COMPARISON OF MACHINE LEARNING ALGORITHMS FOR ECG

PROJECT REPORT

Submitted by

SUDARSHAN V (201EC519)
ASFAQ AHMED M (201EC503)
THIBESH V (201EC522)

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

BANNARI AMMAN INSTITUTE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University, Chennai)

SATHYAMANGALAM-638401

ANNA UNIVERSITY: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report **"COMPARISON OF MACHINE LEARNING ALGORITHMS FOR ECG"** is the bonafide work of **"SUDARSHAN V, ASFAQ AHMED M, THIBESH V"** who carried out the project work under my supervision.

SIGNATURE

Dr. POONGODI C
Professor &
Head of the Department,
Department of Electronics and
Communication Engineering.

SIGNATURE

Mrs. SRITHA P
Assistant professor
Level III,
Department of Electrical and
Electronics Engineering.

Submitted for Project Viva Voce examination held on ……………….

Internal Examiner 1

Internal Examiner 2

DECLARATION

We affirm that the project work titled **"COMPARISON OF MACHINE LEARNING ALGORITHMS FOR ECG"** being submitted in partial fulfilment for the award of the degree of **Bachelor of Engineering** in **'Electronics and Communication Engineering'** is the record of original work done by us under the guidance of Mrs **Sritha P, Assistant Professor III**. It has not formed a part of any other project work(s) submitted for the award of any degree or diploma, either in this or any other University.

(Signature of candidate)	(Signature of candidate)	(Signature of candidate)
SUDARSHAN V	**ASFAQ AHMED M**	**THIBESH V**
201EC519	**201EC503**	**201EC522**

ACKNOWLEDGEMENT

We would like to enunciate heartfelt thanks to our esteemed Chairman **Thiru. S. V. Balasubramaniam**, Trustee **Dr. M. P. Vijayakumar**, and the respected Principal **Dr. C. Palanisamy** for providing excellent facilities and support during the course of study in this institute.

We are grateful to **Dr Poongodi C**, **Prof & Head, Electronics and Communication Engineering** for her valuable suggestions to carry out the project work successfully.

We wish to express our sincere thanks to Faculty guide **Sritha P, Assistant Professor Level III, Electrical and Electronics Engineering** for her constructive ideas, inspirations, encouragement, excellent guidance and much needed technical support extended to complete our project work.

We would like to thank our friends, faculty and non-teaching staff who have directly and indirectly contributed to the success of this project.

SUDARSHAN V

ASFAQ AHMED M

THIBESH V

ABSTRACT:

An Electrocardiogram records the electrical signal from the heart to check for different heart conditions by placing electrodes in the patient's Chest. Abnormalities in the heart will be detected by comparing these signal waveforms. In our project we are Comparing two machine learning algorithms to get the output accuracy of ecg signal. For that we are getting a number of ecg signals of Arrhythmia patients from Kaggle databases, which was uploaded by research personnels in MIT. We get these data and preprocess them for signal extraction, then we feed them into two different algorithms namely Support Vector Machine Algorithm and Capsule Network Algorithm. After getting the output ecg signal the comparison will take place. Since the Introduction of Capsule Network there has been an upsurge in Machine learning, it has better performance than the previous machine learning Networks like CNN. We recent advanced technologies to get the accurate output and performance of providing precise output of ecg data. So, the final output of our project is to give the Ecg data with greater accuracy by using machine learning algorithms.

Keywords:

Electrocardiogram, Arrhythmia, Comparing, Accuracy.

CONTENTS:

INTRODUCTION:

An electrocardiogram is the equipment used for medical purposes and it can capture the ecg signal from the heart to check for various heart conditions. ECG waveforms are characterized by major waves including P, QRS, and T. where QRS stands for ventricular depolarization. Determining the QRS complexes can provide a lot of useful data based on heart rate, heart rhythm, and ectopic beat information. As a result, QRS compound detection, also known as R-wave detection or simply peak detection, is useful in the medical field. This unwanted noise of ecg signal such as baseline wander (BW), power line interference (PLI), muscle artifact (MA), and instrument noise (IN). There are more methods in the technological world for this issues to overcome these noises. But to get the desired output very accurately in our project, In our project the main frame of the work is to compare two algorithms in ML technology called as Support vector machines and Capsule Networks in our project. To compare algorithms the noisy data of ecg will be obtained from the Kaggle databases. As ECG data is received, it is preprocessed to remove noise. After the noise removal finally the ECG data is fed to ML algorithms to get the highly accurate outputs and algorithms will be compared.

LITERATURE REVIEW:

Traditional methods:

According to the analysis of the slope, width and amplitude of the QRS complex, which is widely used in the literature, extensive research for the accurate analyses of QRS complexes, including the extensive work by Pan and Tompkins many developments made on the algorithm. Other ECG peak detection methods include first derivative and second derivative. Poli R, Cagnoni S, and Valli G present QRS analyzing algorithms involving polynomial filters, whose relevant parameters are selected by a genetic algorithm. For Afonso VX, Tompkins WJ, Nguyen TQ, and Luo S, the electrocardiogram signal was decomposed with a filter bank, and multiple features were combined with heuristic decision rules to detect heart beats results.

Lot of the journals have been published using wavelet transforms to detect his electrocardiogram beats. Apart from these issues, various types of noise can be present in the ECG signal, including muscle noise, electrode movement artifacts, power line disturbances, and baseline drift. To successfully address these issues, there is an existing technique called Variable Frequency Complex Decomposition abbreviated as VFCDM for ecg reconstruction and heart rate detection. Reconstructing the ECG signal between different VFCDM subbands to remove noise-related components and achieve better peak analyzing. VFCDM is a time-frequency analysis technique used in a variety of physiological signal processing. The motive of the VFCDM reconstruction stage is to combine only the components that represent the dynamics of the ECG. There is also another usual way to remove noise is using a high pass filter with a cutoff frequency of 0.5 to 0.6 Hz is used.

Existing algorithms:

ECG signals use the duration and shape of each waveform and the spacing between different peaks to determine heart disease. In our project we introduce the new algorithm which was proposed to better analyze his ECG signals using two-event moving average abbreviated as TERMA and fractional Fourier transform abbreviated as FrFT. The TERMA algorithm finds peaks of interest in a specific region of PQRS signal. FrFT, in the other side, rotates the ECG signal in the time-frequency domain to reveal the positions of various peaks. The output accuracy of the implemented algorithm outperforms that of state of the art algorithms. Additionally, we trained a ML model using calculated peaks, durations between different peaks, and other ECG signal characteristics to automatically classify heart disease by ML algorithms. Most of the available studies using the MIT-BIH database consist of only 48 patients. TERMA is used in economics to find different events in trading, while moving averages help find signals contained in a particular event. Therefore, these averages can also be used for ECG signals containing events such as P waves, QRS complexes and T waves. These waves repeat at regular time intervals. Time-frequency analysis is also important due to the high variability of the P wave, QRS complex and T wave.

What is Arrhythmia?

An arrhythmia is an irregular heartbeat. Arrhythmia affected patients can get heart beat faster or slower than in people who are not affected by arrhythmia. Abnormal heartbeats have several causes, and treatment depends on the cause.

Supraventricular arrhythmias:

An arrhythmia that begins in the atria (upper chambers of the heart). "Supra" means above. It affects the heart upper chamber.

Ventricular Arrhythmias:

Arrhythmias that begin in the heart's conduction system, such as the ventricles (lower chambers of the heart), the sinus node, the atrioventricular nodes. It makes the lower chamber or heart twitch but it can not pump.

Causes:

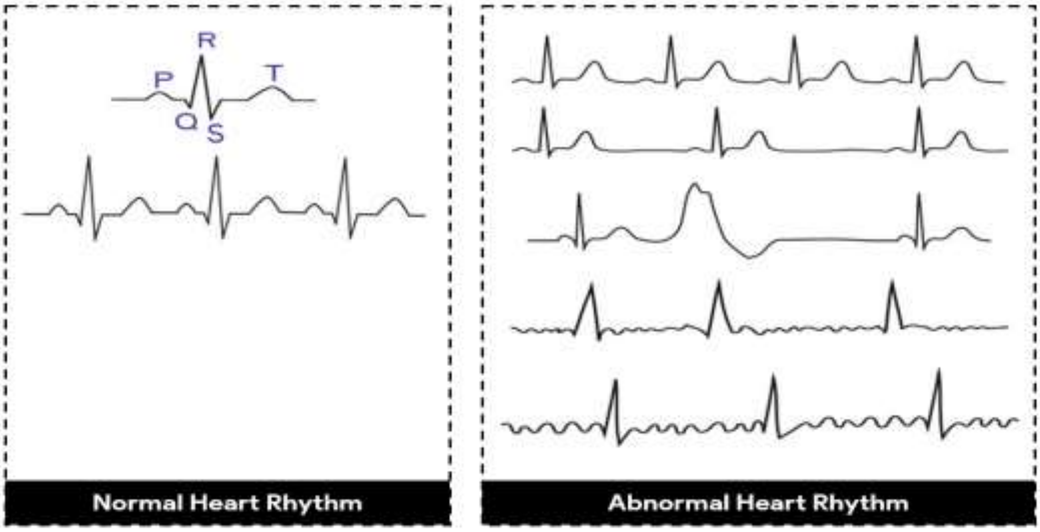
Arrhythmia can occur due to coronary heart disease. Hypersensitive tissue of the heart, Hypertension, Changes in the heart muscle, Defective valve. B. Sodium or potassium imbalance such as electrolyte imbalance in the blood, Heart attack injury, Healing process after heart surgery and other illnesses.

Diagnostic procedure:

There are some of the methods to diagnose the heart arrhythmia are

- Electrocardiogram
- outpatient monitor
- stress test
- echocardiogram
- cardiac catheter
- Electrophysiological studies
- tilt table test

Image below is the graph of Normal and Arrhythmia affected patients.

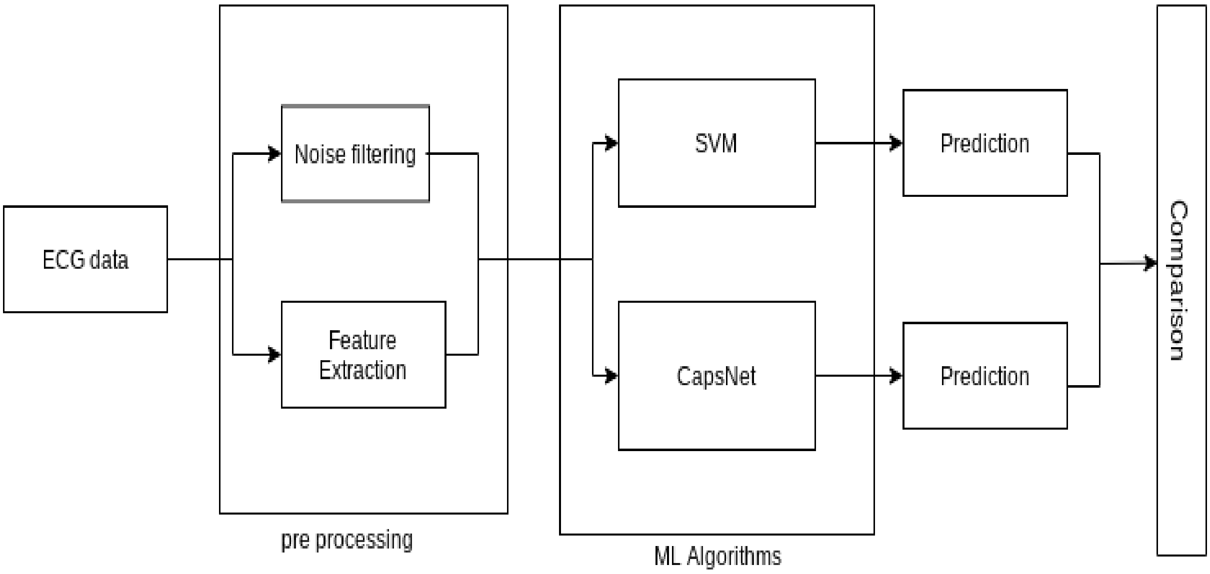


There are some most commonly identified arrhythmias are mentioned below

le	**Description**
	Left bundle branch block beat
	Right bundle branch block beat
	Atrial premature beat
	Aberrated atrial premature beat
	Nodal (junctional) premature beat
	raventricular premature or ectopic beat (atrial or nodal)
	Premature ventricular contraction
	R-on-T premature ventricular contraction
	Atrial escape beat
	Nodal (junctional) escape beat
	Supraventricular escape beat (atrial or nodal)
	Ventricular escape beat
	Paced beat

BLOCK DIAGRAM:

The below block diagram demonstrates the overall project layout



SOFTWARE REQUIREMENTS:

We are using the software called as Jupyter notebook in our project to implement the algorithms.

1. **Visual Studio Code**

Debugging, task execution, and version control are monitored by the simplified code editor Visual Studio Code. It tries to give developers only the tools they require for a short cycle of code-build-debugging and leaves more sophisticated processes to IDEs with more features, like Visual Studio IDE.

0. **Jupyter Notebook**

Jupyter Notebook is a web programme for designing and sharing computational documents. It creates a straightforward, efficient, document-focused experience. The most recent web based IDE for code, data, and notebooks is Jupyterlab. Users can configure and arrange machine learning and scientific computing workflows using the interface's flexibility. A modular layout encourages expansions to increase and develop the functionality.

DATABASE:

1. **MIT-BIH Dataset**

This Arrhythmia database contains 48 and 30-minute excerpts of 2-lead ambulatory ECG signals from 47 subjects examined by the BIH Arrhythmia Laboratory. Twenty-three recordings were randomized from a set of 4000 his 24-

hour ambulatory ECG recordings gathered from a mixed population of inpatients 60% and outpatients 40% at BIH in Boston, rice field. The another 25 recordings, selected from the same set, contained uncommon but medically significant arrhythmias, poorly shows in a small random sample. This directory has the MIT-BIH arrhythmia database. Since PhysioNet's launch in the year of 1999, about half of this database (25 out of 48 complete records plus reference annotation files for all 48 records) is freely available in this database. The remaining 23 waveform files are only available in the MIT-BIH Patients datasets.

Training and test:

The datasets are separated by two groups, one for training the signals and another is for testing. This split of the data is chosen to compensate for the presence of different types of heartbeats and the counts of signals in each dataset. Take into account the split between patients. The subject used to build or optimize a classifier is different from the subject used to evaluate it. Models using the same patient heart rate for both training the signal and testing the signal have been shown to be biased, and the results are not reproducible in the world.

PROPOSED METHODOLOGY:

In our implementation of project we are using ML algorithms as a proposed methodology. We get the ECG data and preprocess them for noise filtration and signal extraction, then we feed them into two different algorithms used Support vector machine algorithm and Capsule network algorithm then get the predicted output of each algorithm and Comparing the output data.

Why Machine Learning?

Machine learning is a type under AI and a branch of computational science focused on determining and interpreting patterns and structures in data, and learning, reasoning, and learning outside of human interaction. enable decision making. In simple words, machine learning allows users to feed vast amounts of data into a computer's algorithms, have the computer analyze it, and make data-driven suggestions and decisions based solely on the input data. Once fixes are identified, algorithms can integrate this data to develop future decision making.

Working of Machine Learning?

<u>Machine learning consists of three parts:</u>

A computational algorithm that is central to the decision. Variables and characteristics that make up the decision. Basic knowledge for which the answer is known allows the system to Train. First, the model is given parameter data for which the already known answers. The algorithm is then run and adjusted until the algorithm's prediction matches the known answer. At this point, more and more data is input to help the system learn and process higher-level computational decisions.

Why Python?

ML software is designed to be used in AI applications. Also, with the help of machine learning, artificial intelligence can learn and improve its predictions, capabilities, and user experience. Both AI and MI can advance the Entrepreneurship of many companies in the current world. More and more companies are starting to implement AI and ML to stay ahead of the competition.

As a technology, machine learning uses different languages and tools to increase productivity. One of the best languages for MI is Python. This language has been adopted by many non programmers for every day tasks. Python has the feature to run in multiple operating systems so that the Python has the ability to work in machine learning.

Working of PYTHON:

Python's origins can be traced back to the 1980s when Guido van Rossum started working on it. It is an open source programming language. Python is versatile, code readable. It is an object-oriented language, so we can write clear, concise code and use it in small and enterprise projects. Hence, it is the reason for using python to implement machine learning.

Types of Noises in ECG:

Baseline Wander:

Baseline wander is the effect in which the baseline of a signal appears to move up and down instead of straight. This shifts the entire signal from its normal base. In ECG signals, baseline wander is caused by improper electrodes, patient motion, and respiration process. A typical electrocardiogram signal affected by baseline wander. The baseline gait frequency content is in the 0.5 Hz range, but increased body movement during exercise testing increases the baseline gait frequency content. Therefore, since the baseline signal is a low-frequency signal, hence we can estimate and remove the baseline of the ECG signal using high-pass zero-phase finite impulse response filtering with a cutoff frequency of 0.5 Hz.

Powerline Interferences:

It is a common source of noise in ECGs, as are other bioelectric signals recorded from the body surface. Such noise is characterized by 50 sinu interference that can be accompanied by many harmonics. Such narrowband noise makes the ECG analysis and interpretation more difficult, as the display of low-amplitude waveforms becomes unreliable and spurious can be introduced. Powerline interference must be removed from the ECG signal to completely overwhelm low-frequency ECG waves such as P-waves and T-waves. A typical ECG signal affected by electrode movement artifacts.

White Gaussian noise:

White Gaussian Noise is a fundamental level of noise. It adds to the noise that may be inherent in information systems. It refers to the knowledge of consistent performance across information system frequency bands. This is analogous to white with equal radiation at all frequencies in the visible spectrum. This is reason because it has a normal splitting with zero mean in the time domain. Broadband noise comes from many environmental noise issue sources, some of them are thermal oscillations of atoms in conductors, shot noise, blackbody effects from the ground and other heat bodies. It is often used as a channel model where the only impediment to communication is the linear addition of broadband or white noise with constant spectral density (expressed in Watts or hertz of bandwidth) and Gaussian amplitude distribution. This model does not account for fading, frequency selectivity, interference, nonlinearities, or dispersion. However, before considering these other phenomena, In our project we can generate a basic and tractable mathematical model. AWGN channels are excellent models for many advanced communication links. However, in terrestrial way modeling this noise is extensively used to monitoring.

Muscle artifacts:

ECG artifacts are defined as changes in the electrocardiogram that are unrelated to cardiac electrical activity. Electrocardiogram components such as baselines and waveforms can be distorted as a result of artifacts. Motion artifacts result from swaying due to rhythmic movement. Examples of motion artifacts include tremor without apparent cause, Parkinson's disease, cerebellar or intentional tremor, anxiety, hyperthyroidism, multiple sclerosis, and amphetamines, xanthines, lithium, benzodiazepines, or chills, cardiopulmonary resuscitation with chest compressions, and patients who move their limbs during the exam causing sudden irregularities in the ECG baseline resembling premature contractions, or ECG waveforms or other supraventricular and ventricular The patient's muscles quiver.

Preprocessing:

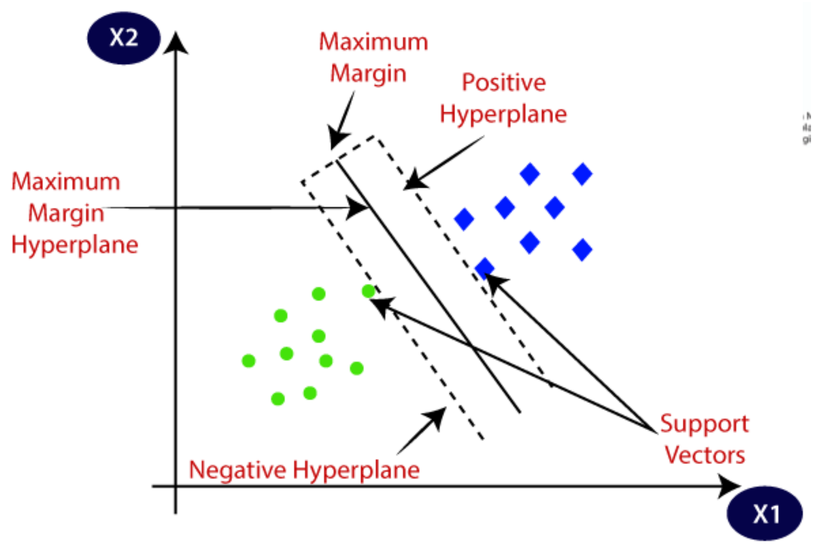
Each ECG signal was divided into non-overlapping segments of 3600 samples (10 s at 360 Hz) and tagged with the dominant label within the time window. Normal sinus rhythm class, and unclassifiable signal outlier class (Q). The number of classes was later reduced to five, including a normal beat, two well-defined arrhythmia classes, a difficult to classify confluent arrhythmia class, and an outlier class.

Normal sinus rhythm, supraventricular beat, premature ventricular beat, fused beat, and unclassifiable beat. This resulted in approximately 105,000 observations, each representing one of his heartbeats. These heartbeats were then plotted over time and saved as a 2D grayscale image consisting. In our project we resize the image to reduce memory usage, converted it to a numpy array and added it. Pixel values are divided by 255 to ensure that each element in the numpy array is between 0 and 1.

SUPPORT VECTOR MACHINE(SVM):

Support Vector Machine is one of the most famous Supervised Learning algorithms that's used for Classification in addition to Regression troubles. However, primarily, it's miles used for Classification troubles in Machine Learning(ML).

The aim of the SVM set of rules is to create the fine line or choice boundary that could segregate n-dimensional area into lessons in order that we are able to effortlessly position the brand new information factor in the precise class withinside the future. This fine choice boundary is also called as hyperplane. Support vector machine chooses the intense vectors that assist in growing the hyperplane. These excessive instances are also called as assist vectors, and as a result the set of rules is called a SVM. Consider the underneath diagram wherein there are exclusive classes which can be labeled the usage of a choice boundary or hyperplane:



Types of SVM:

SVM can be of two types:

Linear SVM:

Linear SVM is used for linearly separable statistics, because of this that if a dataset may be categorised into instructions with the aid of using the usage of a unmarried instantly line, then such statistics is named as linearly separable statistics, and the classifier is used as Linear support vector machine classifier.

Non-linear SVM:

Non-Linear support vector machine is used for non-linearly separated records, this means that if a dataset can not be labeled with the aid of using the use of a immediately line, then such records is called as non-linear records and the classifier used is referred to as Non-linear support vector machine classifier.

Working of Support Vector machine algorithm:

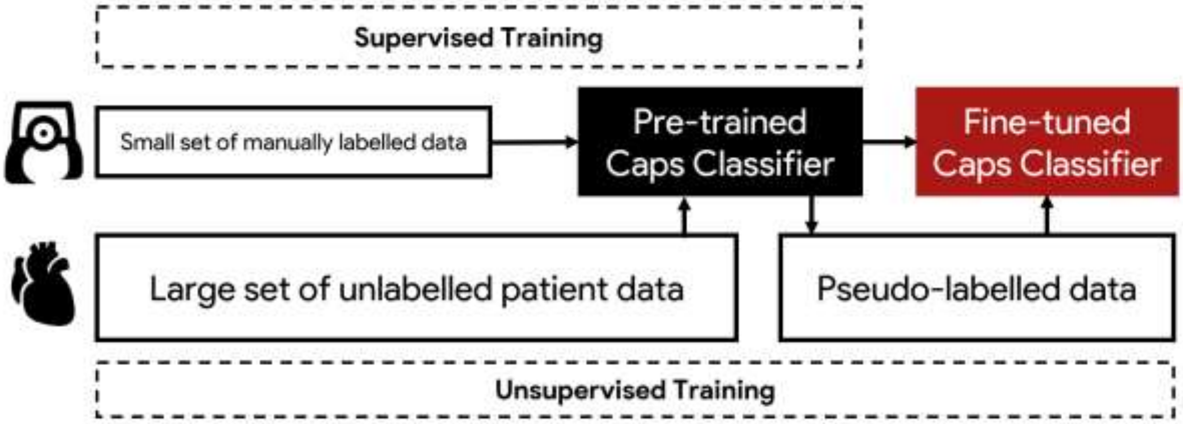
Linear SVM:

The operating of the SVM set of rules may be understood by the usage of an example. Suppose we've got a dataset with tags and the dataset has traits x1 and x2. In our project we want a classifier which could classify the coordinate pair (x1,x2) as inexperienced or blue. Consider the subsequent illustration.

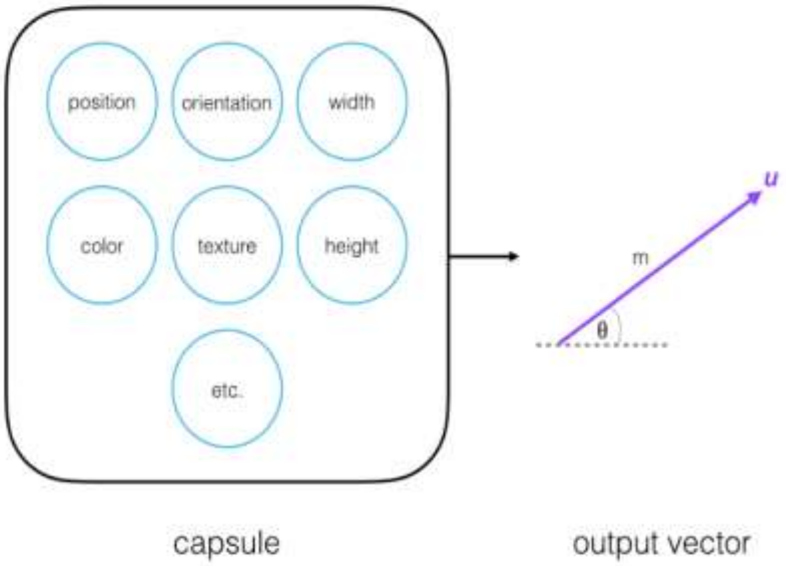
Since that is a dimensional space, we are able to classify without difficulty separate those lessons from the usage of only a direct line. However, there can be a couple of strains which could separate those lessons. Consider the subsequent illustration. Therefore, the SVM set of rules enables discovering the pleasant line or selection boundary. This gold standard boundary or area is known as a hyperplane. The SVM set of rules unearths the nearest factors of hetero strains in each units. These factors are known as guide vectors. The distance between the vector and the hyperplane is known as margin. The main motive of SVM is to maximise that margin. The hyperplane with the most essential margin known as the gold standard hyperplane. Support vector machines are a supervised mastering technique of algorithms used for each category and regression problems. However, its miles are particularly used for system mastering category problems. Support vector machine algorithms may be used for face detection, photograph category, textual content category, etc. Therefore, after a literature survey and analysis, we selected a support vector machine because of the set of rules to put in force the electrocardiogram contrast project.

CAPSULE NETWORK:

Architecture:



Since the advent of capsule networks, machine learning(ML) has been developed, surpassing previous machine learning networks such as Convolutional neural networks. These new technologies are used to improve accuracy and performance in providing accurate output of ECG data. Capsule Neural Network is a machine learning system and a type of artificial neural network used to better model hierarchical relationships. The idea is to add structures called "capsules" to a convolutional neural network and reuse the outputs of some of these capsules to form a more robust, high-level representation of the capsules. The output is a vector consisting of the probability of an analyzing and the pose of that observation. This vector is similar to classification using localization in CNN. Capsnet takes advantage of the fact that viewing angle changes have a non-linear effect at the pixel level, but a line is damaged at the object level for image recognition. As for research and determining, the capsule network is another algorithm for implementing projects. MIT-BIH is a data set available in his Kaggle database that acquires a series of ECG signals from arrhythmia patients.Hence, In our project we are using these type of datasets which was uploaded by research staff at MIT.



Groups of neurons representing numerous photograph properties. CNNs are true at detecting features, much less powerful at exploring spatial relationships amongst features.

Every pill outputs a vector u, with a value and orientation.

RESULTS AND DISCUSSION:

SUPPORT VECTOR MACHINE(SVM):

As per the discussion we have imported the libraries for SVM for processing and we have also declared the methods and variables.

Imports:

In [1]:
```
import numpy as np
import pandas as pd
import os

import matplotlib.pyplot as plt
import csv
import itertools
import collections

import pywt
from scipy import stats

from sklearn.utils import resample
from sklearn.model_selection import train_test_split

from keras import regularizers


%matplotlib inline
```

In [2]:
```
plt.rcParams["figure.figsize"] = (30,6)
plt.rcParams['lines.linewidth'] = 1
plt.rcParams['lines.color'] = 'b'
plt.rcParams['axes.grid'] = True
```

Method definitions:

In [3]:
```
def denoise(data):
    w = pywt.Wavelet('sym4')
    maxlev = pywt.dwt_max_level(len(data), w.dec_len)
    threshold = 0.04 # Threshold for filtering

    coeffs = pywt.wavedec(data, 'sym4', level=maxlev)
    for i in range(1, len(coeffs)):
        coeffs[i] = pywt.threshold(coeffs[i], threshold*max(coeffs[i]))

    datarec = pywt.waverec(coeffs, 'sym4')

    return datarec
```

Variable Definitions:

```
In [4]:
path = '../input/mitbit-arrhythmia-database//mitbih_database/'
window_size = 180
maximum_counting = 10000

classes = ['N','A','V','F','/']
n_classes = len(classes)
count_classes = [0]*n_classes

X = list()
y = list()
```

After the above processes we have Extracted the data for preprocessing.

Data Extraction and Preprocessing:

In [7]:

```
# Records
for r in range(0,len(records)):
    signals = []

    with open(records[r], 'rt') as csvfile:
        spamreader = csv.reader(csvfile, delimiter=',', quotechar='|') # read CSV file\
        row_index = -1
        for row in spamreader:
            if(row_index >= 0):
                signals.insert(row_index, int(row[1]))
            row_index += 1

    # Plot an example to the signals
    if r is 6:
        # Plot each patient's signal
        plt.title(records[6] + " Wave")
        plt.plot(signals[0:700])
        plt.show()

    signals = denoise(signals)
    # Plot an example to the signals
    if r is 6:
        # Plot each patient's signal
        plt.title(records[6] + " wave after denoised")
        plt.plot(signals[0:700])
        plt.show()

    signals = stats.zscore(signals)
    # Plot an example to the signals
    if r is 6:
        # Plot each patient's signal
        plt.title(records[6] + " wave after z-score normalization ")
        plt.plot(signals[0:700])
        plt.show()

    # Read anotations: R position and Arrhythmia class
    example_beat_printed = False
    with open(annotations[r], 'r') as fileID:
        data = fileID.readlines()
        beat = list()

        for d in range(1, len(data)): # 0 index is Chart Head
            splitted = data[d].split(' ') #The split() method splits a string into a list.
            splitted = filter(None, splitted)
            next(splitted) # Time... Clipping
            pos = int(next(splitted)) # Sample ID
            arrhythmia_type = next(splitted) # Type
            if(arrhythmia_type in classes):
                arrhythmia_index = classes.index(arrhythmia_type)
          # if count_classes[arrhythmia_index] > maximum_counting: # avoid overfitting
                #    pass
            #else:
                count_classes[arrhythmia_index] += 1
                if(window_size <= pos and pos < (len(signals) - window_size)):
                    beat = signals[pos-window_size:pos+window_size]     ## REPLACE WITH R-PEAK DETECTION
                    # Plot an example to a beat
                    if r is 6 and not example_beat_printed:
                        plt.title("A Beat from " + records[6] + " Wave")
                        plt.plot(beat)
                        plt.show()
                        example_beat_printed = True

                    X.append(beat)
                    y.append(arrhythmia_index)

# data shape
print(np.shape(X), np.shape(y))
```

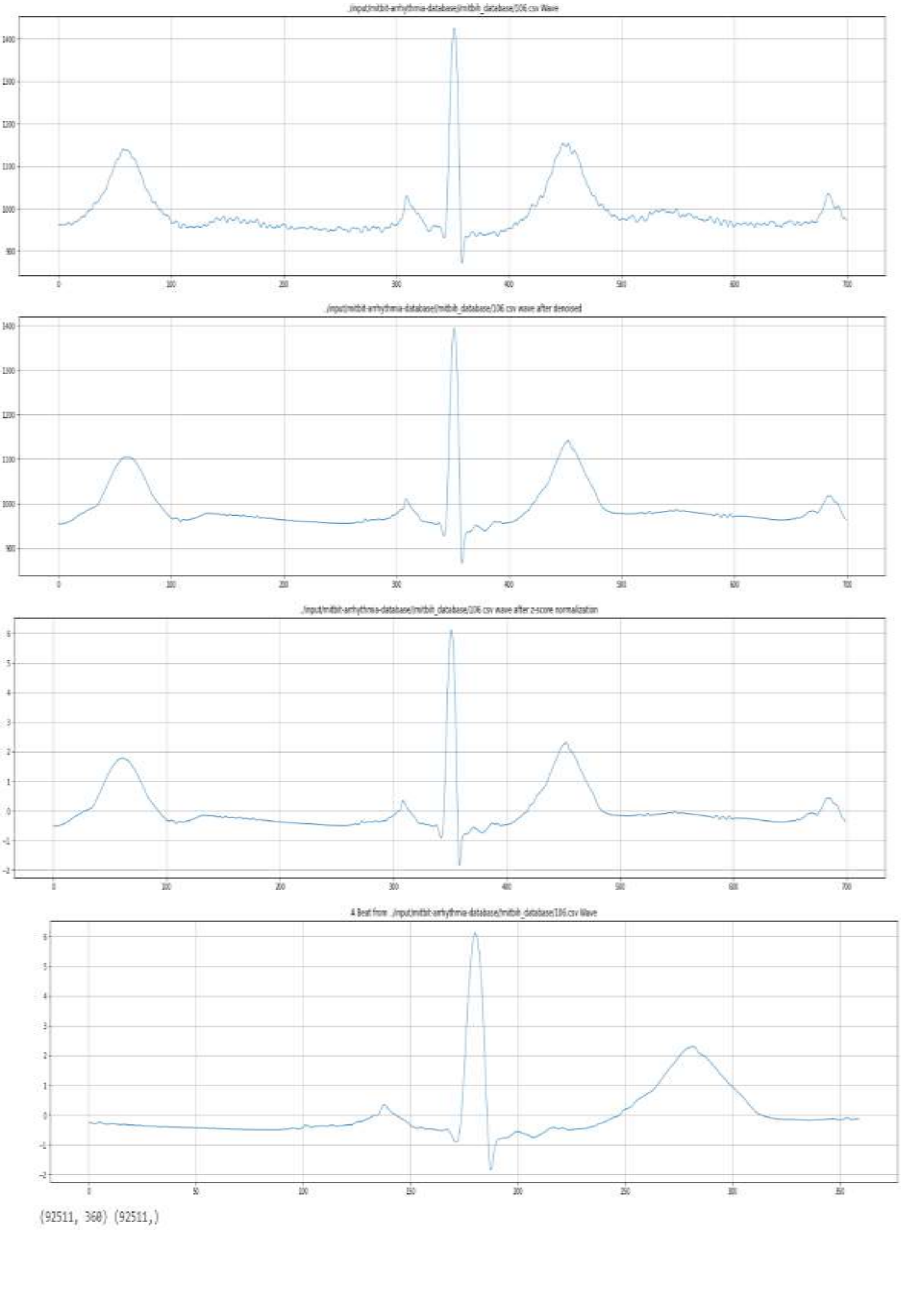
After the data extraction and preprocessing we can the Patient's data.

data_set - DataFrame

Index	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
5	15728773	Male	27	58000	0
6	15598044	Female	27	84000	0
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0
10	15570769	Female	26	80000	0
11	15606274	Female	26	52000	0
12	15746139	Male	20	86000	0
13	15704987	Male	32	18000	0
14	15628972	Male	18	82000	0

Format Resize Background color Column min/max Save and Close Close

The below graph demonstrates the signals of Implementation of Support Vector machines.

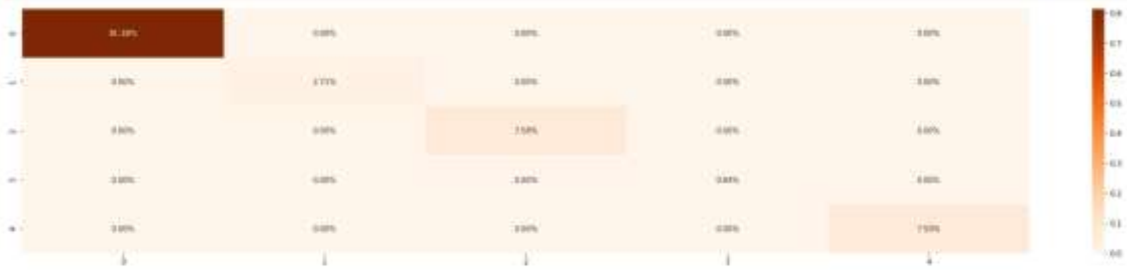


(92511, 360) (92511,)

As we have mentioned in our paper the Support Vector Machine is trained and the output result was obtained. This is achieved by using python code to train SVM. Training my SVM

```
In [11]:
print("Trainig My SVM ...!!!")
from sklearn.svm import SVC
import seaborn as sns
from sklearn import metrics
#Create a svm Classifier
#classifier = SVC(kernel='linear', decision_function_shape='ovo') # Linear Kernel
classifier = SVC(kernel='linear')

#Train the model using the training sets
classifier.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = classifier.predict(X_test)

print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

Trainig My SVM ...!!!
Accuracy: 1.0
```

```
In [12]:
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm/np.sum(cm), annot=True,fmt='.2%', cmap='Oranges')
plt.savefig('Confusion_matrice.png')
```



Capsule network training: (Arrhythmia data) :

After finalizing the performance of the capsule network using the MNIST benchmark dataset, we trained the capsule network using preprocessed arrhythmia images generated from the MIT BIH dataset. First, the same network parameters were used in a new data set. The only addition to the network was an extra CNN layer to reduce the image size. This was necessary to keep the training time within reasonable limits. However, initial tests using arrhythmia images showed that the network was not trained well and the loss did not converge over multiple epochs with different stack sizes. To compensate for this, L2 regularization was used for the CNN layer. Additionally, the number of primary capsules was reduced to 16, reducing the size of the network. The size of the primary CNN kernel was also reduced from 256 used in the MNIST network to 128, which helped reduce the size. Also, while using the Adam optimizer, the learning rate decreased from 1e-3 to 1e-4, a deliberate decrease. A learning rate of 1e-3 was found to be insufficient for model training in preliminary tests using the arrhythmia patients data set.

Capsule Network training Results:

Approximately 12 million training parameters were required for the Arrhythmia training network. The highest addition was due to the size of the training image which required a large output dense layer. The larger image resolution was required to capture the fine details in the Arrhythmia signal. With the modified network with L2 Regularizer, the model was trained using the training images and labels that were created in the Preprocessing stage. The batch training loss over two epochs and batch training accuracy over two epochs is shown in the figure below. The loss converges to an order of $2x10^{-4}$ during two epochs of the training while the training accuracy rises to approximately 80%. However, on testing data, this trained network resulted in very poor accuracy of 20%.

This low testing accuracy could be attributed to multiple factors including:

- Model overfitting
- Pre-processed images not being able to capture ECG signal details accurately due to possible data loss during the image conversion process
- Large network size causes unstable training and convergence.

Additional testing was done by modifying other parameters.

The following results show the loss propagation and accuracy for one epoch training with the hyperparameter changes as outlined in the table. However, the desired testing accuracy could not be achieved with these test cases either.

Capsule Network Results:

Effective on MNIST data
98% test accuracy after one epoch

Ineffective on arrhythmia data
80% training accuracy
20% validation accuracy

Discussions:

While conversion of electrocardiogram signals to 2D images for use with capsules was an interesting approach, likely not the best for this problem. Level of detail required to separate classes in images requires high resolution accuracy, impractical without significant computing power. Extra conversion work was not justified when the Conv1D network outperformed the capsule. Pivoting was important when the capsule network was at a dead end, change in approach allowed us to powerfully implement a semi-supervised routine 13% increase in accuracy.

CODE:

```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import csv
import itertools
import collections
import pywt
from scipy import stats
from sklearn.utils import resample
from sklearn.model_selection import train_test_split
from keras import regularizers
%matplotlib inline

plt.rcParams["figure.figsize"] = (30,6)
plt.rcParams['lines.linewidth'] = 1
plt.rcParams['lines.color'] = 'b'
plt.rcParams['axes.grid'] = True

def denoise(data):
    w = pywt.Wavelet('sym4')
    maxlev = pywt.dwt_max_level(len(data), w.dec_len)
    threshold = 0.04 # Threshold for filtering
    coeffs = pywt.wavedec(data, 'sym4', level=maxlev)
    for i in range(1, len(coeffs)):
        coeffs[i] = pywt.threshold(coeffs[i], threshold*max(coeffs[i]))
    datarec = pywt.waverec(coeffs, 'sym4')
    return datarec

path = '../input/mitbit-arrhythmia-database//mitbih_database/'
window_size = 180
maximum_counting = 10000
```

```
classes = ['N','A','V','F','/']
n_classes = len(classes)
count_classes = [0]*n_classes
X = list()
y = list()

# Read files
filenames = next(os.walk(path))[2]
# Split and save .csv , .txt
records = list()
annotations = list()
filenames.sort()

# segrefating filenames and annotations
for f in filenames:
    filename, file_extension = os.path.splitext(f)
    # *.csv
    if(file_extension == '.csv'):
        records.append(path + filename + file_extension)
    # *.txt
    else:
        annotations.append(path + filename + file_extension)

# Records
for r in range(0,len(records)):
    signals = []
    with open(records[r], 'rt') as csvfile:
        spamreader = csv.reader(csvfile, delimiter=',', quotechar='|') # read CSV file\
        row_index = -1
        for row in spamreader:
            if(row_index >= 0):
                signals.insert(row_index, int(row[1]))
            row_index += 1
```

```
# Plot an example to the signals
if r is 6:
    # Plot each patient's signal
    plt.title(records[6] + " Wave")
    plt.plot(signals[0:700])
    plt.show()
signals = denoise(signals)
# Plot an example to the signals
if r is 6:
    # Plot each patient's signal
    plt.title(records[6] + " wave after denoised")
    plt.plot(signals[0:700])
    plt.show()
signals = stats.zscore(signals)
# Plot an example to the signals
if r is 6:
    # Plot each patient's signal
    plt.title(records[6] + " wave after z-score normalization ")
    plt.plot(signals[0:700])
    plt.show()

    # Read anotations: R position and Arrhythmia class
example_beat_printed = False
with open(annotations[r], 'r') as fileID:
    data = fileID.readlines()
    beat = list()
    for d in range(1, len(data)): # 0 index is Chart Head
        splitted = data[d].split(' ') #The split() method splits a string into a list.
        splitted = filter(None, splitted)
        next(splitted) # Time... Clipping
        pos = int(next(splitted)) # Sample ID
        arrhythmia_type = next(splitted) # Type
```

```
            if(arrhythmia_type in classes):
                arrhythmia_index = classes.index(arrhythmia_type)

           # if count_classes[arrhythmia_index] > maximum_counting: # avoid overfitting
                #    pass
            #else:
                count_classes[arrhythmia_index] += 1
                if(window_size <= pos and pos < (len(signals) - window_size)):
                    beat = signals[pos-window_size:pos+window_size]     ## REPLACE WITH R-
PEAK DETECTION
                    # Plot an example to a beat
                    if r is 6 and not example_beat_printed:
                        plt.title("A Beat from " + records[6] + " Wave")
                        plt.plot(beat)
                        plt.show()
                        example_beat_printed = True
                    X.append(beat)
                    y.append(arrhythmia_index)
  # data shape
  print(np.shape(X), np.shape(y))

  for i in range(0,len(X)):
        X[i] = np.append (X[i],y[i] )
  #        X[i].append(y[i])
  print(np.shape(X))

  X_train_df = pd.DataFrame(X)
  per_class = X_train_df[X_train_df.shape[1]-1].value_counts()
  print(per_class)
  plt.figure(figsize=(30,10))
  my_circle=plt.Circle( (0,0), 0.8, color='white')
  plt.pie(per_class, labels=['N','A','V','F','/'],
colors=['tab:orange','tab:purple','tab:olive','tab:green','tab:blue'],autopct='%1.1f%%')
  p=plt.gcf()
```

```
p.gca().add_artist(my_circle)
plt.show()

X_train,X_test,y_train, y_test = train_test_split(X,y, test_size=0.2)
print("X_train : ", np.shape(X_train))
print("X_test  : ", np.shape(X_test))

print("Trainig My SVM ...!!!")
from sklearn.svm import SVC
import seaborn as sns
from sklearn import metrics
#Create a svm Classifier
#classifier = SVC(kernel='linear', decision_function_shape='ovo') # Linear Kernel
classifier = SVC(kernel='linear')
#Train the model using the training sets
classifier.fit(X_train, y_train)
#Predict the response for test dataset
y_pred = classifier.predict(X_test)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm/np.sum(cm), annot=True,fmt='.2%', cmap='Oranges')
plt.savefig('Confusion_matrice.png')
```

ADVANTAGES:

- Used for better quality prediction for classification.
- While multilayer AI networks may accurately recognise signals and patterns that are largely unrecognisable to human interpreters.
- These approaches can be applied to data obtained from standard 12-lead ECGs or from single-lead or multi lead mobile or wearable ECG technologies.
- History of heart beat data can be analysed over a period of years.

FUTURE SCOPE:

In the world of emerging technologies, heart beat monitoring is becoming increasingly more common in smart wearable devices. Our software would be more helpful for these types of technologies.

Changes in heart beat can be detected for live ecg monitoring used in sport wearables.

CONCLUSION:

An Electrocardiogram is often used for the medical purpose to diagnose and monitor conditions affecting the heart. It is a basic non-invasive tool to assist in diagnosis of heart disease. It analyzes the heart rate and rhythm of the patient. It is a primary procedure for patients with heart disease. Our project helps in a way of producing accurate results of ecg signals and producing classified data. It is used as advanced solution for better health care facilities. Finally the ecg signal is filtered without the noise using our project by Preprocessing and after that the implementation of machine learning algorithms gives the ECG signal with good accuracy.

REFERENCES:

1. Rajpurkar P, Hannun Awni Y, Haghpanahi M, Bourn C and Ng Y A 2017 Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks (Preprint arXiv:1707.01836v1).

2. Janez Demšar. Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research, 7:1–30, 2006. ISSN 1532-4435.

3. G Stiglic, P Kocbek, N Fijacko, M Zitnik, K Verbert, and L Cilar. Interpretability of machine learning based prediction models in healthcare. arxiv 2020. arXiv preprint arXiv:2002.08596, 2019.

4. Liu, C., Yang, M., Di, J., Xing, Y., Li, Y., & Li, J. (2019). Wearable ECG: History, key technologies and future challenges. Chinese Journal of Biomedical Engineering, 38(6), 641–652.

5. Redmond, S. J., Xie, Y., Chang, D., Basilakis, J., & Lovell, N. H. (2012). Electrocardiogram signal quality measures for unsupervised telehealth environments. Physiological Measurement, 33(9), 1517–1533.

6. Zhang, Y., Liu, C., Wei, S., Wei, C., & Liu, F. (2014). ECG quality assessment based on a kernel support vector machine and genetic algorithm with a feature matrix. Journal of Zhejiang University Science C, 15(7), 564–573.

7. Kužílek, J., Huptych, M., Chudáček, V., Spilka, J., & Lhotská, L. (2011). Data driven approach to ECG signal quality assessment using multistep SVM classification. In Computing in Cardiology (pp. 453–455)

8. Zhang, Y., Wei, S., Zhang, L., & Liu, C. (2019). Comparing the performance of random forest, SVM and their variants for ECG quality assessment combined with nonlinear features. Journal of Medical and Biological Engineering, 39(3), 381–392.

CERTIFICATES:

ASFAQ AHMED M

SRI RAMAKRISHNA INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

(Accredited by NACC with 'A' Grade, Approved by AICTE,
All UG Programs are Accredited by NBA &
Affiliated to Anna University, Chennai)
Pachapalayam, Perur Chettipalayam, Coimbatore-10

Certificate of Participation

International Conference

on

Electronics, Communication and Networking (ICECN-2022)

This is to certify that Mr./~~Ms.~~/~~Dr.~~ ASFAQ AHMED M of BANNARI AMMAN INSTITUTE OF TECHNOLOGY, SATHYAMANGALAM has presented a paper entitled COMPARISON OF MACHINE LEARNING ALGORITHMS FOR ECG in the *"International Conference on Electronics, Communication and Networking (ICECN-2022)"* organized by the Department of Electronics and Communication Engineering, Sri Ramakrishna Institute of Technology at Coimbatore -10 on November 2nd 2022.

Dr. S. Mary Praveena	Dr. S. Anila
Coordinator	HOD/ECE & Convenor

THIBESH V

SRI RAMAKRISHNA INSTITUTE OF TECHNOLOGY
(An Autonomous Institution)
(Accredited by NACC with 'A' Grade, Approved by AICTE, All UG Programs are Accredited by NBA & Affiliated to Anna University, Chennai)
Pachapalayam, Perur Chettipalayam, Coimbatore-10

Certificate of Participation

International Conference
on
Electronics, Communication and Networking (ICECN-2022)

This is to certify that Mr./~~Ms.~~/~~Dr.~~ THIBESH V of BANNARI AMMAN INSTITUTE OF TECHNOLOGY, SATHYAMANGALAM has presented a paper entitled COMPARISON OF MACHINE LEARNING ALGORITHMS FOR ECG in the *"International Conference on Electronics, Communication and Networking (ICECN-2022)"* organized by the Department of Electronics and Communication Engineering, Sri Ramakrishna Institute of Technology at Coimbatore - 10 on November 2^{nd} 2022.

Dr. S. Mary Praveena
Coordinator

Dr. S. Anila
HOD/ECE & Convenor

SUDARSHAN V

SRI RAMAKRISHNA INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

(Accredited by NACC with 'A' Grade, Approved by AICTE,
All UG Programs are Accredited by NBA &
Affiliated to Anna University, Chennai)
Pachapalayam, Perur Chettipalayam, Coimbatore-10

Certificate of Participation

International Conference

on

Electronics, Communication and Networking (ICECN-2022)

This is to certify that Mr./~~Ms.~~/~~Dr.~~ SUDARSHAN V of BANNARI AMMAN INSTITUTE OF TECHNOLOGY, SATHYAMANGALAM has presented a paper entitled COMPARISON OF MACHINE LEARNING ALGORITHMS FOR ECG in the ***"International Conference on Electronics, Communication and Networking (ICECN-2022)"*** organized by the Department of Electronics and Communication Engineering, Sri Ramakrishna Institute of Technology at Coimbatore - 10 on November 2nd 2022.

Dr. S. Mary Praveena
Coordinator

Dr. S. Anila
HOD/ECE & Convenor

PHOTOGRAPHS:

Normal ECG graph with PQRS plots:

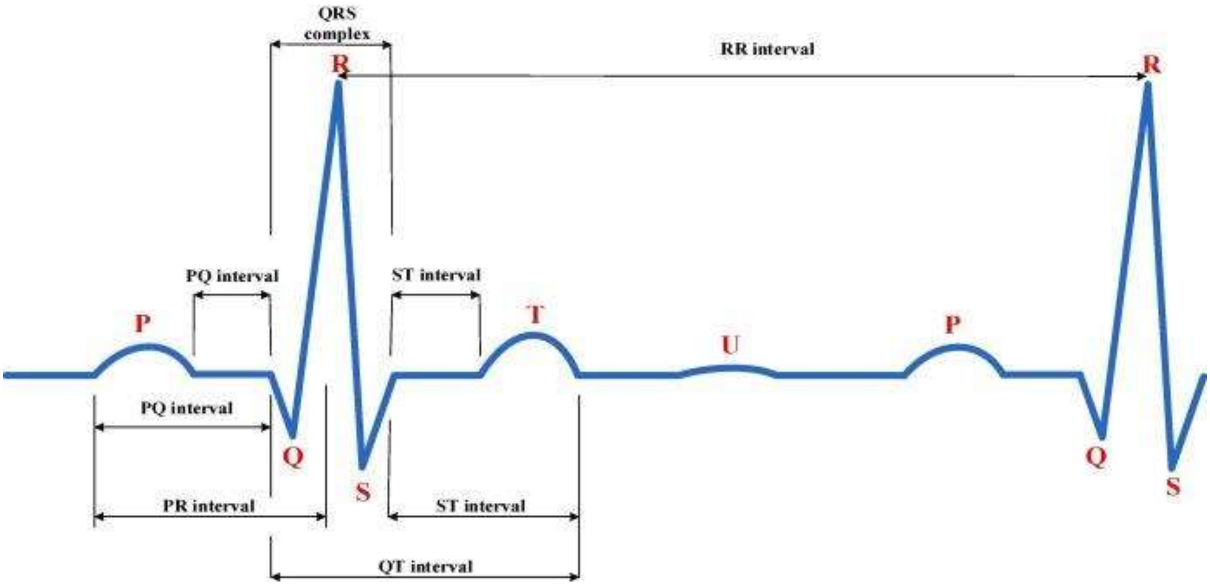
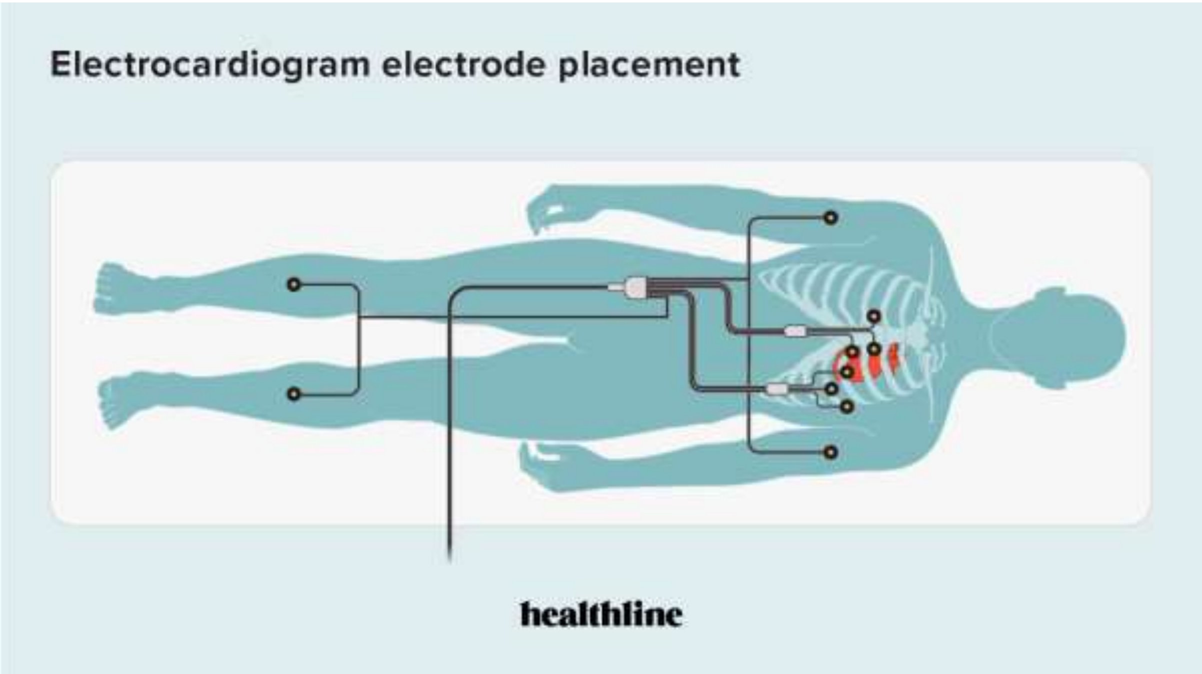


Image of getting ECG data from patient's heart:



ECG data without noise and with noise:

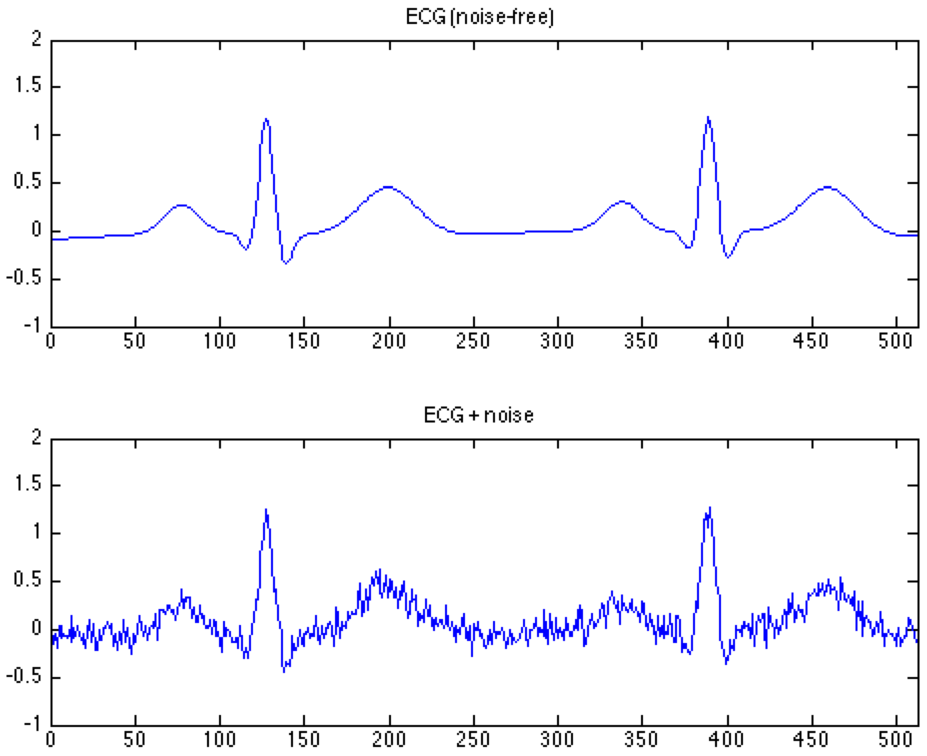


Image of SVM algorithm:

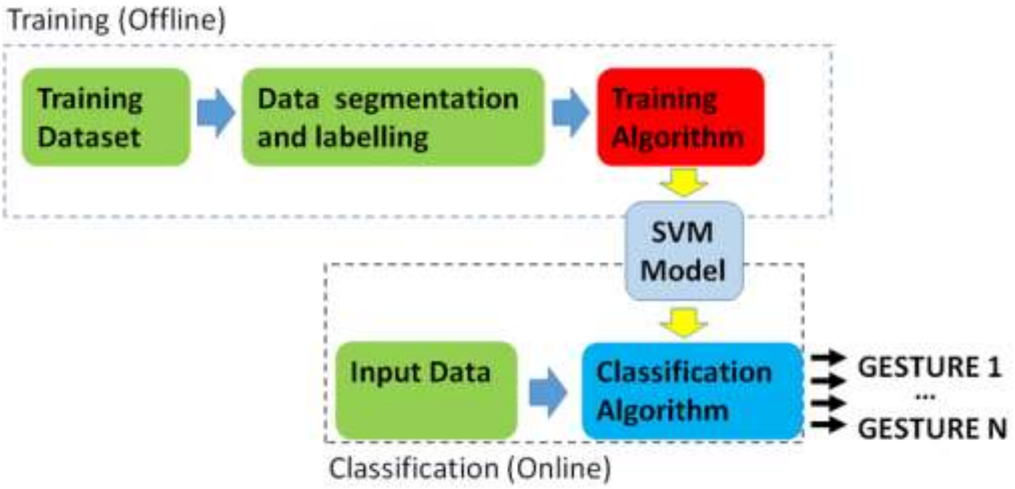


Image of convolutional neural network:

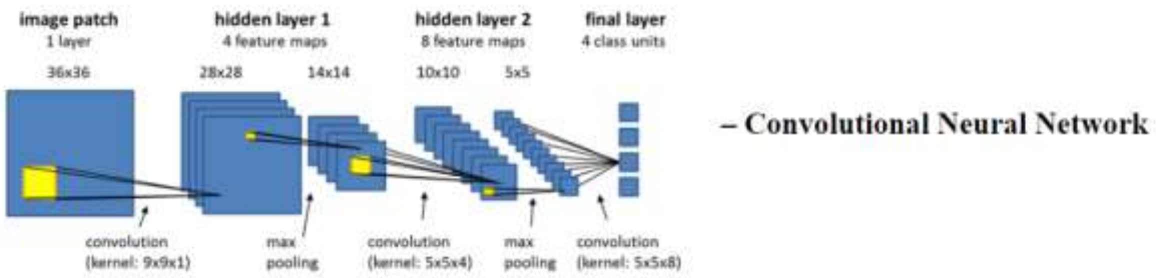


Image of capsule network:

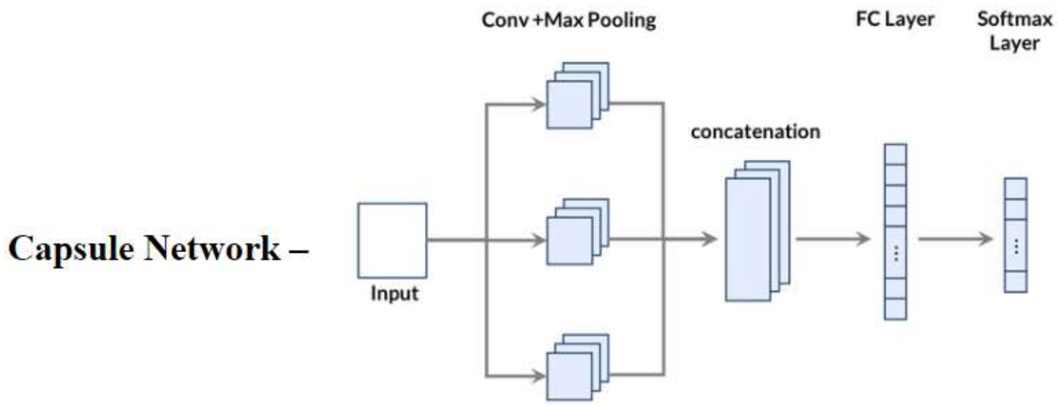
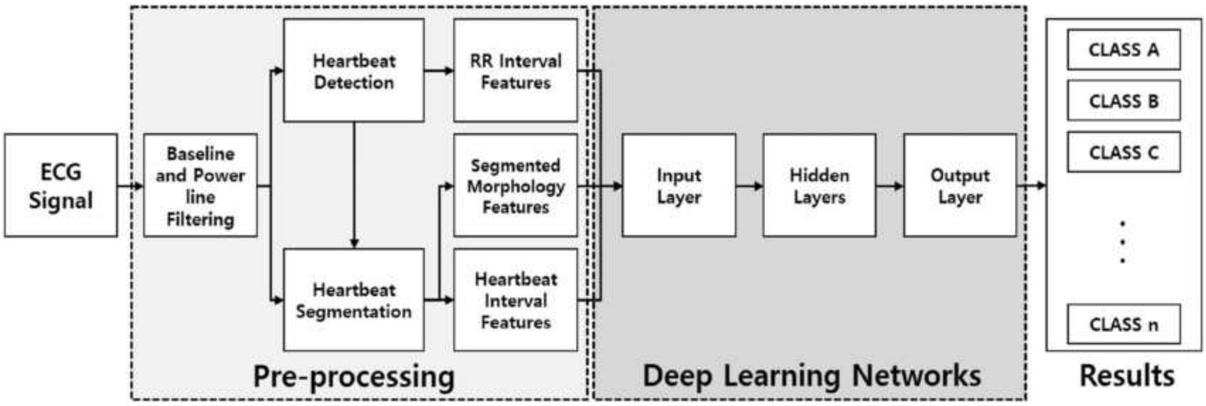


Image of Machine learning Training:

Similar DL Classification Network:


ECG Signal
Baseline and Power line Filtering
Heartbeat Detection
Heartbeat Segmentation
RR Interval Features
Segmented Morphology Features
Heartbeat Interval Features
Pre-processing
Input Layer
Hidden Layers
Output Layer
Deep Learning Networks
CLASS A
CLASS B
CLASS C
CLASS n
Results