SLEEP APNEA DETECTION USING SINGLE LEAD ECG

A PROJECT REPORT

Submitted by

SRI JANANI A 211420104261

SRI VARSHAA N 211420104268

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University,Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“SLEEP APNEA DETECTION USING SINGLE LEAD ECG”** is the bonafide work of **SRI JANANI A (211420104261), SRI VARSHAA.N (211420104268),** who carried out the project work under my supervision.

Signature of the HOD with date	Signature of the Supervisor with date
Dr L.JABASHEELA M.E.,Ph.D.,	**Dr.A.HEMLATHADHEVI M.E.,Ph.D.,**
Professor and Head,	Associate Professor,
Department of Computer Science and Engineering, Panimalar Engineering College, Chennai- 123	Department of Computer Science and Engineering, Panimalar Engineering College, Chennai- 123

Submitted for the Project Viva– Voce examination held on ____________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **SRI JANANI A [211420104261], SRI VARSHAA N [211420104268],** here by declare that this project report titled "**SLEEP APNEA DETECTION USING SINGLE LEAD ECG",** under the guidance of **Dr.A.HEMLATHADHEVI M.E.,Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

SRI JANANI A[211420104261]

SRI VARSHAA N[211420104268]

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI , Dr. C . SAKTHIKUMAR , M.E. , Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department , **Dr. L .JABASHEELA, M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide **Dr.A.HEMLATHADHEVI, M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

Our sincere thanks to **MRS.V.A.PRABHA, SCIENTIST-E, C-DAC, CHENNAI**, for this kind support. Finally, I would like to thank all those who were directly or indirectly helpful in carrying out this research.

NAME OF THE STUDENTS

SRI JANANI A(211420104261)

SRI VARSHAA N (211420104268)

ABSTRACT

Sleep apnea is a common sleep breathing disorder (SBD) in which patients suffer from stopping or decreasing airflow to the lungs for more than 10 sec. Accurate detection of sleep apnea episodes is an important step in devising appropriate therapies and management strategies. This article provides a comprehensive analysis of machine learning and deep learning algorithms on 70 recordings of the PhysioNet ECG Sleep Apnea v1.0.0 dataset. First, electrocardiogram signals were pre-processed and segmented and then machine learning and deep learning methods were applied for sleep apnea detection. Among conventional machine learning algorithms, linear and quadratic discriminate analyses, logistic regression, Gaussian naïve Bayes, Gaussian process, support vector machines, *k*-nearest neighbor, decision tree, extra tree, random forest, AdaBoost, gradient boosting, multi-layer perceptron, and majority voting were implemented. Among deep algorithms, convolutional networks (Alex-Net, VGG16, VGG19 ZF-Net), recurrent networks (LSTM, bidirectional LSTM, gated recurrent unit), and hybrid convolutional-recurrent networks were implemented. The available data were divided into a training set to adjust the model parameters, a validation set to adjust hyperparameters, avoid overfitting, and improve the generalizability of the models, and a test set to evaluate the generalizability of the models on unseen data. This procedure was then repeated in a fivefold cross-validation scheme so that all the recordings were once located in the test set. It was found that the best detection performance is achieved by hybrid deep models where the best accuracy, sensitivity, and specificity were 88.13%, 84.26%, and 92.27%, respectively. This study provides valuable information on how different machine learning and deep learning algorithms perform in the detection of sleep apnea and can further be extended toward the detection of other sleep events.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

We proposed detecting Sleep Apnea Detection from Single-Lead ECG. The advancement of smart wearables technologies has provided a unique opportunity for sleep and health monitoring. However, wearable technologies rely on accurate and real-time monitoring algorithms. This presents a fully integrated system for the detection and prevention of sleep apnea in infants. Apnea has been one of the leading causes of death worldwide with an approximation of about 200 deaths of premature neonatal infants a year. Currently, for the diagnosis of apnea the patients need to go through overnight sleep study in the laboratory, which is very expensive. The proposed device will be a solution for both monitoring and preventing the condition using accurate readings and also judging and giving suggestions on when the patient needs medical help using cloud and artificial intelligence. And all this done in the respective homes of the patients. The detected sleep apnea events can be integrated with existing healthcare systems, electronic health records (EHR), or telemedicine platforms to facilitate communication between patients, healthcare providers, and sleep specialists. This allows for further evaluation, diagnosis, and treatment planning based on the detected events. Overall, sleep apnea detection using single-lead ECG offers a non-invasive and convenient method for screening, diagnosing, and monitoring sleep-related breathing disorders. By leveraging advanced signal processing techniques and machine learning algorithms, it holds promise for improving the accuracy and accessibility of sleep apnea diagnosis, ultimately leading to better management of this prevalent sleep disorder.

1.2 PROBLEM DEFINITION

This is to develop a reliable and accurate method for detecting sleep apnea using data from a single-lead electrocardiogram (ECG) signal .It limits the input data to a single-lead ECG signal, which may have fewer channels and lower spatial resolution compared to multi-lead ECG or PSG recordings, posing challenges for feature extraction and signal processing. There is a need for non-invasive and cost-effective methods for detecting sleep apnea that can be easily implemented in a clinical practice and home settings for continuous and easy monitoring of patients. It is to develop a non-invasive method for sleep apnea detection that is comfortable for patients and can be used in home-based monitoring setups. This is to achieve high accuracy in detecting sleep apnea events based on single-lead ECG signals, minimizing false positives and false negatives. The system is designed to be user-friendly and easy to operate by both patients and healthcare professionals. It involves enhancing the interpretability of the detection system to provide meaningful insights to healthcare providers and facilitate clinical decision-making. It also addresses user acceptance issues and ensures that patients and healthcare professionals trust and are willing to adopt the new technology for sleep apnea detection.

1.3 LITERATURE REVIEW

Sleep accounts for about one-third of the human life span and plays an important role in health and quality of life. Two main stages of sleep are called rapid eye movement (REM) and non-REM (NREM) . REM sleep compared with NREM sleep is associated with increased sympathetic activity, cardiovascular instability, and hemodynamic variations. During NREM, oxygen consumption and heart rate decrease as well as the blood pressure, while in the REM stage, blood pressure and heart rate surge. Sleep apnea may occur during any stage of sleep, but it is most dominant during the REM. This is mainly because the muscles in the upper airway are further relaxed during the REM stage of sleep. There has been considerable research on the development of automatic algorithms for the detection of sleep apnea from a variety of physiological signals among which ECG has shown the most promising results in terms of both convenience and precision. Apnea detection based on a singlelead electrocardiogram (ECG) provides the opportunity for the design and development of wearable technologies based on a single sensor.

Conventional machine learning algorithms are among the first automatic algorithms used for the detection of apnea events. Given the complexity of physiological signals and the limited feature extraction capability of conventional machine learning algorithms, the research focus has recently been shifted toward the more complex deep learning models

In this paper, we provide a fair and unbiased comparison between different conventional machine learning and deep learning algorithms in the detection of sleep apnea occurrence from a single-lead ECG. All the experiments are performed on the same dataset and under the same setting to be able to properly evaluate and compare the performances of different algorithms. Unlike most studies that tune their model hyperparameters based on the same data used for final evaluation, we used three sets of data: a training set to train the model parameters, a validation set to find the model optimum hyperparameters, and a test set to evaluate the generalizability of the developed models on unseen data. In this the performance of a few deep learning methods was analyzed for the detection of sleep apnea. In this article, we have significantly extended our comparisons by studying the performance of 14 different conventional machine learning methods and 19 different deep learning methods. A detailed comparison between different models as well as eight other state-of-the-art techniques is provided. We also performed a feature importance analysis and demonstrated which ECG features are most effective for the detection of apnea episodes.

CHAPTER 2

SYSTEM ANALYSIS

2.1.EXISTING SYSTEM

Technologies presented in recent literatures for apnea include monitoring the oxygen levels using sensors and, in some cases, include chest belt, straingauge, etc. These methods are not suitable for neonatal respiratory monitoring because babies cannot wear these devices due to the sensitive nature of their skin. The proposed device uses a spo2 sensor, that represents a very small form factor and consumes very small amount of power which makes it suitable for daily home usage. An algorithm used with the MAX30102 (For Spo2)output signal can make up for the error associated with Spo2readings with ambient temperature changes. By this the device could help notify the variations in the heart rate of the baby which is caused due to the variations in oxygen levels leading to complications.

2.2.PROPOSED SYSTEM

We provide a fair and unbiased comparison between different conventional machine learning and deep learning algorithms in the detection of sleep apnea occurrence from a single-lead ECG. All the experiments are performed on the same dataset and under the same setting to be able to properly evaluate and compare the performances of different algorithms. Unlike most studies that tune their model hyperparameters based on the same data used for final evaluation, we used three sets of data :a training set to train the model parameters, a validation set to find the model optimum hyperparameters, and a test set to evaluate the generalizability of the developed models on unseen data. where the performance of a few deep learning methods was analyzed for the detection of sleep apnea. We also performed a feature importance analysis and demonstrated which ECG features are most effective for the detection of apnea episodes.

2.3.IMPLEMENTATION ENVIROMENT

2.3.1.SOFTWARE REQUIREMENT

- Operating System : Windows 10 (64 bit)
- Software : Python
- Tools : Anaconda

2.3.2.HARDWARE REQUIREMENT

- Processor: I3 or above
- Memory (RAM): 400 GB or above
- Hard Drive: 500 GB or above
- Webcam -1

2.4.ALGORITHM

CNN (Convolutional Neural Network)

Train the CNN model on the preprocessed data. This involves feeding the ECG segments through the network, computing the loss (difference between predicted and actual labels), and updating the network parameters using optimization algorithms like stochastic gradient descent (SGD) or Adam. Validate the model on a separate validation dataset to ensure it generalizes well. Tune hyperparameters such as learning rate, batch size, and network architecture to improve performance.

VGG16:

From the segments of the preprocessed ECG signal, extract pertinent features. Generally, you would have to rearrange the input data for VGG16 such that it fits the network's expected input size (e.g., 224x224 pixels). Convolutional layers might be used to learn features directly from the raw signal, or hand-crafted features like heart rate variability (HRV) could be employed. Use the VGG16 model that has already been trained as a feature extractor. At the top of the network, remove the fully linked layers and add layers appropriate for detecting sleep apnea. For binary classification, you can add a few dense layers and then a final output layer (apnea or non-apnea). Optionally, use the gathered ECG data to adjust the weights of the newly added layers and the pre-trained VGG16 layers.

ZFNET-BilSTM:

Implementing a ZFNet (a variant of CNN architecture) combined with Bidirectional Long Short-Term Memory (BiLSTM) for sleep apnea detection involves leveraging both CNNs for feature extraction from ECG signals and BiLSTM for sequence modeling to capture temporal dependencies. Create the ZFNet-BiLSTM hybrid architecture. For sequence modeling, you would combine the ZFNet layers first, then the BiLSTM layer or layers. For regularization, you can also add more layers, like dropout and completely linked layers. Utilizing the enhanced and preprocessed ECG data, train the ZFNet-BiLSTM model. To do this, the ECG segments are sent through the network, the loss the difference between the anticipated and real labels is computed, and the network parameters are updated using optimization methods such as Adam or stochastic gradient descent (SGD). To make sure the model generalizes well, validate it using a different validation dataset.

CHAPTER 3

SYSTEM DESIGN

3.1 UML DIAGRAMS

ACTIVITY DIAGRAM

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

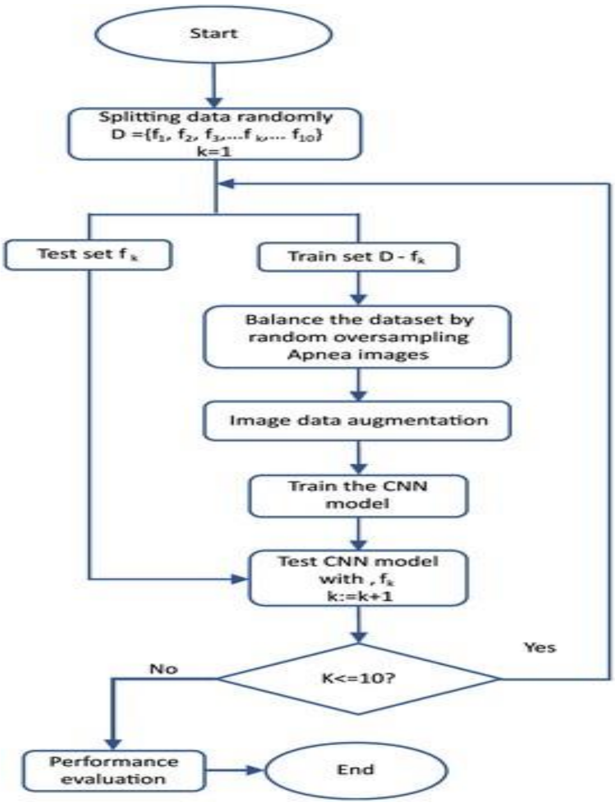

Fig: 3.1.1.Activity diagram for Sleep Apnea

CHAPTER 4

SYSTEM ARCHITECTURE

4.1 ARCHITECTURE OVERVIEW

The architecture diagram for sleep apnea detection using single-lead ECG by using different machine learning and deep learning algorithms. Below is a description of each component in the architecture diagram:

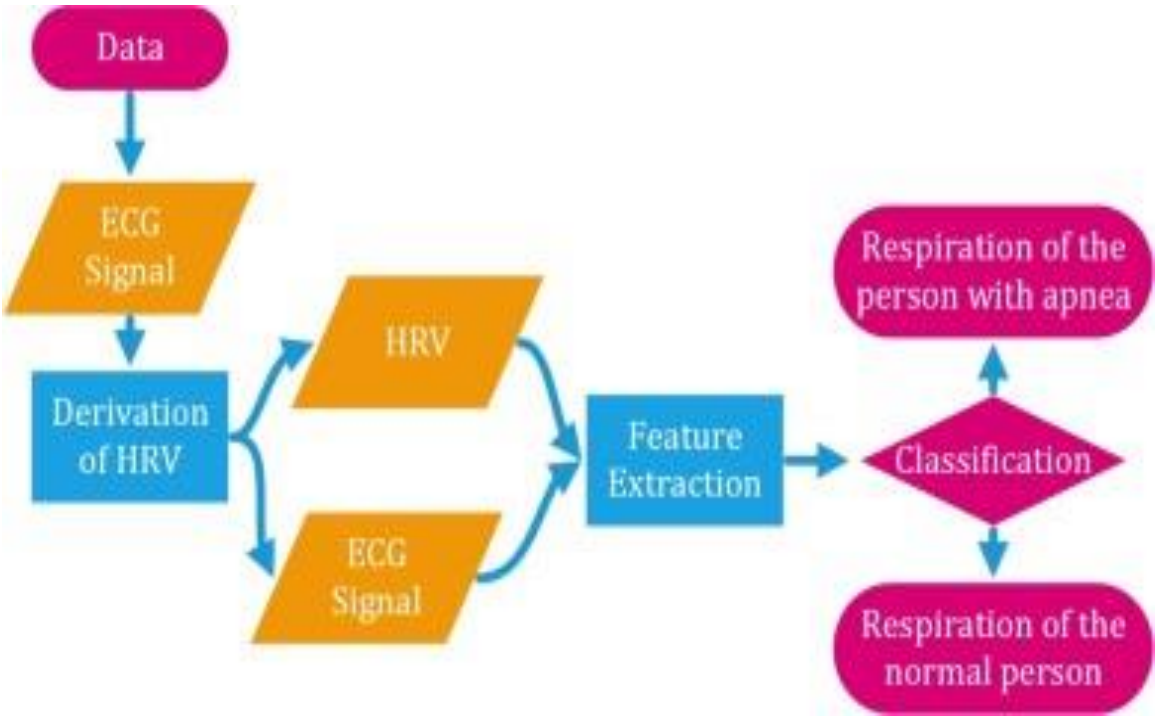

Fig: 4.1.1. System Architecture for Sleep Apnea Detection

Data:

PhysioNet Apnea-ECG Database was used to build our models and to compare their performances. The database contains 70 recordings from 32 individuals which is divided into two groups.35 recordings are used for training and 35 recordings are used for detection.

ECG Signal:

The recorded ECG signals are normalized and the R-peaks are detected to carryout the HRV analysis.

HRVAnalysis:

Heartrate variability (HRV) analysis can play a significant role in sleep apnea detection. HRV refers to the variation in the time interval between consecutive heartbeats, which is influenced by the autonomic nervous system (ANS).

Feature Extraction:

Feature extraction in sleep apnea detection involves identifying and quantifying relevant characteristics or patterns from physiological signals, such as electrocardiogram (ECG), respiratory signals, or oxygen saturation levels, that are indicative of apnea events.

Classification:

The attribute 60-second segments are taken from the collected ECG signals, and sleep apnea is identified in each segment. After a combined analysis of all these signs, sleep apnea is identified in the patient.

4.2 MODULE DESIGN

SPECIFICATION

CLASSIFICATION OF PEAKS

ECG signals were segmented into 1-min intervals. To extract the R–R Intervals from the ECG signal, the Hamilton Rpeak detection method based on an open-source code was employed . A median filter proposed by Chen et al. [34]was used for removing physiologically uninterpretable points. The extracted R–R Intervals were then presented to the developed machine learning algorithms after appropriate processing as is discussed in the Section IV. In addition to the R–R Intervals, the amplitudes of R-peaks were also extracted and presented to the deep models. Cubic interpolation at 3 Hz was applied to sample R–R intervals and R-peak amplitudes at an equal rate. Finally, the interpolated R–R Intervals and R-peak amplitudes were fed to the deep models. To capture the time pattern of the input data when using DRNNs, the input data were segmented into n segments of 60/n s and n-cell DRNNs were designed to learn from the input data (here, n was set to 2).

FEATURE EXTRACTION USING MACHINE LEARNING

Feature extraction, feature selection, and classification are the main steps in conventional machine learning methods. Feature engineering plays a vital role in boosting the performance of these machine learning algorithms. Here, we developed aunified feature engineering framework for the conventional machine learning algorithms . First, ECG signals were pre-processed. Then time, frequency, and nonlinear features of the ECG signal were extracted. In addition to this, principal component analysis (PCA) was applied for the most effective and important features for classification. Here, we applied PCA for dimension reduction. Different machine learning algorithms have been applied for sleep apnea detection. However, the main problem in this field is the lack of a fair and unified comparison between different algorithms. In this study, different well-known machine learning algorithms are compared.

DETECTION USING DEEP LEARNING

The advancement of smart wearables technologies has provided a unique opportunity for sleep and health monitoring. However, wearable technologies rely on accurate and real-time monitoring algorithms. We provided a comprehensive comparison between different machine learning and deep learning algorithms, in a unified framework, for the detection of sleep apnea from single-lead ECG. It was observed that deep learning models outperform conventional machine learning techniques. Among deep learning algorithms, CNN based models performed better than DRNNs in our application where short segments of ECG (1 min) were processed. The best detection performances were obtained using hybrid CNNDRNN architectures among which ZFNet-BiLSTM achieved the highest accuracy and specificity and ZFNet-GRU achieved the best sensitivity. Based on the achieved results, the use of hybrid deep neural networks is recommended for sleep apnea detection from ECG.

CHAPTER 5

SYSTEM IMPLEMENTATION

A.1 Module:1


```
"cells": [
 {
  "cell_type": "markdown",
  "id": "55e19dc9",
  "metadata": {},
  "source": [
   "# Import Libraries"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 1,
  "id": "0e3b8856",
  "metadata": {},
  "outputs": [],
  "source": [
   "import numpy as np\n",
   "import wfdb\n",
   "import hrvanalysis\n",
   "import matplotlib.pyplot as plt\n",
   "from wfdb import get_record_list,plot_wfdb\n",
   "from wfdb.io import rdheader,rdann,rdrecord\n",
   "from wfdb.processing import normalize_bound\n",
   "import  ecgdetectors\n",
   "import scipy\n",
   "from scipy import signal\n",
   "#import emd"
  ]
 },
 {
```

"cell_type": "markdown",
"id": "94dc183f",
"metadata": {},
"source": [
 "# Database detail"
]
},
{
"cell_type": "code",
"execution_count": 2,
"id": "e38c4400",
"metadata": {},
"outputs": [
 {
  "data": {
   "text/plain": [
    "['a01',\n",
    " 'a01er',\n",
    " 'a01r',\n",
    " 'a02',\n",
    " 'a02er',\n",
    " 'a02r',\n",
    " 'a03',\n",
    " 'a03er',\n",
    " 'a03r',\n",
    " 'a04',\n",
    " 'a04er',\n",
    " 'b01',\n",
    " 'b01er',\n",
    " 'b01r',\n",
    " 'b02',\n",
    " 'b03',\n",
    " 'b04',\n",
    " 'b05',\n",
    " 'c01',\n",
    " 'c01er',\n",
    " 'c01r',\n",

" 'c02',\n",
" 'c02er',\n",
" 'c02r',\n",
" 'x01',\n",
" 'x02',\n",
" 'x03',\n",
" 'x04',\n",
" 'x05',\n",
" 'x06',\n",
" 'x07',\n",
" 'x08',\n",
" 'x09',\n",
" 'x10',\n",
" 'x11',\n",
" 'x12',\n",
" 'x13',\n",
" 'x14',\n",
" 'x15',\n",
" 'x16',\n",
" 'x17',\n",

]
},
"execution_count": 2,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"get_record_list('apnea-ecg/1.0.0/',records='all')"
]
},
{
"cell_type": "markdown",
"id": "fb52c33c",
"metadata": {},
"source": [

```
"# Reading ECG record (data, header, annotation)"
]
},
{
"cell_type": "code",
"execution_count": 3,
"id": "68411b30",
"metadata": {},
"outputs": [],
"source": [
"ecg_record_head=rdheader(record_name=r'D:\\sleep_apnea\\apnea-ecg-database-1.0.0/a01',pn_dir=None,rd_segments=True)\n",
"ecg_record_ann=rdann(record_name=r'D:\\sleep_apnea\\apnea-ecg-database-1.0.0/a01',extension='apn',sampto=120000,pn_dir=None)\n",
"ecg_record_sig=rdrecord(record_name=r'D:\\sleep_apnea\\apnea-ecg-database-1.0.0/a01',sampto=120000,pn_dir=None)"]
```

Module-2

```
import numpy as np\n",
"import wfdb\n",
"import hrvanalysis\n",
"import matplotlib.pyplot as plt\n",
"from wfdb import get_record_list,plot_wfdb\n",
"from wfdb.io import rdheader,rdann,rdrecord\n",
"from wfdb.processing import normalize_bound\n",
"import  ecgdetectors\n",
"import scipy\n",
"from scipy import signal\n",
"import pandas as pd\n",
"from scipy.signal import hilbert,find_peaks\n",
"from PyEMD import EMD,Visualisation,CEEMDAN\n",
"from hrvanalysis import remove_outliers, remove_ectopic_beats, interpolate_nan_values\n",
"from scipy.interpolate import interp1d\n",
"from scipy.integrate import trapz\n",
"import os\n",
"import glob\n",
```

"from scipy.stats import skew\n",
"from hrvanalysis import get_frequency_domain_features,get_csi_cvi_features,get_geometrical_features,get_poincare_plot_features\n",
"from scipy.stats import skew\n",
"from scipy.stats import zscore\n",
"import heartpy"
]
},
{
"cell_type": "code",
"execution_count": 72,
"id": "16fb1261",
"metadata": {},
"outputs": [],
"source": [
"def time_domain_features(nn_intervals_list):\n",
"   time_domain_feature={};\n",
"   length_int = len(nn_intervals_list) \n",
"   diff_nni = np.diff(nn_intervals_list)\n",
"   mean_nni=np.mean(nn_intervals_list)\n",
"   median_nni = np.median(nn_intervals_list)\n",
"   range_nni = max(nn_intervals_list) - min(nn_intervals_list)\n",
"   skewness_nni=skew(nn_intervals_list)\n",
"   sdsd = np.std(diff_nni)\n",
"   rmssd = np.sqrt(np.mean(diff_nni ** 2))\n",
"   nni_50 = sum(np.abs(diff_nni) > 0.050)\n",
"   pnni_50 = 100 * nni_50 / length_int\n",
"   nni_20 = sum(np.abs(diff_nni) > .020)\n",
{
"cell_type": "code",
"execution_count": 73,
"id": "70cda66c",
"metadata": {},
"outputs": [],
"source": [
"def frequency_domain_features(nn_intervals_list):\n",
"   frequency_domain_feature={}\n",
"   x =np.cumsum(nn_intervals_list)\n",

```
    "    f = interp1d(x,nn_intervals_list, kind='cubic',fill_value='extrapolate')\n",
    "    \n",
    "    fs = 3\n",
    "    steps = 1 / fs\n",
    "    xx = np.arange(1, np.max(x), steps)\n",
    "    rr_interpolated = f(xx)\n",
    "    v=len(rr_interpolated)/2;\n",
    "    fxx, pxx = 
signal.welch(x=rr_interpolated,window='hann',fs=3,nperseg=v,noverlap=v//2)\n",
    "    cond_vlf = (fxx >= 0) & (fxx < 0.04)\n",
    "    cond_lf = (fxx >= 0.04) & (fxx < 0.15)\n",
    "    cond_hf = (fxx >= 0.15) & (fxx < 0.4)\n",
    "    \n",
    "    \n",
    "    vlf = trapz(pxx[cond_vlf], fxx[cond_vlf])\n",
    "    lf = trapz(pxx[cond_lf], fxx[cond_lf])\n",
    "    hf = trapz(pxx[cond_hf], fxx[cond_hf])\n",
    "    \n",
    "    \n",
    "    total_power = vlf + lf + hf\n",
    "\n",
    "  \n",
    "    peak_vlf = fxx[cond_vlf][np.argmax(pxx[cond_vlf])]\n",
    "    peak_lf = fxx[cond_lf][np.argmax(pxx[cond_lf])]\n",
    "    peak_hf = fxx[cond_hf][np.argmax(pxx[cond_hf])]\n",
    "\n",
    "   \n",
    "    lf_nu = 100 * lf / (lf + hf)\n",
    "    hf_nu = 100 * hf / (lf + hf)\n",
    "\n",
    "   \n",
    "    frequency_domain_feature['Total_power']=total_power\n",
    "    frequency_domain_feature['VLF Power']=vlf\n",
    "    frequency_domain_feature['HF Power']=hf\n",
    "    frequency_domain_feature['LF Power']=lf\n",
    "    frequency_domain_feature['HFnu']=hf_nu\n",
    "    frequency_domain_feature['LFnu']=lf_nu\n",
    "    frequency_domain_feature['LF/HF ratio']=lf/hf\n",
    "    frequency_domain_feature['Peak of VLF']=peak_vlf\n",
```

```
"    frequency_domain_feature['Peak of LF']=peak_lf\n",
"    frequency_domain_feature['peak of hf']=peak_hf\n",
"    return frequency_domain_feature"
 ]
},
```

Module:3

```
  "import pickle\n",
  "import numpy as np\n",
  "import os\n",
  "from keras.callbacks import LearningRateScheduler,EarlyStopping\n",
  "from keras.layers import Dense,Flatten,MaxPooling2D,Conv2D\n",
  "from keras.regularizers import l2\n",
  "from sklearn.model_selection import train_test_split\n",
  "import tensorflow as tf\n",
  "from keras.models import Sequential\n",
  "from sklearn.model_selection import StratifiedKFold\n",
  "import pandas as pd\n",
  "from sklearn.metrics import confusion_matrix\n",
  "from sklearn.metrics import f1_score\n",
  "    "
 ]
},
{
 "cell_type": "code",
 "execution_count": 2,
 "id": "436019e3",
 "metadata": {},
 "outputs": [],
 "source": [
```

```
   "#!pip install tensorflow"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 4,
  "id": "c49d3e50",
  "metadata": {},
  "outputs": [],
  "source": [
   "R_R=pd.read_csv(r'E:\\sleep_apnea\\M3/RR_input.csv')\n",
   "R_amp=pd.read_csv(r'E:\\sleep_apnea\\M3/Ramp_input.csv')\n",
   "Ann_in=pd.read_csv(r'E:\\sleep_apnea\\M3/Ann_input.csv')"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 5,
  "id": "c78a0886",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/html": [
      "<div>\n",
      "<style scoped>\n",
      "    .dataframe tbody tr th:only-of-type {\n",
      "        vertical-align: middle;\n",
      "    }\n",
```

```
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>1</th>\n",
"      <th>2</th>\n",
"      <th>3</th>\n",
"      <th>4</th>\n",
"      <th>5</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>N</td>\n",
"      <td>N</td>\n",
"      <td>N</td>\n",
"      <td>N</td>\n",
"      <td>N</td>\n",
"      <td>N</td>\n",
```

```
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>...</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "      <td>N</td>\n",
    "    </tr>\n",
    "  </tbody>\n",
    "</table>\n",
    "<p>1 rows × 12211 columns</p>\n",
    "</div>"
   "execution_count": 5,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "Ann_in"
 ]
},
```

```
{
 "cell_type": "code",
 "execution_count": 6,
 "id": "f45d79b2",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/html": [
     "<div>\n",
     "<style scoped>\n",
     "    .dataframe tbody tr th:only-of-type {\n",
     "        vertical-align: middle;\n",
     "    }\n",
     "\n",
     "    .dataframe tbody tr th {\n",
     "        vertical-align: top;\n",
     "    }\n",
     "\n",
     "    .dataframe thead th {\n",
     "        text-align: right;\n",
     "    }\n",
     "</style>\n",
     "<table border=\"1\" class=\"dataframe\">\n",
     "  <thead>\n",
     "    <tr style=\"text-align: right;\">\n",
     "      <th></th>\n",
     "      <th>1</th>\n",
     "      <th>2</th>\n",
```

```
"      <th>3</th>\n",
"      <th>4</th>\n",
"      <th>5</th>\n",
"      <th>6</th>\n",
"      <th>7</th>\n",
"      <th>8</th>\n",
"      <th>9</th>\n",
"      <th>10</th>\n",
"      <th>...</th>\n",
 "    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>0.364434</td>\n",
"      <td>0.337631</td>\n",
"      <td>0.479507</td>\n",
"      <td>0.542216</td>\n",
"      <td>0.427492</td>\n",
"      <td>0.589773</td>\n",
"      <td>0.248231</td>\n",
"      <td>0.360798</td>\n",
"      <td>0.238218</td>\n",
"      <td>0.322059</td>\n",
"      <td>...</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>0.636907</td>\n",
```

"   <td>0.942566</td>\n",
"   <td>0.589272</td>\n",
"   <td>0.456118</td>\n",
"   <td>0.495767</td>\n",
"   <td>-0.526756</td>\n",
"   <td>0.201293</td>\n",
"   <td>1.055831</td>\n",
"   <td>0.662136</td>\n",
"   <td>0.447851</td>\n",
"   <td>...</td>\n",
"   <tr>\n",
"   <th>2</th>\n",
"   <td>0.687279</td>\n",
"   <td>0.290031</td>\n",
"   <td>0.686516</td>\n",
"   <td>1.245632</td>\n",
"   <td>0.292241</td>\n",
"   <td>0.264685</td>\n",
"   <td>0.447059</td>\n",
"   <td>0.363134</td>\n",
"   <td>0.371452</td>\n",
"   <td>0.514150</td>\n",
"   <td>...</td>\n",
"   </tr>\n",
"   <tr>\n",
"   <th>3</th>\n",
"   <td>1.092107</td>\n",
"   <td>1.177531</td>\n",
"   <td>0.355396</td>\n",

```
"      <td>0.437273</td>\n",
"      <td>0.426968</td>\n",
"      <td>1.846100</td>\n",
"      <td>0.145463</td>\n",
"      <td>0.348760</td>\n",
"      <td>0.556847</td>\n",
"      <td>...</td>\n",
"      <td>0.331656</td>\n",
"      <td>0.191366</td>\n",
"      <td>0.238225</td>\n",
"      <td>0.153526</td>\n",
"      <td>0.160626</td>\n",
"      <td>0.164168</td>\n",
"      <td>0.300943</td>\n",
"      <td>0.312570</td>\n",
"      <td>0.129185</td>\n",
"      <td>-0.011484</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>...</th>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
```

```
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>175</th>\n",
"      <td>1.890818</td>\n",
"      <td>0.126771</td>\n",
"      <td>0.528686</td>\n",
"      <td>1.296374</td>\n",
"      <td>0.624352</td>\n",
"      <td>0.615634</td>\n",
"      <td>0.144447</td>\n",
"      <td>0.694091</td>\n",
"      <td>0.954221</td>\n",
"      <td>0.220534</td>\n",
"      <td>...</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>176</th>\n",
```

```
"    <td>0.163063</td>\n",
"    <td>-0.398215</td>\n",
"    <td>0.409592</td>\n",
"    <td>0.631174</td>\n",
"    <td>0.276174</td>\n",
"    <td>0.471937</td>\n",
"    <td>0.748590</td>\n",
"    <td>0.323428</td>\n",
"    <td>0.421652</td>\n",
"    <td>0.461786</td>\n",
"    <td>...</td>\n",
"    <td>0.270029</td>\n",
"    <td>0.129616</td>\n",
"    <td>0.195705</td>\n",
"    <td>0.159984</td>\n",
"    <td>0.281186</td>\n",
"    <td>0.209262</td>\n",
"    <td>0.249323</td>\n",
"    <td>0.086014</td>\n",
"    <td>0.112347</td>\n",
"    <td>0.124506</td>\n",
"   </tr>\n",
"   <tr>\n",
"    <th>177</th>\n",
"    <td>0.397967</td>\n",
"    <td>1.738980</td>\n",
"    <td>0.250270</td>\n",
"    <td>0.584795</td>\n",
"    <td>0.214184</td>\n",
```

```
"      <td>0.708686</td>\n",
"      <td>0.176794</td>\n",
"      <td>0.510741</td>\n",
"      <td>0.309275</td>\n",
"      <td>0.101536</td>\n",
"      <td>...</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>178</th>\n",
"      <td>0.648367</td>\n",
"      <td>0.131107</td>\n",
"      <td>0.533536</td>\n",
"      <td>0.483676</td>\n",
"      <td>3.344856</td>\n",
"      <td>0.446122</td>\n",
"      <td>0.230701</td>\n",
"      <td>0.250903</td>\n",
"      <td>0.267959</td>\n",
"      <td>-0.430354</td>\n",
"      <td>...</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>179</th>\n",
"      <td>0.364434</td>\n",
"      <td>0.337631</td>\n",
"      <td>0.479507</td>\n",
"      <td>0.542216</td>\n",
"      <td>0.427492</td>\n",
"      <td>0.589773</td>\n",
```

CHAPTER 6

PERFORMANCE EVALUATION

6.1. PERFORMANCE PARAMETERS

Performance parameters for sleep apnea detection using single-lead ECG typically include metrics that assess the accuracy, reliability, and effectiveness of the detection system. Here are some commonly used performance parameters:

ACCURACY:

Accuracy measures the overall correctness of the detection system and is calculated as the ratio of correctly classified cases (both true positives and true negatives) to the total number of cases.

SENSITIVITY:

Sensitivity, also known as the true positive rate, measures the proportion of actual positive cases (i.e., true apnea events) that are correctly identified by the detection system.

SPECIFICITY:

Specificity, also known as the true negative rate, measures the proportion of actual negative cases (i.e., non-apnea events) that are correctly identified as negative by the detection system.

F-SCORE:

The F-score, also known as the F1 score, is a metric commonly used in machine learning to evaluate the performance of classification algorithms, particularly in binary classification tasks. It considers both the precision (positive predictive value) and recall (sensitivity) of the model to provide a single score that balances these two measures.

Precision (Positive Predictive Value):

Precision measures the proportion of true positive predictions (correctly predicted positive cases) out of all instances predicted as positive

TABULATED RECORDS:

	0	1	2	3	4	5
0	-0.879350	4.696814	-1.279444	-1.005479	-0.511546	0.006653
1	-1.352353	3.195900	-3.376741	-0.437663	-0.064926	-1.114255
2	2.448429	3.933600	-3.542013	0.379828	0.146574	2.190057
3	-0.340360	2.641163	-2.139999	0.141682	0.045764	-0.202830
4	1.453735	3.516680	-2.793656	0.334216	-1.062819	1.209723
...	...	...	...	...	...	...

Fig: 6.1.1.Tabulated record of patients for 1min segments

CONFUSION MATRIX:

A confusion matrix provides a tabular representation of the true positive, true negative, false positive, and false negative counts, facilitating a detailed analysis of the detection system's performance.

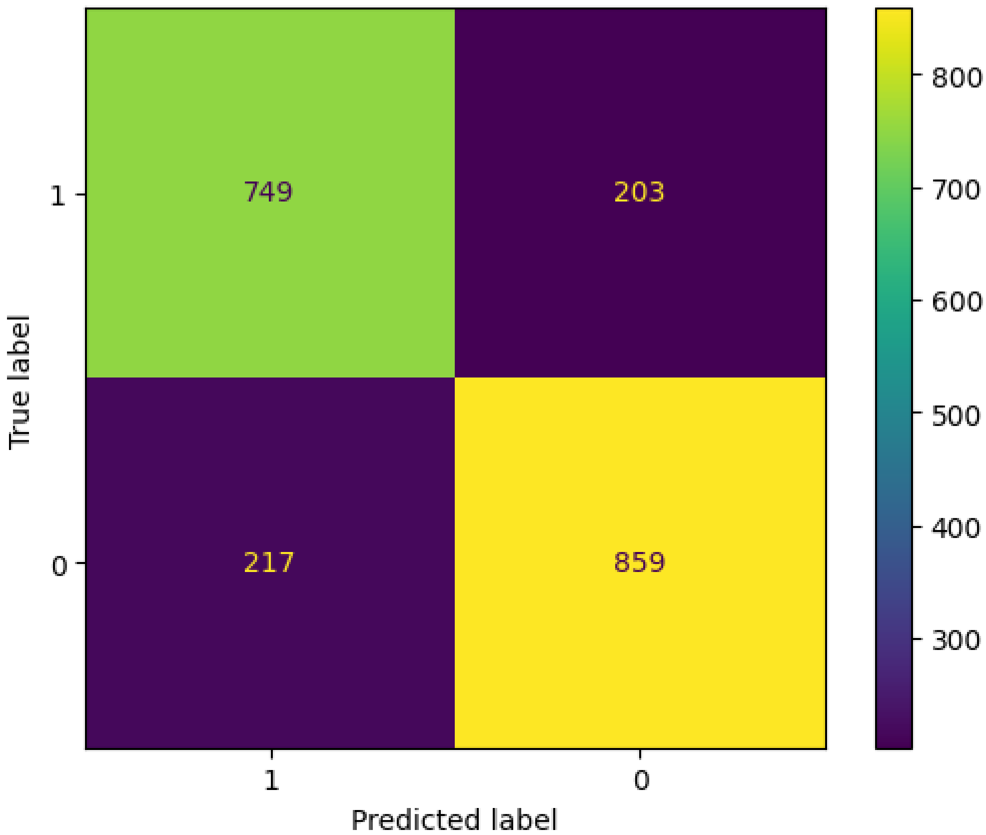

Fig: 6.1.2.Confusion Matrix

6.2. RESULTS AND DISCUSSION

A special possibility for health and sleep monitoring has been made possible by the development of smart wearable devices. Wearable technology, however, depends on precise and instantaneous monitoring algorithms. In order to diagnose sleep apnea from a single-lead ECG, we have presented a thorough comparison in this article between various machine learning and deep learning methods inside a single, cohesive framework. Deep learning models have been found to perform better than traditional machine learning methods. CNN-based models outperformed DRNNs in our application, which analyzed one-minute segments of ECG, among deep learning methods. The hybrid CNNDRNN architectures yielded the best detection performances; ZFNet-BiLSTM demonstrated the highest accuracy and specificity, while ZFNet-GRU demonstrated the best sensitivity. Hybrid deep neural networks are suggested for the identification of sleep apnea from electrocardiograms (ECG) based on the obtained results.

CHAPTER 7

CONCLUSION AND FUTURE WORK

6.1. CONCLUSION

Single-lead electrocardiography (ECG) has the potential to revolutionize the diagnosis and treatment of sleep apnea, a common sleep disease. Researchers have made significant progress in properly recognizing apneic occurrences and assessing their severity using data from a single ECG lead thanks to developments in signal processing algorithms and machine learning approaches. A few benefits of using single-lead ECG for sleep apnea detection are its affordability, ease of use, and potential for at-home monitoring. These techniques, which do not need laborious polysomnography setups, can offer important insights into a patient's sleep patterns and respiratory disruptions by taking advantage of the intrinsic variability in ECG signals during apneic occurrences.

6.2. FUTURE ENHANCEMENT

Detecting sleep apnea using single-lead ECG (Electrocardiogram) is a promising area for future enhancements. Implementing advanced signal processing techniques such as wavelet analysis, time-frequency analysis, and machine learning algorithms can help in better extraction of features associated with sleep apnea events from the ECG signal. Developing more sophisticated algorithms for feature extraction from single-lead ECG signals can enhance the accuracy of sleep apnea detection. This can involve extracting features related to heart rate variability, respiratory patterns, and presence of arrhythmias. Integrating single-lead ECG data with information from other sensors such as accelerometers or pulse oximeters can provide additional contextual data for more accurate detection of sleep apnea events

CHAPTER 8

APPENDICES

8.1 SDG GOALS

Good Health and Well-being (SDG 3):

This goal aims to ensure healthy lives and promote well-being for all ages. Detecting sleep apnea early using single lead ECG can contribute to this goal by enabling timely intervention and treatment, thus improving overall health outcomes.

Infrastructure, Industry, and Innovation (SDG 9):

Using cutting edge technologies, such as single lead ECG for sleep apnea detection, helps to advance sustainable industrialization, resilient infrastructure development, and innovation.

Partnerships for the Goals (SDG 17):

To successfully adopt technologies such as single lead ECG for sleep apnea detection, researchers, policymakers, and healthcare professionals must work together. In order to effectively address health concerns, partnerships can help with knowledge sharing, resource mobilization, and capacity building.

8.2 SCREENSHOTS

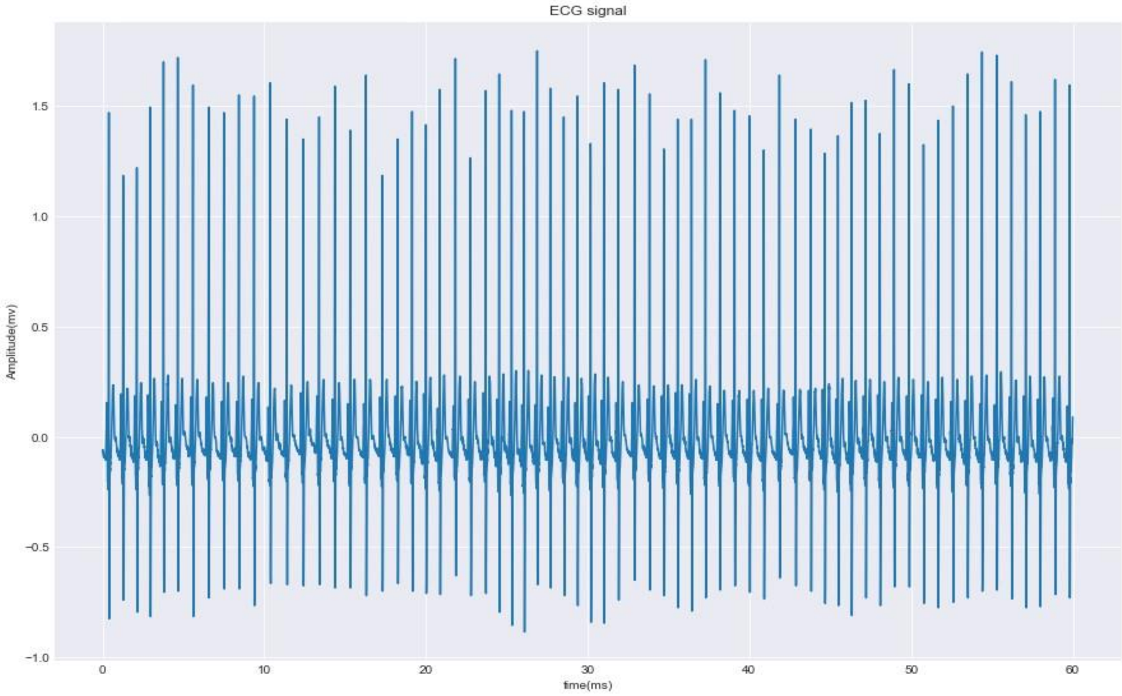

Fig:8.2.1.Screenshot of raw ECG data

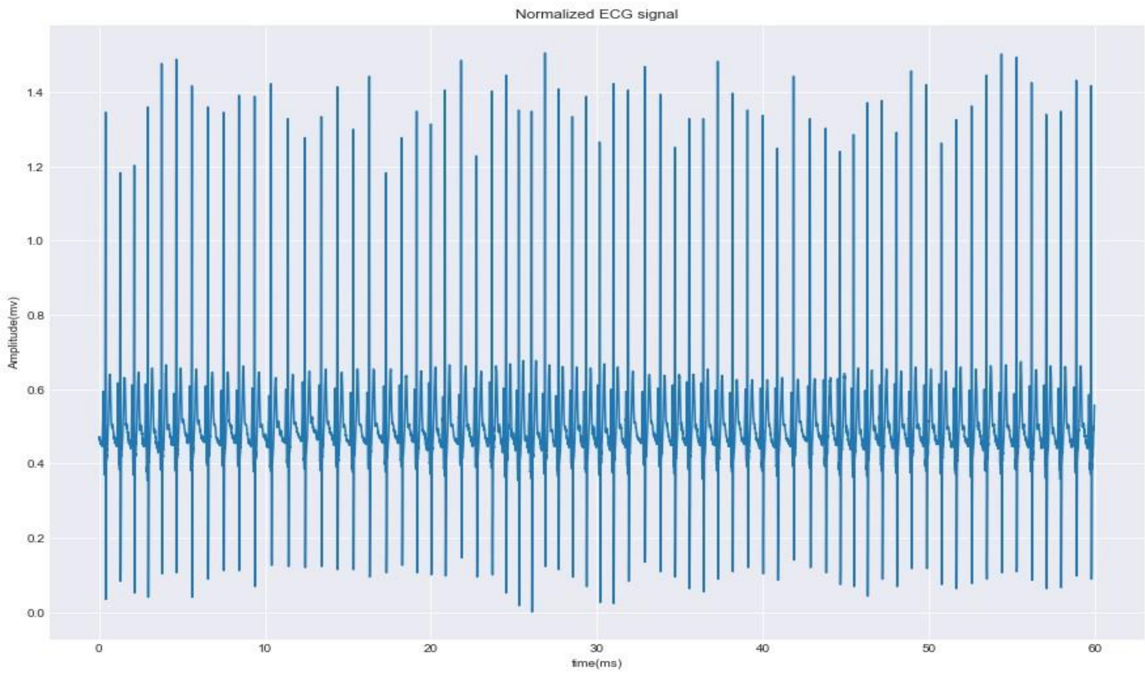

Fig:8.2.2.Screenshot of Normalized ECG

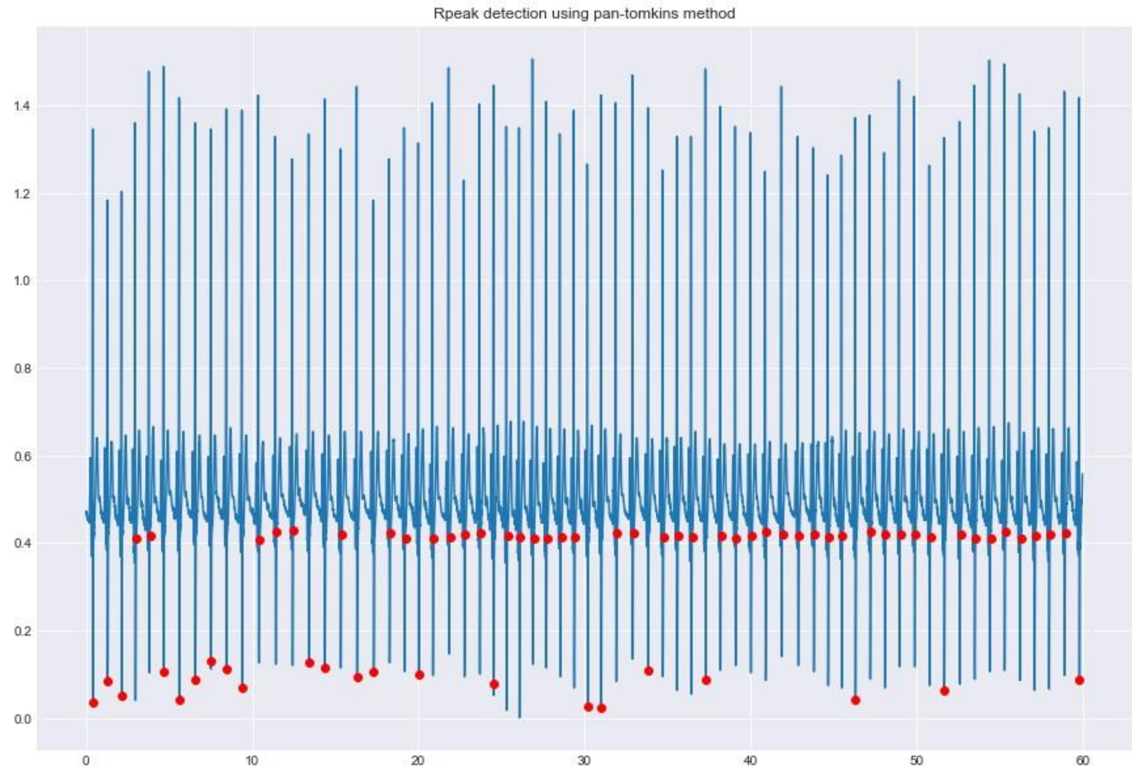

Fig:.8.2.3 Screenshot of R-peak detection using Pan Tomkims method

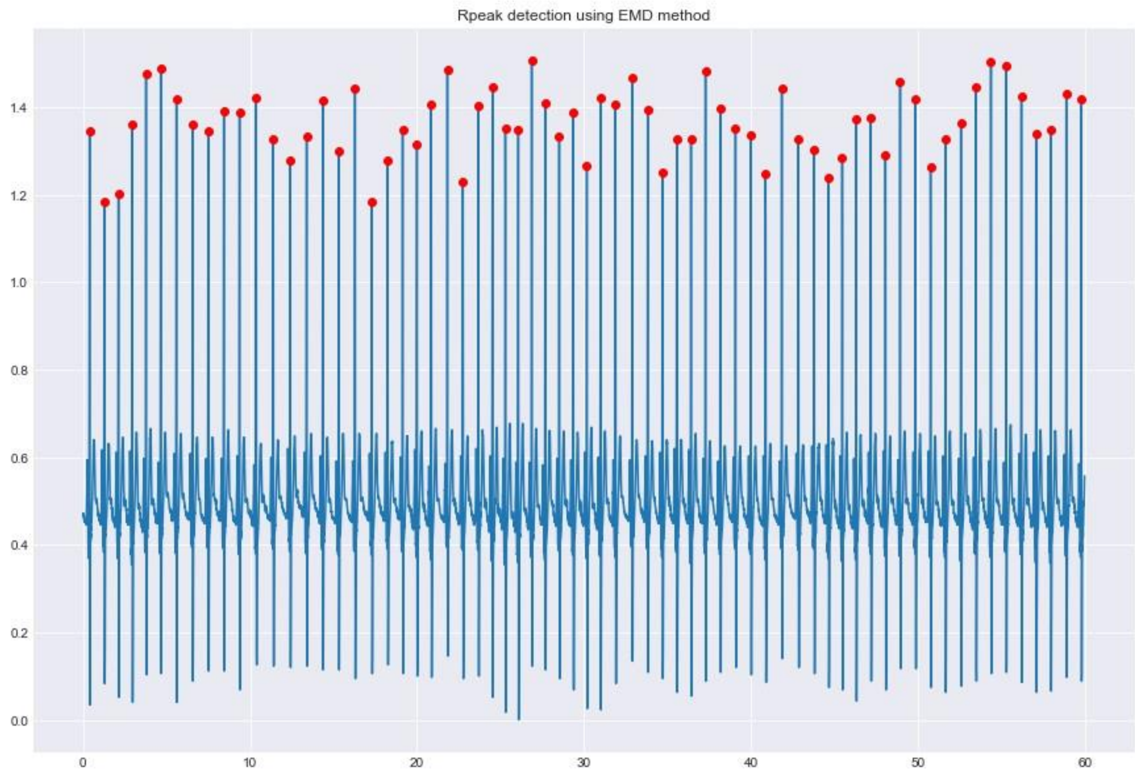

Fig :8.2.4. Screenshot of R-peak detection using EMD method

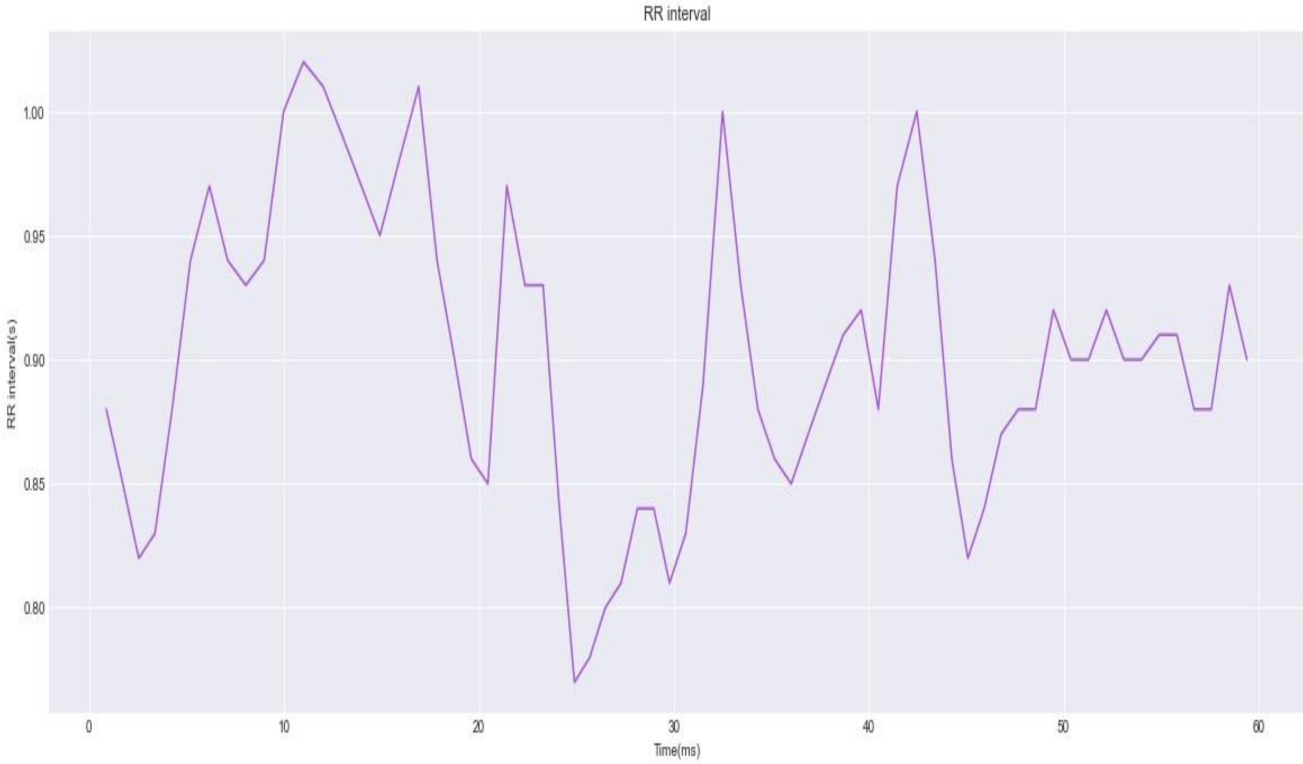

Fig:8.2.5.Screenshot of R-R Interval graph

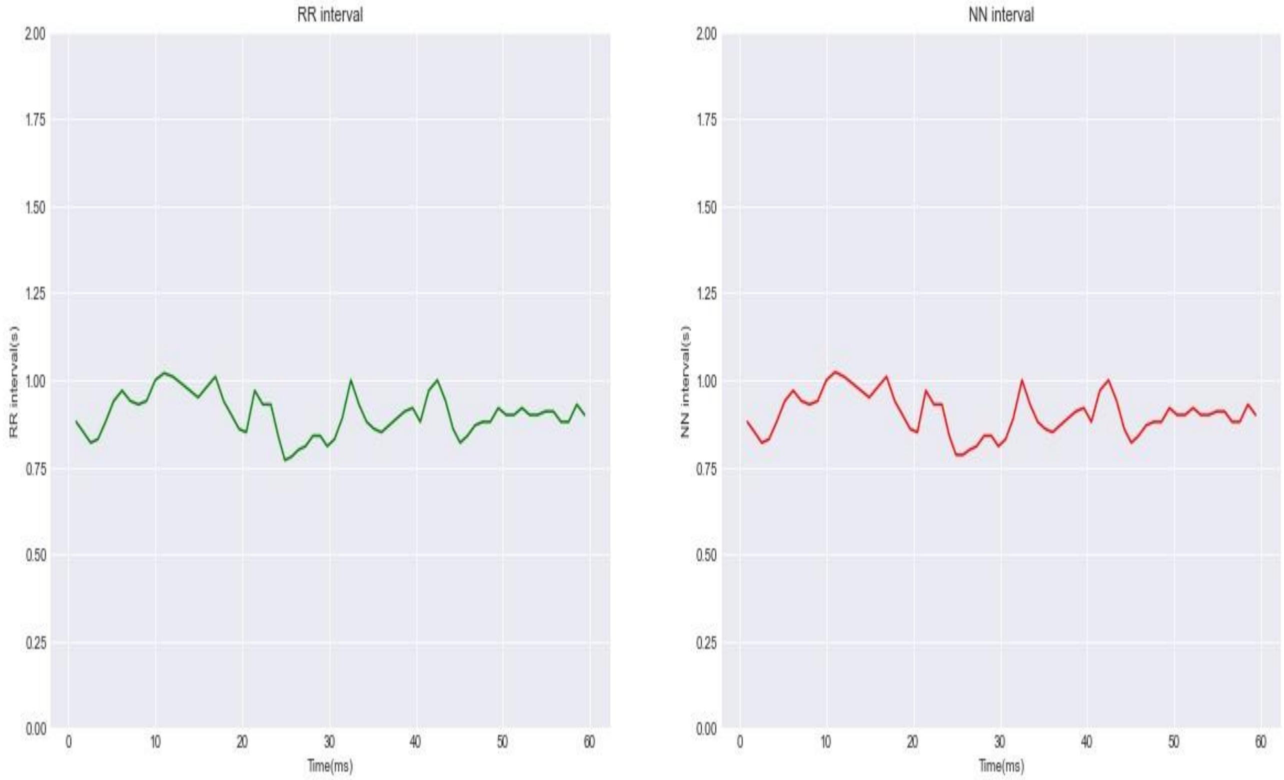

Fig :8.2.6.Screenshot of R-R Interval graph and N-N Inteerval graph

8.3.PLAGIARISM REPORT

SLEEP APNEA DETECTION USING SINGLE-LEAD ECG

ORIGINALITY REPORT

14	5%	3%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

CHAPTER 9

REFERENCES

[1] M. Bahrami and M. Forouzanfar, "Detection of sleep apnea from singlelead ECG: Comparison of deep learning algorithms," in Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA), Jun. 2021, pp. 1–5.

[2] N. Pombo, B. M. C. Silva, A. M. Pinho, and N. Garcia, "Classifier precision analysis for sleep apnea detection using ECG signals," IEEE Access, vol. 8, pp. 200477–200485, 2020.

[3] M. H. Kryger, T. Roth, and W. C. Dement, Principles and Practice of Sleep Medicine, 6th ed. Amsterdam, The Netherlands: Elsevier, 2017.

[4] S. A. Singh and S. Majumder, "A novel approach osa detection using single-lead ECG scalogram based on deep neural network," J. Mech. Med. Biol., vol. 19, no. 4, Jun. 2019, Art. no. 1950026.

[5] D. Dey, S. Chaudhuri, and S. Munshi, "Obstructive sleep apnoea detection using convolutional neural network based deep learning framework," Biomed. Eng. Lett., vol. 8, no. 1, pp. 95–100, Feb. 2018.

[6] J. Acquavella, R. Mehra, M. Bron, J. M.-H. Suomi, and G. P. Hess, "Prevalence of narcolepsy and other sleep disorders and frequency of diagnostic tests from 2013–2016 in insured patients actively seeking care," J. Clin. Sleep Med., vol. 16, no. 8, pp. 1255–1263, Aug. 2020.