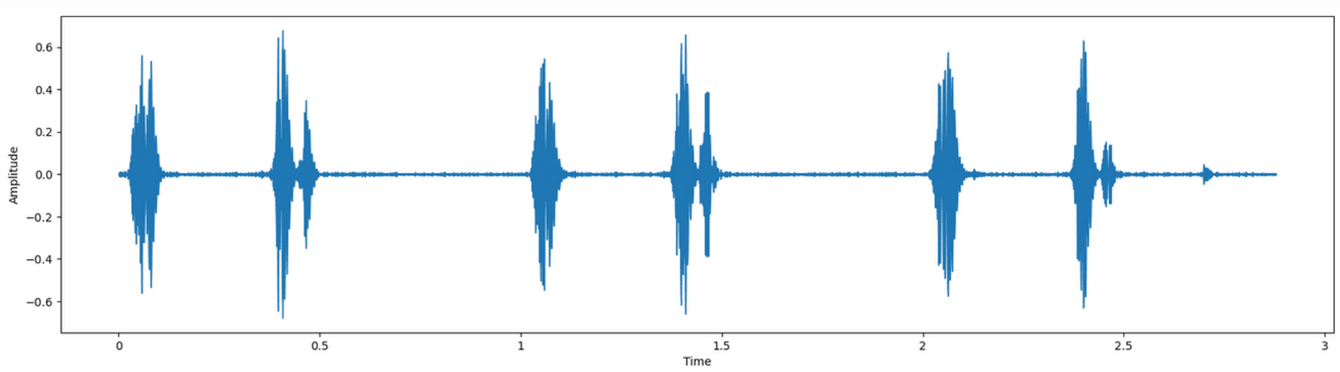
Waveform of audio from the dataset



show_audio_waveform
(Normal_sample)

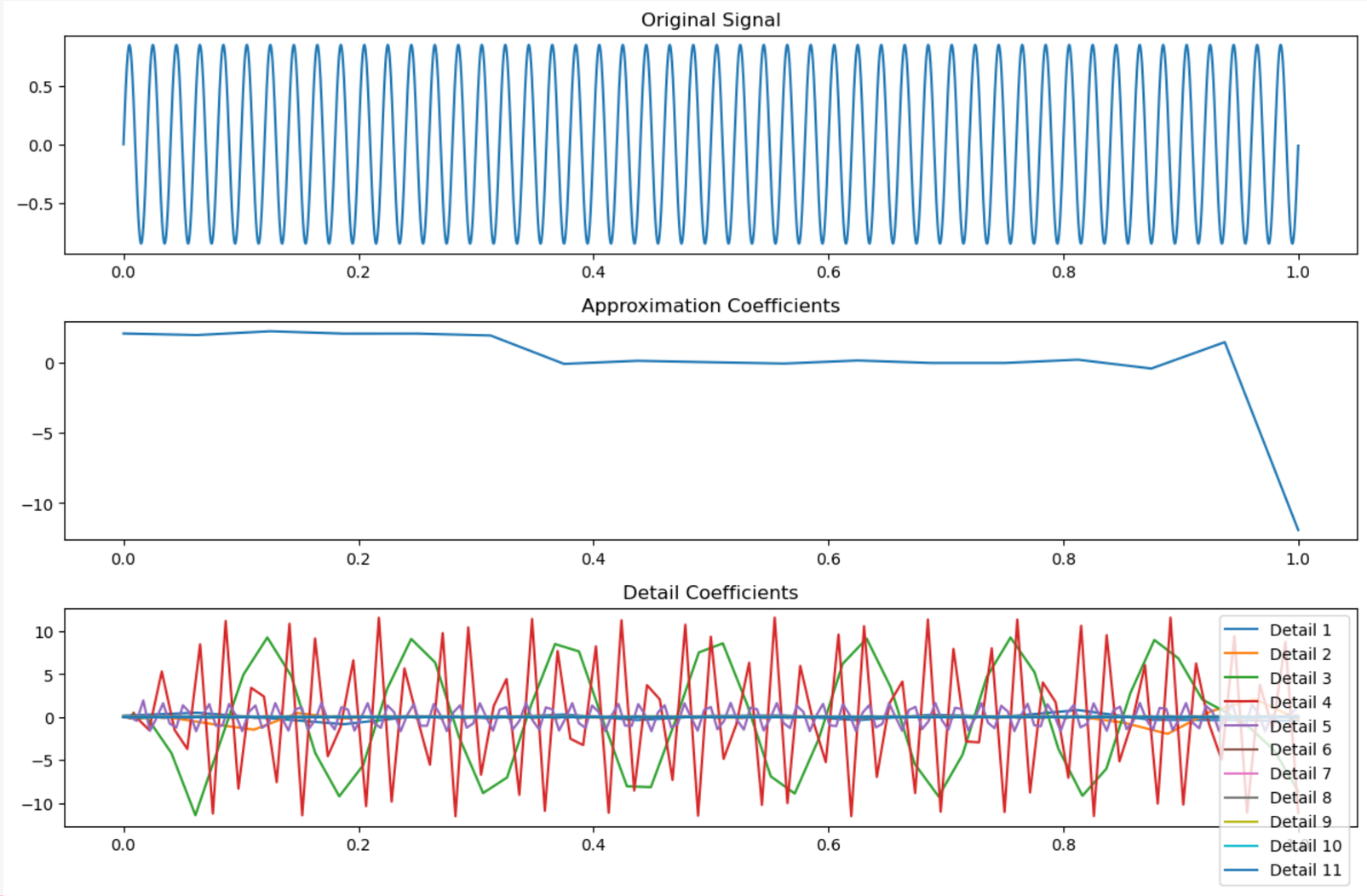


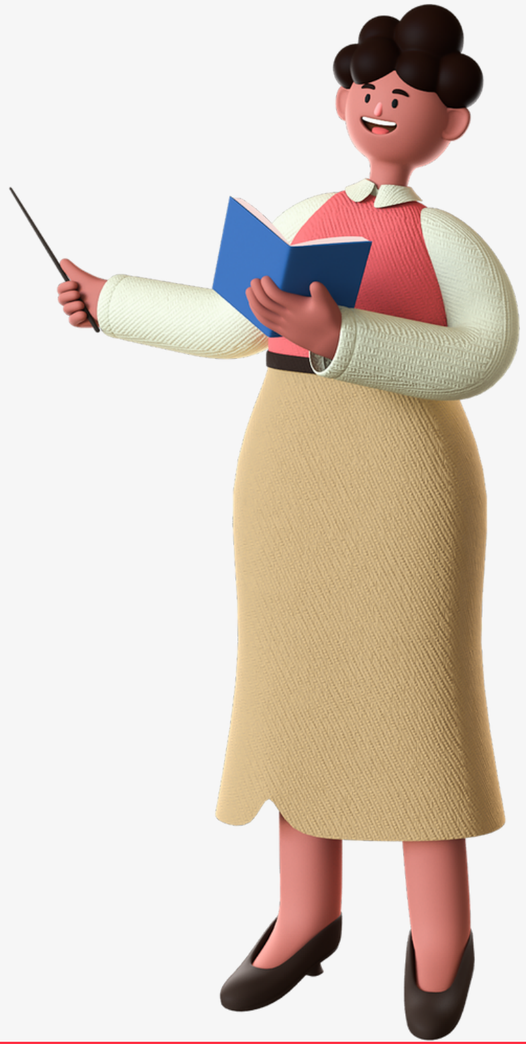
show_audio_waveform
(mr_sample)



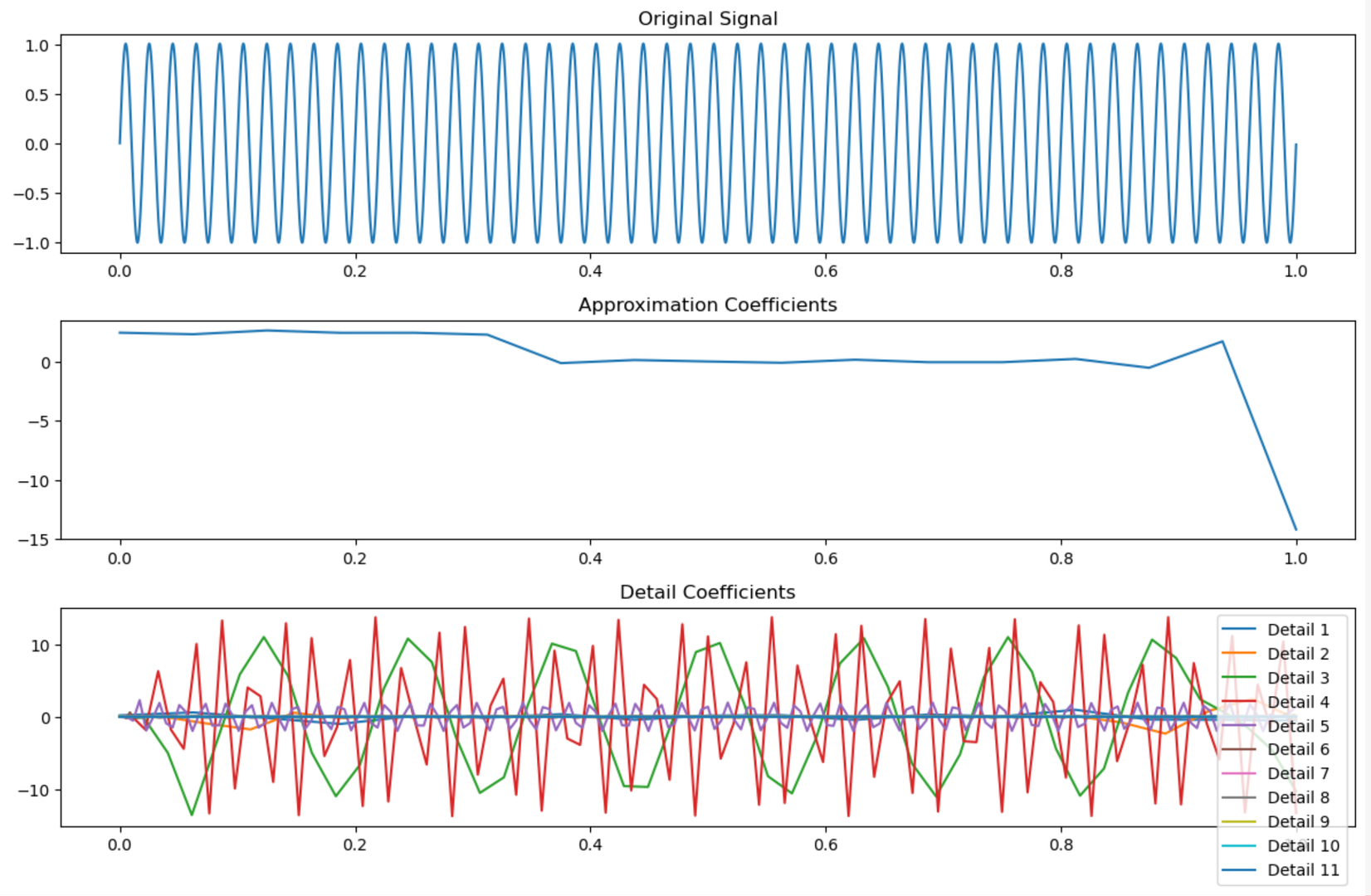
show_audio_waveform
(ms_sample)

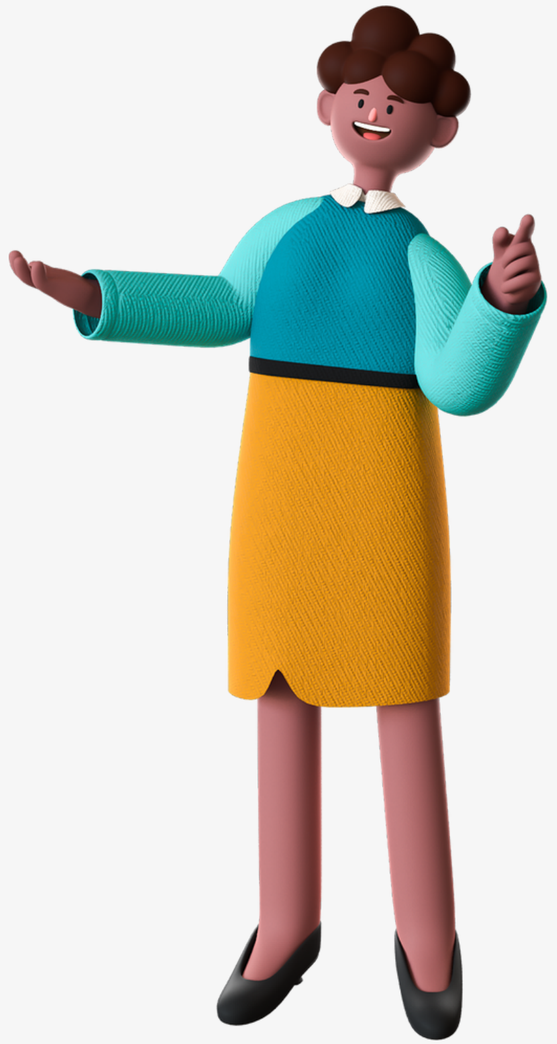
wavelet feature normal sound



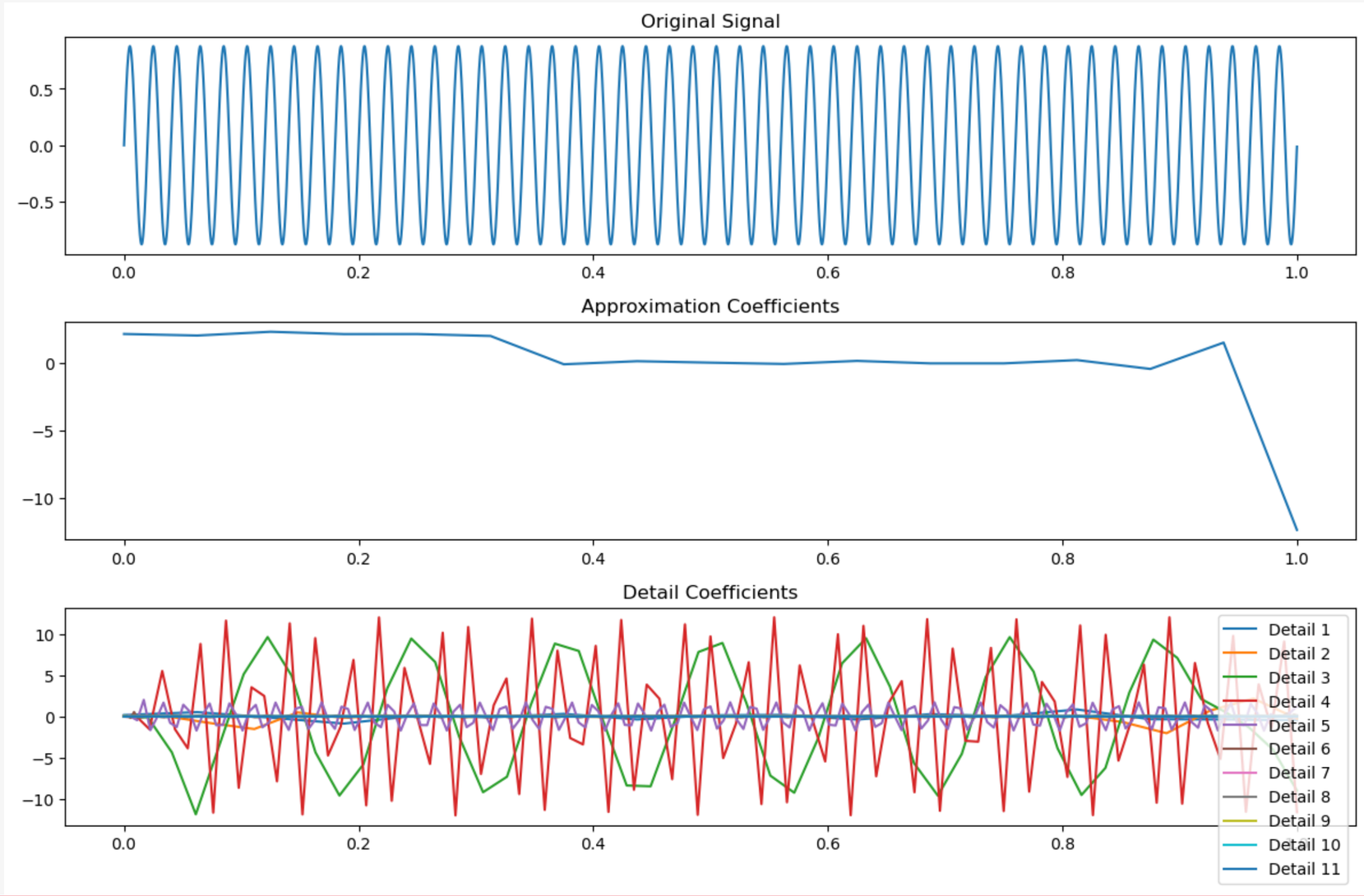


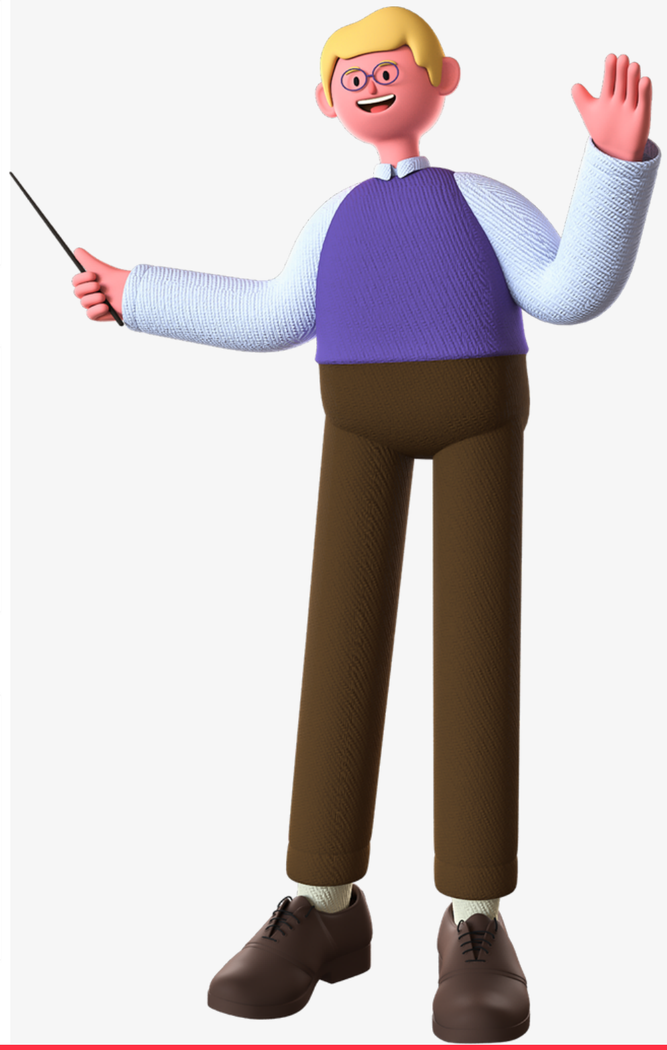
wavelet feature mr sound



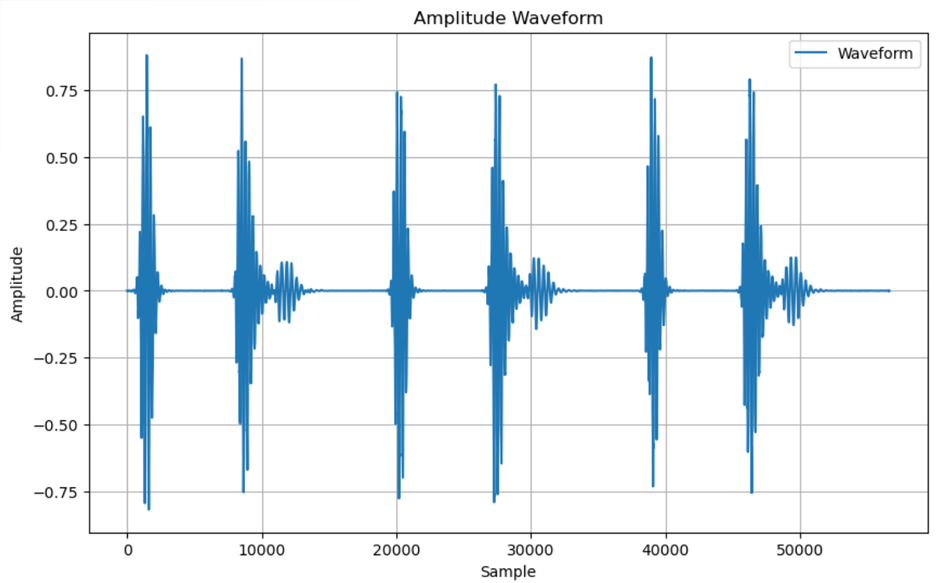


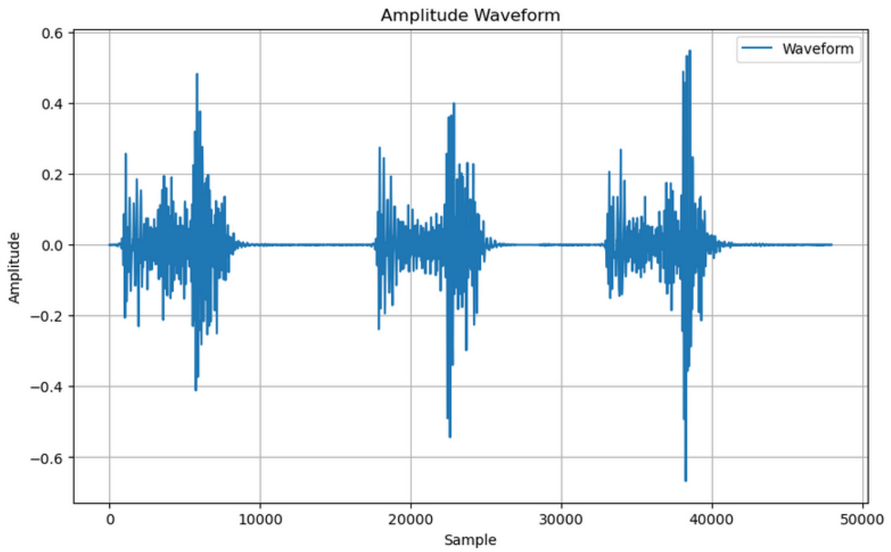
wavelet feature ms sound

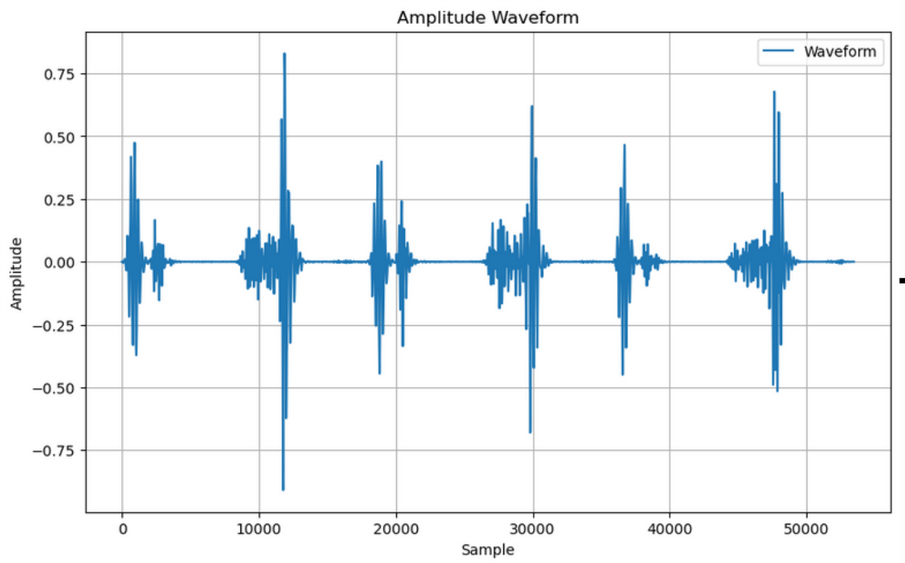




Feature Signal: Amplitude





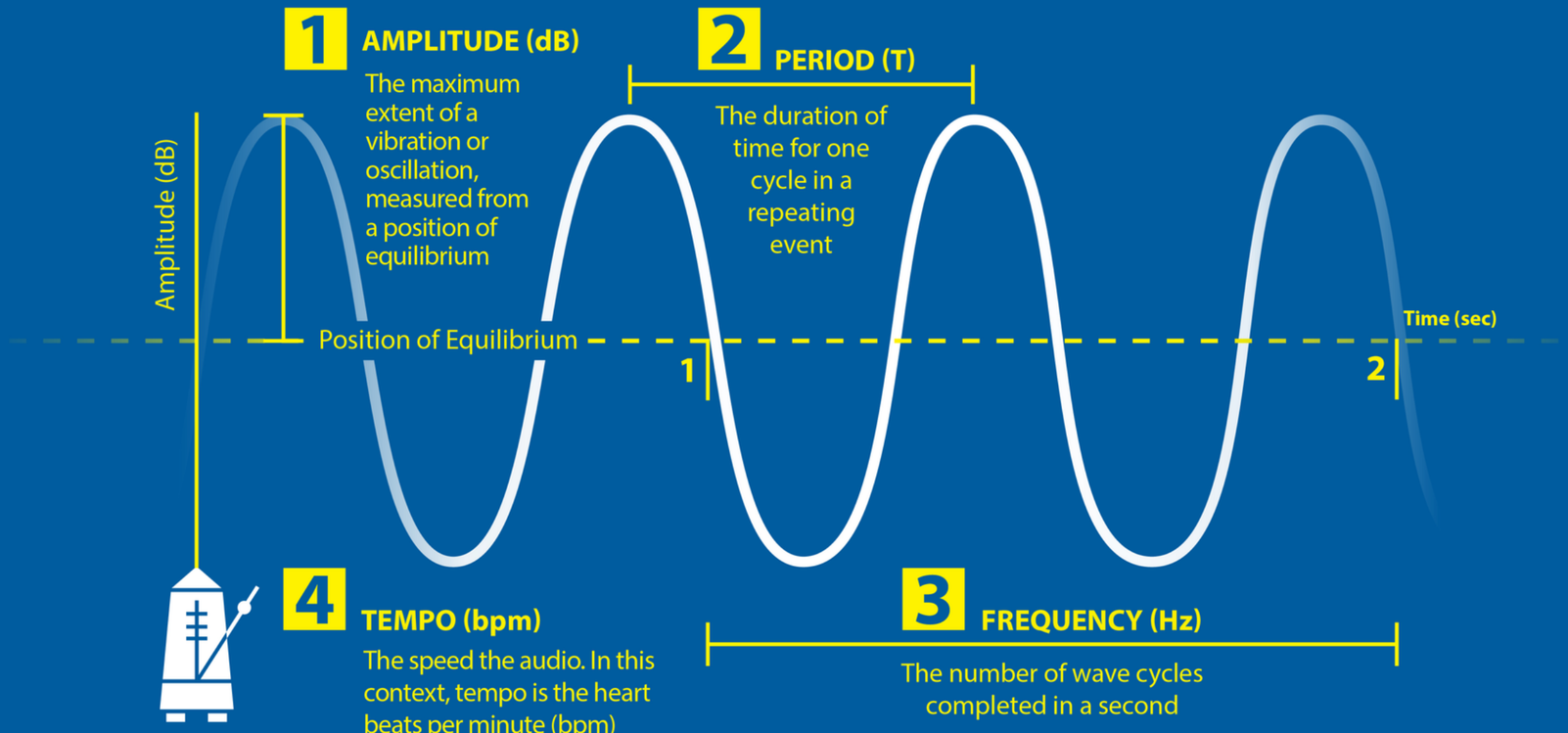


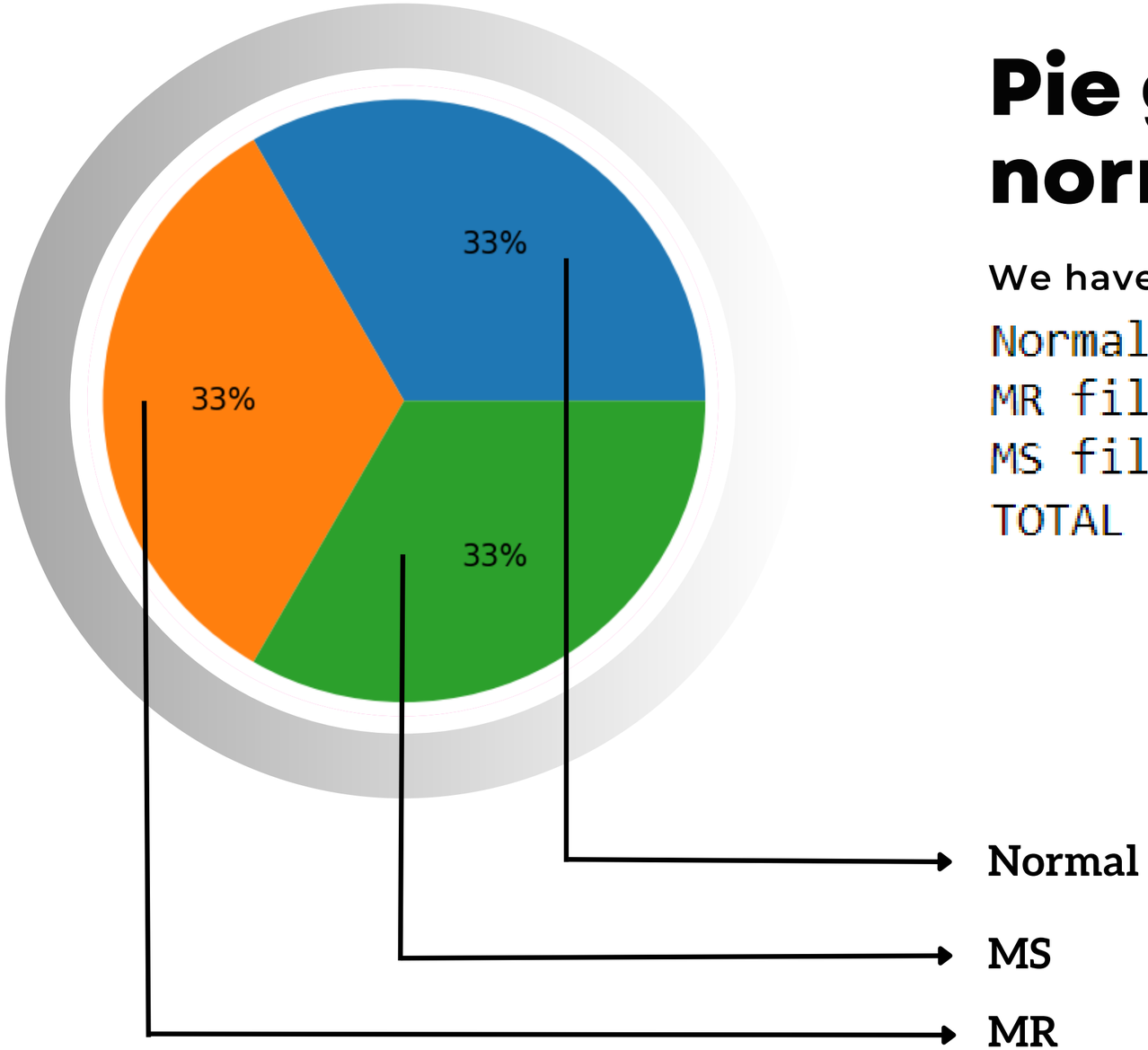
Amplitude normal_sample 0.87926805

Amplitude mr_sample 0.90950817

Amplitude ms_sample 0.6675642

There are 4 features that describe the audio signals above



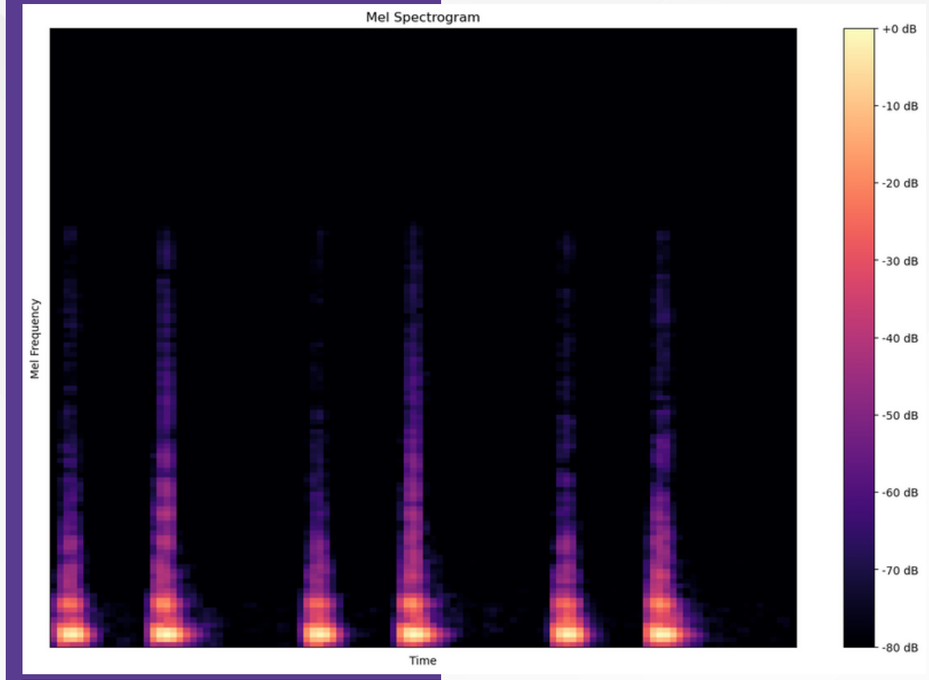


Pie graph of 3labels normal, MR and MS

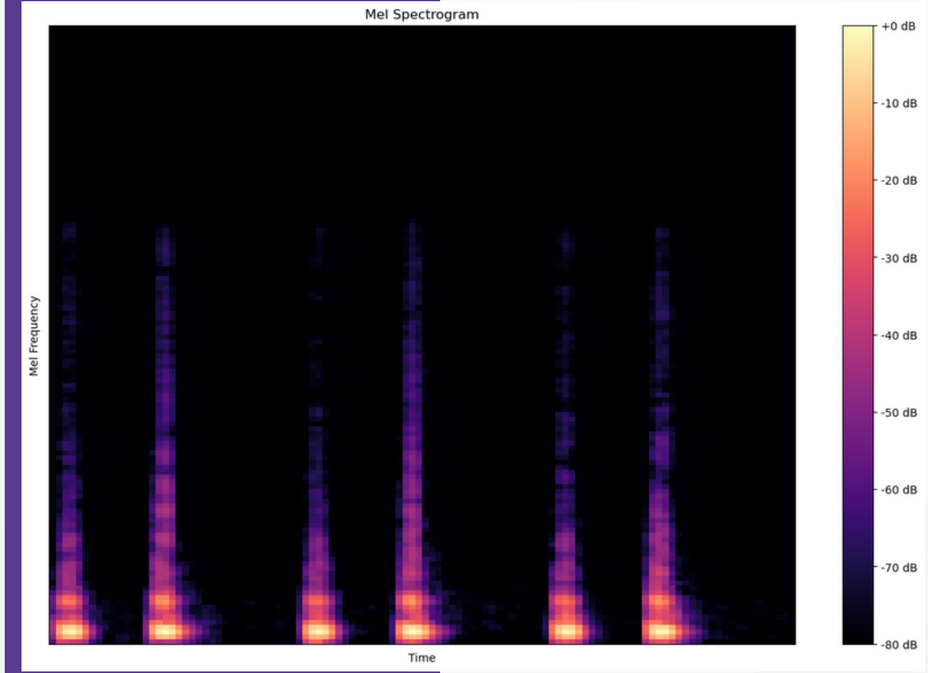
We have Label Data is Balance and the length is:

```
Normal files: 160
MR files: 160
MS files 160
TOTAL TRAIN SOUNDS: 480
```

REPRESENTS A SOUND

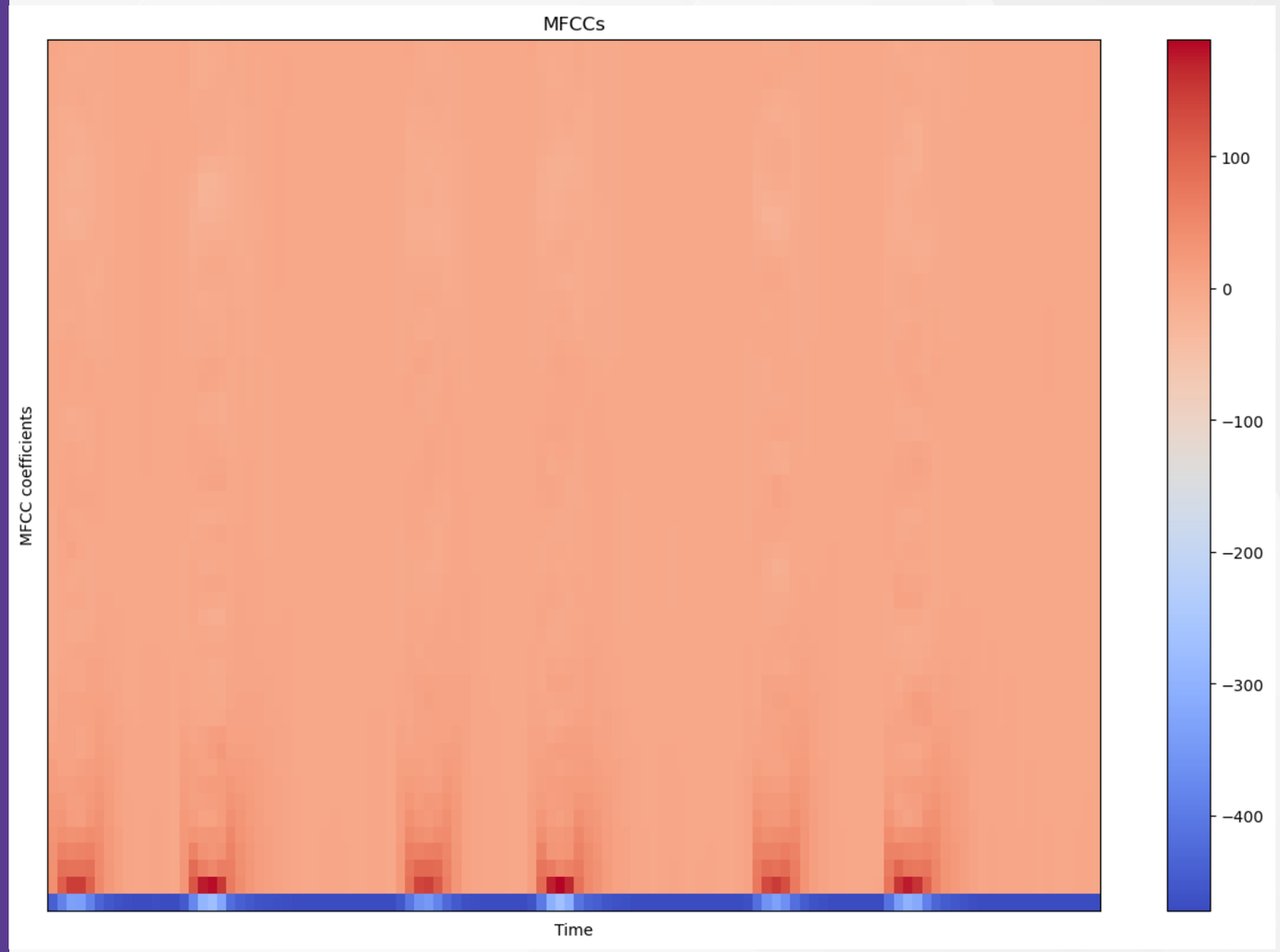


Spectrogram of Normal Sample



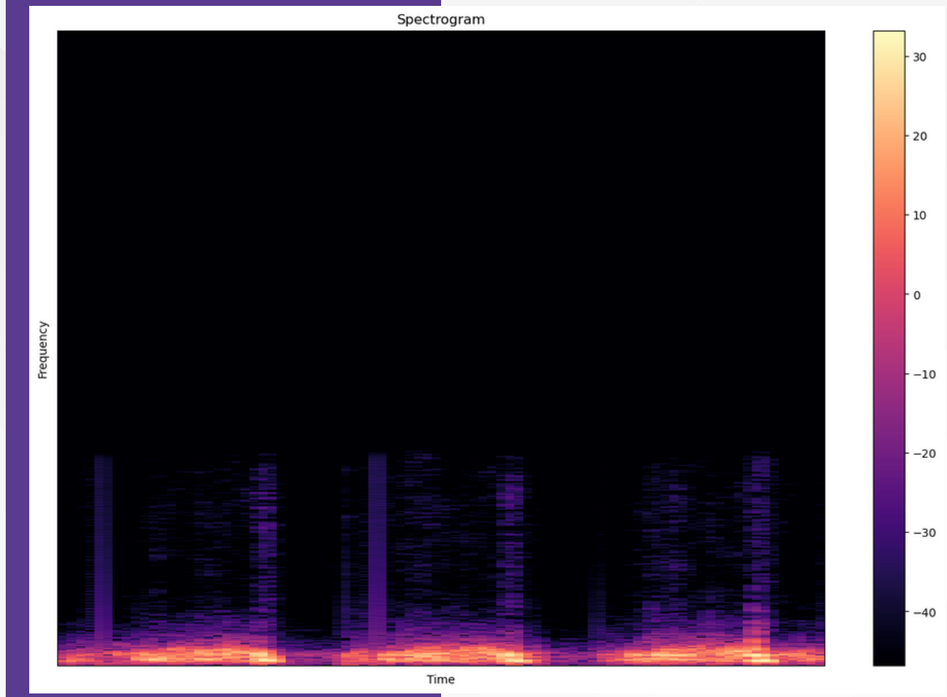
Mel Spectrogram of Normal Sample

REPRESENTS A SOUND

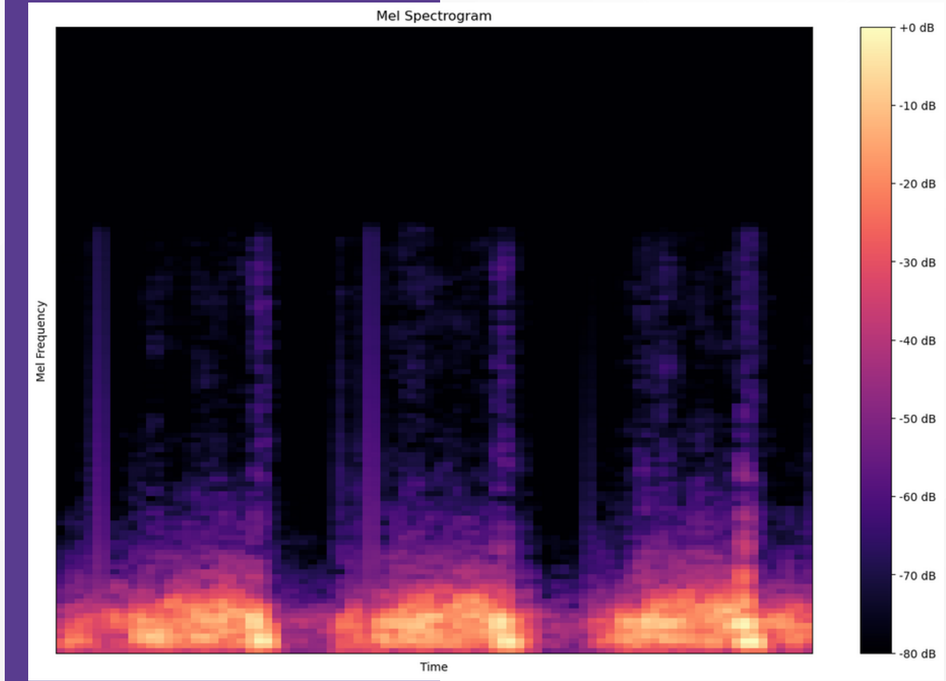


MFCC of Normal Sample

REPRESENTS A SOUND

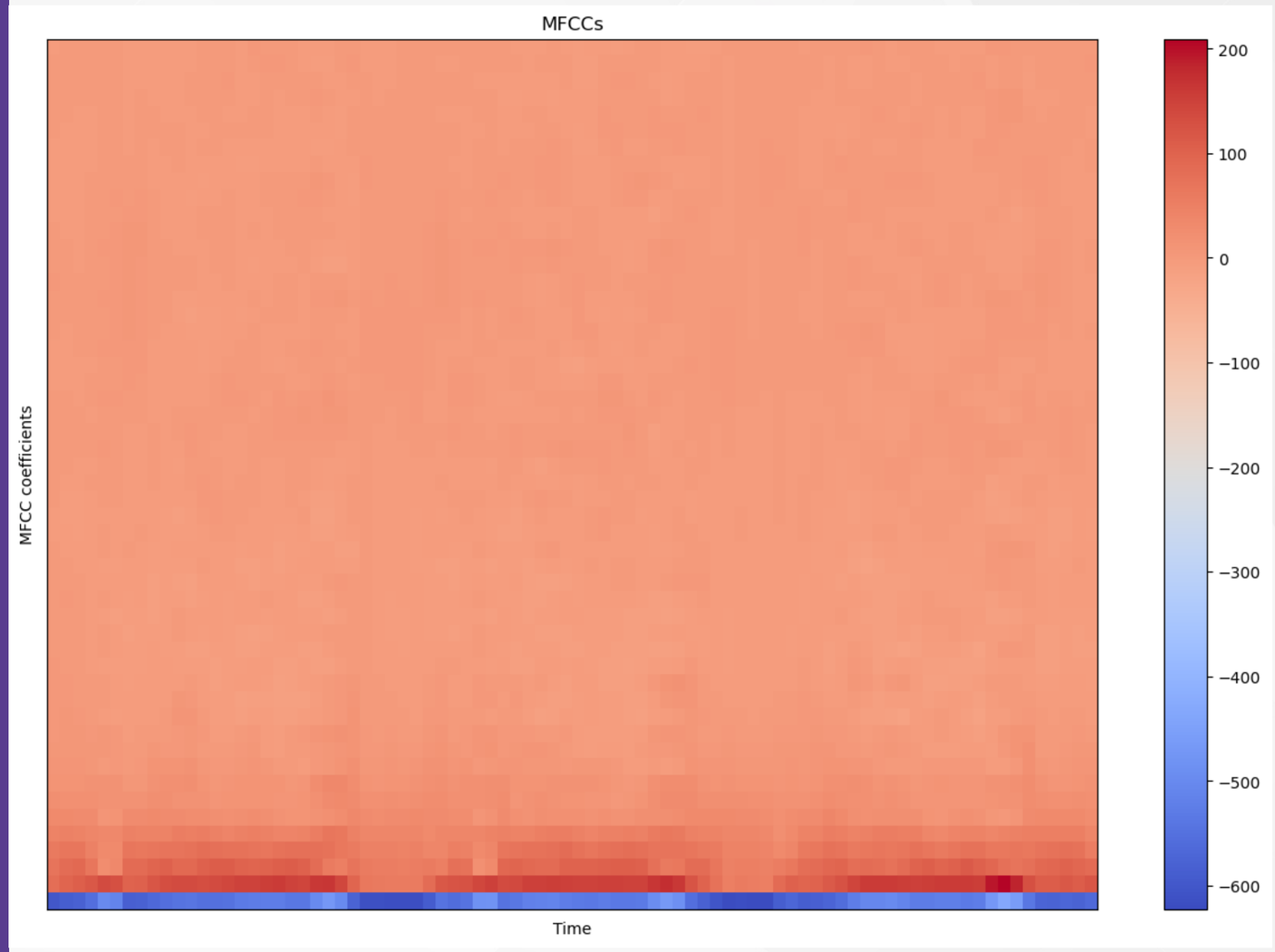


Spectrogram of mr Sample



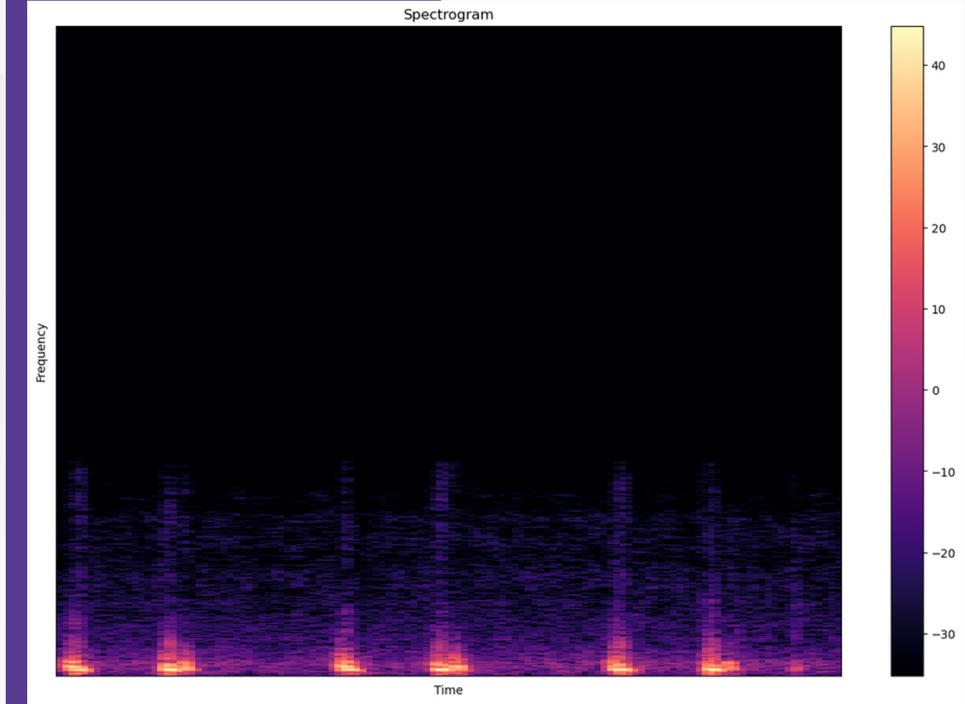
Mel Spectrogram of mr Sample

REPRESENTS A SOUND

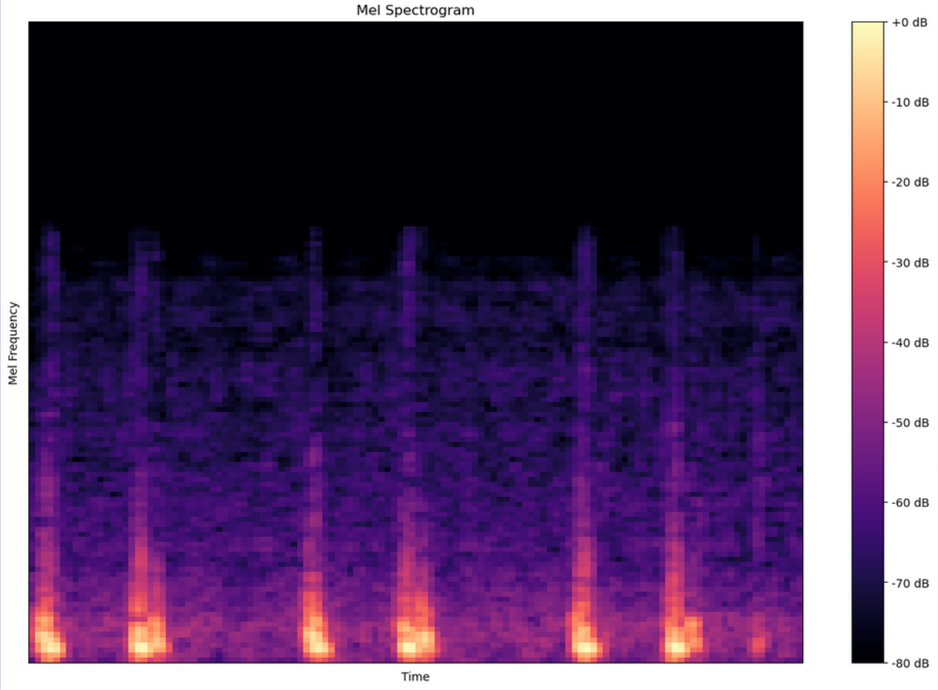


MFCC of mr Sample

REPRESENTS A SOUND

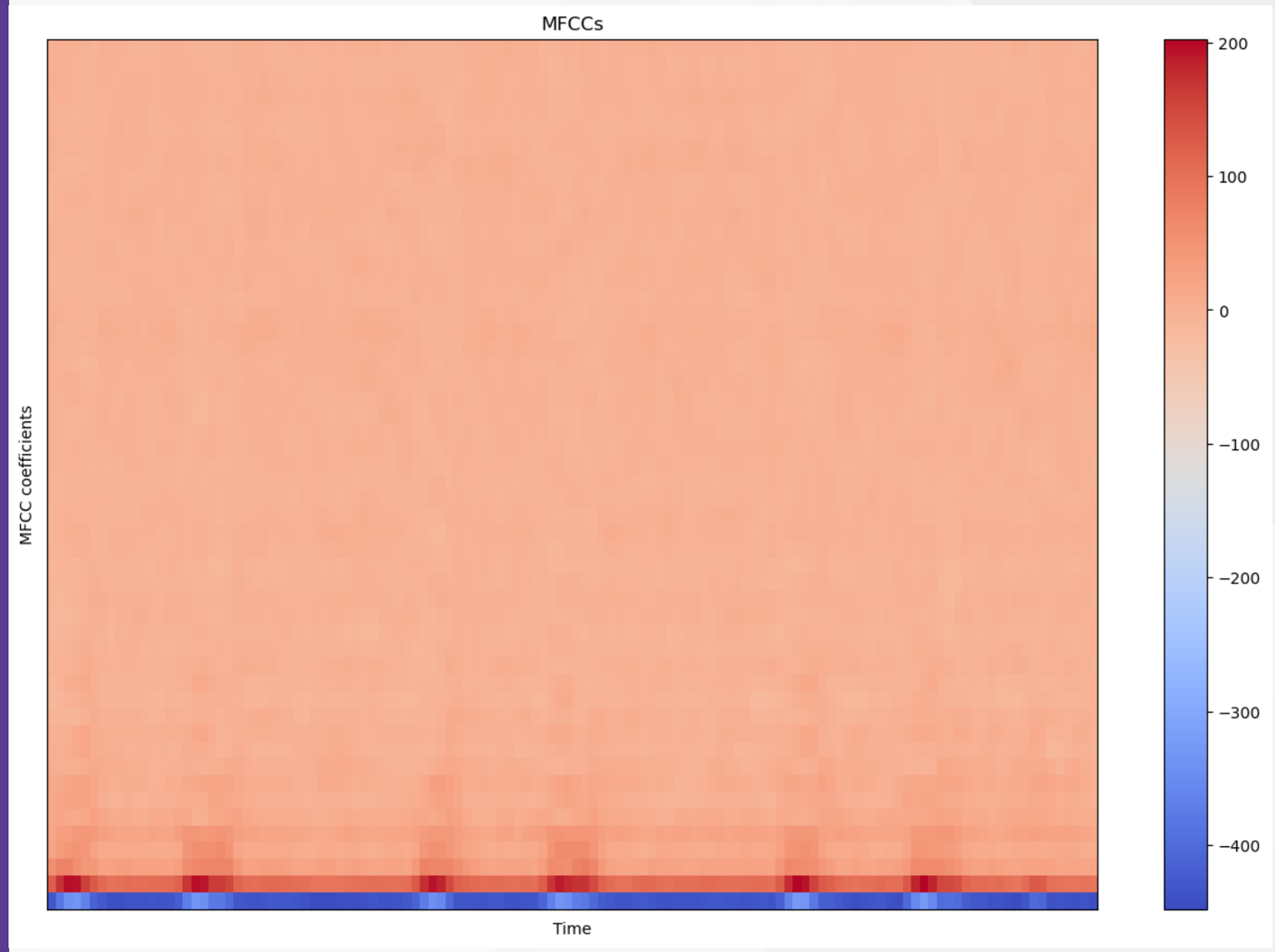


Spectrogram of ms Sample



Mel Spectrogram of ms Sample

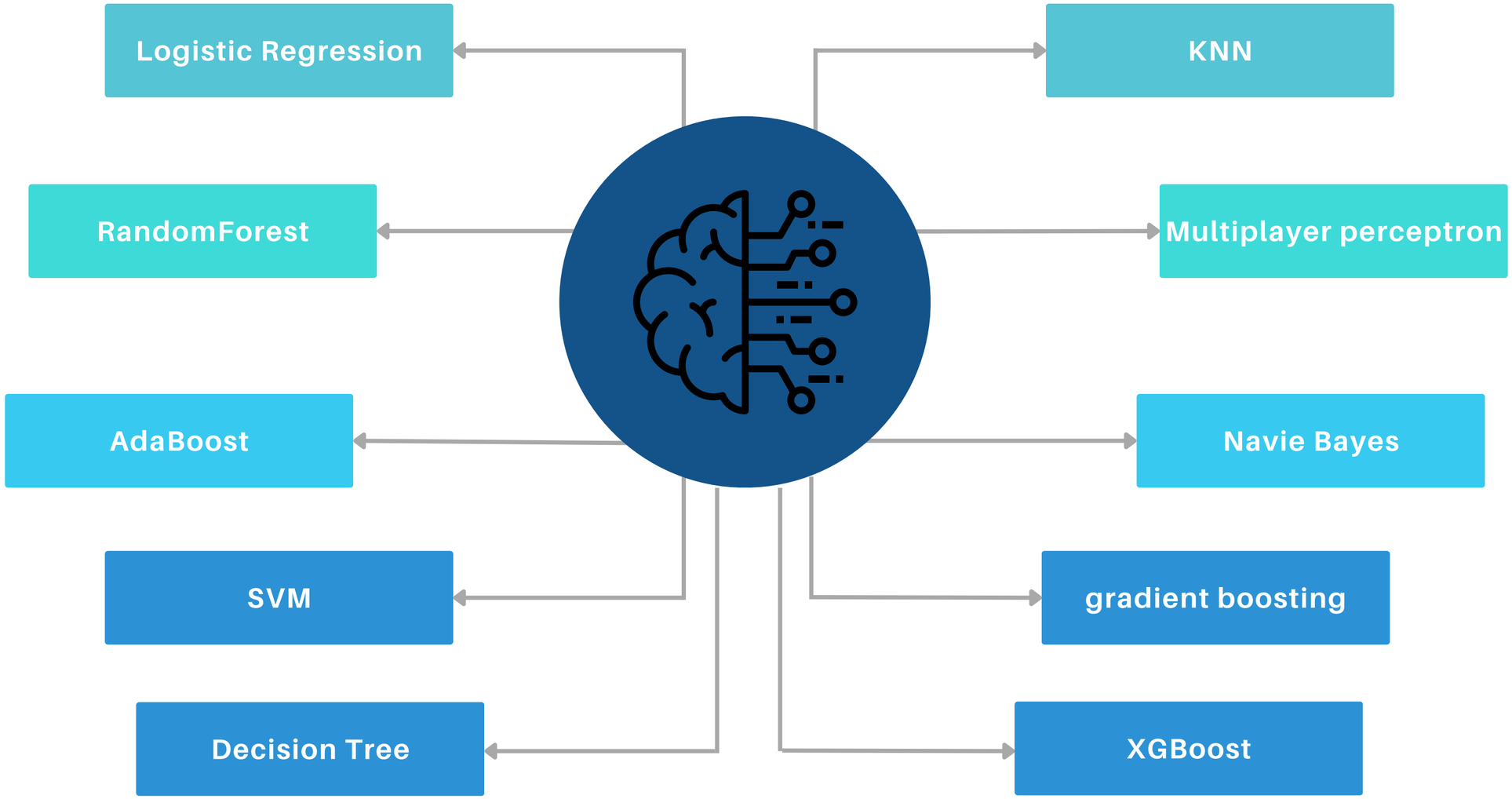
REPRESENTS A SOUND



MFCC of ms Sample

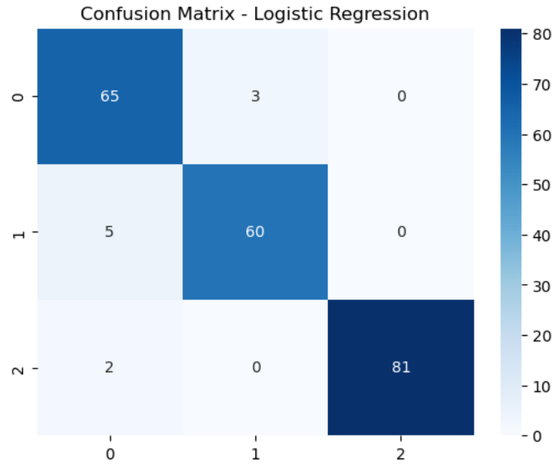
S	M	M
Spectrogram	**Mel-spectrogram**	**MFCC**
A spectrogram is a visual representation of how the frequency content of a signal changes over time. It shows how the intensity of different frequencies in a signal varies as time progresses. Spectrograms are commonly used in audio processing and other fields to analyze and visualize sound data.	A Mel-spectrogram is a type of spectrogram that uses the Mel scale to represent the frequencies of a signal.Mel-spectrograms are used for various audio processing tasks, particularly in speech and audio analysis. Essentially, Mel-spectrograms provide a more human-like representation of audio data, making it easier for machine learning models to work with sound information.	MFCCs are a representation of the short-term power spectrum of a sound signal, commonly used in speech and audio processing. In essence, MFCCs help convert audio signals into a format that makes it easier for machines to analyze and understand spoken language or distinguish different sounds.

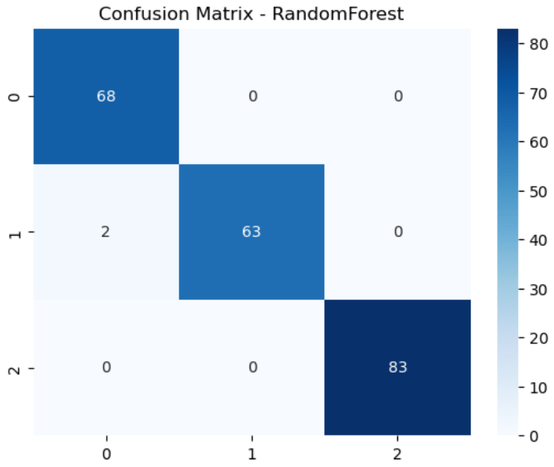
Machine Learning Overview

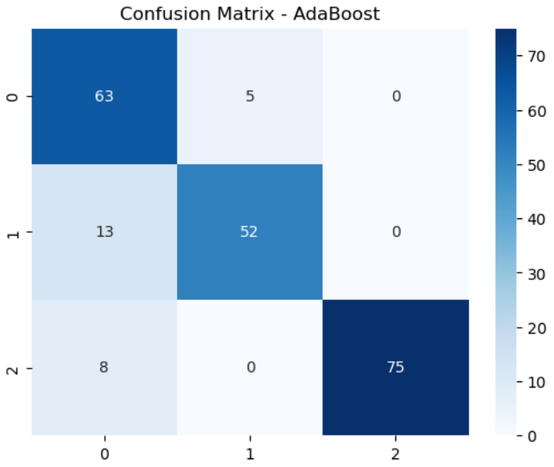


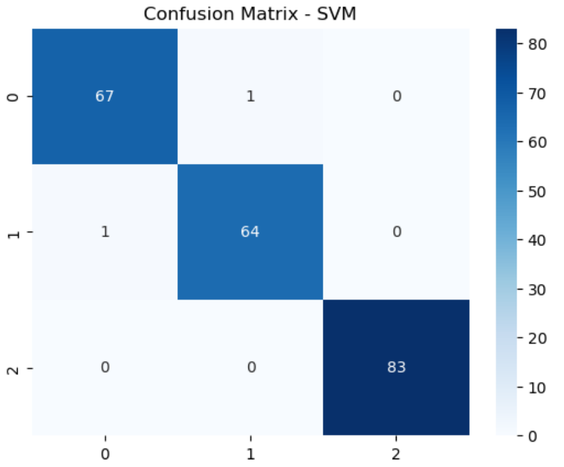
Accuracy Score of Models

	Algorithm	Accuracy
0	Logistic Regression	95.370370
1	Random Forest	99.074074
2	Adaboost	87.962963
3	SVM	99.074074
4	Decision Tree	98.611111
5	KNN	98.148148
6	Multiplayer perceptron	94.907407
7	Navie Bayes	93.518519
8	gradient boosting	99.074074
9	XGBoost	59.722222

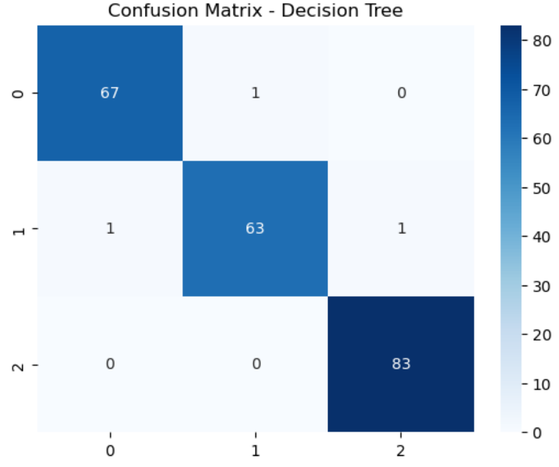


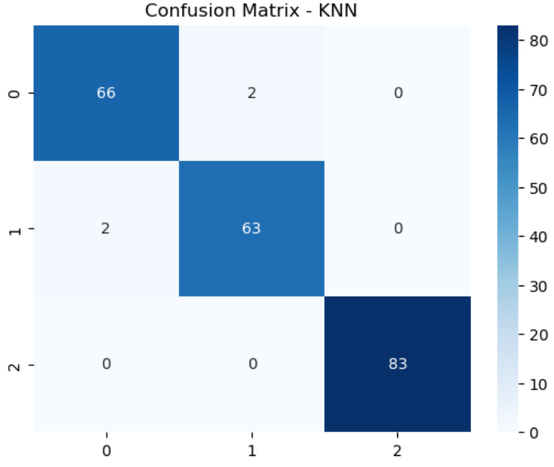


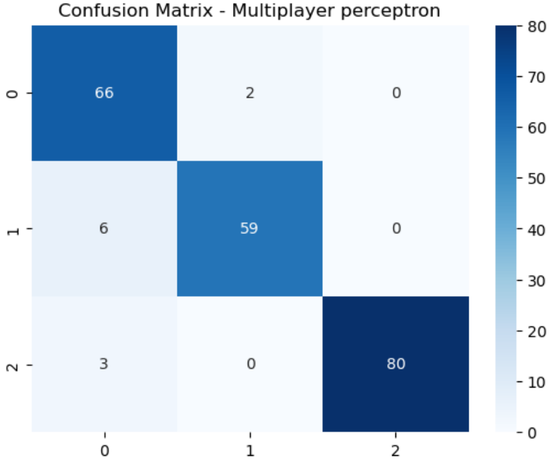


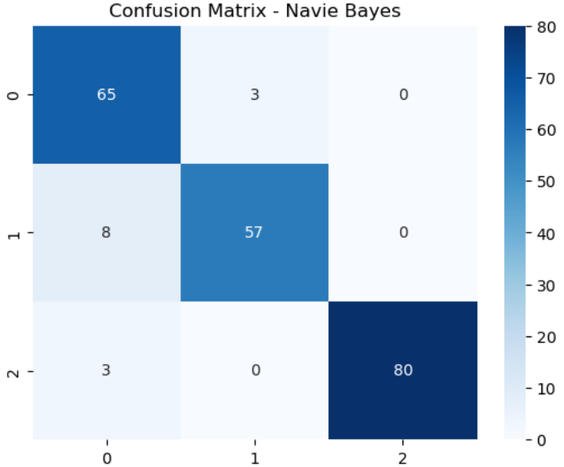


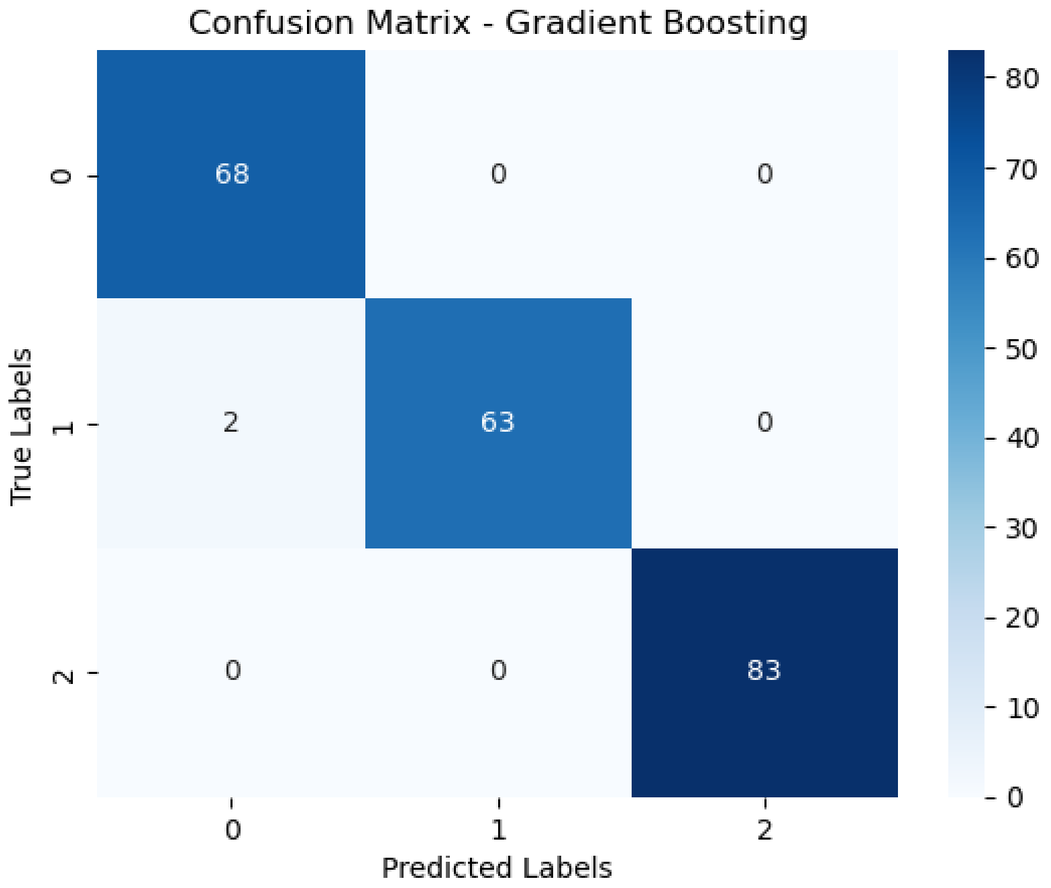
Confusion metric

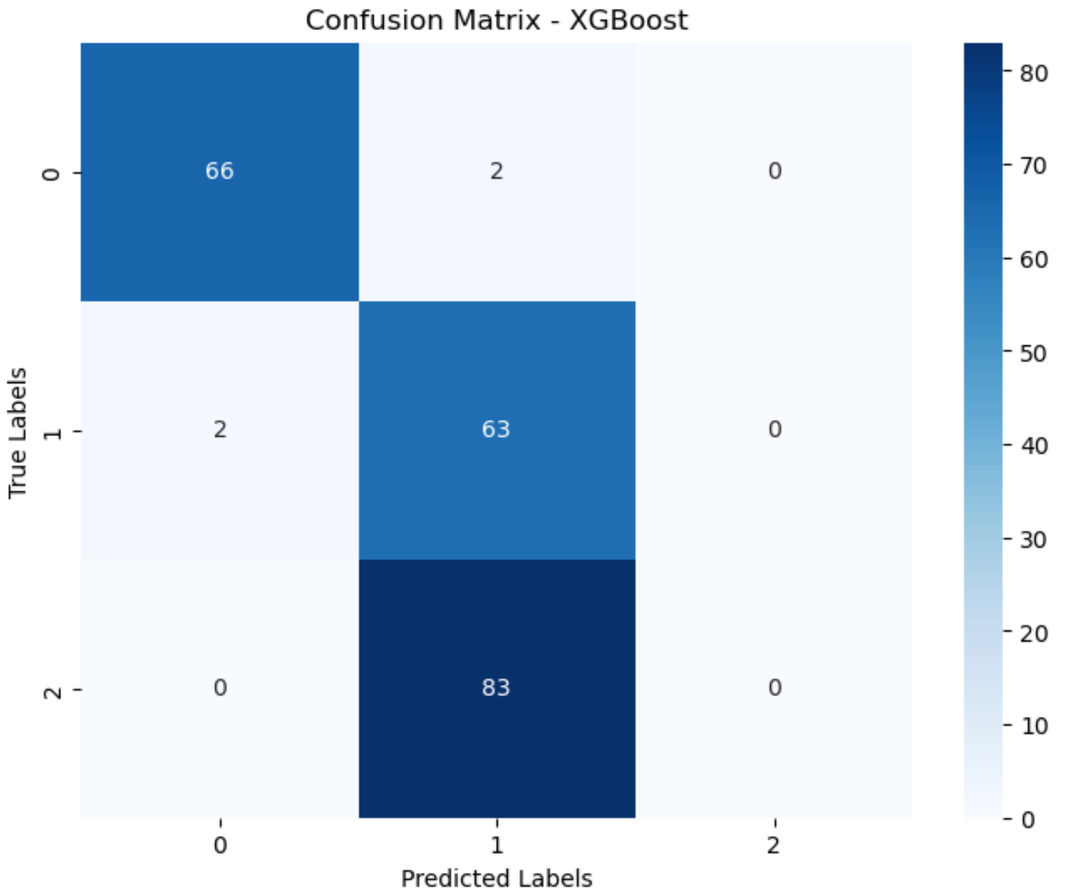




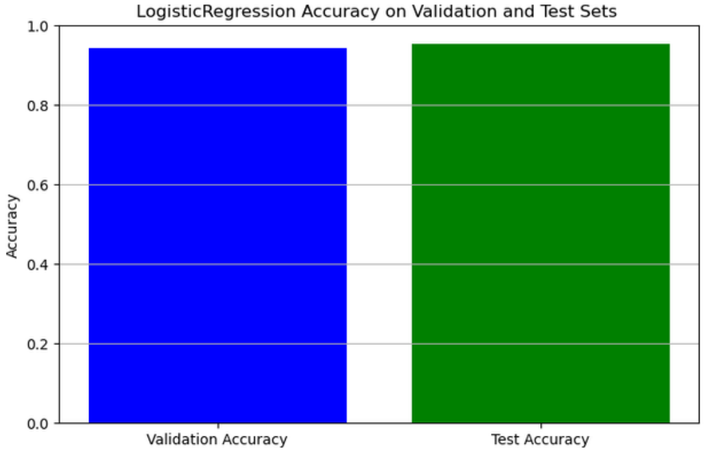


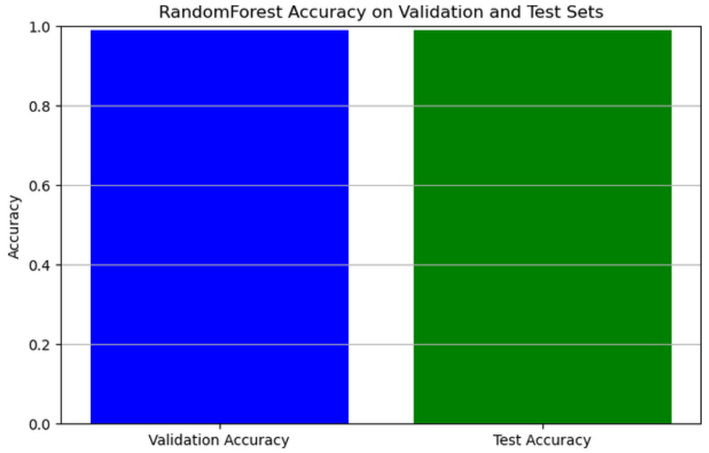


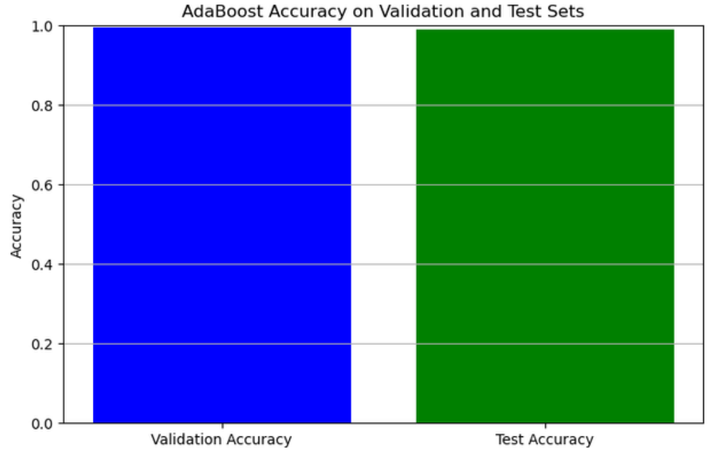

Confusion Matrix - Gradient Boosting
True Labels
Predicted Labels
0
1
2
68
0
0
2
63
0
0
0
83
80
70
60
50
40
30
20
10
0

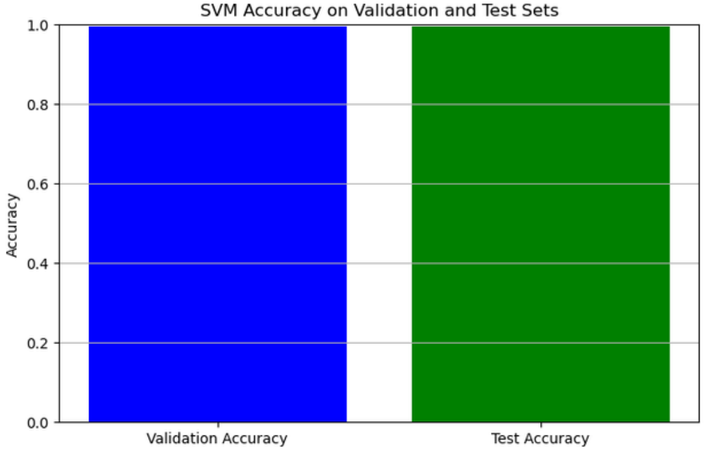

Confusion Matrix - XGBoost
True Labels
Predicted Labels
0
1
2
66
2
0
2
63
0
0
83
0
80
70
60
50
40
30
20
10
0

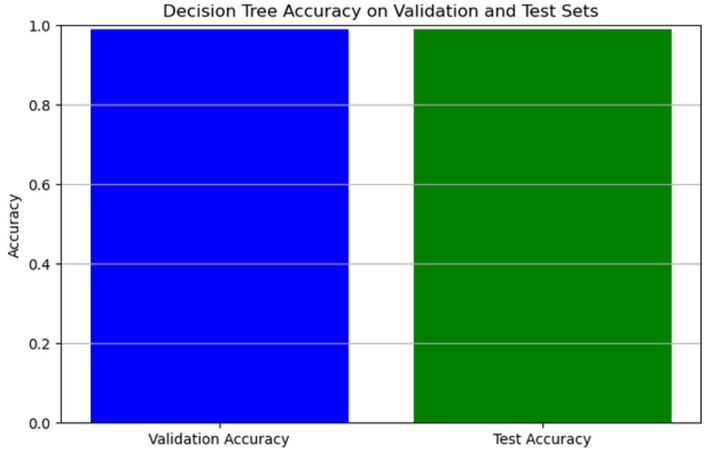
Plot Accuracy of model

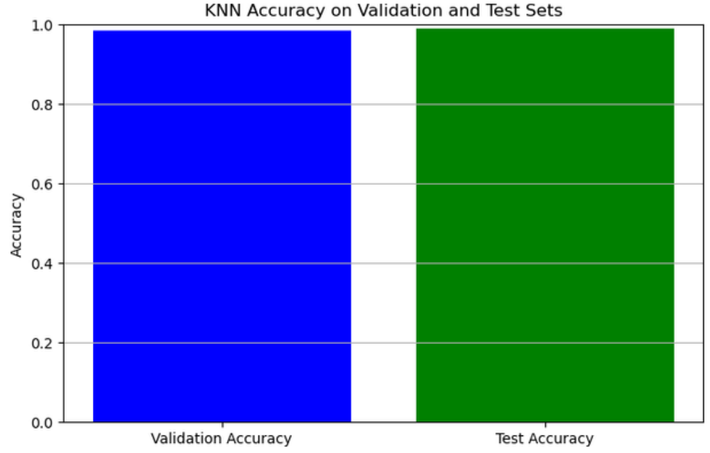


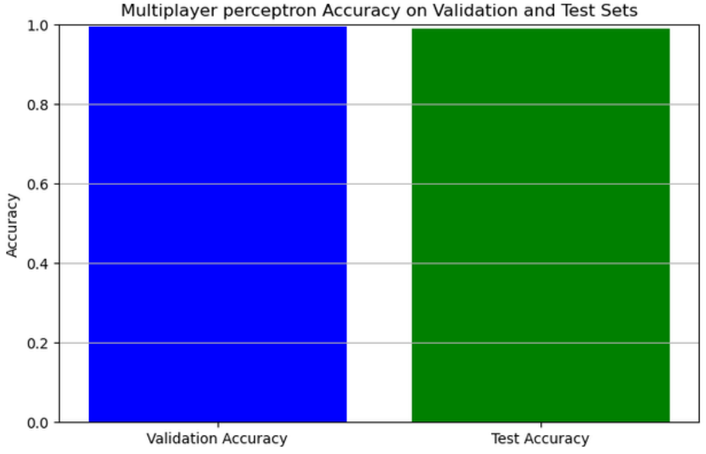


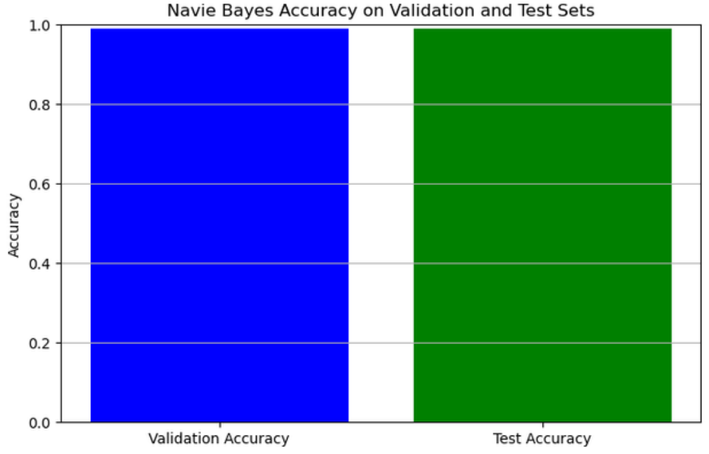


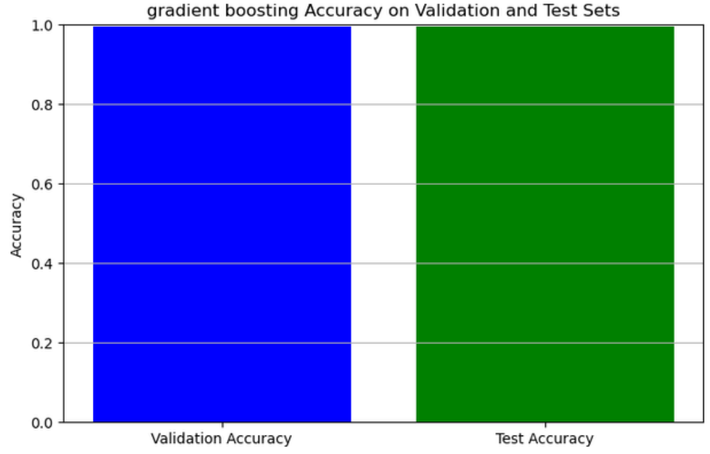




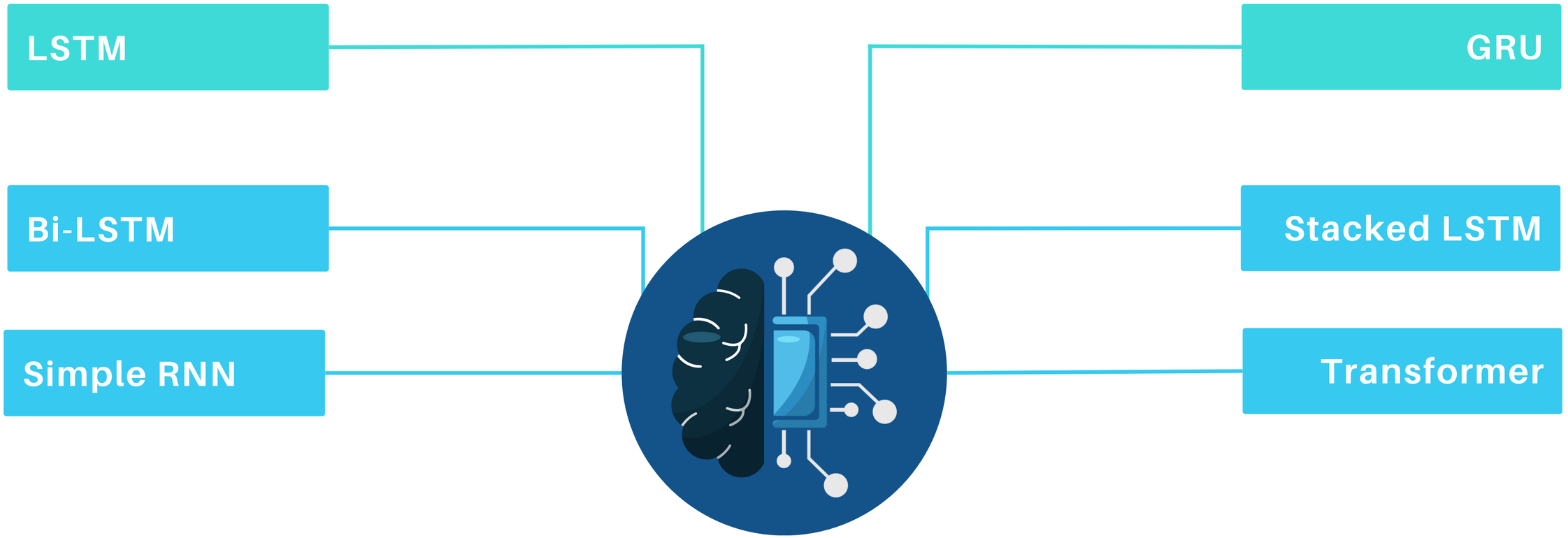






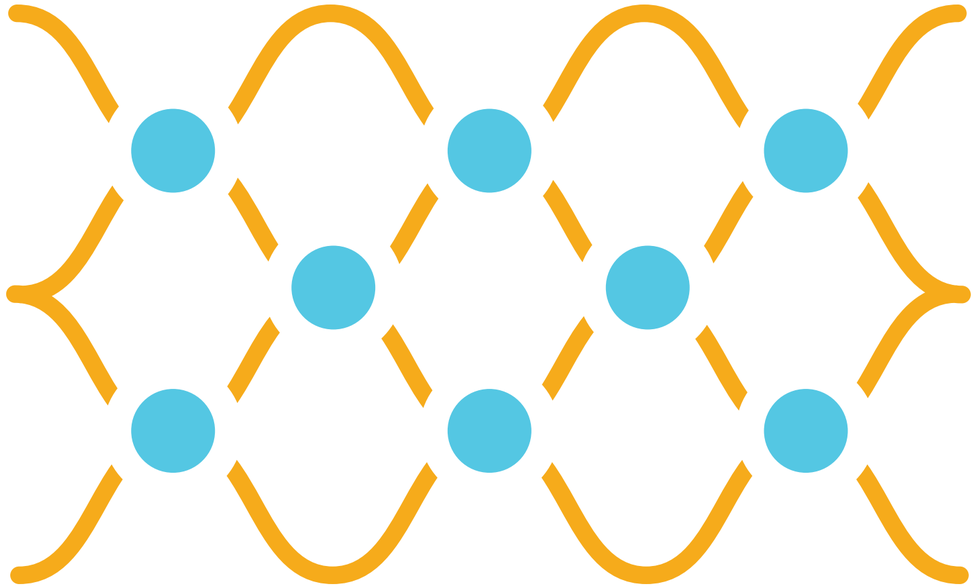


Deep Learning Overview

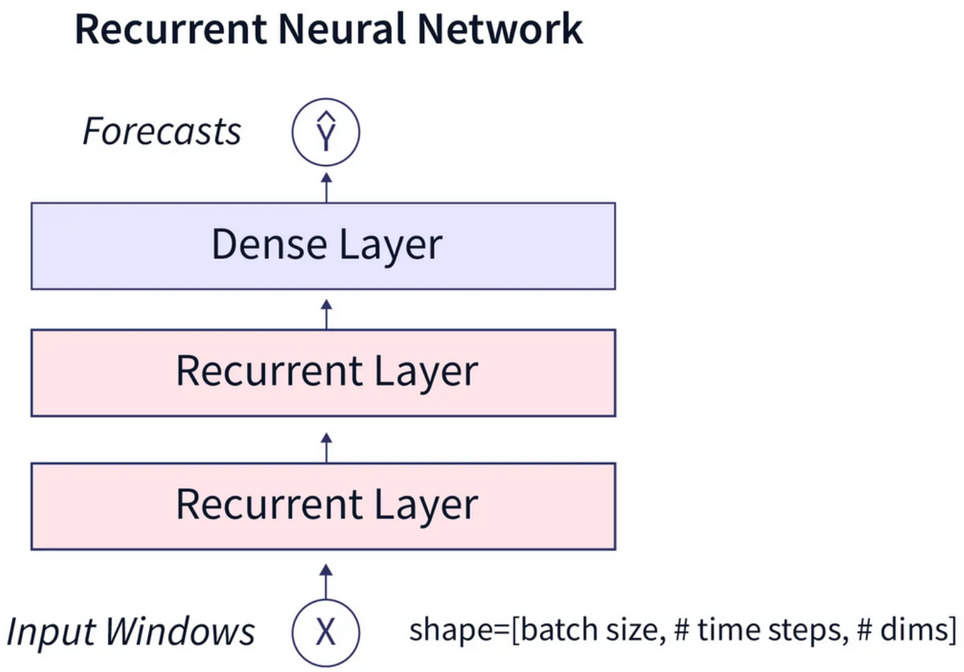


Simple RNN

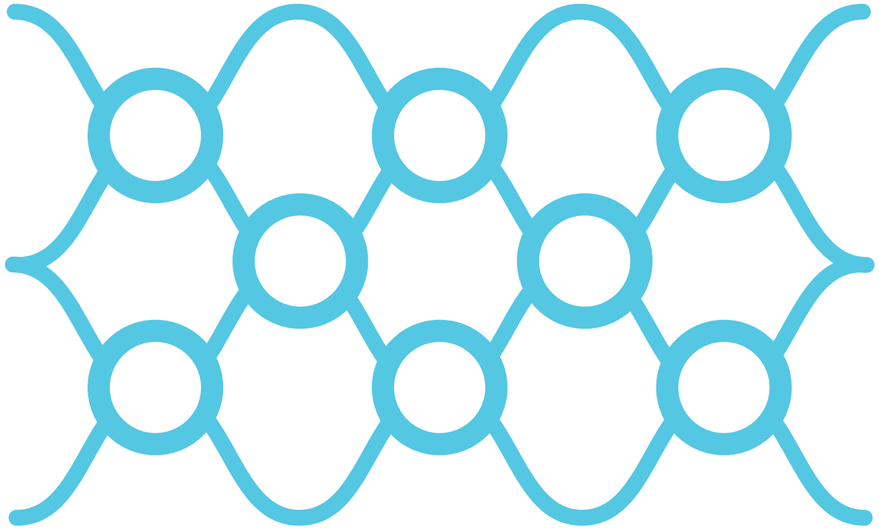
A Simple Recurrent Neural Network (Simple RNN) is a type of artificial neural network designed for processing sequences of data. It's used for tasks that involve sequences, such as time series prediction, natural language processing, and speech recognition. In a Simple RNN, information flows in a loop, where each step in the sequence considers both the current input and the previous step's output. Simple RNNs are used to work with sequential data by capturing patterns and dependencies, but they may not be ideal for long sequences due to their limitations.



Architecture



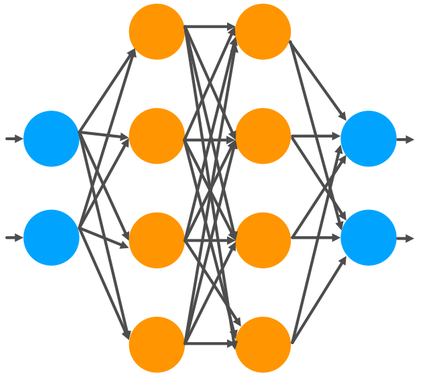
An RNN architecture generally takes a 3-dimensional input, namely batch size, the number of timesteps, and dimensions(can be univariate or multivariate).



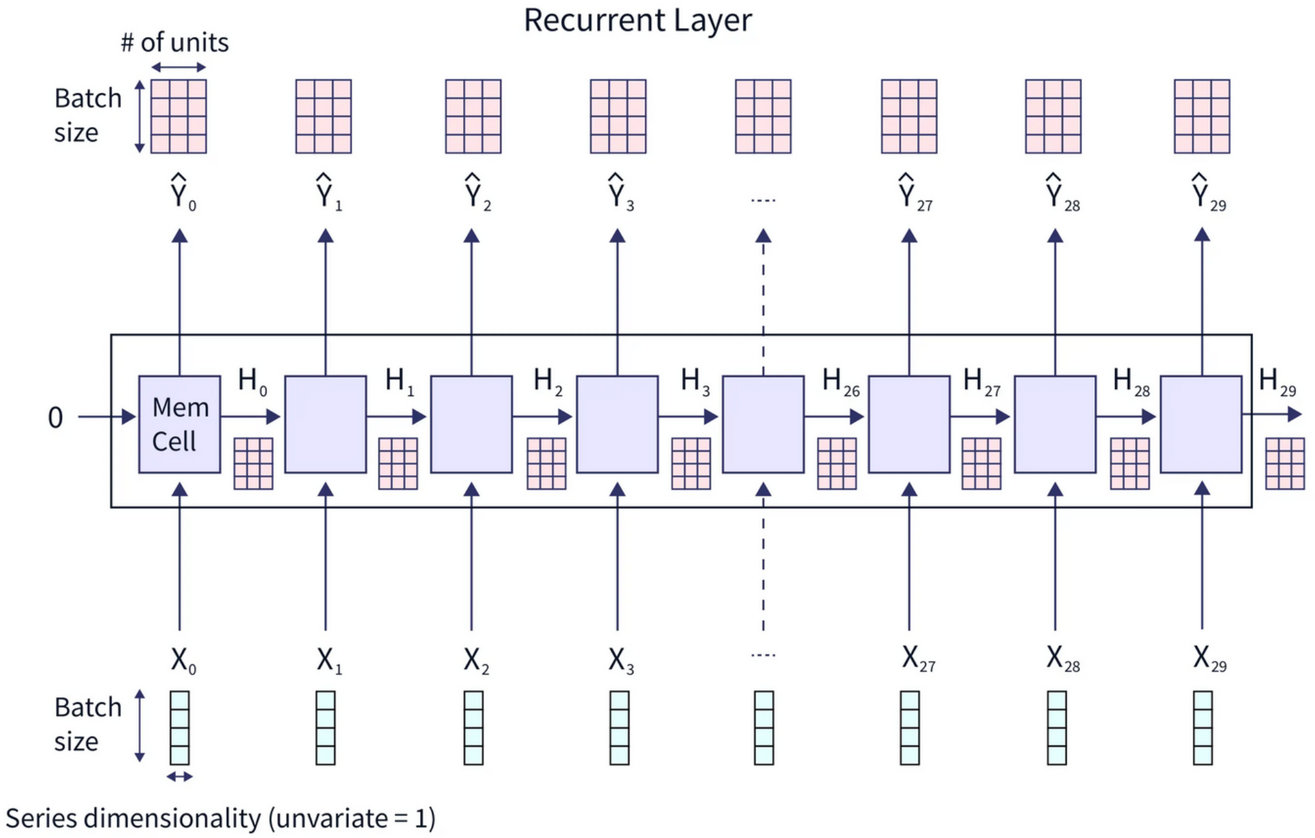
Architecture

A Simple Recurrent Neural Network (Simple RNN) consists of three main components:

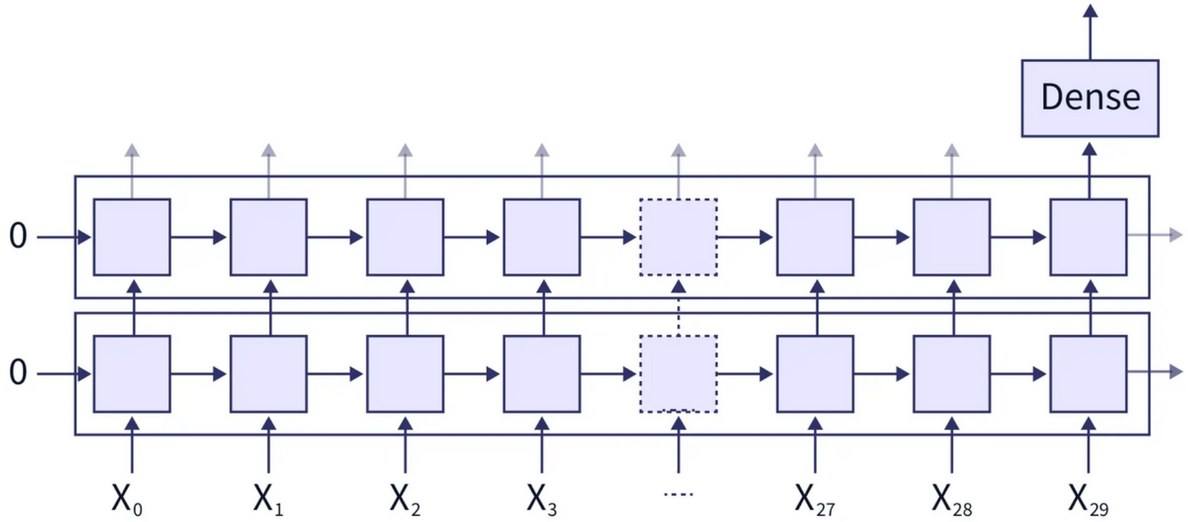
1. Input Layer
2. Hidden Layer (hidden layers are the main features of RNN)
3. Output Layer



- X_0, X_1,.., X_29 are the different timesteps.
- The blue colored boxes are the inputs at different timesteps.
- Each "MemCell" constitutes the hidden layer.
- The orange boxes at the top of the image constitute the units of the hidden layer(3 * 3 in the above case).
- To predict the output at a timestep, we must pass the hidden layer through a dense layer. We will see that next.



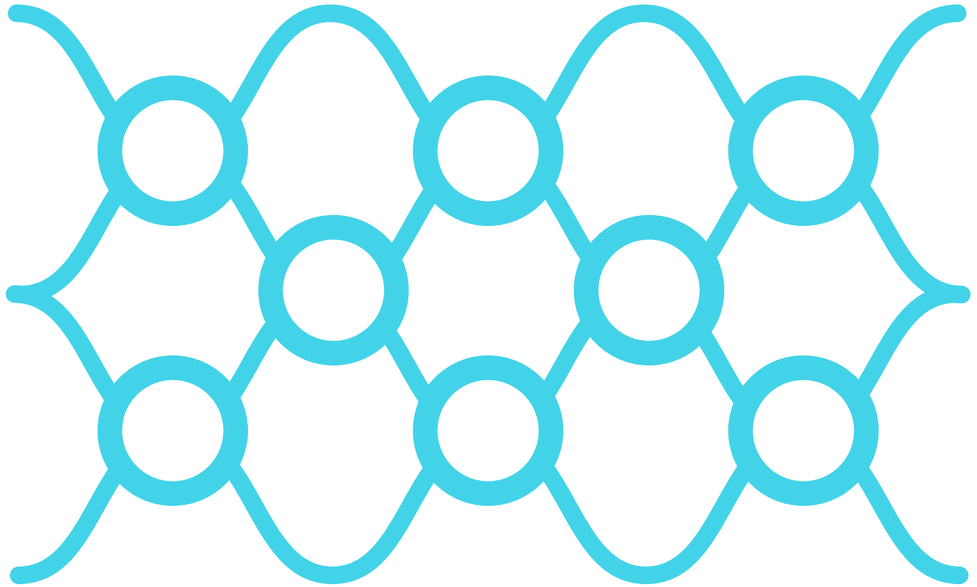
Architecture



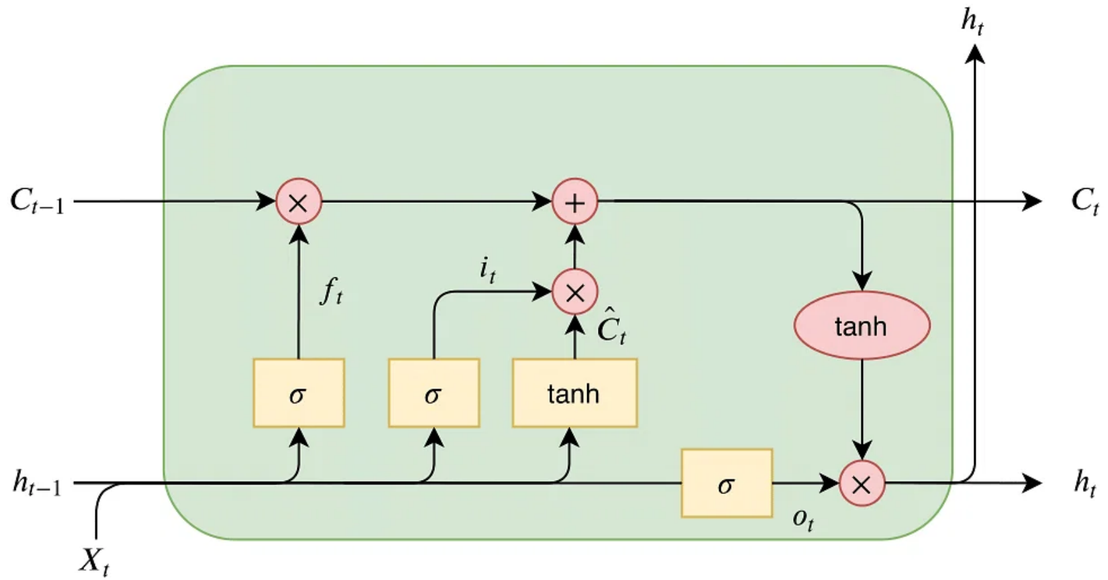
We can see from the above image, there are two recurrent layers and one dense layer. For simplicity, we shall deal with the prediction of the word problem. We have the data for the first 29 timesteps Our task is to predict the word at the 30th timestep. So we need to pass the hidden layer of the rnn architecture in the X_29 timestep through a dense layer to predict the output at the X_30(30th) timestep.

LSTM

LSTM (Long Short-Term Memory) is a type of advanced recurrent neural network designed for processing sequences of data. It's used for tasks that involve sequences, such as natural language processing, speech recognition, and time series prediction. LSTM is a powerful neural network architecture used for tasks involving sequences, known for its ability to handle long sequences and capture complex dependencies in the data.



Architecture

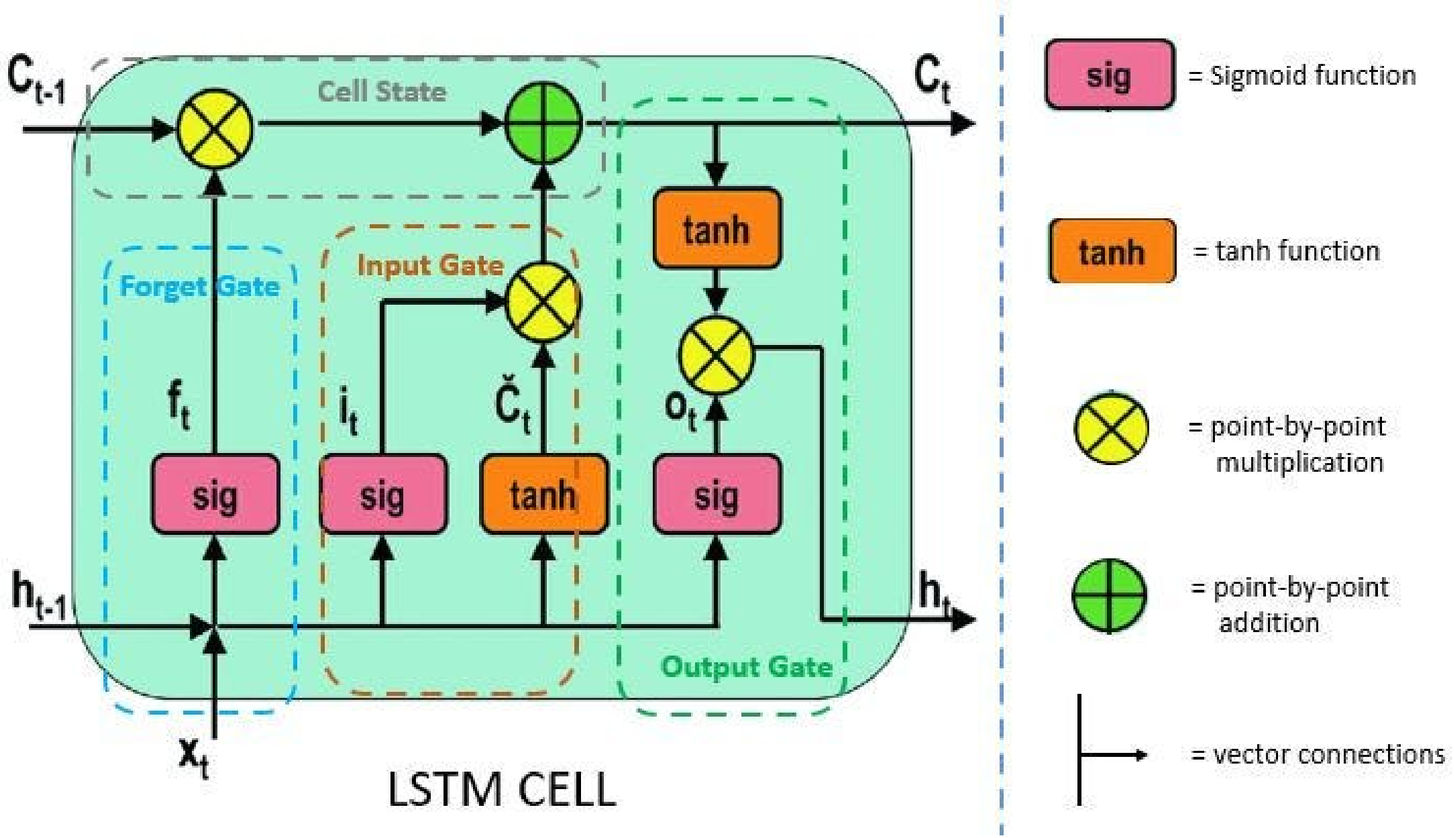


Xt: input time step
ht: output
Ct: cell state
ft: forget gate
it: input gate
Ot: output gate
Äˆt: internal cell state.
Operations inside the light red circle are pointwise.

In LSTM architecture, the outcome of an LSTM at a specific moment in time is influenced by three factors:

- the cell state, which represents the current long-term memory of the network,
- the previous hidden state, which refers to the output from the prior time step,
- and the input data present at the current time step.

Architecture

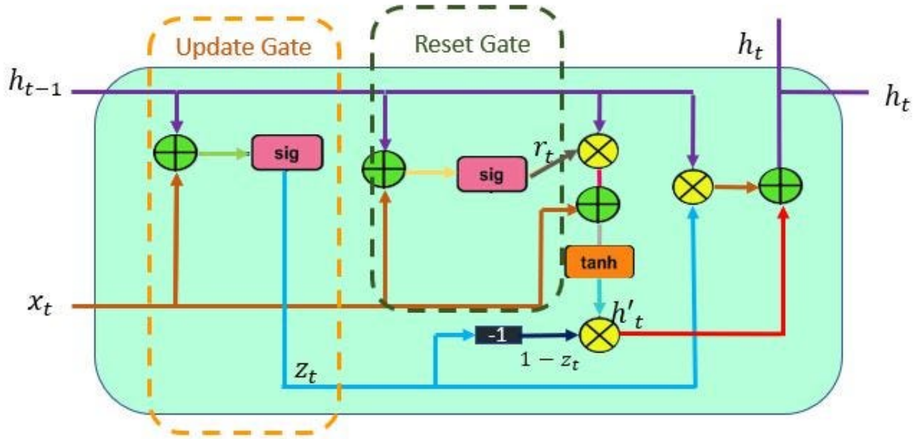


The 3 gates of the LSTM model are the input gate, the forget gate, and the output gate.

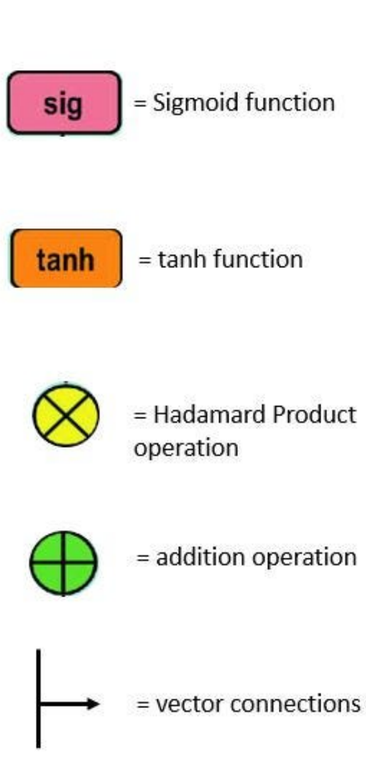
- The input gate decides which new information to add to the cell state.
- The forget gate decides which information to discard from the cell state.
- The output gate decides which information to output from the cell state

GRU Model

The GRU is the newer generation of Recurrent Neural networks and is pretty similar to an LSTM. GRUs got rid of the cell state and used the hidden state to transfer information. It also only has two gates, a reset gate and an update gate.



Gated Recurrent Network (GRU)



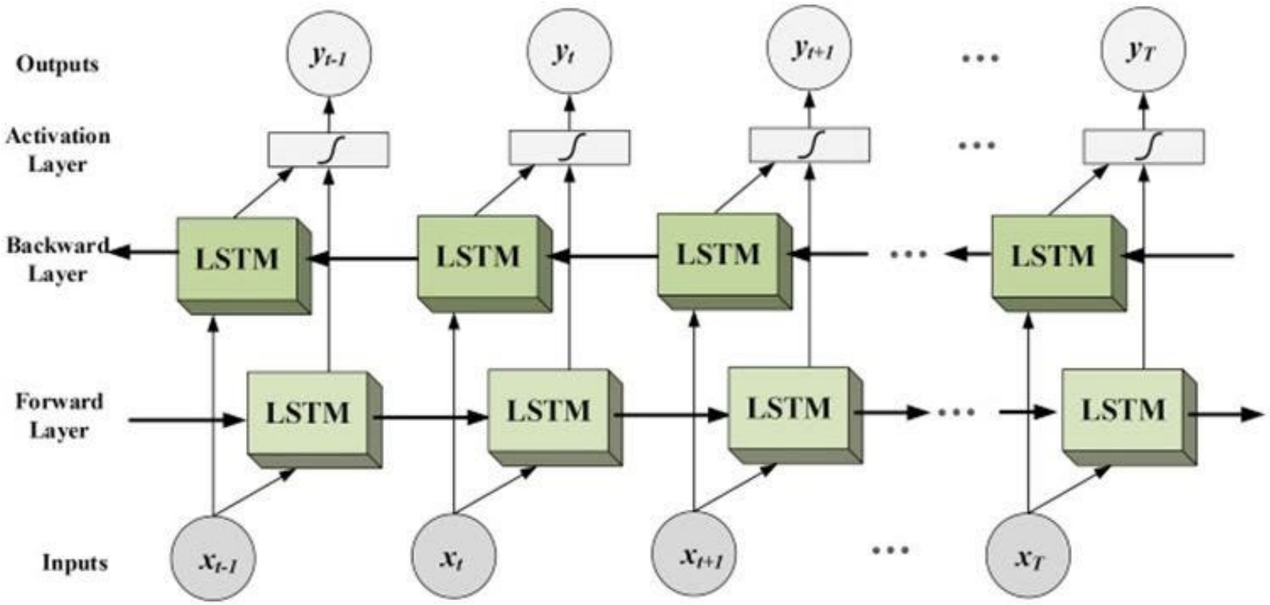
Update Gate

The update gate acts similarly to the forget and input gate of an LSTM. It decides what information to throw away and what new information to add.

Reset Gate

The reset gate is another gate used to decide how much past information to forget.

Bi-LSTM Model

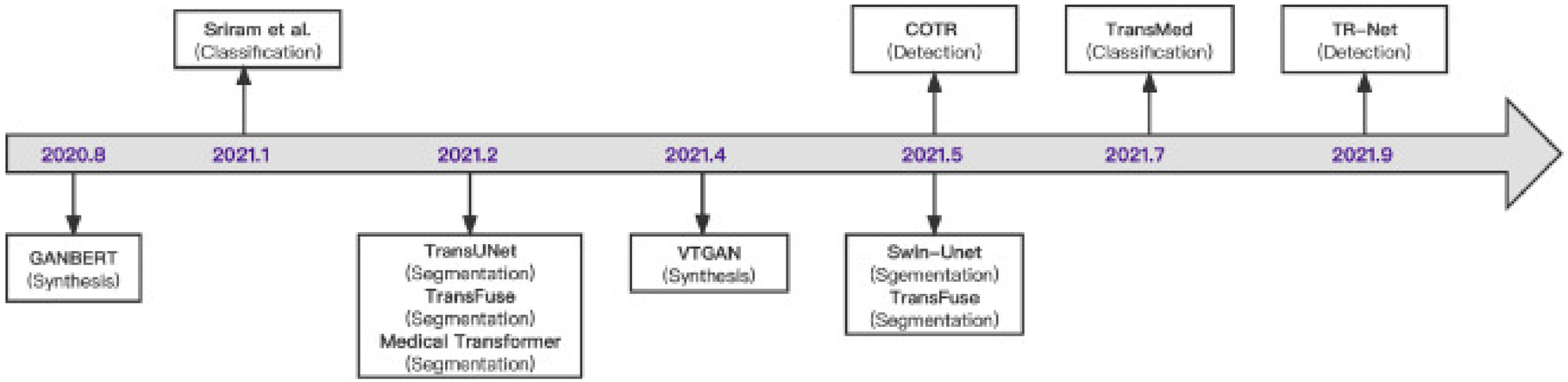


We have seen how LSTM works and we noticed that it works in uni-direction.

Bidirectional long-short term memory networks are advancements of unidirectional LSTM. Bi-LSTM tries to capture information from both sides left to right and right to left. The rest of the concept in Bi-LSTM is the same as LSTM.

Transformer Model

Transformers have dominated the field of NLP, with applications in areas including speech recognition, synthesis , text to speech translation , and natural language generation.



The development of transformers in medical image analysis. Selected methods are displayed relating to classification, detection, segmentation, and synthesis applications.

Preliminaries

A typical transformer leverages the attention mechanism in neural networks. Hence, we start by introducing the core principle of the attention mechanism, followed by a detailed description of how the transformer works.

Attention mechanism

For information exploration, human beings usually leverage their “attention mechanism” to filter out irrelevant information while focusing on the meaningful parts of the data encountered in daily life. Inspired by this observation, researchers have designed attention mechanisms for deep learning that sift through homogeneous data while paying attention to the most significant components or elements.

Attention mechanism in computer vision

Similar concepts have been developed in the field of CV.

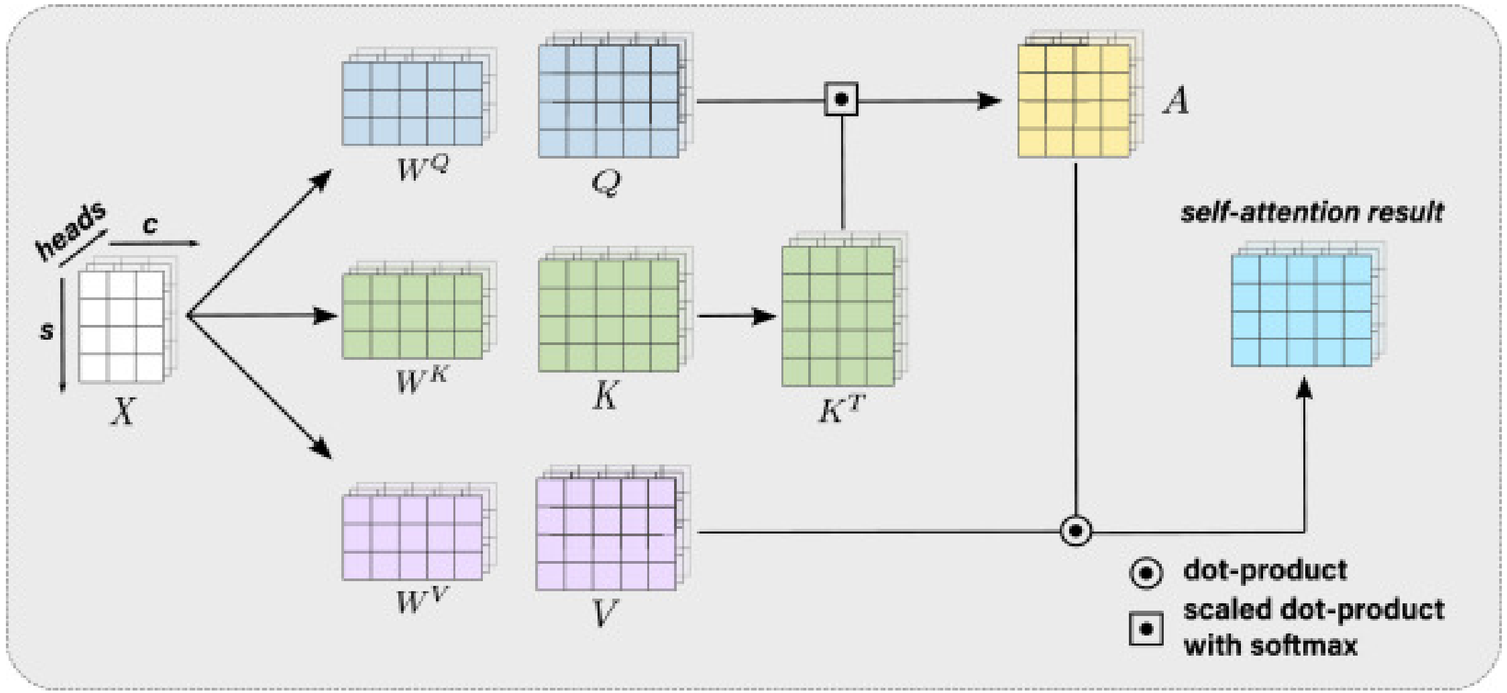
Self-attention.The attention mechanism was re-defined as a function working with queries, keys, and values derived from the input vectors of the module.

Multi-head self-attention. It was shown in that applying multiple self-attentions to the same input could better capture hierarchical features. These self-attention layers work similarly to multiple kernels in convolution layers.

$$Z_i = \text{Attention}\left(Q \times W_i^Q, K \times W_i^K, V \times W_i^V\right),$$

$$\text{MultiHead}\left(Q, K, V\right) = \text{Concat}\left(Z_1, \cdots, Z_h\right) W^O,$$

A brief illustration of a self-attention mechanism.



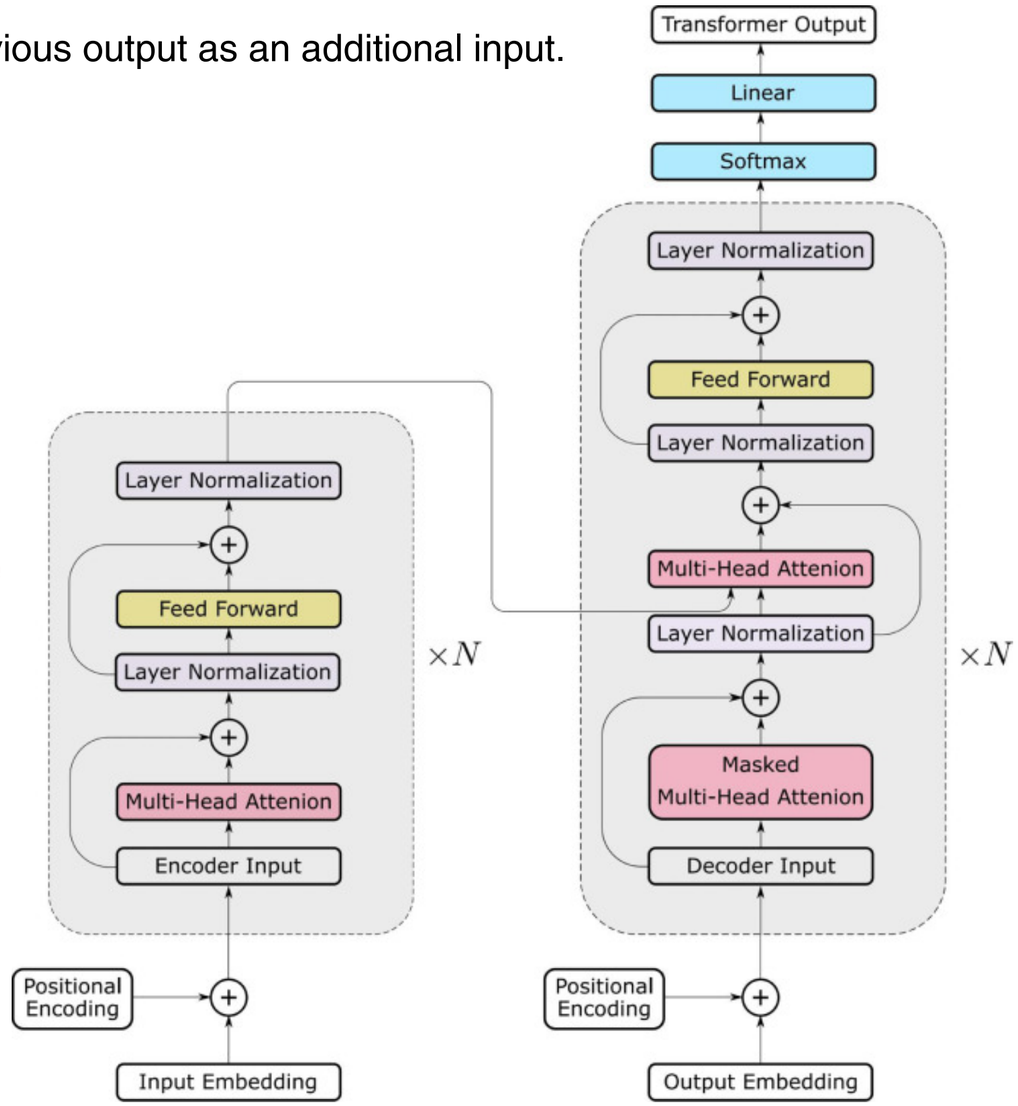
where W_i^Q, W_i^J, W_i^V denote linear projection matrices that map matrices Q, K, V into different subspaces, respectively.

Architecture

A typical transformer network with an encoder–decoder structure. The encoder maps an input sequence $\{x_1, \cdots, x_n\}$ to an output sequence $\{z_1, \cdots, z_n\}$ of the same length. The decoder generates the output $\{y_1, \cdots, y_m\}$ from the encoded representation in an element-wise manner and takes the previous output as an additional input.

A typical transformer architecture is

A brief illustration of a typical transformer architecture



Encoder

The encoder in a typical transformer has n= 6 stacked blocks consisting of two types of layers, i.e., the multi-head attention layer and the feed-forward layer. Residual connections and layer normalization layers are combined with the aforementioned layers. Concretely, in each block, the multi-head attention is first calculated, followed by a layer-wise normalization, calculating the sum of the input and output of the multi-head attention. This is followed by a feed-forward layer, then a layer-wise normalization of the sum of the feed-forward layer's input and output.

Decoder

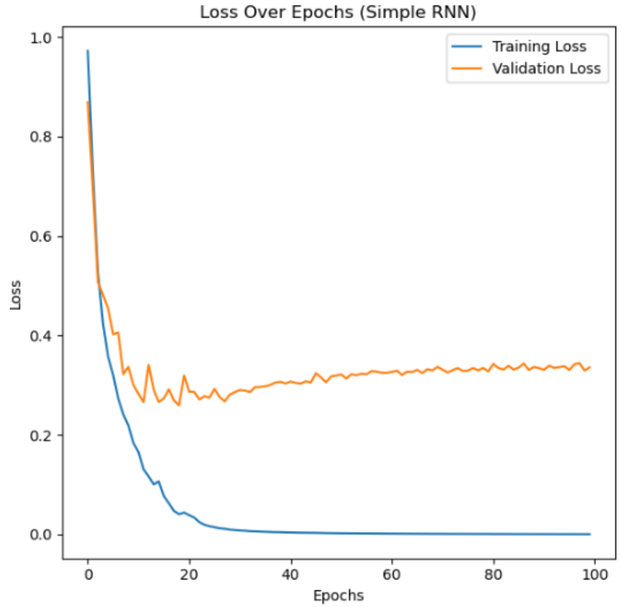
The decoder also has n=6 blocks, similar to the encoder, with some minor modifications. Specifically, an additional self-attention layer is inserted on top of the encoded output. Masking is employed in the first self-attention layer to block subsequent contributions to the state of the previous position, as the prediction is based on a known state. A linear layer and a Softmax layer are inserted after the output of the decoder to generate the final output.

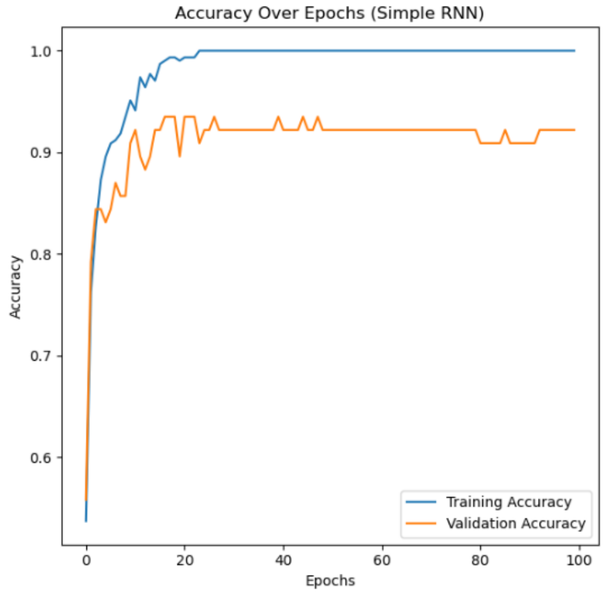
Transformers in medical image applications

Applications of transformers in medical image analysis

Transformers have been widely used in full-stack clinical applications. In this section, we first introduce transformer-based medical image analysis applications, including classification, segmentation, image-to-image translation, detection, registration, and video-based applications.

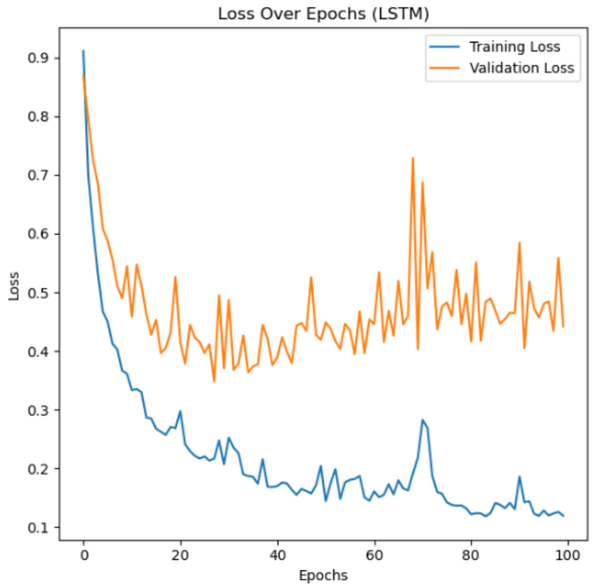
Simple RNN Model

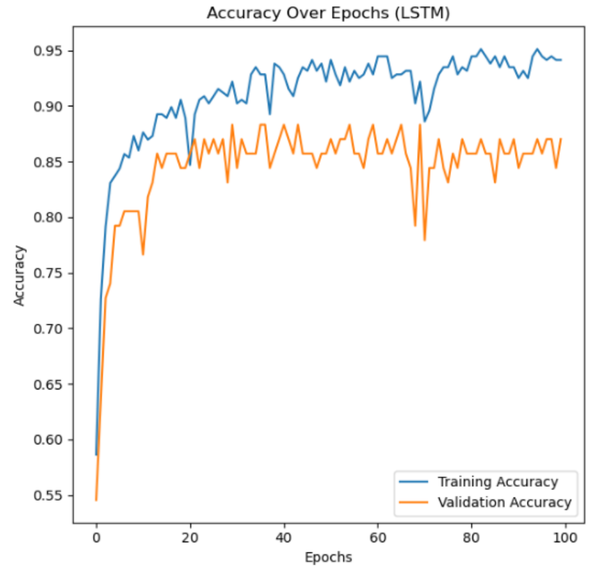




Simple RNN Accuracy: 73.96%

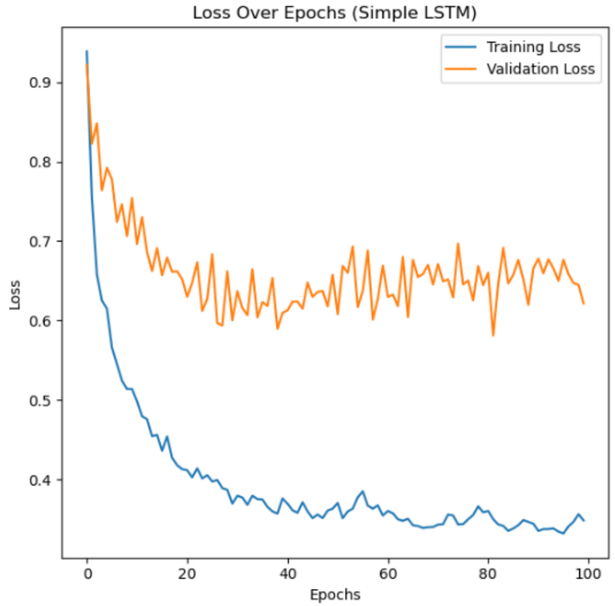
LSTM Model

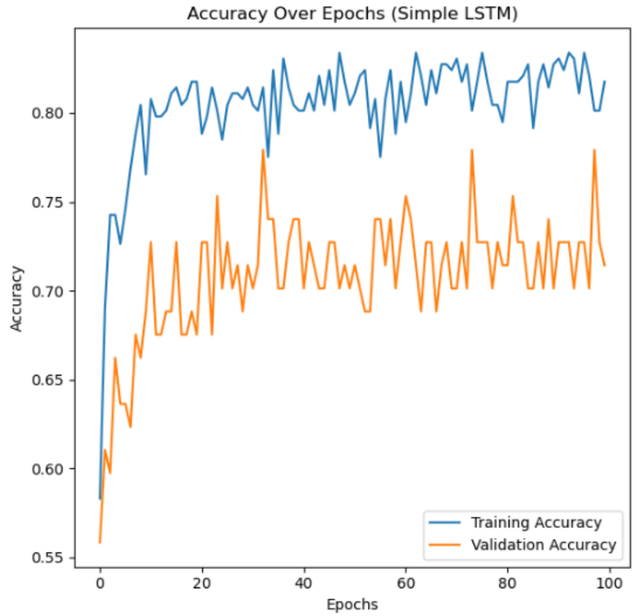




LSTM Accuracy: 97.92%

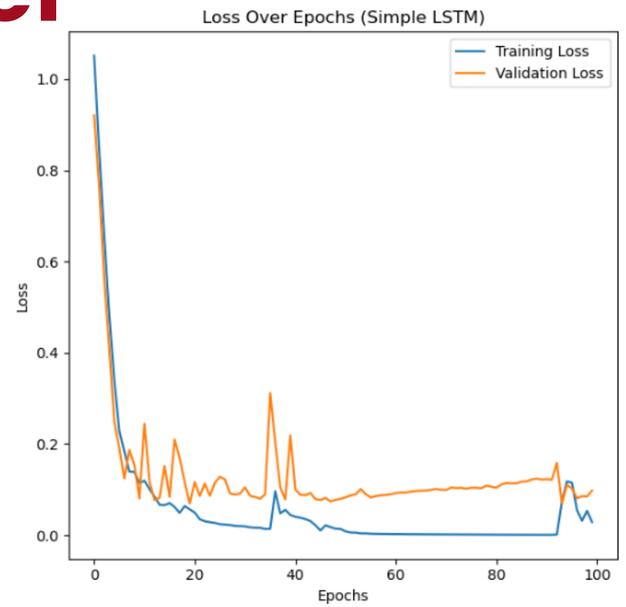
GRU Model

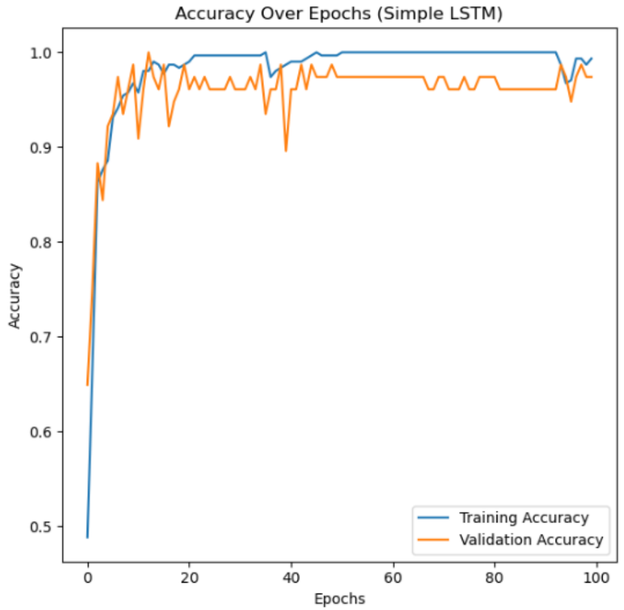




GRU Accuracy: 93.75%

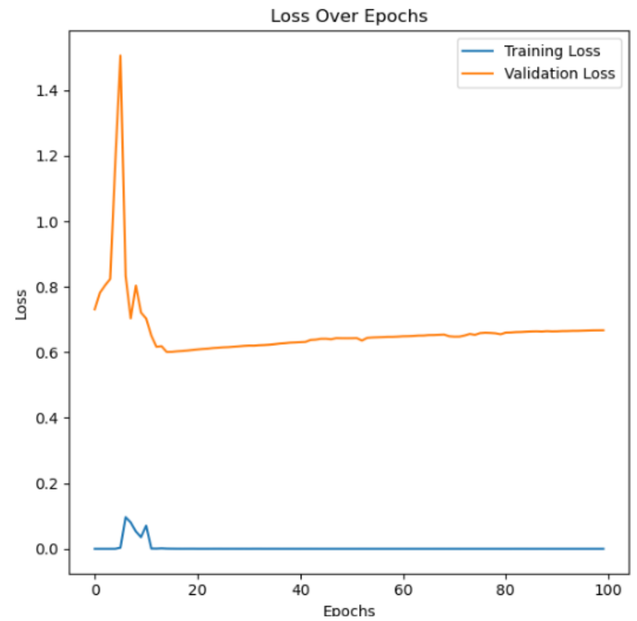
Bi-LSTM Model

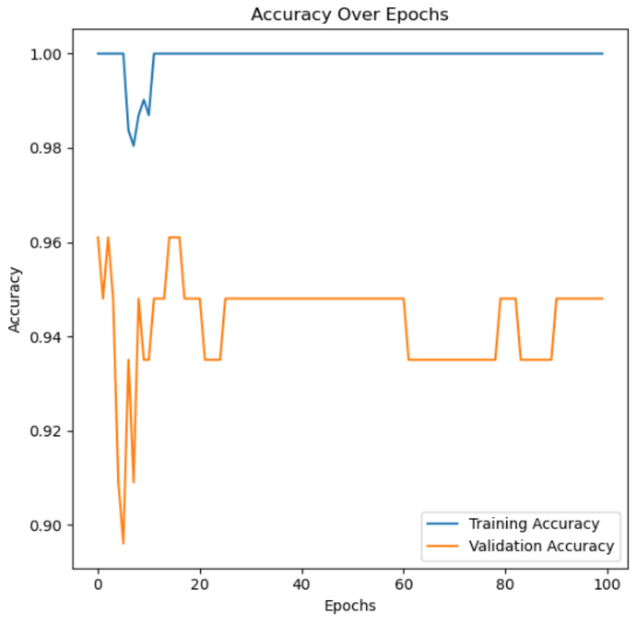




Bidirectional LSTM Accuracy: 96.88%

Transformer Model

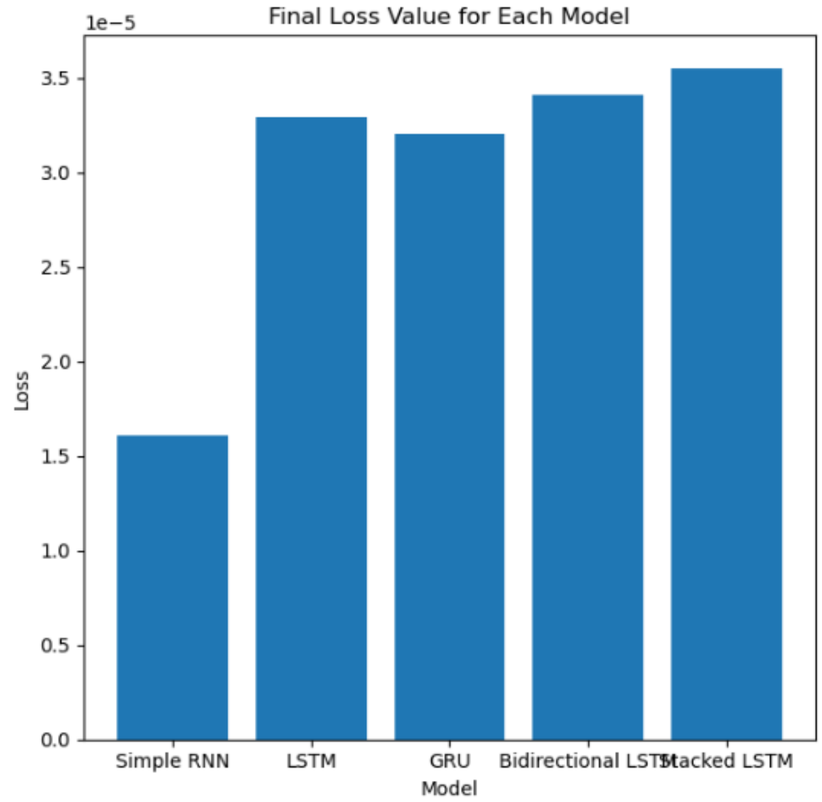


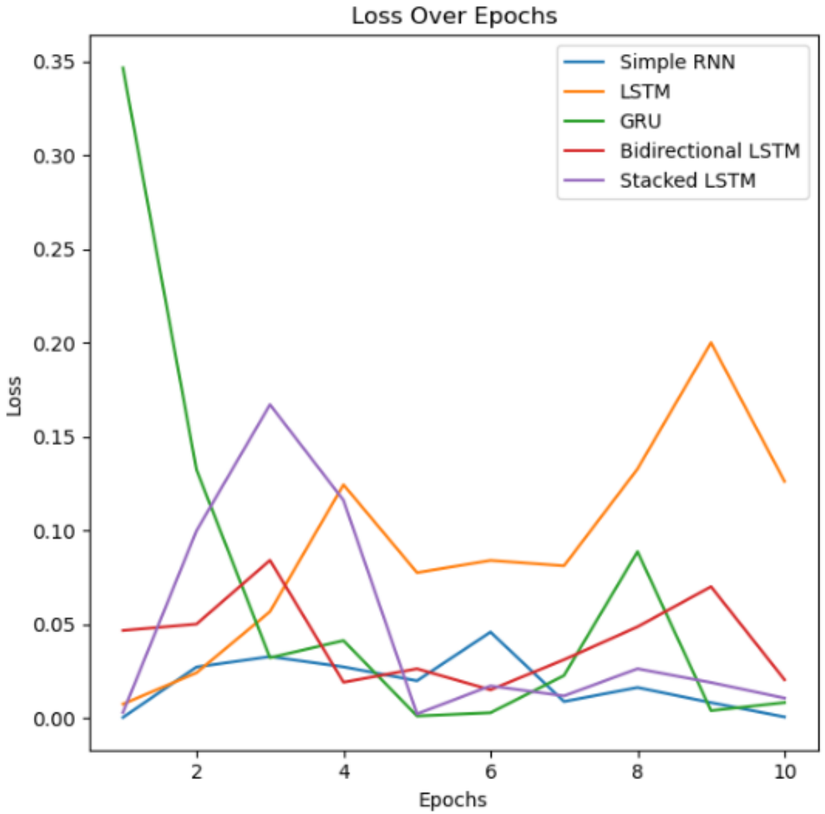


Transformer Accuracy: 96.88%

Result

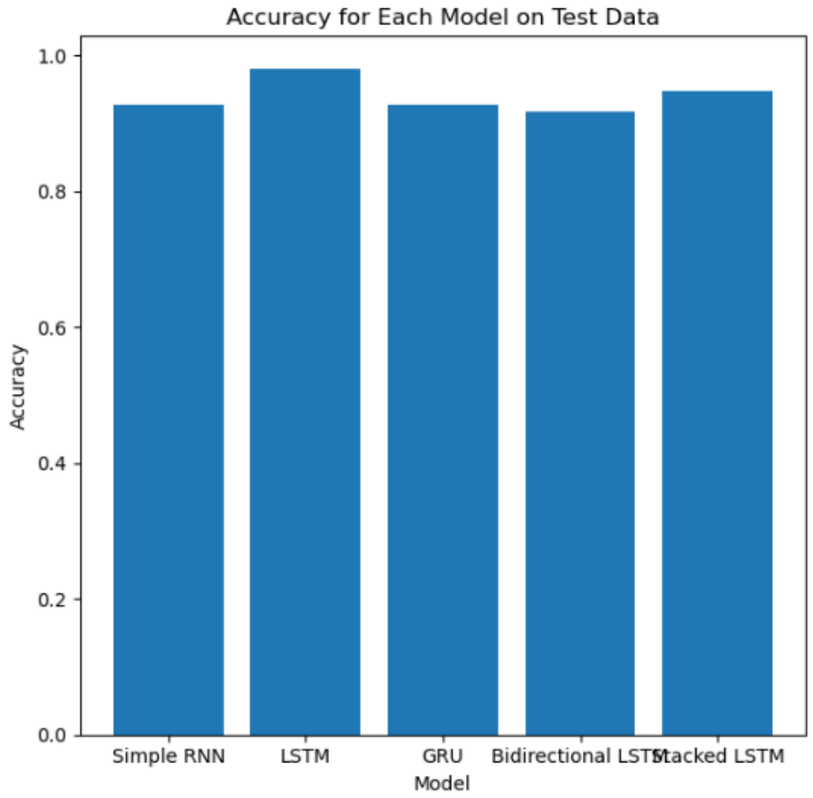
Loss function

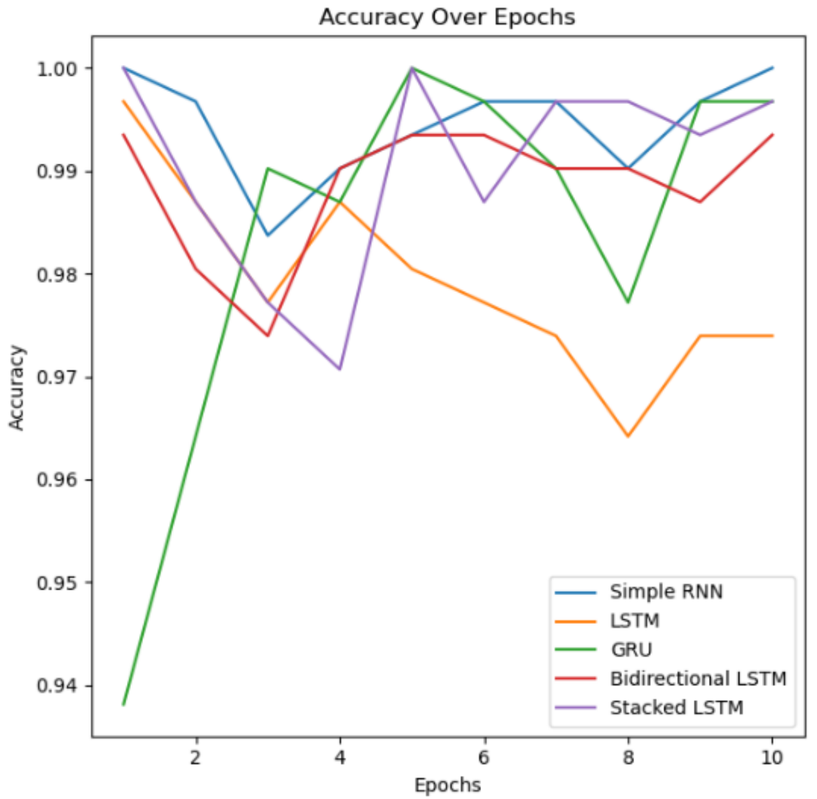




Result

Loss function



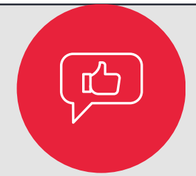


Result

Analysis of the Results Achieved Through Machine Learning

Accuracy

The Machine learning model (Random Forest,SVM,gradient boosting)achieved an accuracy of 99.07%, demonstrating its potential for heart sound classification.



Speed

The classification process using machine learning was 70% faster than traditional methods, allowing for quicker diagnosis and treatment.

Future work

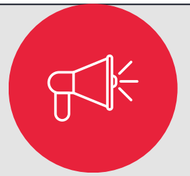
Exploration of Potential Future Advancements in Heart Sound Classification Using Deep Learning

Adapting to Different Languages

Creating deep learning models that can accurately classify heart sounds in multiple languages.

Real-Time Heart Sound Classification

Developing models that can provide real-time heart sound classification, helping healthcare providers diagnose heart conditions faster.



Remote Heart Sound Analysis

Creating models that can accurately classify heart sounds captured remotely, making it easier to monitor patient health from a distance.

Conclusion

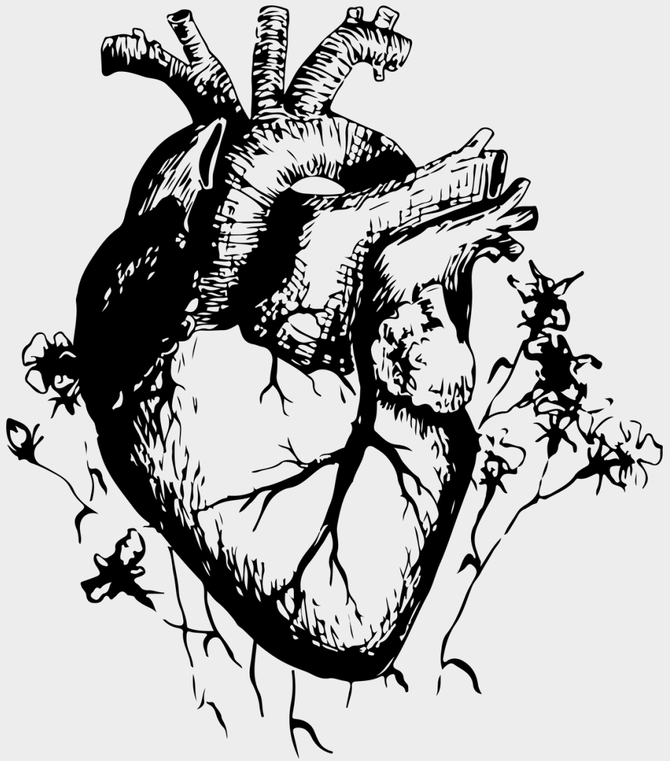
1

1 Implications of the Study

Heart sound classification using machine learning can improve the accuracy and speed of heart disease detection, leading to better patient outcomes.

2 Future Research Directions

Further research should focus on enhancing the accuracy of deep learning models, integrating them into clinical settings, and improving their accessibility to healthcare practitioners.



Thank You