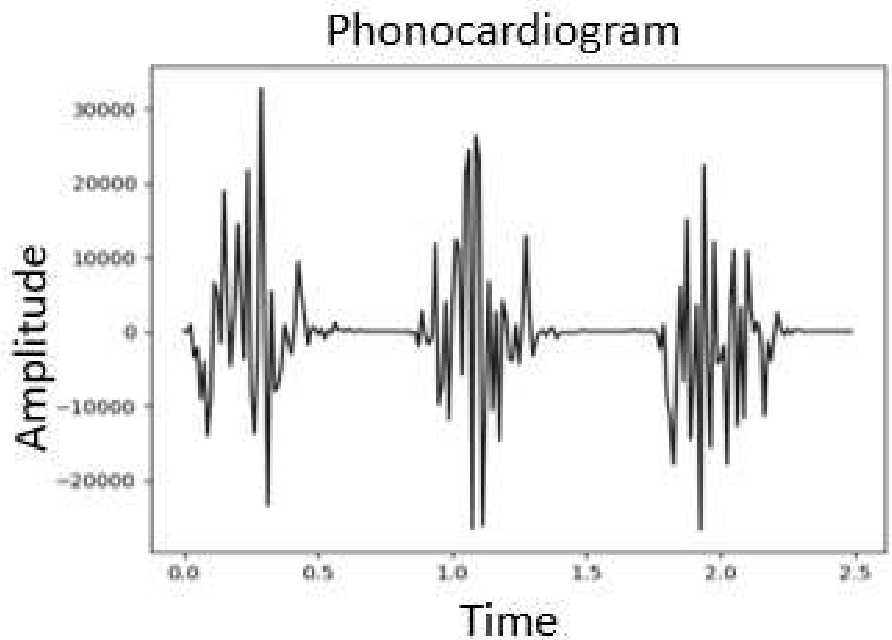
Deep learning in Time Series Signal

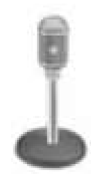
What is 1D Signal?

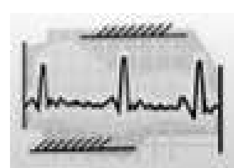
A 1D "Signal" describes how some physical quantity varies over time or space.



Examples of Signals

- Sound pressure
- Radio broadcast
- Songs
- Electrocardiogram



What is 1D Signal Processing?

Manipulating a signal to change its characteristics or extract information

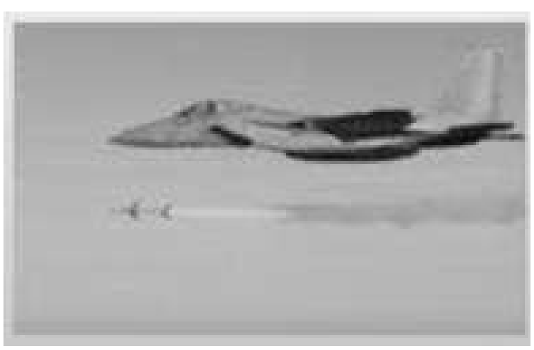
Performed by:

- Computer
- Special purpose integrated circuits
- Analog electrical circuits

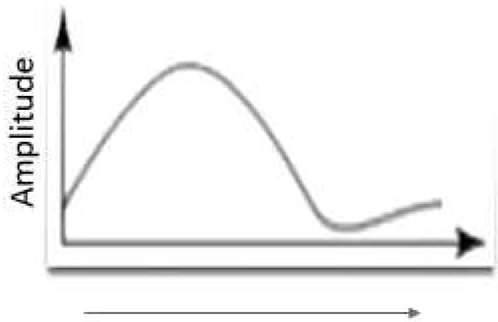
Applications

- Consumer electronics like HDTV, cell phones,…
- Transportation like GPS, engine control, tracking…
- Medical Monitoring like(EEG, ECG, EMG,…)
- Military like Target tracking, surveillance,…
- Remote sensing like Astronomy, climate monitoring, weather forecasting,…



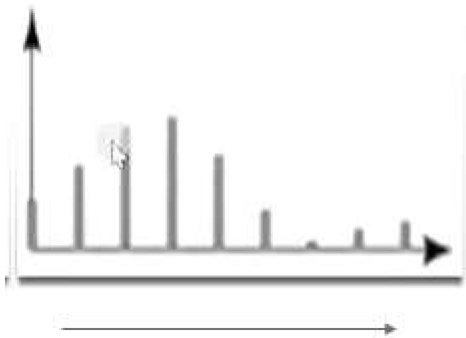



PCM in Signal



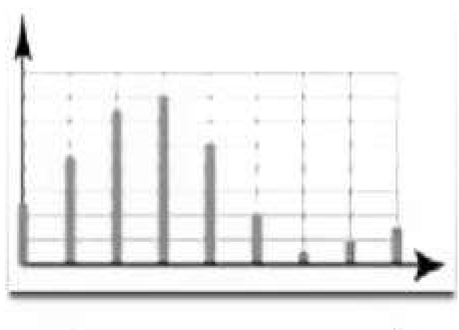
Time: 1sec

Analog Signal



Time: 1sec
Sampling rate: 9

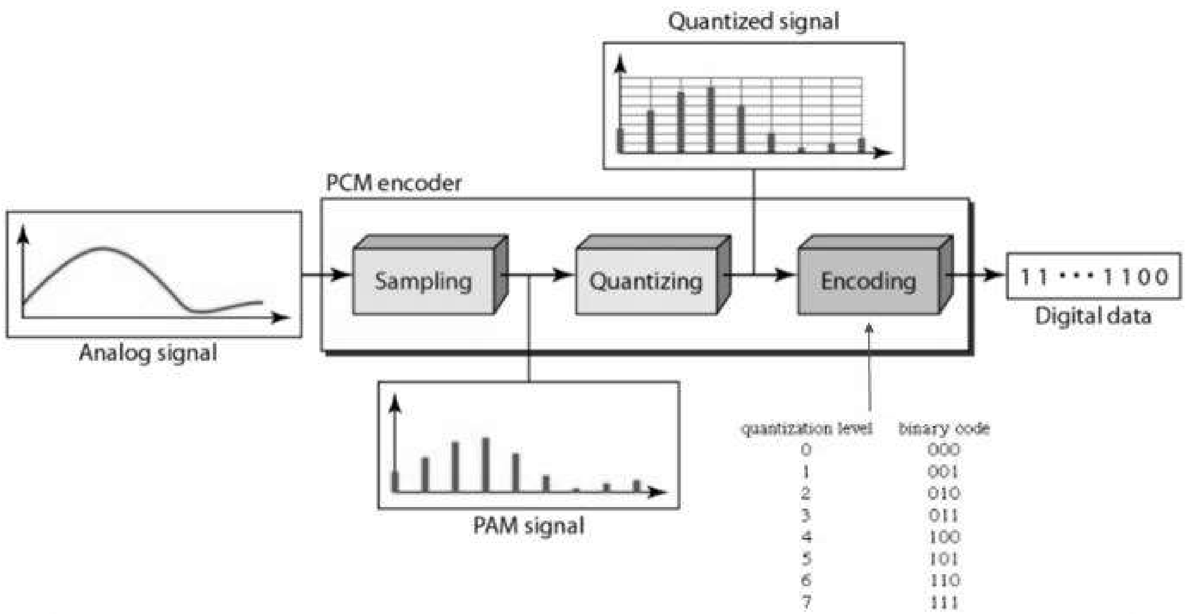
Sampled Signal



Time: 1sec
Sampling rate: 9
Bitpersample:8

Quantized Signal

PCM in Signal

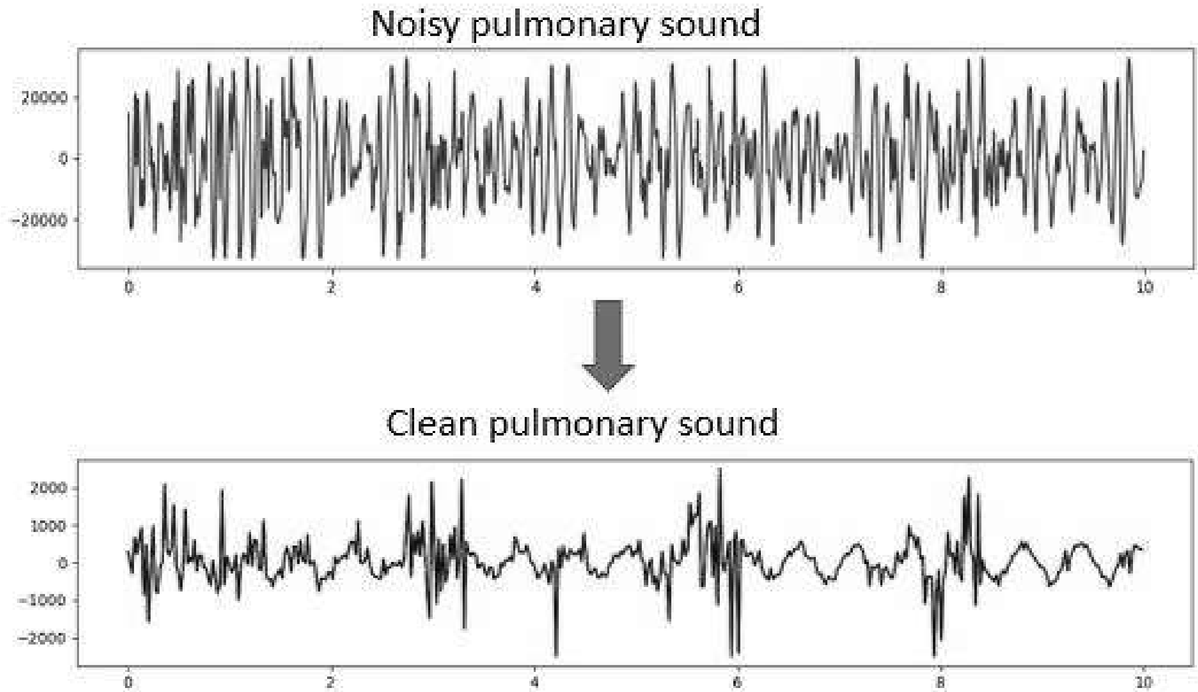


Typical 1D Signal Processing Problems

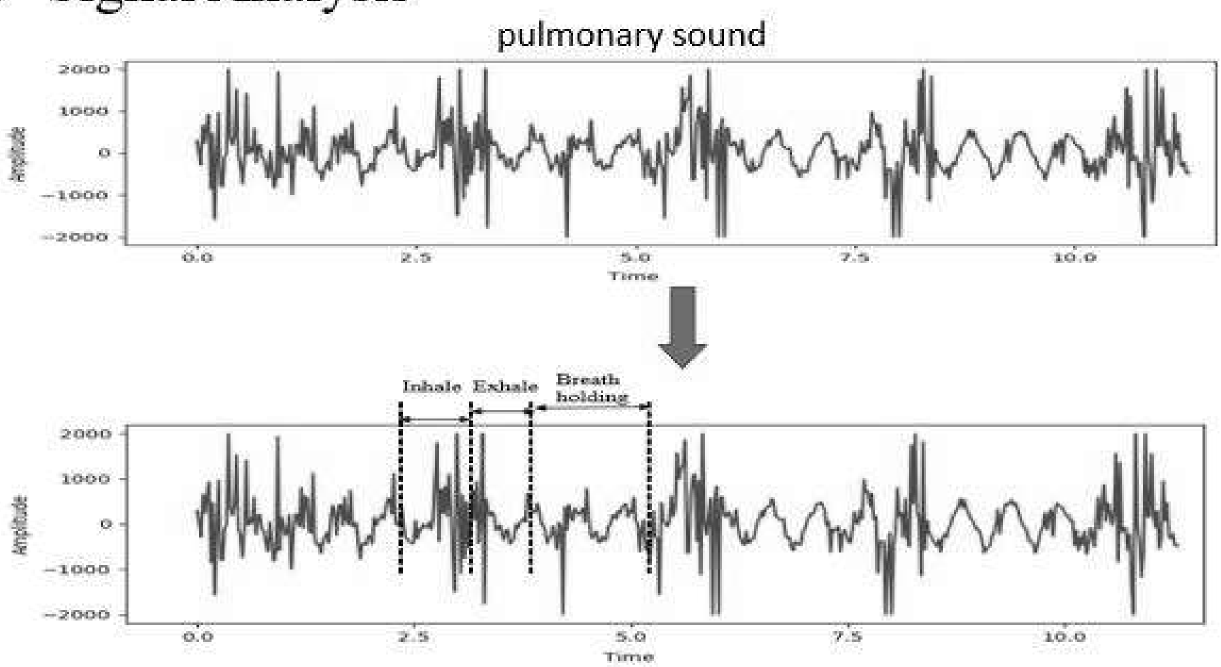
1. **Signal Filtering**
2. **Signal Analysis**
3. **Extracting an indirect quantity from measured signal**

Typical 1D Signal Processing Problems

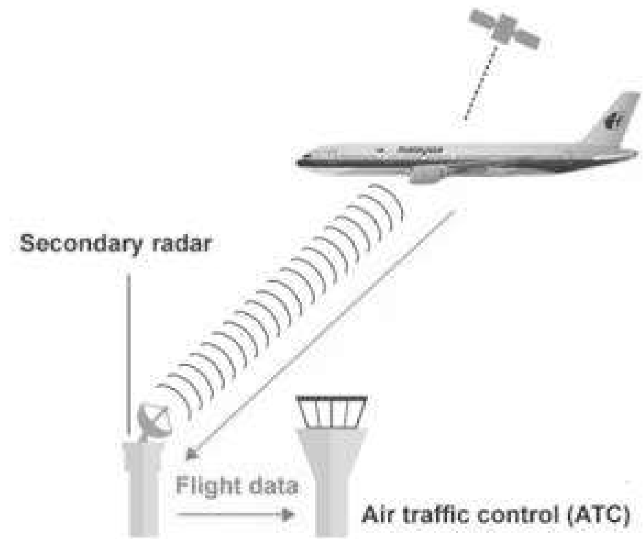
1. Signal Filtering



2. Signal Analysis



3. Extracting an indirect quantity from measured signal

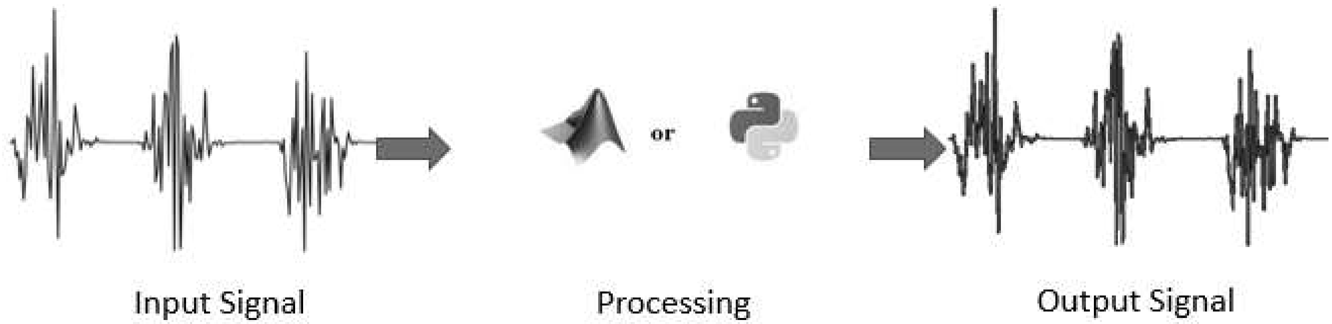


E.g. Determining the aircraft position and velocity from radar signal

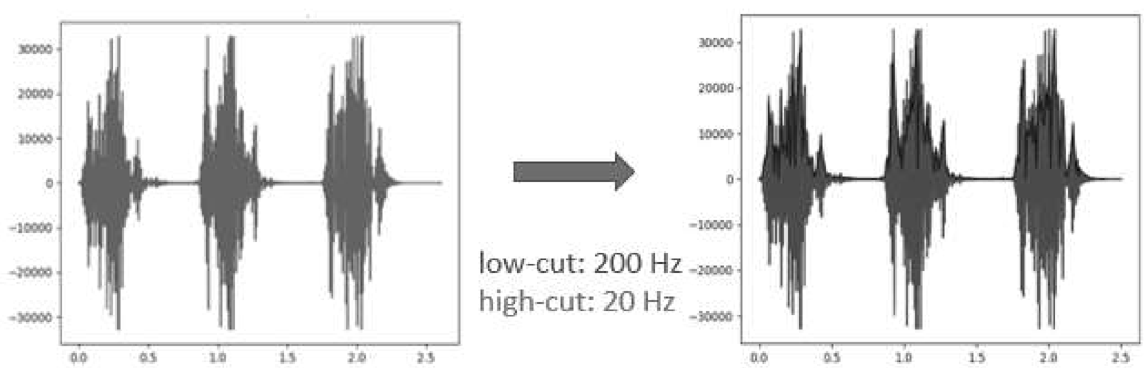
Signal Processing Philosophy

1. Model plays a fundamental role
   - Characterize "signal" and "noise"
   - Relate desired quantity to measured data
2. Model derives from prior knowledge
   - Physics, biology, etc.
3. Processing procedure based on models
4. Modeling issues
   - Poor model   Poor performance
   - Model complexity vs performance

1D Signal Processing



1D Signal Filtering



Represents the graph of Aortic Stenosis cardiac PCG signals amplitude with respect to time-domain before and after filtering. Also, it represents these signals with discriminating elements in a different colour. In these cardiographs, red colour denotes noise in these cardio signals, green colour denotes the extracted cardio signals after filtering and blue represents the peaks in the signals.

Original *PCG* signals

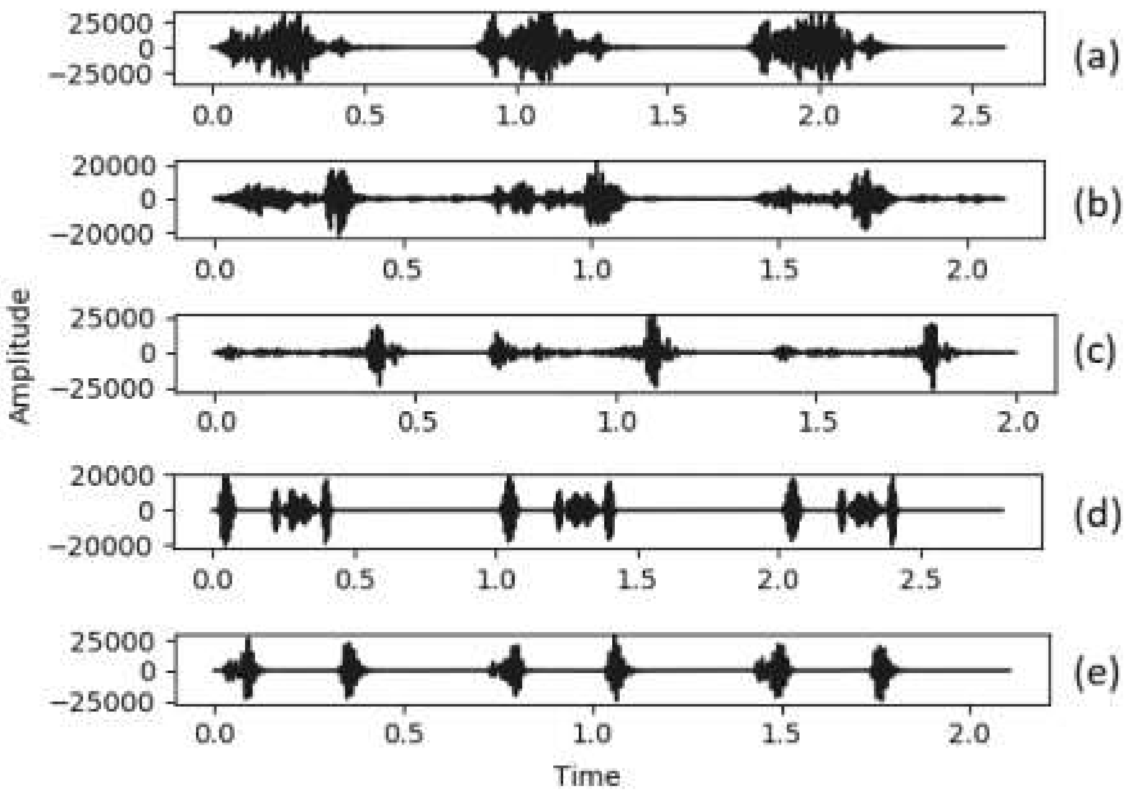


Fig.: Represents the graph of PCG signals amplitude with respect to time-domain. (a) Graph of Aortic Stenosis cardiac PCG signal (b) Graph of Mitral Regurgitation cardiac PCG signal (c) Graph of Mitral Stenosis cardiac PCG signal (d) Graph of Mitral valve prolapse cardiac PCG signal (e) Graph of Normal cardiac PCG signal.

Filtered PCG signals

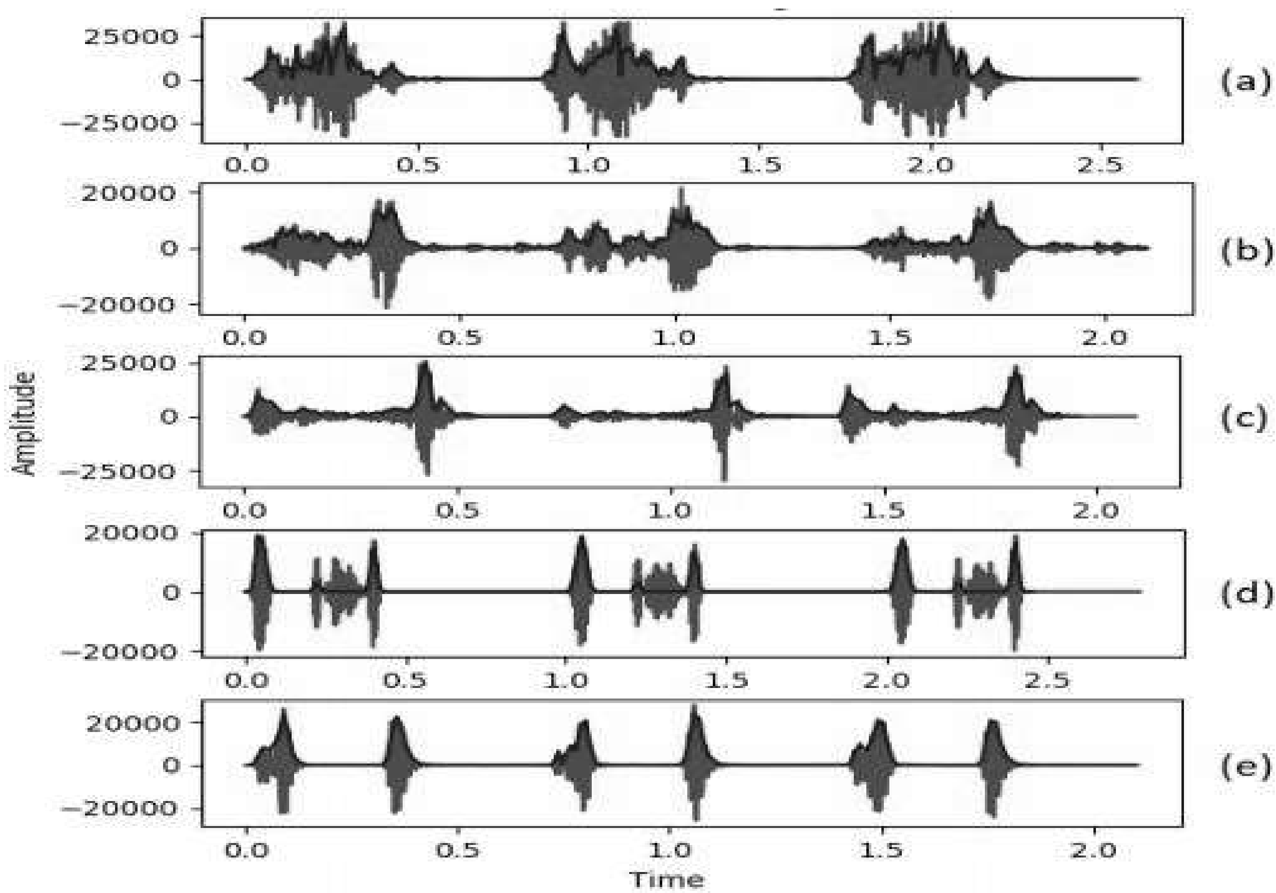


Fig: Represents the graph of PCG signals amplitude with respect to time-domain after signal processing. Also, it represents these signals with discriminating elements in a different colour. In these cardiographs, red colour denotes noise in these cardio signals, green colour denotes the extracted cardio signals after signal-processing and blue represents the peaks in the signals. (a) Graph of Aortic Stenosis cardiac PCG signal (b) Graph of Mitral Regurgitation cardiac PCG signal (c) Graph of Mitral Stenosis cardiac PCG signal (d) Graph of Mitral valve prolapse cardiac PCG signal (e) Graph of Normal cardiac PCG signal.

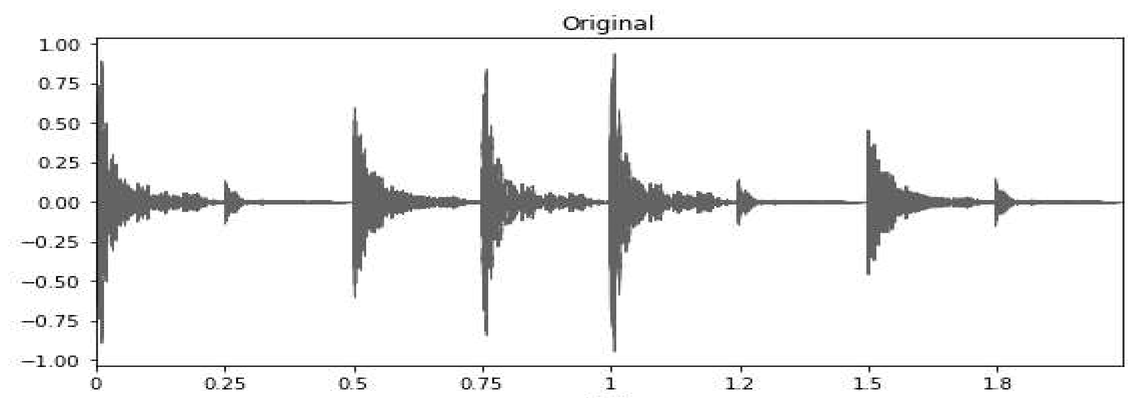
1D Signal Augmentation

Data augmentation is a strategy that enables practitioners to significantly increase the diversity of **data** available for training models, without actually collecting new **data**

Major Types of 1D Data augmentation

1. **Noise Injection**
2. **Shifting Time**
3. **Changing Pitch**
4. **Changing Speed**

Noise Injection



Represents the graph of 1D signals amplitude with respect to time-domain.

There is a given signal represented as $I=[i_1, i_2, i_3, \ldots, i_n]$

Same dimension of noise (random values) is selected

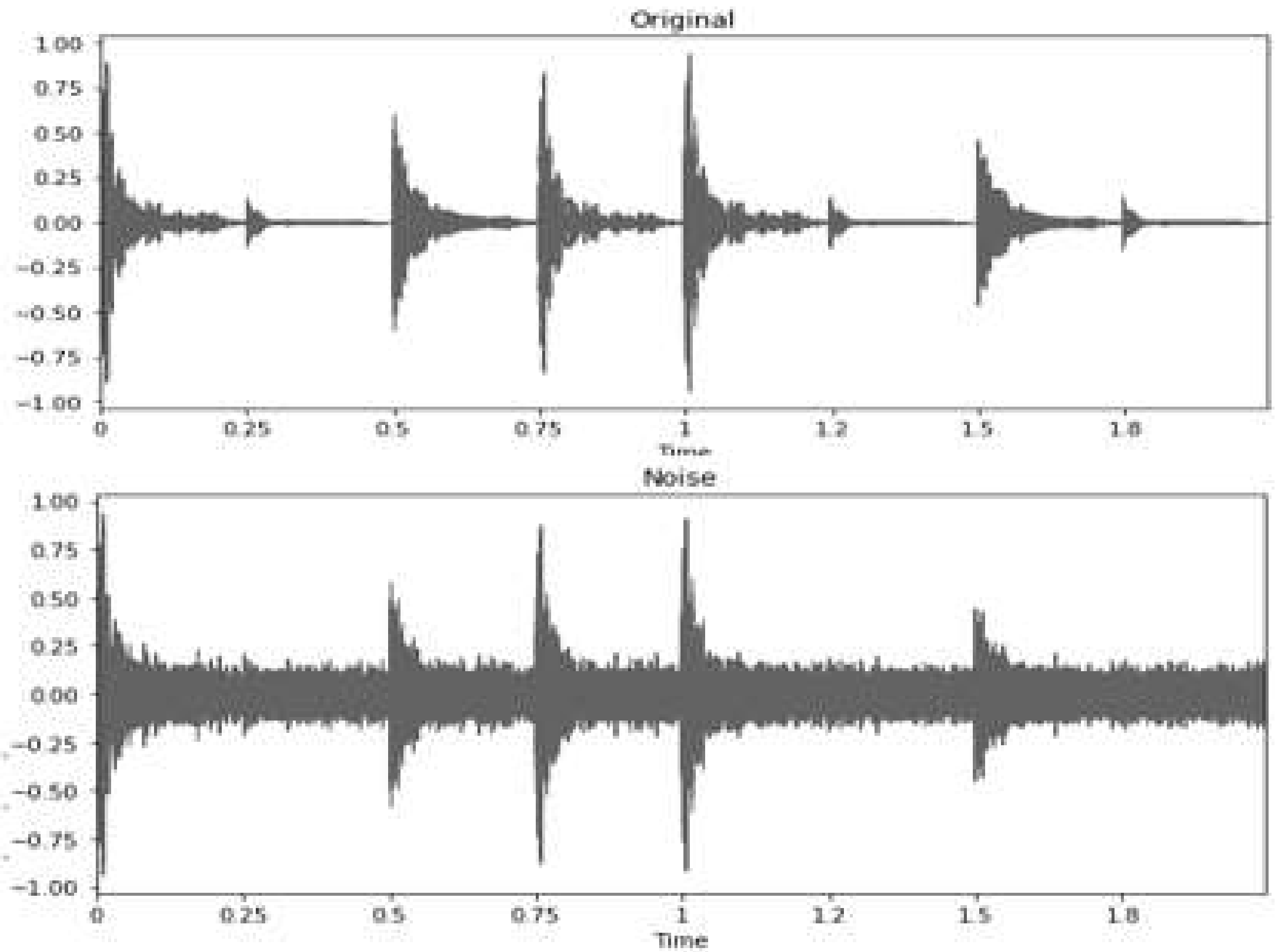
Represented as $\beta = (\beta_1, \beta_2, \beta_3, \ldots, \beta_n)$

Then, Augmentation signal α is calculated

and their relation is $\alpha = I + \beta * \lambda$.

Where, deformations control parameter is represented as λ

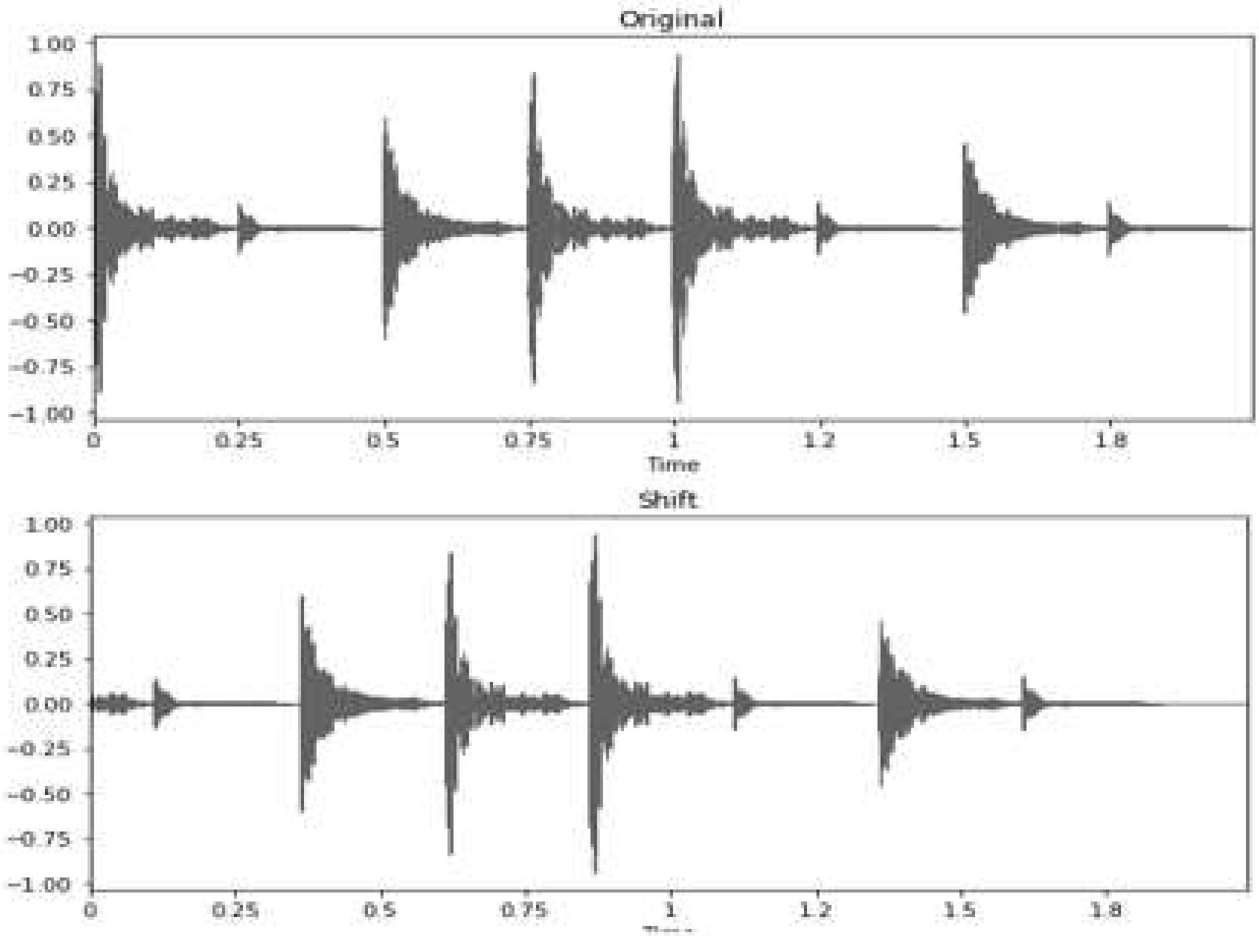
1. Noise Injection



Represents the graph of 1D signals amplitude with respect to time-domain before and after noise injection.

There is a given signal represented as $I=[i_1, i_2, i_3, \ldots, i_n]$ where i is the amplitude of the signal with the dimension of signal length n. Here the same dimension of background deformations is randomly selected and represented as $\beta = (\beta_1, \beta_2, \beta_3, \ldots, \beta_n)$ and deformations control parameter is represented as λ. From these audio deformations, α is calculated and their relation is $\alpha = I + \beta * \lambda$. Where, background deformations β belong to the interval (0, 1) and deformations control parameter λ is taken as 1000 Hz to convert the deformation value into three-digit value.

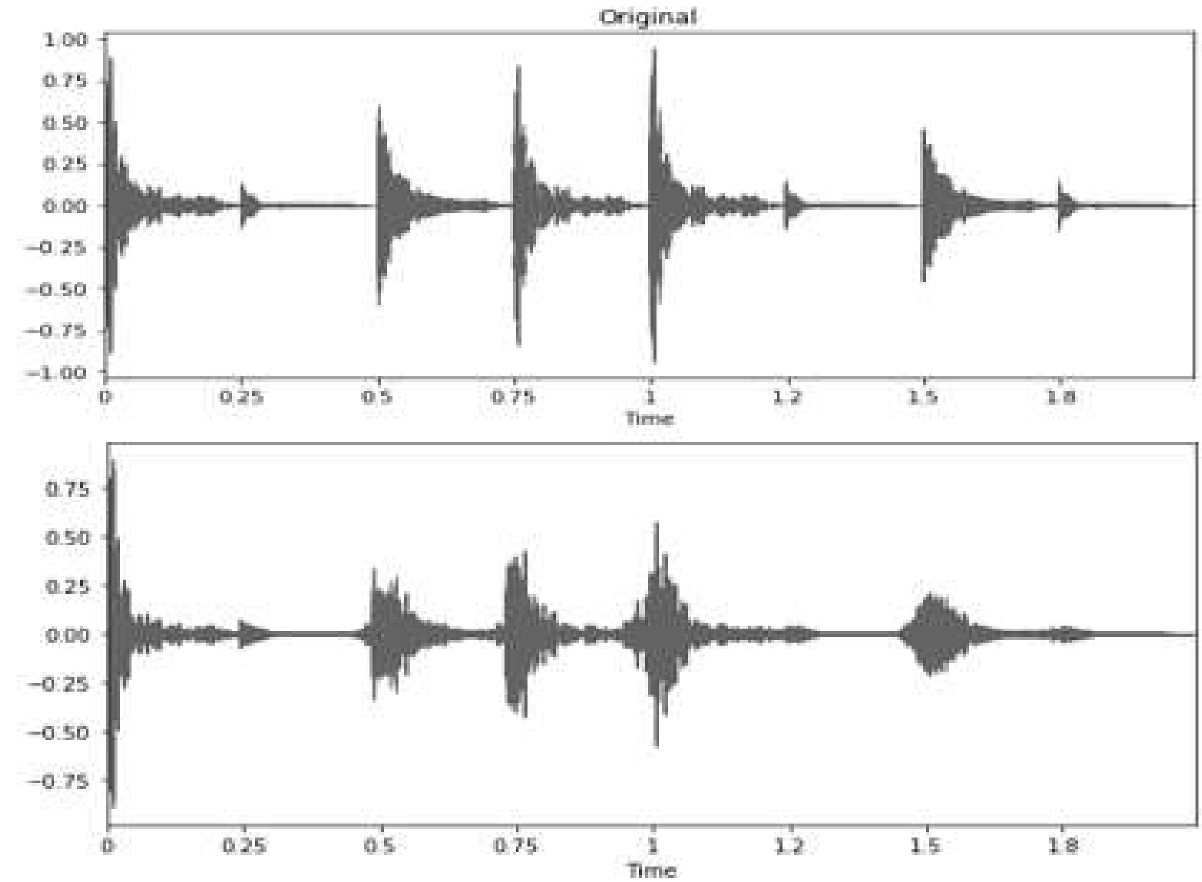
2. Shifting Time



Represents the graph of 1D signals amplitude with respect to time-domain before and after shifting time

There is a given signal represented as $I=[i_1, i_2, i_3, \ldots, i_n]$ where i is the amplitude of the signal with the dimension of signal length n. Here the randomly selected and represented as $\beta = (\beta_1, \beta_2, \beta_3, \ldots, \beta_n)$ of maximum time shift * sampling rate and From these time shifted signal , α is calculated and their relation is $\alpha = I + \beta$ for left shift.

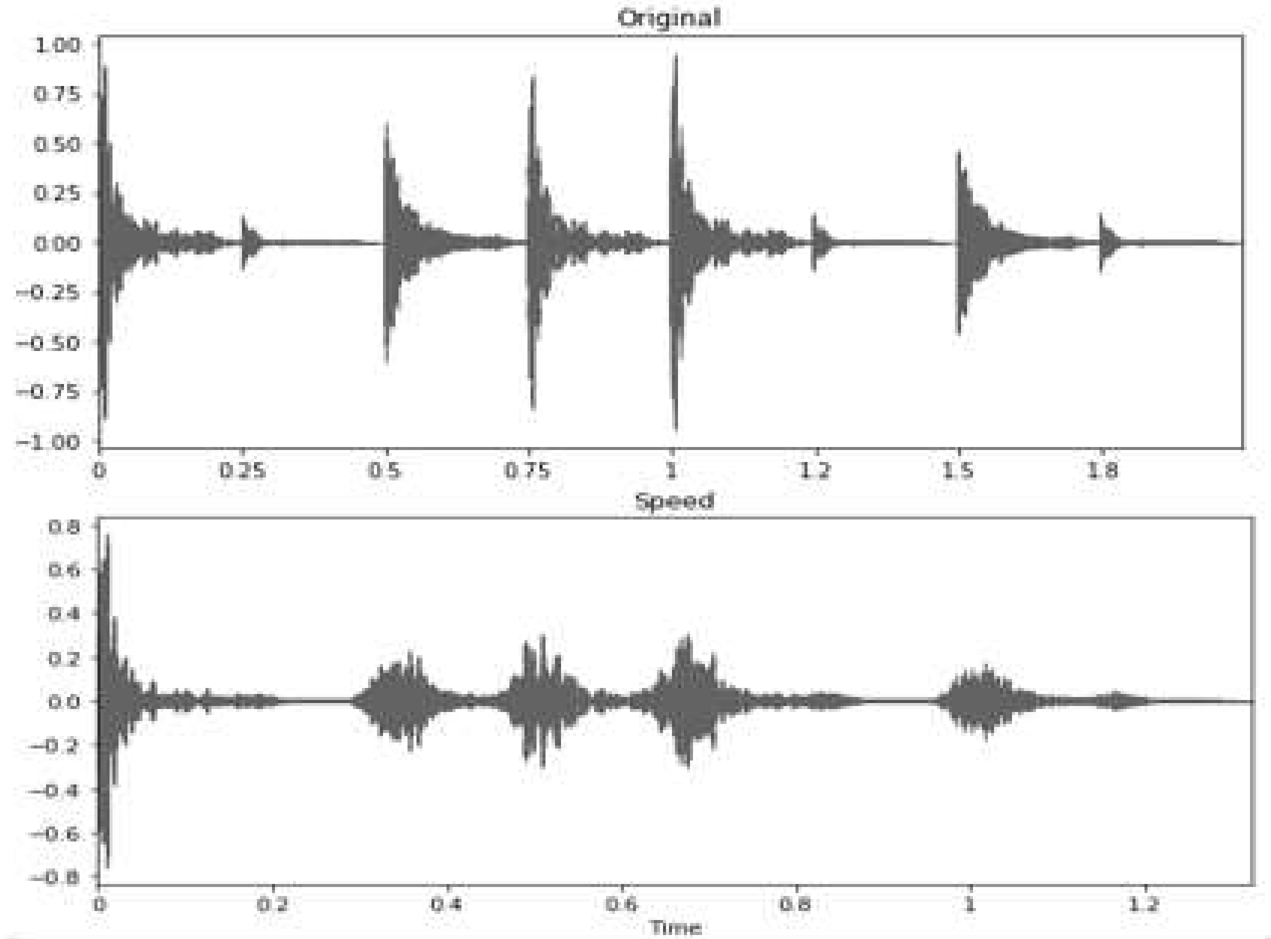
3. Changing Pitch



Represents the graph of 1D signals amplitude with respect to time-domain before and after changing pitch

There is a given signal represented as $I=[i_1, i_2, i_3, \ldots, i_n]$ where i is the amplitude of the signal with the dimension of signal length n. Here the pitch control parameter is represented as *λ and sampling rate as s.* From these time shifted signal , α is calculated and their relation is $\alpha = fx\ (I, s, \lambda,)$.

4. Changing Speed



Represents the graph of 1D signals amplitude with respect to time-domain before and after changing speed.

There is a given signal represented as $I=[i_1, i_2, i_3, \ldots, i_n]$ where i is the amplitude of the signal with the dimension of signal length n. Here the speed control parameter is represented as λ. From these time shifted signal , α is calculated and their relation is $\alpha = fx\ (I, \lambda,)$.

1D Signal Features extraction

Feature extraction is used to find important points for discriminating the data among the different classes from the input data.

Feature extraction increases the accuracy of learned models by extracting features from the input data.

Statistical-based feature extraction

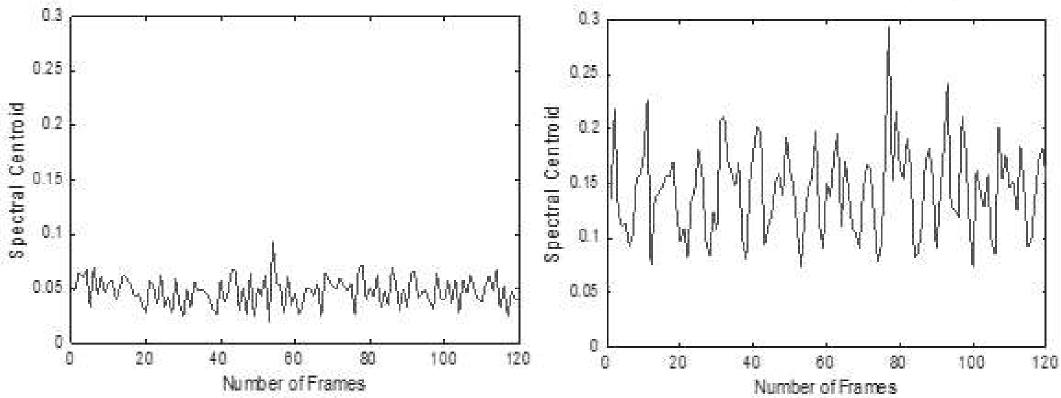
1. Spectral Centroid
2. Energy Entropy
3. Spectral Roll off
4. Spectral Flux
5. Volume
6. Zero Crossing Rate

1. ***Spectral Centroid***

$$Centroid = \frac{\sum_{n=0}^{N-1} f(n)\, x(n)}{\sum_{n=0}^{N-1} x(n)}$$

x (n) is the weighted frequency of bin number

f (n) is the center frequency of bin

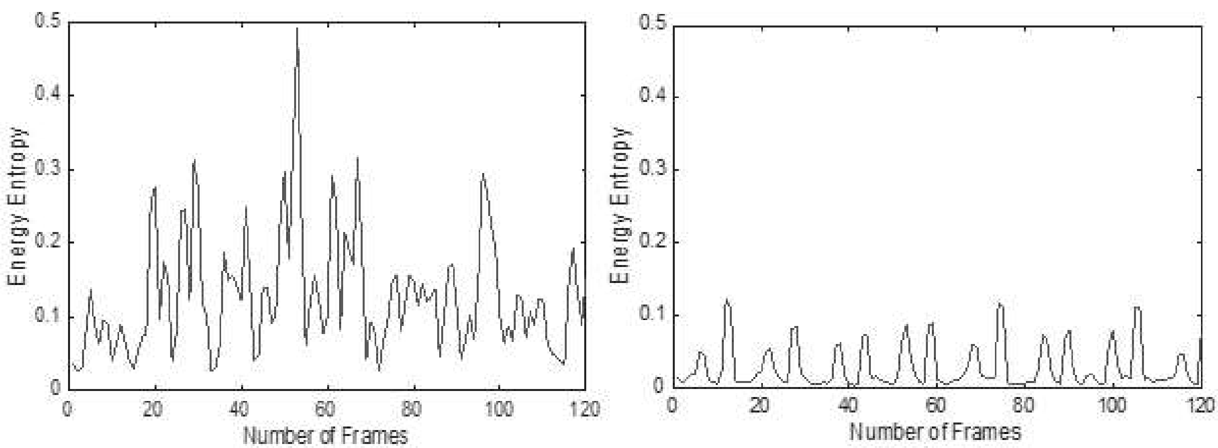


(a) (b)

Spectral Centroid: (a) Normal heart sound (b) Abnormal heart Sound

2. ***Energy Entropy***

$$I_j = -\sum_{i=1\ldots k} \sigma_i^2 \, log_2 \sigma_i^2$$

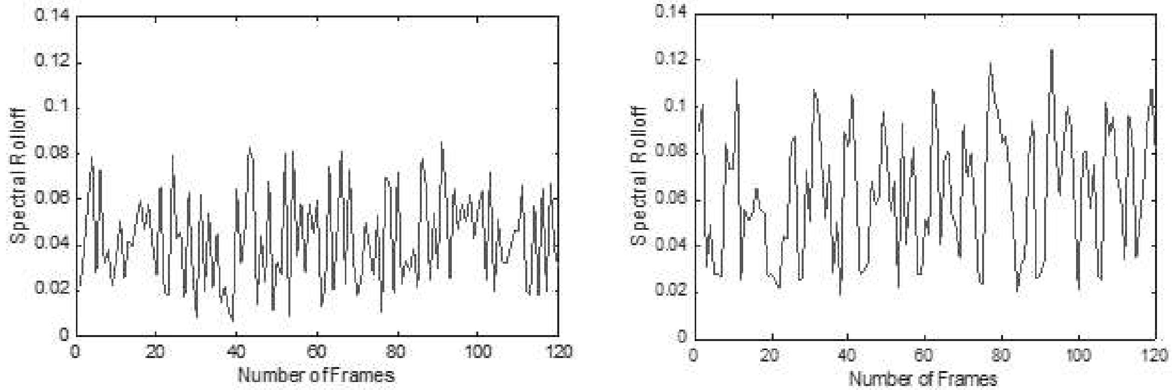


(a) (b)

Energy Entropy : (a) Normal heart sound (b) Abnormal heart Sound

3. Spectral Roll off

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 \sum_{n=1}^{N} M_t[n]$$

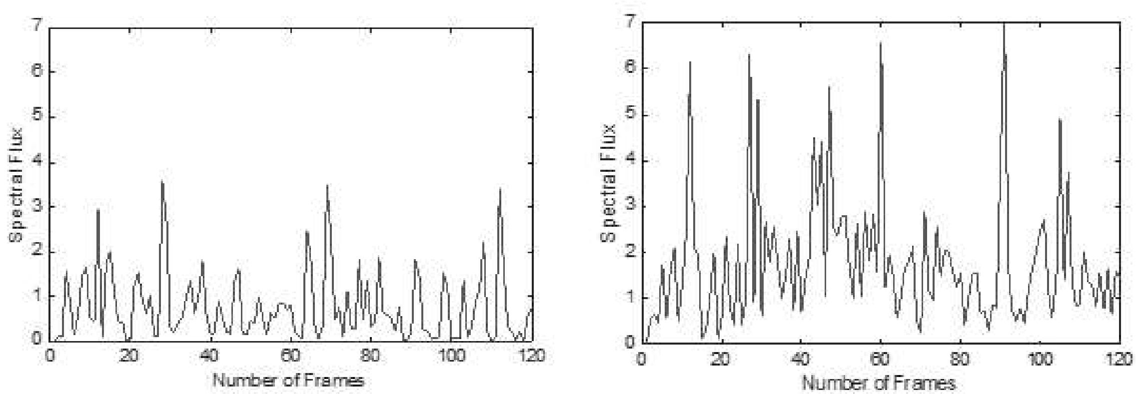


(a) (b)

Spectral Roll off : (a) Normal heart sound (b) Abnormal heart Sound

4. Spectral Flux

$$F_j = \sum_{k=0\ldots S-1} (N_{j,k} - N_{j-1,k})^2$$

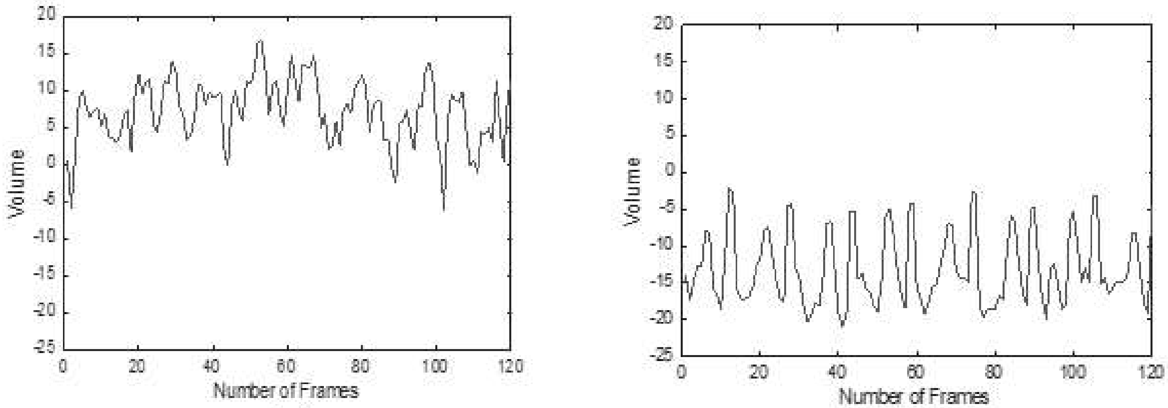


(a) (b)

Spectral Flux : (a) Normal heart sound (b) Abnormal heart Sound

5. Volume

$$volume = 10 * \log_{10} \sum_{i=1}^{n} {s_i}^2$$

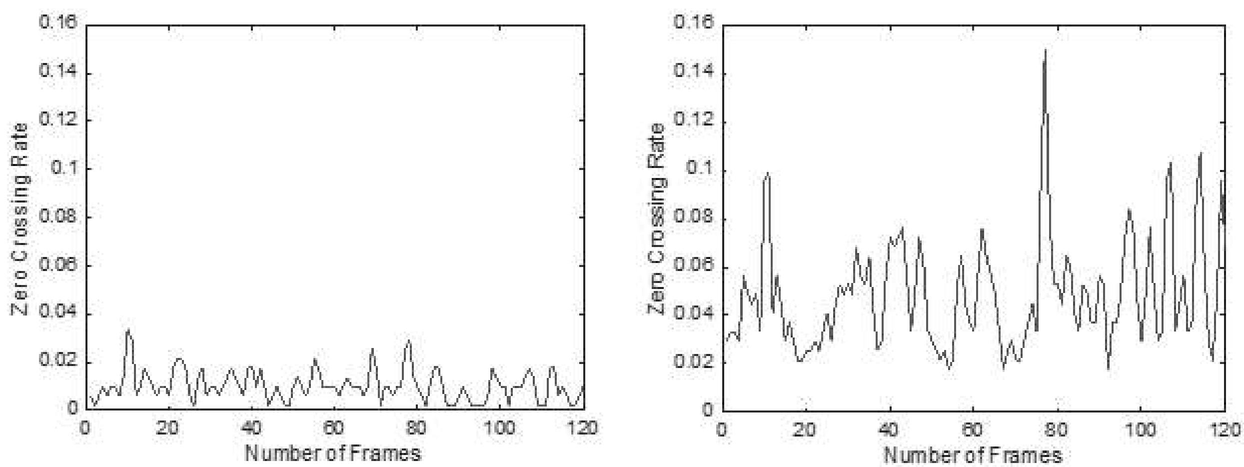


(a) (b)

Spectral Flux : (a) Normal heart sound (b) Abnormal heart Sound

6. Zero Crossing Rate

$$Z_j = \frac{1}{2S} \sum_{i=1...S} |sgn(x_i) - sgn(x_{i-1})|$$



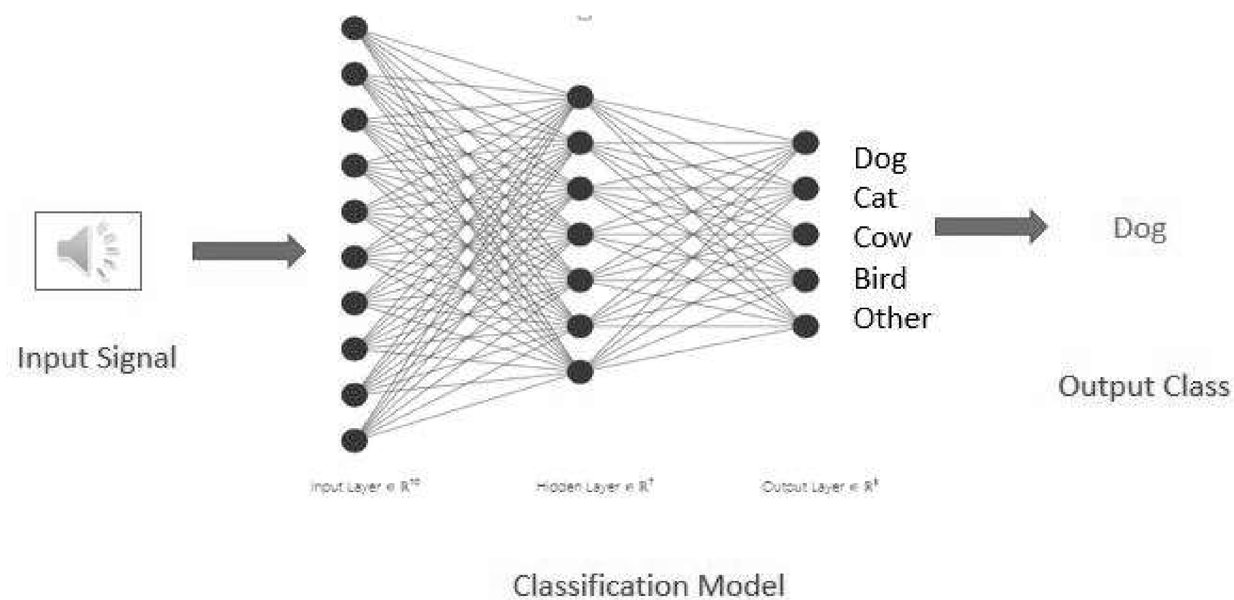
(a) (b)

Zero Crossing Rate : (a) Normal heart sound (b) Abnormal heart Sound

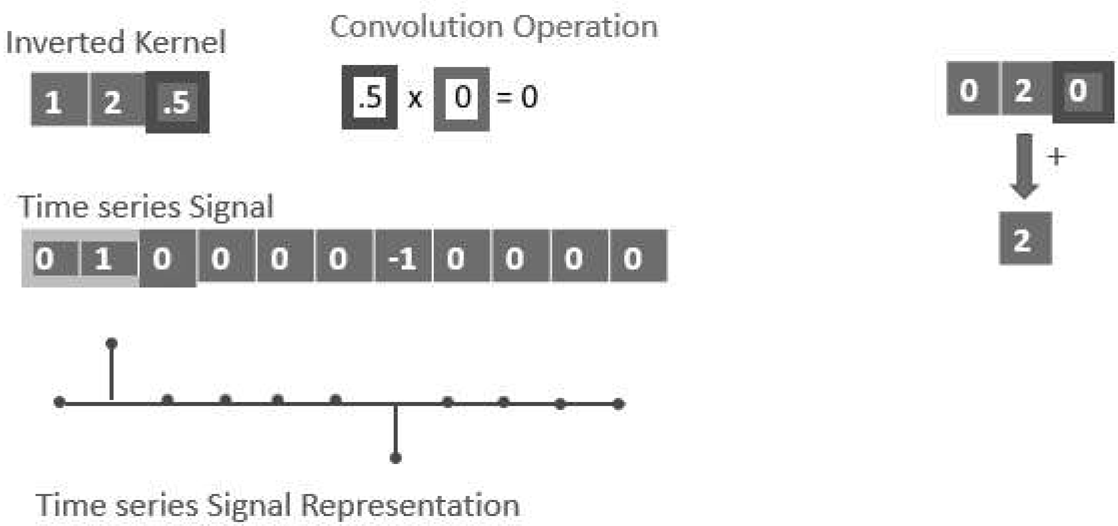
Other types of features used in 1D signal

1. Fourier transform
2. Spectrogram
3. Sort-Time Fourier Transform
4. Mel Filter Bank
5. Mel Frequency Cepstral Coefficients

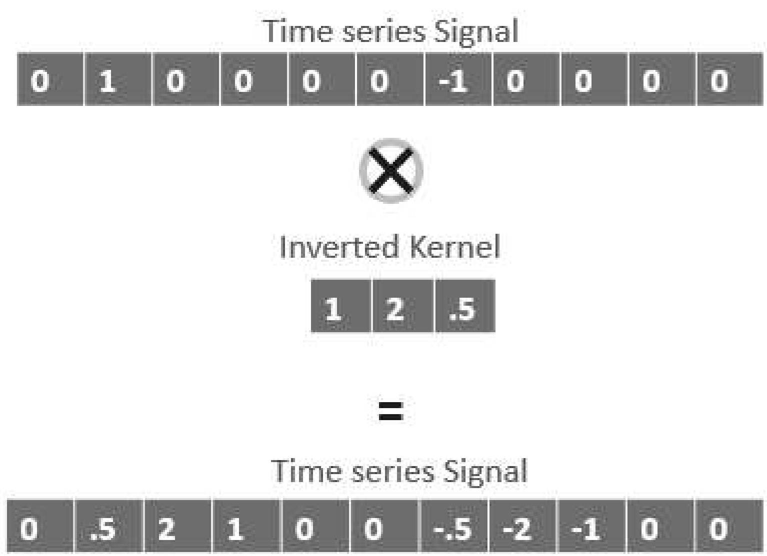
Deep learning in 1D

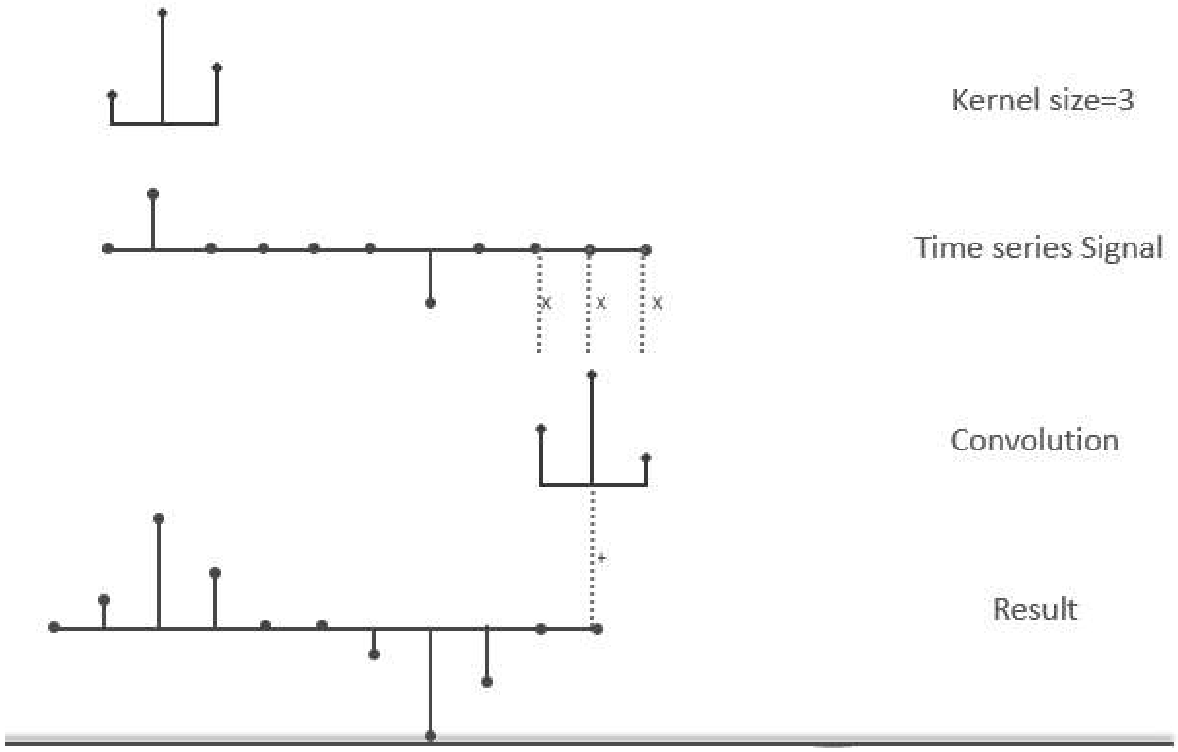


Convolution in Time Series Signal

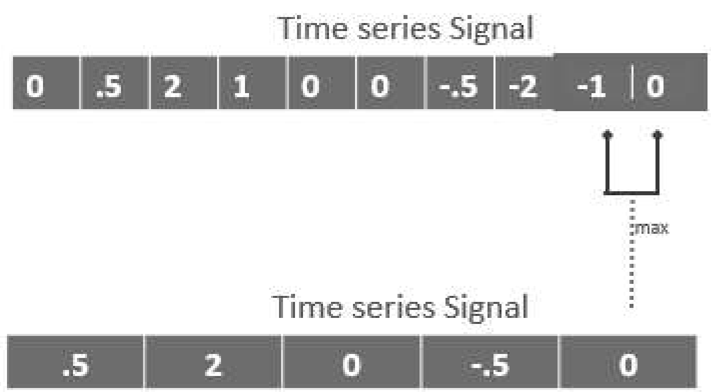


Convolution in Time Series Signal

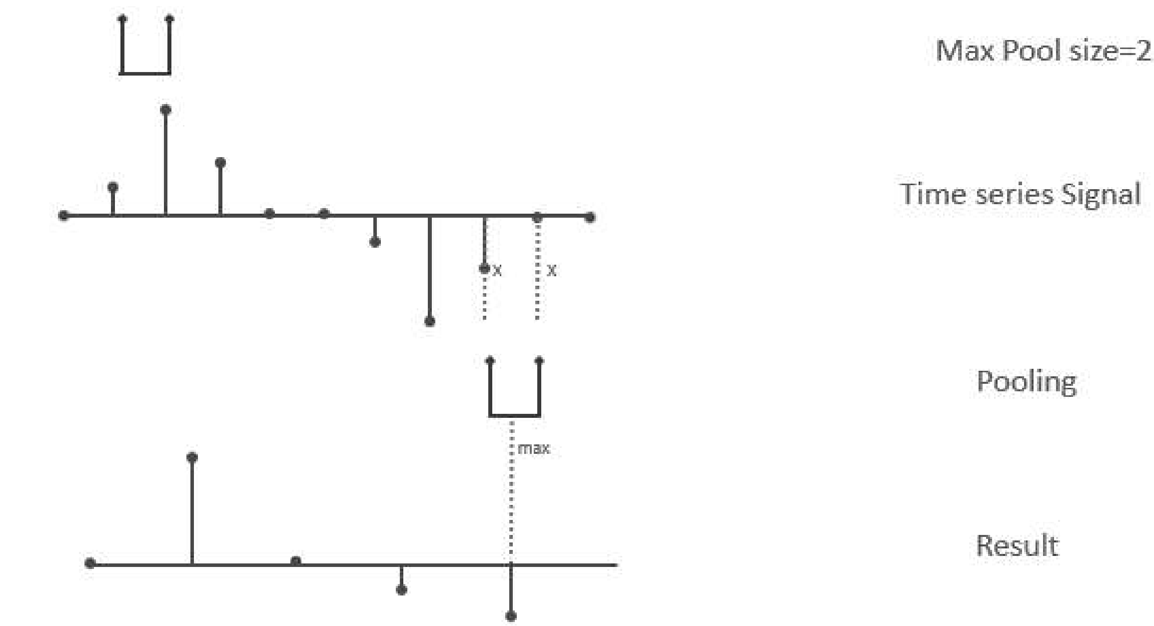




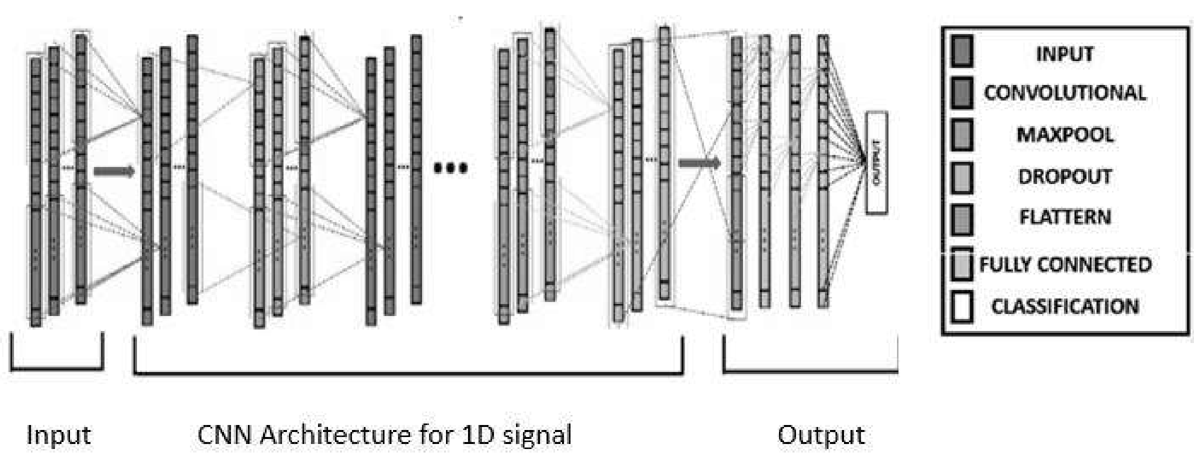
Max Pooling in Time Series Signal



Max Pooling in Time Series Signal



Time Series Signal Classifications using CNN



Deep learning for Signal Classifications using CNN

There is a given signal represented as $I=[i_1, i_2, i_3, \ldots, i_n]$ where i is the amplitude of the signal with the dimension of signal length *20000*. It is classified into 5 classes.

```
def cnn_model(size):
  m = Sequential()
  m.add(Conv1D(filters=8, kernel_size=3, strides=3, padding='valid', activation='relu', input_shape=size))
  m.add(Conv1D(filters=16, kernel_size=3, strides=3, padding='valid', activation='relu'))
  m.add(Conv1D(filters=32, kernel_size=3, strides=3, padding='valid', activation='relu'))
  m.add(Conv1D(filters=64, kernel_size=3, strides=3, padding='valid', activation='relu'))
  m.add(Conv1D(filters=128, kernel_size=3, strides=3, padding='valid', activation='relu'))
  m.add(MaxPooling1D(pool_size=2))
  m.add(Dropout(0.5))
  m.add(Flatten())
  m.add(Dense(128, activation='relu'))
  m.add(Dense(5, activation='softmax'))
  m.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
  m.summary()
  return m

model=cnn_model((20000,1))
```

```
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_38 (Conv1D)           (None, 6666, 8)           32
_________________________________________________________________
conv1d_39 (Conv1D)           (None, 2222, 16)          400
_________________________________________________________________
conv1d_40 (Conv1D)           (None, 740, 32)           1568
_________________________________________________________________
conv1d_41 (Conv1D)           (None, 246, 64)           6208
_________________________________________________________________
conv1d_42 (Conv1D)           (None, 82, 128)           24704
_________________________________________________________________
max_pooling1d_8 (MaxPooling1 (None, 41, 128)           0
_________________________________________________________________
dropout_9 (Dropout)          (None, 41, 128)           0
_________________________________________________________________
flatten_8 (Flatten)          (None, 5248)              0
_________________________________________________________________
dense_15 (Dense)             (None, 128)               671872
_________________________________________________________________
dense_16 (Dense)             (None, 5)                 645
=================================================================
Total params: 705,429
Trainable params: 705,429
Non-trainable params: 0
_________________________________________________________________
```

Deep learning for Signal Classifications using RNN

There is a given signal represented as $I=[i_1, i_2, i_3, \ldots, i_n]$ where i is the amplitude of the signal with the dimension of signal length *20000*. It is classified into 5 classes.

```
def rnn_model(size):
  m = Sequential()
  m.add(LSTM(128,return_sequences=True,input_shape=size))
  m.add(LSTM(128,return_sequences=True))
  m.add(Dropout(0.5))
  m.add(TimeDistributed(Dense(128, activation='relu')))
  m.add(TimeDistributed(Dense(16, activation='relu')))
  m.add(Flatten())
  m.add(Dense(128, activation='relu'))
  m.add(Dense(5, activation='softmax'))
  m.compile(loss='categorical_crossentropy', optimizer='adadelta', metrics=['accuracy'])
  m.summary()
  return m

model=rnn_model((20000,1))
```

```
Model: "sequential_14"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_9 (LSTM)                (None, 20000, 128)        66560
_________________________________________________________________
lstm_10 (LSTM)               (None, 20000, 128)        131584
_________________________________________________________________
dropout_14 (Dropout)         (None, 20000, 128)        0
_________________________________________________________________
time_distributed_8 (TimeDist (None, 20000, 128)        16512
_________________________________________________________________
time_distributed_9 (TimeDist (None, 20000, 16)         2064
_________________________________________________________________
flatten_12 (Flatten)         (None, 320000)            0
_________________________________________________________________
dense_30 (Dense)             (None, 128)               40960128
_________________________________________________________________
dense_31 (Dense)             (None, 5)                 645
=================================================================
Total params: 41,177,493
Trainable params: 41,177,493
Non-trainable params: 0
_________________________________________________________________
```