Quantifying Severity of Symptoms in Patients with Parkinson's Disease Using Wearable Sensors and Machine Learning

By

Umar Khan

2018-NUST-MSCS-8 00000277753

Supervisor

Dr. Qaiser Riaz

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science (MS CS)

Department of Computing

School of Electrical Engineering and Computer Science (SEECS)

National University of Sciences and Technology (NUST)

Islamabad, Pakistan.

(July 2022)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled “Quantifying Severity of Symptoms in Patients with Parkinson's Disease Using Wearable Sensors and Machine Learning " written by UMAR KHAN, (Registration No 00000277753), of SEECS has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: 

Name of Advisor: Dr. Qaiser Riaz

Date: 22-Jul-2022

HOD/Associate Dean: _____

Date: _____

Signature (Dean/Principle): _____

Date: _____

Approval

It is certified that the contents and form of the thesis entitled "Quantifying Severity of Symptoms in Patients with Parkinson's Disease Using Wearable Sensors and Machine Learning" submitted by UMAR KHAN, have been found satisfactory for the requirement of the degree.

Advisor: Dr. Qaiser Riaz
Signature: _____
Date: 22-Jul-2022

Committee Member 1: Dr. Muhammad Zeeshan
Signature: _____
Date: 22-Jul-2022

Committee Member 2: Dr. Mehdi Hussain
Signature: _____
Date: 22-Jul-2022

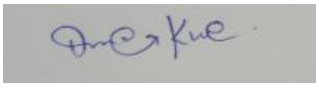
Dedication

I dedicate this thesis to the people in medical sciences, who are working on making lives easier for us all.

Certificate of Originality

I hereby declare that this submission titled "Quantifying Severity of Symptoms in Patients with Parkinson's Disease Using Wearable Sensors and Machine Learning" is my own work. To the best of my knowledge, it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation, and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Author's Name: Umar Khan

Signature: 

Acknowledgement

I’m in debt of my mother, my wife, and my supervisor, for their infinite support, for they pushed me when I needed it the most.

Table of Contents

List of Abbreviations and Symbols

Abbreviations

IMU	Inertial Measurement Unit
PD	Parkinson’s Disease
PS	Parkinson’s Syndrome
FOG	Freeze of Gait
EEG	Electroencephalogram
EMG	Electromyography
RF	Random Forest
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
GRU	Gated Recurrent Units

List of Tables

List of Figures

Abstract

Parkinson's disease has been an active area of research from a long time. Although no cure for it has been discovered yet, but certain medicines, surgical treatments, or other therapies can sometimes provide some relieve with some of the symptoms. Most prominent Parkinson's symptoms include Tremors, Dyskinesia, and Bradykinesia. Detection and prediction of these symptoms holds big significance as it can help in avoiding accidents, injuries etc., by generating alerts or generating cues to help reduce amplitude of anomaly. Our research focuses on detection of Parkinson's and on quantifying severity of Parkinson's symptoms using inertial data. For this purpose, we have used inertial dataset shared in the levodopa study**,** where inertial data is collected from three different body locations i.e., upper two limbs and waist, using 3 different sensors i.e., An IMU (GeneActiv), a smart watch (Pebble) and a smartphone (Samsung Galaxy S2). We present a 2-stage anomaly detection and classification pipeline where in the first stage (binary classification) an input signal segment is classified as either anomalous or as a normal signal. If the input signal is anomalous, then it passes through second stage (multi-label classification) where we categorize the type of anomalies present in the signal and thus quantify the severity of symptoms. We also present a performance comparison by using different preprocessing hyperparameters (segment sizes and segment labelling methodologies), of machine learning (Random Forest) vs deep learning (HARDenseRNN) based techniques. Overall, we present results on 3 different datasets (sourced from 3 different sensor [GeneActiv IMU, Pebble smartwatch, Samsung Smartphone]), where for each dataset we make 6 different preprocessing configurations by using 3 different segment sizes (50, 150, and 250 datapoints) and 2 different segment labeling methodologies (using mode as the segment label and identifying signal as anomalous if it contains any anomalous portion in it). Then we have 2 models for machine learning and 2 for deep learning, one for each of the two stages of pipeline, for each of which we present a detailed comparative performance analysis, for each of the six-preprocessing configuration and for all three sensors. For machine learning models we are using feature engineering on raw data and present top performing features as well, whereas we feed raw inertial data (magnitude of raw values) to the deep learning models.

Keywords: Anomaly Detection, Anomaly Classification, Parkinson's, Tremors, Dyskinesia, Bradykinesia.

Chapter 1

Introduction

Can our smartwatches and smartphones help create alerts and cues for patients with Parkinson's?

Our smartwatches and phones are equipped with sensor like accelerometer, which generates inertial data. This inertial data hides a lot of hidden potential within itself. It has been used in many different application areas, like terrain classification[1], human emotion classification, human soft biometrics prediction etc. Previous works has shown great potential for using inertial data to detect Parkinson's. In our study, we use inertial data from smartphone and smartwatches to detect and classify different Parkinson's symptoms.

1.1. Parkinson's Disease

Parkinson's is a neurological disorder, which is caused by nerve cell degeneration in the substantia nigra, a portion of the brain that controls movement, and has no discovered cure yet [2]. Even before cure, there isn't any X-Ray or Blood test that can be used to diagnose Parkinson's. People suffering from Parkinson, experiences several types of disorders with their daily motion-based activities[3]. Some of these anomalies includes Freeze of Gait (a person might freeze meanwhile walking or performing a daily routine task), Tremors, Dyskinesia (uncontrolled fastness of movement or too much uncontrolled movement), and Bradykinesia (uncontrolled slowness of movements). These anomalies can give rise to extremely dangerous scenarios, e.g., a patient suffering from Parkinson's may experience a bradykinesia episode while he was crossing the street, or a person dealing with a sharp object e.g., a knife while cooking, may experience a Dyskinesia attack, and may hurt themselves or others around them. If these anomalies are timely detected, we can reduce the damage in above scenarios by a big margin [4]. With timely detection, we can raise alarms, alerting the patient so that they may sit and secure themselves and even alert the people in surroundings of patient, and we can also generate cues which can reduce the amplitude of the anomaly[5].

1.2. Inertial Data

Inertial data has been a focus of many recent research studies, that are trying to unveil all the hidden potential from it[6]. Inertial motion sensors can be used in a variety of situations due to the universal presence of motion, vibration, and shock. They are used in many practical applications covering multiple domains ranging from robotics to navigation systems, from soft

biometric (gender, age, height) prediction to person re-identification, from emotion detection to medical research etc [7]. In addition to the ubiquitous presence of motion, vibrations, and shock, which are various sources of inertial data, the readily availability of Inertial Measurement Units (IMU's) also open doors to many new horizons. IMU's makes use of accelerometer, gyroscope and sometimes magnetometer, with all these sensors being readily available even in a common smartphone. For our study, we have only used data from IMU's with accelerometer, producing time series data with each point being a 3D data point i.e., 3D (x, y, and z axis) acceleration[8].

Figure 1.1: Inertial Sensors

1.3. General Methodology

From literature review, we found that sometimes raw inertial data is first pre-processed and then used for machine learning model training and inference and is sometime fed directly to a deep learning model in its raw form. Raw signal preprocessing includes steps like signal noise removal, signal segmentation, and feature computation. For noise removal, multiple studies [9][10][8][11] have used simple moving average with window size of 9. The cleaned signal is then decomposed into various segments using different techniques. Some studies [9] utilize peak and valley detection method for signal segmentation, where the part between two consecutive peaks/valleys is considered as a single segment. Some Other studies like [12] decomposes signal by dividing it into windows of certain length e.g., 1 second, with a stride of e.g., 20%. This means that each window will have an 80% overlap with the previous one.

These data segments now need to be labeled, which is sometimes a very straight forward task, but sometimes requires a bit more thoughtful approach. Cases where the output label stays the same for the whole signal e.g., gender of a person, as in [9], are straight forward, as the same label is applied to every signal segment. Cases where the label for the whole signal is not same, e.g., tremor occurrence within gait of patient suffering from Parkinson's, requires more sophisticated segment labelling methods. With non-homogeneous segments, methods like

using the mode as the overall segment label are commonly used, as in [11], or treating the segment as an anomalous one if even a little portion of it was anomalous, as in [12].

This segmented data is then used for feature computation. The feature list can contain feature from time, frequency, or wavelet domain etc., as in [8]. These features are computed for each segment and the resultant dataset is then used for model training and inference [13]. This although has been known to improve model performance in terms of accuracy, but at the same time it increases the processing time. This is where feeding raw data into a deep learning model has shown its edge over machine learning model, as shown in [14]. With deep learning model, the raw data is first segmented and labeled accordingly, and is then directly used for model training and inference.

1.4. Past Work and Our Main Contributions

Although previous works has shown great potential in using Inertial data to detect Parkinson's disease in subjects, classification of anomalies in patients who are already diagnosed with Parkinson's, is still an open area of research [4], [15], [16]. In our presented study, we have used inertial data to detect and classify anomalous events in patients suffering from Parkinson's disease. We have also explored different Pre-Processing dimensions (data segment size, data segment labeling methodology, most efficient features to use) to find the most efficient ones and have also presented a comparative performance analysis of using machine learning models vs using deep learning models for our purpose [17], [18]. Overall, following are our main contributions

- Multiple preprocessing hypermeters are evaluated, and optimal ones are presented.
- Optimal feature set for Parkinson's anomaly detection and classification using machine learning models.
- Parkinson's anomaly detection and classification using only a single feature (magnitude of raw inertial values) with our newly proposed deep learning model.
- Comparative analysis of machine learning vs deep learning models.

1.5. Problem Statement

Anomalous event detection and anomaly classification in patients suffering from Parkinson's is an open area of research [19]. Where majority of previous work, is focused on Parkinson's detection, we present an approach towards detection and classification of anomalous events from inertial data of patients suffering with Parkinson's. By identifying the various anomalies present in a signal, we quantify the severity of the disease. In our study, we explore inertial

data from different sensors and from different body locations and present a two-stage pipeline for anomaly detection and classification. We also explore various pre-processing parameters like data segment sizes, and data segment labeling methodologies. We also present optimal feature set that gives best performance results with machine learning model (Random Forest). We then present, use of deep learning models (HARDenseRNN) with raw inertial data, using different data segment sizes, and data segment labeling methodologies, for anomaly detection and classification [20]. We then present a comparative analysis of machine learning models vs deep learning models in terms of accuracy and inference speeds.

1.6. Research Hypothesis

Our Research hypothesis is as follows:

Deep learning models (like HARDenseRNN) can outperform traditional machine learning models (like Random Forest) for anomalous event (Tremors, Dyskinesia, and Bradykinesia) detection and classification, using relatively bigger segment sizes, and treating segment as anomalous if it contains any anomaly, from inertial data of patients suffering from Parkinson's disease.

1.7. Research Questions

Our research problem is aimed towards finding optimal pipeline for anomalous event detection and classification (identifying various anomalies present in a signal – thus quantifying the severity of symptoms) from inertial data of patients suffering from Parkinson's disease. To support our above stated hypothesis and present an answer towards our research problem, we formulated the following research questions

- **Research Question 1:** What are the optimal pre-processing parameters for anomalous event detection and classification in inertial data of patients suffering from Parkinson's?

  Objective: The objective of this question is to find the optimal value of Pre-Processing parameters i.e., the window size for signal segmentation, and segment labeling methods.

- **Research Question 2:** What are the optimal features, to be used with machine learning models, for anomalous event detection and classification in inertial data of patients suffering from Parkinson's?

Objective: The objective of this question is to find the optimal features (top 10) from time, frequency, and wavelet domain, so that they may be used with machine learning models.

- **Research Question 3:** Performance comparison of deep learning and machine learning based models.

  Objective: The objective of this question is to perform a comparative analysis of the two techniques (machine learning vs deep learning models), using various pre-processing parameters (as discussed in research question 1). The answer here is meant to contribute towards an optimal model, which can be used in real life applications.

1.8. Overview of Proposed Methodology

In our study, we present a 2-stage pipeline for anomaly detection and classification. The first stage, the input signal segment goes through a binary classification model, which classifies the signal segment as anomalous or normal. If the signal segment is classified as an anomalous one, then it passes through a multi-label classification model in second stage, using which we identify the type of anomaly present in the signal. Using the identified anomalies, we then quantify the severity of symptoms present in a signal.

we present finding in two dimensions i.e., machine learning models (Random Forest) and deep learning models (HARDenseRNN). For both dimensions, the raw data is first decomposed into smaller segments. For this purpose, we have explored 3 different window sizes (50, 150, and 250 data points) with a stride of 20%, and 2 different segment labeling methodologies i.e., using mode as segment label, or treating segment as anomalous if it contains any anomalous portion.

For machine learning models, this raw data is used for feature computation. A total of 150 features are extracted including statistical features and features from time, frequency, and wavelet domain. The extracted feature set is then handled for class imbalance, using Random Under Sampling. The resultant dataset is then split into train and test datasets with a 70%-30% division and is then normalized using zscore. Model evaluation is done using k-fold cross validation with k=10. Top 10 optimal features are also presented split based feature ranking.

For deep learning models, the decomposed raw data is directly used for model training and evaluation. Here, only a single raw feature is used i.e., the magnitude of the input 3D vector. This feature is then normalized using zscore, and then reshaped into a single vector. So, a 250

data points of a single segment are converted into a 1D vector of size 250. This vector is then labeled, and the two above mentioned labeling methodologies are explored. This data is then divided into train and test dataset which are then used for model training and evaluation respectively.

In the end, we present optimal pre-processing parameter values, and a comparative analysis of machine learning and deep learning models.

1.9. Structure of the Thesis

The rest of this thesis is organized as follows:

- Chapter 1: Research problem is introduced in the beginning of this section followed by main contributions of this work.
- Chapter 2: Presents detailed literature review. Analysis of different modalities used to detect Parkinson's Disease and the anomalies that the patient suffers.
- Chapter 3: Presents the methodology of the proposed system that detects and classifies anomalous events in Parkinson's patients. The methodology contains summary of data preparation, preprocessing configurations, feature computation, model (Random Forest and HARDenseRNN) training and evaluation.
- Chapter 4: Contains the results of this research work. In this section we present performance of 72 different models, along with their performance wise comparative analysis. We comment on the best-found preprocessing configuration, top 10 features for Random Forest, and present comparative analysis of machine learning (Random Forest) vs deep learning model (HARDenseRNN).
- Chapter 5: Presents the discussion of results, comparison with related techniques and limitations of proposed approach.
- Chapter 6: Presents the conclusion of the research and possible dimensions of future work.

1.10. Summary of Chapter

The main object of chapter 1 was to share the motivation behind this research. Based on existing methodologies being used while working with inertial data, we defined our research hypothesis and identified research questions as well. The research questions play an important role to illustrate the requirements of the solution regarding the contributions of the research.

Literature Review

2.1. Chapter Overview

In this chapter, we discuss existing efforts made towards the detection of Parkinson's Disease (PD) and classification of its symptoms. We also explore the role that different modality i.e., the inertial data, EEG data, EMG data, Speech data and Vision data, play in automating the detection of PD and measuring the severity of its symptoms. Later in this thesis, we use this related work to support the proposed methodology for anomalous event detection and quantifying severity of symptoms in patients suffering from Parkinson's.

2.2. Background

Inertial data provides information about the motion of subject under consideration, in a time series format, collected through sensors that are commonly referred as Inertial Measurement Units (IMU's). Merging machine and deep learning with the inertial data has become a contributor for major breakthroughs in multiple domains [21]. In some cases, inertial data even surpasses a human observer, e.g., a human observer might fail to distinguish between a person suffering from early-stage Parkinson's and a healthy old person, as the gait of both suffer from almost the same anomalies [2].

Over the past decade multiple studies have shown, that a lot of hidden information is encoded in a subject's kinematic information. In the following literature review, we discuss some very interesting dimensions and use cases of inertial data, EEG data, EMG data, Speech data and Vision data along with machine learning and deep learning techniques used to make such advancements possible.

2.3. Anomalous Event (Parkinson's Disorders) Detection, Prediction and Classification Using Inertial Data

Parkinson's Disease (PD) and Parkinson's Syndrome (PS) have been termed as common neurodegenerative disorders that most commonly appear in the elderly. The research in [22] focuses on detecting Bradykinesia which is a common motor symptom that occurs in both PD and PS. The main work in the paper is the collection of upper limb movement signals of the patients suffering from PD and PS. This data is then passed through the feature extraction phase and fed to a multilayer perceptron (MLP) to detect Bradykinesia. The detection accuracy in this paper is 85% for both PS and PD.

Freezing of the gait (FOG) arises in PD as the disease progresses towards its final stages. It severely effects the gait and becomes a cause of falling in the patients with PD/PS. If these gait phases are detected, they can lead towards preventing the risk of falling in patients PD. [23] utilize the inertial sensors to detect gait phases based on feature like, acceleration, position of the foot, rate of turn and velocity. They introduce a GaitScore, which is an evaluation measure that separates a normal gait from the motion phases affected by FOG. This helps in detecting FOG. They have been able to detect FOG with a sensitivity of 97% and specificity of 87%.

Remote analysis of the axial impairments in patients with PD/PS has been very limited, however, it is crucial to detect these impairments in PD. Wearable sensors are a promising technology to aid this cause. [24] focuses on utilizing wearable inertial sensors and integrating the data with machine learning to compute Posture Instability/Gait Difficulty score (PIGD score) in PD patients. The sensors were placed under the thighs, the collected data was passed through the feature extraction phase and then fed to Support Vector Machine (SVM) to predict the PIGD score. The regression model yeilds RMSE of 0.27 in ON state of patients with FOG, and a RMSE of 0.22 in the OFF state. Similarly in patients without FOG for ON and OFF states the results were RMSE = 0.19 and RMSE = 0.21, respectively.

To verify the validity of a wearable inertial sensors, so that they can be used for home-based monitoring of PD patients is crucial. [25] claims, that a 3D motion capture system can be used as a validity measure, if we plan to perform sensor-based gait analysis in PD patients. To analyze the performance, they compare the results of mobile-based inertial sensors against the gold standard movement system. The results showed that both sensors work almost same except for a few cases, where movement system outperforms. The overall results show that the data collected from wearable inertial sensors is valid for capturing the gait parameters.

[26] has worked on detecting the signs of hand tremor in PD patients. To perform the experiment, they utilize the wearable inertial sensors. The sensors are applied on the wrists of patients with PD and patients without PD. The data of the inertial sensors is passed through a 9-layer CNN model, to predict the hand tremor. The results have shown that the method is highly effective with an accuracy of 97.32%.

[27] aims to study the effect that levodopa has on the phase coordination index (PCI) and the gait asymmetry of patients with PD. They measure the motor symptom and the gait parameters using the inertial sensors and then compute the correlation between their severity. To measure

the severity of motor symptoms Unified Parkinson's Disease Rating Scale Part III has been used. They have computed the posture instability and gait difficult score (PIGD) to measure severity of the gait parameters. The study showed that patients that took levodopa had an improved PCI.

Identifying the stride borders is a crucial step towards gait analysis pipelines that utilize the mobile inertial sensors. [28] focuses on stride segmentation by evaluating the existing models on real time, free-living gait. They have utilized a comprehensive dataset that includes the labeled stride of 28 PD patients. For stride segmentation, Hidden Markov Model (HMM) has been used. The proposed approach has provided an F1 score of 92.1% which has outperformed the existing approaches.

[29] have worked on finding out the validity of the wearable inertial sensors, which are used for monitoring gait impairment in both supervised and unsupervised situations. For this, they developed a solution that analyses two phase movement using 3D full body kinematics, recorded through inertial sensors and a smartphone. Later the validity is computed against the optoelectronic criterion amongst the population of PD patients. The results show that the wearable sensors provide valid data.

[12] explores potential of using inertial data from human gait, to detect and predict freeze of gait anomaly in patients suffering from Parkinson, using different merging thresholds for anomalous episodes. Human gait data was collected from 11 males suffering with Parkinson's, who experience FOG episode at least once a week. The collected data was decomposed into segments of 1 second with a stride of 0.2 seconds, i.e., 80% overlap between consecutive segments. For detection the target class contained all segments that had any anomalous (FOG) event inside them. For prediction the target class contained all segments that were in the window of 2 seconds before any FOG appearance.

From each segment, a total of 850 features were computed, from which top 10 (containing both frequency and wavelet domain features) were selected using Relief-R feature ranking. These features were then fed into a decision tree ensemble of 100 trees with max_split = 5. For training, random under sampling boosting (RUSBoost) was used. Model was validated using leave-one-freezer-out cross validation, where data from a single subject (who experienced FOG) was left in the test set while all other data was used for training. This model, with the same thresholds, was trained on data with different merging threshold consideration.

It was observed that the best detection results were achieved using a merging threshold of 2 seconds, whereas the best prediction results were achieved using a merging threshold of 0 seconds (no merging). For window-level detection, very small difference in detection models is observed, across different thresholds, whereas only a slight difference is observed in prediction models.

[30] proposes a method to differentiate among healthy individuals and individuals suffering from Parkinson's (in the initial stages of disease), enabling early disease detection, using inertial data from IMU's.

The data was collected from 27 participants by placing inertial sensors at the back of the hand and on the forearm. These features were used to train four different classifiers i.e., KNN (k = 3), Random Forest (120 trees), Naïve Bayes (with normal kernel), and Support Vector Machines (with polynomial kernel). SVM classifier fed with 10% and 20% of all features achieved the highest accuracy and sensitivity, whereas highest specificity and precision was achieved from Random Forest classifier when fed with 20% of all features.

These classifiers were evaluated with two different PD groups. The first group had all individuals with Parkinson's (HY = 1 or 2) vs healthy individuals, whereas in the second group only those patients were included that had HY = 1, against healthy individuals. Better sensitivity and accuracy were achieved for the second group having Parkinson's subjects with HY=1.

[2] proposes use of neural networks to detect Parkinson's disease in early stages, using IMU's. They show the potential of using neural networks for three cases: detection of advance stage Parkinson's disease, classifying different stages of Parkinson's, and differentiating patients with early-stage Parkinson's disease from normal elderly participants (normal elderly participant shows gait issues due to disorder other than Parkinson's like joint osteoarthritis or sarcopenia).

Data was collected from 32 patients suffering from Parkinson's and from 16 age and sex matched healthy elderly control subjects, using 5 APDM OPAL system wearable IMU's, attached on participants waist (1 IMU), shanks (2 IMU's) and lower arms (2 IMU's). 6D data was collected. It was observed that patients with different stages of Parkinson's tended to have varied IMU responses, although from it was hard to differentiate between healthy elderly participants and participants with Parkinson's just from the exploratory data analysis.

The collected data was segmented according to Mid-Swing events, which were then normalized to 100 points for each gait cycle. Finally, the data from all sensors were collected into a single array to form the input. This data was passed through a selection layer, in which a selection matrix was used to pick appropriate data for the two neural networks. For the first neural network, data from the participant affected side was selected, as advanced PD cases exhibits anomalies especially on the affected side. For the second neural network, angular velocities from both arms and waist were selected, as at an early stage, the PD usually affects the unilateral side, especially the arm movements. In addition, early stages of PD tend to have more unstable trunk movements than healthy individuals.

The selected data was used to train and evaluate the neural network. The proposed methodology consists of two sub-networks. The first neural network classifies the signal as an advanced staged PD or Not an advanced staged PD. If the signal is classified as not an advanced staged PD, then it is passed to the second neural network which further classifies the signal as an early-stage PD or a non-PD signal. The model architecture consists of input layer, three convolutional layers, flatten and then 3 fully connected layers, and then the output layer. ReLU activation is used for convolutional and the fully connected layers, and a sigmoid activation function was used in the output layer for binary classification. Although both models have the same architecture, but they have different parameter as they were trained individually. Promising results from both models are presented and shows potential of neural networks applications.

2.4. Anomalous Event (Parkinson's Disorders) Detection, Prediction and Classification Using EEG Data

The current detection and monitoring measures for PD require physical examination and laborious experiments by medical experts which is a slow and sometimes fail to capture the disease at its prodromal stage. To overcome this, [31] have proposed a deep learning model that classifies the rest-state EEG (electroencephalogram recording of brain activity) of patients with PD and healthy audience. The model is a CRNN, which is a combination of CNN and RNN with GRUs. The 1D CNN is used for the extraction of spatio-temporal feature from the EEG data. These features are then forwarded to the RNN, that makes the predictions. The model successfully classified PD patients from the healthy patients with an accuracy of 99.2%, precision of 98.9% and recall of 99.4%.

The common physical characteristics of PD are motor symptoms like slow movement and tremors. Whereas the non-motor symptoms include anxiety, depression, and insomnia. [32] has

developed a deep learning-based approach to diagnose a patient with PD. They utilize the readings of 3 spatial channels of resting state EEG as the dataset. The data is trained using the Artificial Neural Net (ANN). The model was successfully able to screen the patients with PD with an accuracy of 98%, sensitivity of 97% and specificity of 100%.

Patients with PD, sometimes suffer a disturbance in their regular gait and become unable to take a step forward, this difficulty is medically termed as Freeze of Gait (FoG). Recent research points out that, during the occurrence of FoG, the basal ganglia region of the brain has an increase in the beta frequency. However, the synchronization between different brain regions and frequency bands in PD patients are rare. [33] utilize the non-linear dynamics, for the analysis of brain EEG generated brain waves of three categories of PD patients, based on the severity of FoG. The EEG synchronization amplitude is higher amongst different brain regions when the FoG severity is higher.

The results are consistent across all the frequency bands (alpha, beta, gamma, theta), which means that the severity of FoG is independent of the motor tasks and can be analyzed based on stronger EEG networks. This approach can be used to attain further insights in PD and how EEG network connections can play a role.

[34] propose that non-invasive methods like the EEG recording of the patients with PD can serve as an alternative biomarker for the diagnosis of the PD. To serve this purpose, they have developed a deep learning model to predict the PD diagnosis. For training, the EEG data of 16 healthy controls (HC) and 15 PD patients is collected. EEG data is then passed through Gabor transform to create spectrograms; they are then passed through a 2D convolution network (2D-CNN) for predictions. Using 10-fold cross validation the model was successfully able to classify among three classes (healthy, PD, and patients without medication). The total classification accuracy was 99.46%.

The neural behavioral and physiological changes in the early stages of PD are very subtle, hence hard to be caught in a diagnosis. [35] utilize the EEG recordings, to get insights about the brain activities which are helpful in the diagnosis of PD. Rather than using traditional machine learning they have combined soothed pseudo-Wigner Vile Distribution (SPWVD) with a 2D convolutional Neural Network (2D-CNN). The first step is to pass the EEG readings through SPWVD to attain time-frequency representation. These are 2D plots, that are passed into the CNN for training, two datasets have been used for the purpose, both providing PD prediction accuracies of 100% and 97% respectively.

2.5. Anomalous Event (Parkinson's Disorders) Detection, Prediction and Classification Using EMG Data

PD can be detected based on the assessment of the neuromuscular movement, which can be attained through surface electromyography (EMG – measures muscle response to a nerve stimulation), as it provides reading for the movements of the arms and wrists. [36] focuses on using pretrained deep transfer learning and conventional machine learning models to generate an automated PD detection system using the sEMG readings as the data. Initially they have created feature vectors by stacking the features extracted by AlexNet, VGG-f and CaffeNet. These features helped in tackling the overfitting situation and made the model robust against the added noise.

The size of the feature vector was further reduced using ROC and signal to noise methods. Finally, to predict PD, this pre-processed data is passed to Support Vector Machine (SVM) with radial basis function (RBF) kernel. The model has been successful with a PD prediction accuracy of 99%.

[37] targets on detecting the motor fluctuations in patients with PD by using surface EMG readings with 24-hour accelerometry (ACC) obtained through a wearable smart device. They measured 7 PD patients that took medication, once, and twice (before and after dDBS reprogramming) measured 9 patients with directional deep brain stimulation (dDBS). The comparison of EMG and ACC was made against the clinical rating score and the home diaries of the patients. The results showed that both EMG and ACC were considerably correlated with the condition of the PD patient as measured through motor score of Unified Parkinson's Disease Rating Scale. And this correlation was decreased after the dDBS reprogramming. This monitored data when compared with the home diaries showed 91% concordance with the tremors, 76% with rigidity and 74% with dyskinesia. The results show that a wearable device providing EMG and ACC readings is a robust way of monitoring patients by providing dynamic information.

EMG is a cheap and effective alternative for the diagnosis of PD in patients, however, the process cannot be carried out manually. [38] utilize the upper limb movements attained via EMG readings to automate the process of PD detection and classification. The dataset was created by collecting the EMG readings of the flexor carpi radialis and the biceps brachii muscle of 15 different PD patients and 10 healthy controls (HC). The raw data was passed through preprocessing phase to extract time and frequency features. On this data, a multi-class

SVM is trained that classifies amongst 4 different cases of PD (normal, early, moderate, severe PD). The model has performed successfully with prediction accuracies of 90%, 91.7%, 95% and 96.6%.

The fluctuations in PD patients worsens at advanced stages and they become abrupt between the OFF and ON events. Currently, the motor symptoms are measured based on the UPDRS, however, it is long and tedious method. [39] focuses on the development of an automated, unobtrusive method for measuring the motor symptoms in PD patients. They have collected surface EMG data from the wrists of 45 PD patients. The hypothesis was based on collecting data when the patient taps, both with and without using the dopaminergic medicine. The final aim was to predict the UPDRS score by utilizing regression-based machine learning models. The random forest regressor outperformed other models by providing a correlation of 0.739 between the actual and the predicted score.

The EMG readings burst during the period of muscle relaxation and contraction; these are helpful for detecting neuromuscular disorders such as the PD. [40] proposed a new approach, they segment the EEG signal during the time of muscle contraction and relaxation and note the onset and offset of the EEG burst. These values are then preprocessed through Discrete Wavelet Transform (DWT) for the extraction of features. These time and frequency features are used to train Hidden Markov Model (HMM) to classify between contraction and relaxation. These segmentations of EMG are performed on the patients with PD and Healthy subjects. The results were based on the ECOTECH database where HMM with 2 states combined with 3 Gaussians and Log Wavelet Decomposition with Energy providing the best accuracy of 100%.

2.6. Anomalous Event (Parkinson's Disorders) Detection, Prediction and Classification Using Speech Data

The early stages of PD include in the change of voice of the patients, the detection of these changes can lead towards the diagnosis of the disease at an early stage before it becomes severe and cause physical impairments. [41] have worked on exploring the static and dynamic speech features that can aid in PD detection. A comparative analysis between the speech transitions of a healthy person and a PD patient shows many differences. Inspired by the analysis, [41] has proposed a bidirectional LSTM so it can capture the dynamics of time-series features to detect PD from the speech data. The energy-based transitions in the speech enable in identifying the speech and non-speech features, that serve as the dynamic features. 10-fold cross validation

and dataset split without overlapping samples were the two approaches used to test the hypothesis. The model was able to achieve an F-1 score of 80.70%.

Dysarthria is one amongst the first symptom of PD, [42] propose an approach to diagnose this symptom using deep learning, because of the advantage that raw data can be fed directly to them with any feature extraction. The authors have used the voice notes of 50 patients suffering from PD and 50 healthy subjects as the data for training. The deep learning model selected is an ensemble of CNNs, the base CNN is trained through a multiple fine-tuning approach. The fine-tuning approach closes the gap between the input pre-trained and the target predictions. Both training and testing have been performed individually on each vowel. The testing has been done using 10-fold cross validation. The interesting fact is that the model was able to distinguish between the vowels of PD patients and those of the healthy people. The best accuracy amongst all the vowels was 99%, 93.3% specificity and 86.2% sensitivity.

Speech has been proved helpful for the detection of PD, [43] have performed the task of detecting PD by utilizing a combination of Adaptive Crow Search Algorithm (ACSA) with deep learning that is utilized for the purpose of optimal feature selection naming it CROWD. Stack sparse autoencoder deep neural net has been used as the deep learning network and a public dataset for PD has been used. The initial step in the research was to clean the data and handle the missing values. Post cleaning, this data is passed to ACSA for finding the scrunched feature vector. This feature vector is passed through the stack sparse autoencoder and finally a compressed feature vector is generated. This approach is compared against multiple state of the art algorithms for PD classification, and it outstands with a prediction accuracy of 96%.

[44] have used the biomedical signal processing (MSP) to detect PD in patients by monitoring the voice disorder. Convolutional Neural Nets (CNN) and Artificial Neural Nets (ANN) are employed to classify between the healthy subjects and patients with PD based on signal processing of their voice features. They have used to publicly available UCI machine learning repository datasets to accomplish the task. Dataset's come with 22 and 45 acoustic features respectively. The approach was a success with CNN being the model providing classification accuracy of 93% on the first dataset with 22 acoustic features.

2.7. Anomalous Event (Parkinson's Disorders) Detection, Prediction and Classification Using Vision Data

Patients suffering from PD have difficulties as they shift their posture from siting to standing state. This impairment holds a risk of falling in the patients and is a common cause for the

occurrence of dementia. [45] use vision based markerless pose estimation data to estimate the Movement Disorder Society Unified PD Rating Scale (MDS-UPDRS) ratings for section 3.9 which means arising from hair. 447 videos collected via smart devices under medically monitored situations have been analyzed under this research. Every video contains an action from section 3.9 and is labeled by the MDS-UPDRS score.

For pose estimation key points from each frame, OpenPose a deep learning library was used. The result was timeseries signals for each key point. Features that captured the movement information were extracted from these signals. These features were used for training a random forest classifier. In 79% of the videos, using leave-one-out cross validation the model was able to classify exactly as done by the UPDRS. The classification of PD was done with a sensitivity of 62.8% and specificity of 90.3%.

Gait impairment is one of the most common symptoms in patients suffering from PD, this impairment is classified under section 3.10 in MDS-UPDRS. [46] have estimated the severity in the gait of PD patients using vision-based data. The system allows to catch error-prone ratings and provide early ratings when a medical physician is not available. Total 729 videos have been used for training the system to detect the severity of gait impairment. 6 features were extracted from the time-series signals of the movement key points that had been extracted from the raw video data. These 6 features were used as training data in a random forest classifier which was evaluated used 10-fold cross validation. There was a high correlation amongst the classifications made by the model and the UPDRS labels provided by the clinicians. The classifier attained a balanced accuracy of 50%. In one of the cases 95% of the UPDRS ratings were a match of the markings made by the clinician.

[47] To avoid the misdiagnosis or late diagnosis of PD this research focuses on presenting a multimodal non-invasive approach that collects the patterns of gait and eye fixation modality through the video descriptors to estimate the severity of PD and allow the early diagnosis of PD. This approach can capture any abnormalities in the gait and eye movements. The video data of these movements is preprocessed, the kinematic features are extracted using dense optical flow and the deep features are extraction using CNN. Then a temporal mean covariance of these features is computed and passed to random forest classifier for training and getting classification predictions. 13 healthy subjects and 13 patients with PD were part of the experiment, the results gave an average accuracy of 100%.

2.8. Comparative Analysis

This chapter has been aimed towards exploring various methodologies and pipelines being used when working with inertial data for different applications. A brief overview can be seen below in Figure 2.1 and in Table 2.1.

Table 2.1: Summarized Comparative Analysis of Research for PD Using Different Modalities

Research for Parkinson's Disease Using Inertial Sensor Data			
Paper Name	**Objective**	**Modelling**	**Results**
[22]	Detection of Bradykinesia	Multi-Layer Perceptron	85% Accuracy
[23]	Detection of Freeze of Gait	Gait Phase Detection Algorithm	97% Sensitivity, 87% Specificity
[24]	Analysis of Axial Impairments	Support Vector Machines	0.27 RMSE
[26]	Hand Tremor Detcetion	Convolutional Neural Net	97.32% Accuracy
[12]	Detection of Freeze of Gait	Random Under Sample Boosting	82.4 Sensitivity, 88.3 Specificity
[30]	Early Stage PD Detection	Support Vector Machines	80% Accuracy
[2]	Early Stage PD Detection	Artificial Neural Network	92.72% Accuracy

Research for Parkinson's Disease Using EEG Data			
Paper Name	**Objective**	**Modelling**	**Results**
[31]	Classification between Healthy subject and PD Patient	CRNN (CNN+RNN)	99.2% Accuracy
[32]	Detection of Patients Suffering PD	Aritificial Neural Net (ANN)	100% Accuracy
[34]	Detection of Patients Suffering PD	Gabor Transform & 2D CNN	99.46% Accuracy
[35]	Detection of Patients Suffering PD	Soothed Pseudo-Wigner Vile Distribution & 2D-CNN	97% Accuracy

Research for Parkinson's Disease Using EMG Data			
Paper Name	**Objective**	**Modelling**	**Results**
[36]	Automated Detection of PD	Support Vector Machine (RBF-Kernel)	99% Accuracy
[38]	Detection of PD in Patients	Multi-Class Support Vector Machine	90%, 91%, 95%, 96% (Acc on 4 classes)
[39]	Automated Method to Measure Motor Symptoms	Random forest Regressor	0.739 Correlation
[40]	Early Stage Detection of PD	Hidden Markov Model	100% Accuracy

Research for Parkinson's Disease Using Speech Data			
Paper Name	**Objective**	**Modelling**	**Results**
[41]	Early Stage Detection of PD	Bidirectional LSTM	80.70% F1-Score
[42]	Detection of Dysarthria	Fine-tuned Ensemble CNNs	99% Accuracy
[43]	Detection of PD	Adaptive CROW Search Algorithm & CNN	96% Accuracy
[44]	Detection of PD	Convolutional Neural Net	93% Accuracy

Research for Parkinson's Disease Using Vision Based Data			
Paper Name	**Objective**	**Modelling**	**Results**
[45]	PoseEstimation for PD Severity Detection	Random Forest Classifier	62.8% Sensitivity, 90.3% Specificity
[46]	Detection of Severity in Gait Impairment	Random Forest Classifier	50% Balanced Accuracy
[47]	Early Stage Detection of PD	Convolutional Neural Net	100% Accuracy

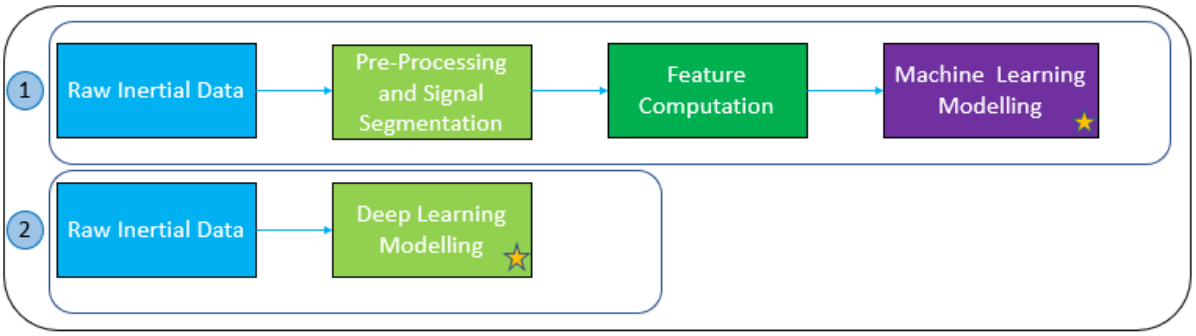


Figure 2.1: Summary of generalized methodology being followed in the previous works while working with inertial data. (1) Generalized methodology for machine learning based models. (2) Generalized methodology for deep learning-based models.

As can be seen above, the basic processing pipeline can be divided into two types: one where machine learning models are being utilized and one with deep learning models. With machine learning models [48]–[50], we observe raw data being pre-processed and decomposed into individual segments, which is then used for feature computation, which in turn are used to train the model. Here factors like [i] techniques used for signal decomposition, [ii] Feature computation, and [iii] deciding upon label for each segment in cases where a single segment contains data from multiple classes (as in cases with Parkinson's detection) etc., plays a major role in the modelling efficiency and effectiveness. Whereas with deep learning models, raw data is utilized and fed into the model and here too factors like signal decomposition and window label selection plays an important role in the model efficiency and effectiveness.

In our work we try to shine light on effect of different sized segments along with effect of different window label selection techniques. Further we show this for both machine learning and deep learning models and present a comparison of both.

Chapter 3

Methodology

In the previous section, we have reviewed different methodologies for working with inertial data in different application areas. In this work we focus on detection of an anomalous event from inertial data of patients suffering from Parkinson's disease [16]. Specifically, we focus on Tremor (involuntary quivering movement), Dyskinesia (Too much or extra movement), and Bradykinesia (slowness of movement) detection [12]. In the following section we shine light on various pre-processing, feature engineering, model training and model evaluation techniques that we have used for anomalous event detection in inertial data of patients with Parkinson's.

3.1. Anomaly Detection and Classification

We propose a two-stage methodology for anomalous event detection from inertial data of patients suffering from Parkinson's. In the first stage we classify the signal segment as an anomalous one or as a normal one. If the segment is classified as an anomalous segment, then it is passed through the second stage where we classify the type of anomaly that has occurred. Second stage involved multi-label classification, and thus identifies all the anomalies present in a signal, using which we make quantify the severity of symptoms. If the first stage classifies the signal as normal, the second stage is skipped, as no anomaly has occurred.

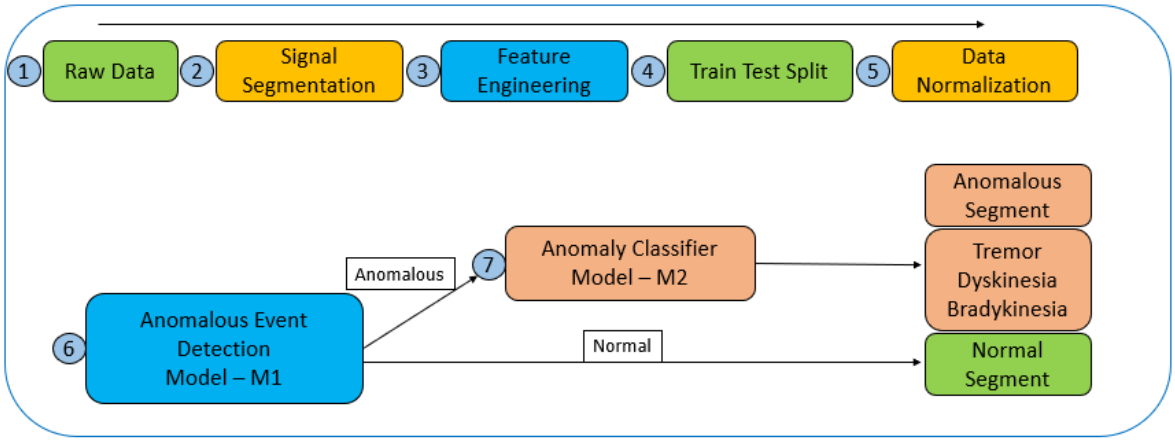


Figure 3.1: The proposed 2 stage methodology for anomaly detection and classification.

The first stage is a binary classifier, where we classify the signal as an anomalous one, or a normal one. For this purpose, the three output labels (Tremor, Dyskinesia, and Bradykinesia) from the dataset are merged into a single binary label, where a 1 represent occurrence of an anomaly and a 0 represent a normal signal segment or the absence of anomaly. The three input

labels i.e., Tremor, Dyskinesia, and Bradykinesia are first converted into binary variables (1's and 0's). We then create our final label for signal classification (anomalous or normal) by a simple OR operation on these three columns i.e., if any of the anomaly has occurred then the signal is anomalous, and the signal is normal, only if no anomaly has occurred. We then evaluate different preprocessing parameters (described in next sub-sections), perform feature engineering, and evaluate use of machine learning and deep learning model for classifying the input inertial signal as anomalous or normal.

Table 3.1: Data preparation for anomaly detection. Here in the first 3 columns, a 1 represents presence of anomaly, while a 0 represents its absence. The final label is Anomalous, if any of the 3 anomalies are present in a signal segment.

Tremor	Dyskinesia	Bradykinesia	Anomalous Event detection
1	1	1	Anomalous
1	1	0	Anomalous
1	0	1	Anomalous
1	0	0	Anomalous
0	1	1	Anomalous
0	1	0	Anomalous
0	0	1	Anomalous
0	0	0	Normal

Now if the signal is classified as an anomalous signal, then it passes through the second stage, which is multi label classification stage. Unlike normal classification, where the classes are mutually exclusive, here we deal with mutually non-exclusive classes or labels. This means that a single segment can be classified as a one with tremor and with dyskinesia. This is useful to quantify the severity of symptoms, i.e., a person with tremor and dyskinesia, would have a higher severity score as compared to the one with only dyskinesia. For this multi label classification stage, we then again explore different preprocessing parameters, features, and different modelling techniques. A performance comparison of all the evaluations, and best-found hyper parameters, features, and modelling technique are shared in the results section.

Table 3.1: Data preparation for anomaly classification. Here the first column represents possible outcomes from the anomaly detection model (stage 1). The last column is the final output label column for the anomaly classifier.

Anomalous Event detection	Tremor	Dyskinesia	Bradykinesia	Anomaly Classifier
Anomalous	1	1	1	All
	1	1	0	Tremor & Dyskinesia
	1	0	1	Tremor & Bradykinesia
	1	0	0	Tremor
	0	1	1	Dyskinesia & BradyKinesia
	0	1	0	Dyskinesia
	0	0	1	BradyKinesia
Normal	0	0	0	None

3.2. Dataset

3.2.1. Levodopa Response Study

For our study, we have used the dataset from levodopa response study, as shared in [5]**.** The shared data focuses on motor fluctuations-based anomalies observed in patients suffering from Parkinson's disease. Three anomalies are considered in this dataset i.e., Tremors, Dyskinesia, and Bradykinesia. Here Tremors refers to involuntary movements, Dyskinesia refers to too much or extra movement, and Bradykinesia refers to slowness of movement. The great thing about this data is that the raw data has been made available via open data repository.

3.2.2. Participants Characteristics

For data collection, 31 subjects with Parkinson's and Hoehn and Yahr score ranging from II to IV, participated in a 4-day study. Here Hoehn and Yahr scale is used to measure how Parkinson's symptoms progress and the level of disability. The subjects included in the study were men and women between age of 30 and 80 years, diagnosed with PD, currently taking L-Dopa, with at least mild dyskinesia and motor fluctuations. Subjects with other serious neurological disorder like epilepsy, brain tumor, hydrocephalus were excluded from the study.

3.2.3. Sensors Used

Data was collected using 3 different sensors from 3 different body locations. Geneactiv wristwatch was wore on the most affected upper limb, a pebble smartwatch was to be wore on the least affected upper limb, and a Samsung galaxy s2 smartphone was attached to the waist. 3D (x, y, and z axis) acceleration data were collected at a frequency of 50Hz. In addition to the 3D data, vector magnitude is also reported in the dataset. Labeled data, representing symptom severity (for tremors) and presence (for dyskinesia and dyskinesia), is also provided for each limb and each motor task, just as generated by a clinician.

Figure 3.2: Sensors Placement [5]

3.2.4. Data Collection Process

The data collection was a 4-day activity. On the first day, subjects involved in the study performed multiple motor tasks, in an on-medication state, while in the laboratory. These tasks include standing, walking in a straight line, walking in a straight line while counting backwards, walking up the stairs, walking down the stairs, walking through narrow corridors, finger to nose movements (twice with each hand), alternating hand movements, drawing, typing, pouring water from a bottle, sit to stand to sit activity, arranging paper sheets in a folder, nuts and bolts assembly, and folding a towel three times. These activities were repeated after every 30 minutes for 3 to 4 hours, making a total of around 6 to 8 repetitions of each activity. A trained clinician was tasked with the annotation task, where he identified severity (for tremors) and occurrence (for dyskinesia and bradykinesia) of anomalies in each sensor's data.

On the second and third day, the participants went home with the sensors, and performed their daily activities as per usual. This generated 2 days of unlabeled data as well. On the fourth day, the participants again came to the laboratory, but in a non-medicated state, and performed several motor tasks (just like on the first day). After this first testing phase on day 4, the patients took their scheduled medication dose, and then performed 5 to 7 repetitions of the motor tasks. This data was also annotated by a trained clinician.

3.2.5. Data Distribution

The distribution of tremor severity, and absence/presence of dyskinesia and bradykinesia is shown below

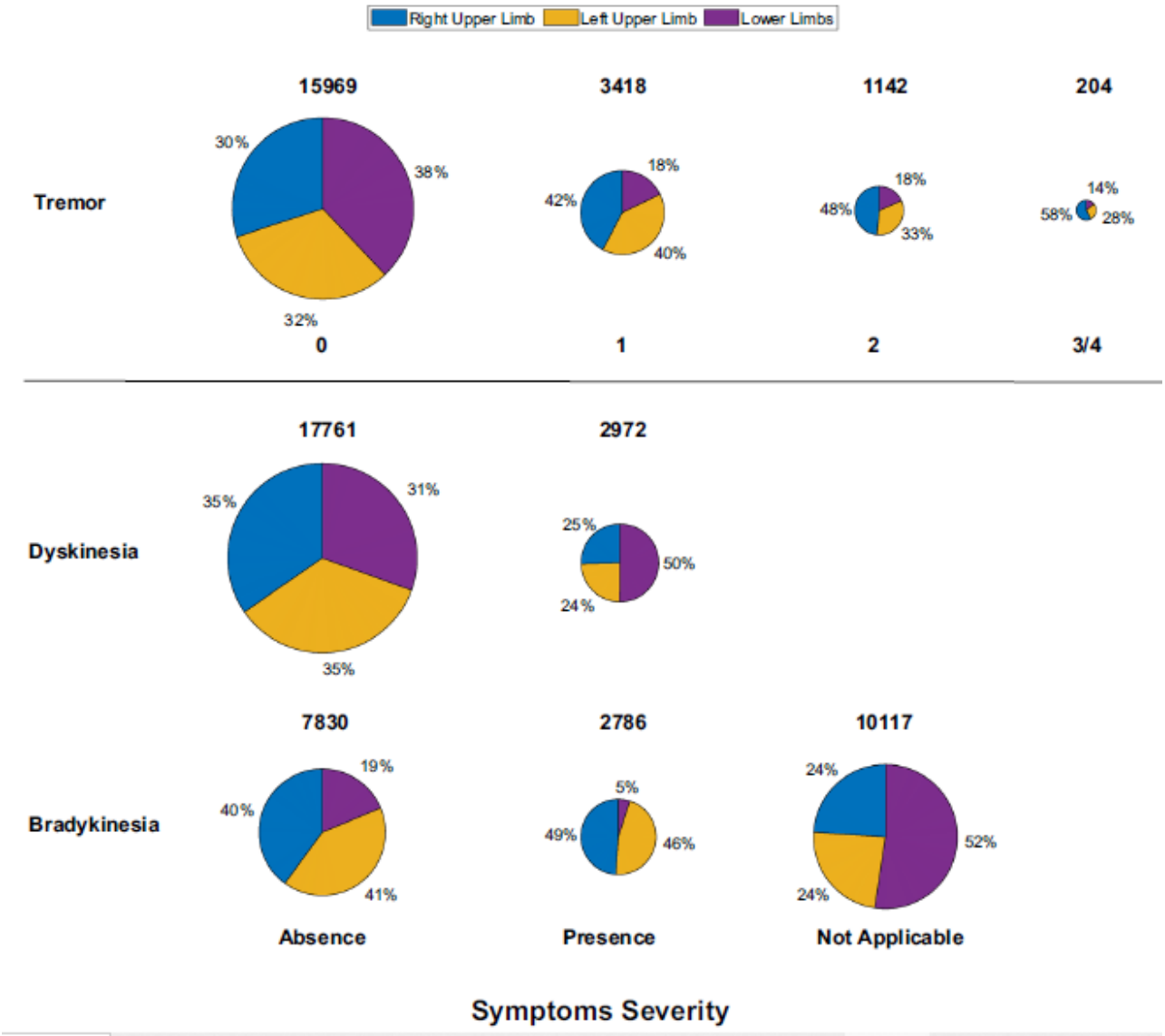


Figure 3.3: Data distribution for the three anomalies (Tremors, Dyskinesia, Bradykinesia) [5].

3.3. Pre-Processing

Inertial data is collected from Inertial measurement units (IMU's), which consists of sensors like accelerometer (acceleration) and gyroscope (angular velocity), and results in 6D (6-Dimensional) data points i.e., 3D acceleration and 3D angular velocity. For our work we have used dataset shared in Levodopa response study, which only consists of 3D accelerometer readings.

The raw input signal is firstly decomposed into smaller segments. For this purpose, we used a fix sized window with a stride of 20%, as demonstrated in Figure 3.1. A window of size X is placed on top of the input data, and the underlying datapoints results in a single segment. Next, this window is moved forward with a stride of 20%, and the next segment is collected. A stride

of 20% means an overlap of 80% between two consecutive segments. Here, we also explored different window sizes of 50, 150, and 250 datapoints to find out the best performing one. The next step is to assign label to the decomposed segment. For this purpose, we have explored two different labeling approaches. In the first one, the segment label is as per the mode of that segment. Here, mode refers to the most frequent label of the segment. So, if in a segment, the anomalous data points are more frequent than the normal ones, then the segment is an anomalous one, else it is a normal segment. In the second labeling approach, we treat the segment as an anomalous one if it contains any anomalous data points, else it is treated as a normal segment (zero anomalous data points). On a whole, we have explored three different window sizes, each with two different labelling approaches, and have shared their performance wise comparative analysis.

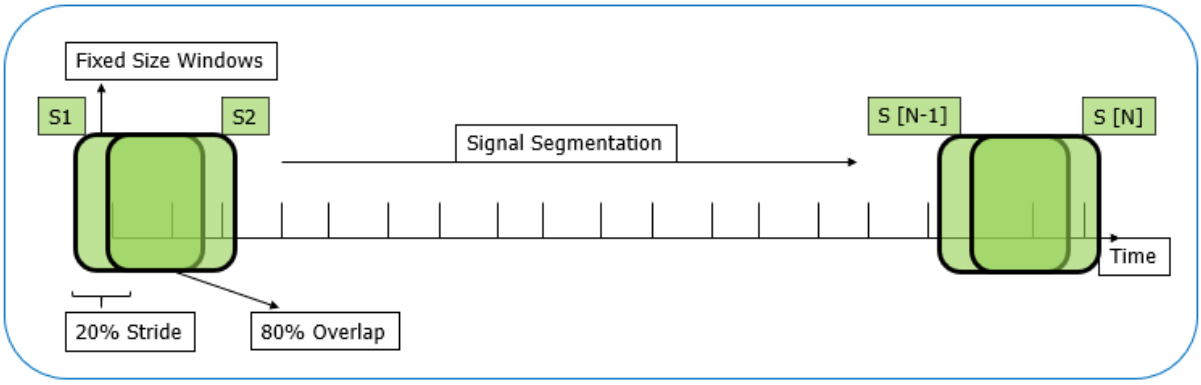


Figure 3.1: Signal segmentation. A fixed segment/window size is defined (50, 150, 250), which is then moved over the whole signal with a stride of 20%.

This gives us six different preprocessing configurations, as shown below

Table 3.2: The 6 preprocessing configurations explored. Here F refers to the segment/window size and S refers to the labeling methodologies.

Configurations	Description
F250-S1	Window Size = 250, Labeled using S1 methodology
F150-S1	Window Size = 150, Labeled using S1 methodology
F50-S1	Window Size = 50, Labeled using S1 methodology
F250-S2	Window Size = 250, Labeled using S2 methodology
F150-S2	Window Size = 150, Labeled using S2 methodology
F50-S2	Window Size = 50, Labeled using S2 methodology
* S1 : Segment is anamalous, if it contains any anomalous portion * S2 : Mode is used as the segment label	

3.4. Machine Learning

In this section we present a two-stage machine learning based pipeline which is used for anomaly detection and classification [51]. The whole pipeline includes feature computation from raw data, and then using those features for Random Forest training model training.

3.4.1. Feature Computation

The segments of raw data are then used for feature computation. A total of 150 features are extracted from each segment. These include time domain, frequency domain, and wavelet domain features. A complete list of computed features can be found in Figure 3.3. Note that we are exploring different window sizes and labeling methodologies as well, so that will give us 6 different datasets of signal segments. We then compute features from each of these 6 datasets. We have used MATLAB for signal segmentation as well as for feature computation. These computed features are then used for model training and evaluation, which is done using Python. In the results section, we have also shared the top 10 features found, for each of the sensor (GeneActiv, Pebble, and Phone), and for each of the six preprocessing configurations.

Table 3.3: Computed Features. Here each feature is computed across all the 3 axes (x, y and z).

Features	Domain	Equation
Mean	T	$\mu = \frac{1}{n}\sum_{k=1}^{n} X_k$
Median	T	$[(n+1)/2]$th value
Standard Deviation; Variance	T	$\sigma = \sqrt{\frac{\sum(x_i-\mu)^2}{N}}$; $s^2 = \frac{\sum(x_i-\bar{x})^2}{n-1}$
Maximum	T	$max(x)$
Minimum	T	$min(x)$
Index of Maximum	T	Index of max value in stride signal
Index of Minimum	T	Index of min value in stride signal
Skewness	T	$s = \frac{E(x-\mu)^3}{\sigma^3}$
Kurtosis	T	$k = \frac{E(x-\mu)^4}{\sigma^4}$
Entropy	T	$ent = \sum_{k=1}^{n}(P_k)log_2(P_k)$ where $P_k = \frac{\frac{S_k}{max(S_k)}}{\sum_{k=1}^{n}\frac{S_k}{max(S_k)}}$
Root Mean Square	T	$rms = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\lvert X_k\rvert^2}$
Energy	T	$eng = \sum_{k=1}^{n}\lvert X_k\rvert^2$
Power	T	$pow = \frac{\sqrt{\sum_{k=1}^{n}\lvert X_k\rvert^2}}{n}$
Mean Absolute Deviation	T	$mad = \frac{1}{n}\sum_{k=1}^{n}(X_k - \mu)$
Interquartile Range	T	$iqr = Q_3 - Q_1$ where $Q_1 = \frac{n}{4}$ and $Q_3 = \frac{3n}{4}$
Signal Magnitude Area	T	$sma = \frac{1}{t}(\int_0^t \lvert x(t)\rvert dt + \int_0^t \lvert y(t)\rvert dt + \int_0^t \lvert z(t)\rvert dt)$
Zero Crossing Rate	T	$zcr = \frac{1}{T}\sum_{k=1}^{T}\lvert s(k) - s(k-1)\rvert$
Slope Sign Change	T	$ssc = \sum_{k=1}^{T}\lvert s(k) - s(k-1)\rvert if x! = 0$
Waveform Length	T	$wfl = \sum_{k=1}^{T}(s(k) - s(k-1))$
FFT Coefficients	F	$fft(k)$ where k = [1,15]
Mean	F	$\mu_F = \frac{1}{n}\sum_{k=1}^{n} fft_k$
Maximum	F	$max(fft)$
Magnitude	F	$mag_F = max(2 * \lvert fft\rvert)$
Energy	F	$eng_F = \sum_{k=1}^{n}\lvert fft_k\rvert^2$
Band Power of Signal	F	Average power in the signal, fft
Sum of Squares of COEF	W	$\lvert\lvert cD_k\rvert\rvert^2$ where k = [1,n]
Sum of Absolute COEF	W	$\lvert\lvert cD_k\rvert\rvert$ where k = [1,n]

3.4.2. Model Training and Evaluation

After we have computed features from raw data of all participants, we then merge them into a single Pandas DataFrame. Next all the three labels (Tremor, Dyskinesia, Bradykinesia) are converted into binary labels (1's and 0's). Then for the first stage of signal classification as an anomalous signal or a normal signal, we apply OR operation on these three labels, to get binary labels, representing the signal as either anomalous or normal. Here 1's represents occurrence of an anomalous event, and the 0's represents normal events or the absence of an anomalous event. Afterwards, the data is divided into train and test sets using a percentage split of 70%-30% respectively, and is then normalized using zscore normalization technique, which gets mean of all values to 0 with a standard deviation of 1, thus resulting in more accurate and faster convergence

$$z = \frac{x - \mu}{\sigma}$$

From data exploration it was found that a high-class imbalance exists between the anomalous and normal data, with majority of datapoints belonging to normal class. To deal with class imbalance we have used Random Under Sampler, which under samples the normal class by randomly picking samples without replacement.

The prepared data is then used to train a Random Forest classifier (decision tree-based classifier), for binary classification of the input signal segment as an anomalous one or a normal one (stage 1). For Random Forest we have used Gini Criteria with 100 number of trees (n_estimators). We use 10-fold cross validation technique for model training and evaluation. For evaluation we have used performance KPI's like accuracy, AUC, Recall, Precision, F1 score, and confusion matrix. Afterwards, we get feature importance and top-10 feature list.

If stage 1 classifies the input signal segment as an anomalous one, then it passes through second stage, where we further classify the type of anomaly (Tremor, Dyskinesia, Bradykinesia) present in the signal. This is a multi-label classification problem and is handled using label powerset approach. Using label powerset approach, we transform the multi-label classification problem to a multi-class classification problem with a single multi-class classifier (Random Forest). To achieve this, we transform the output array of nx3 (Tremor, Dyskinesia, Bradykinesia) into a nx1 array, by assigning a unique class label to each of the possible combination of the three anomalies, as shown below

Table 3.4: Label Powerset approach demonstration. Each output label corresponds to a unique combination of the input columns.

Using Phone Sensor			
Tremor	Dyskinesia	Bradykinesia	Output Label
0	0	1	0
0	1	0	1
1	0	0	2
1	0	1	3
1	1	0	4

The prepared data is normalized using z score and is then used to train and evaluate a Random Forest Classifier with Gini index and 100 trees. Model is trained and evaluated using 10-fold cross validation and is evaluated for all the performance metrics as used for stage 1 model.

For Machine learning based pipeline, we used MATLAB for feature computation. Afterwards we trained Random Forest using Anaconda's Python distribution, on Jupyter Notebook, with Pycaret (used for model training and testing) and Imblearn (used for implementing random under sampling). Feature computation, Random Forest model training and execution was all performed on a local system, with window 10 operating system, and with 24 GB of RAM.

3.5. Deep Learning

Here too, we have a two-stage pipeline. In the first stage, the input signal segment is classified as an anomalous signal or as a normal one, using a deep learning based binary classifier. If the signal segment is classified as an anomalous signal, then it is passed through the second stage, which performs multi-label classification using a deep learning-based model.

Unlike before, here only a single feature, magnitude of the raw acceleration values, is passed to the deep learning model for training and inference. This reduces the overall processing time and computation cost, which was being used for feature computation, as described above while using machine learning based techniques. Just like before, this deep learning-based pipeline is also trained and evaluated using six different data configurations, made using three different window sizes for segmentation, and two different segment labeling methodologies.

3.5.1. Network Architecture

In our study, we propose a deep learning network (HARDenseRNN), which is combination of convolutional neural networks (CNN) and recurrent neural networks (RNN). Previously, a combination of CNN and RNN based networks, has shown great potential in areas like computer vision, where spatial features are first extracted from input data using a CNN, which are then fed into a RNN based network for capturing temporal relations. In our proposed architecture, we first pass the raw sensor input through a CNN based network for spatial feature extraction, which are then concatenated with the raw input signal, and then passed to a RNN based network for capturing temporal features. These features are then passed through Dense (Fully connected layers) for making predictions. The concatenation of raw input signal, with the feature map generated by CNN, ensures that RNN can take advantage of both the raw signal, and the CNN features, for temporal feature extraction. To the best of our knowledge, such concatenation of CNN feature maps with the raw input signal, before passing to RNN model, has never been experimented before for any classification problem. In our experiments, we have observed that this concatenation stage, significantly improves the overall performance.

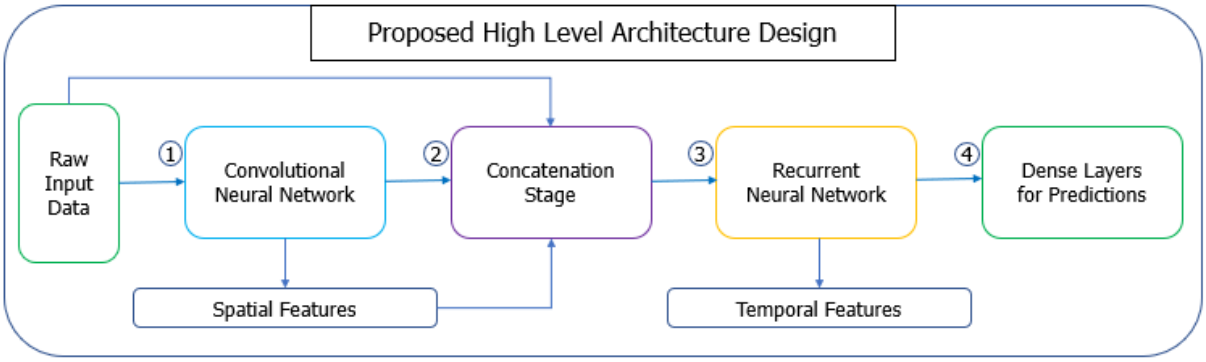


Figure 3.4: High Level Architecture Design of Proposed Deep Learning Network

The convolutional neural network part of our proposed architecture utilizes 2 InceptionResNet inspired convolutional neural network for spatial feature extraction. The benefit of using InceptionNet is that it utilizes variable sized kernels and is thus able to extract both local and global features. It also makes use of 1x1 convolutions which also reduces the computation cost. The benefit of using ResNet is that it makes use of the residual connections which enables the network to bypass a module if it does not contribute to the final classification. The raw input signal is convolved with 10 1x1 kernel and is then convolved with 5 different sized kernels (1x1,1x3,1x5,1x7,1x9) followed by max-pooling and concatenation layers. The generated feature maps are then concatenated with the raw input features using residual connection. To ensure consistent spatial dimensions for concatenation, we make use of 64, 1x1 sized kernel convolutions. The generated feature maps from the first InceptionResNet module passes through a dropout layer and a batch normalization layer before being passed to the second InceptionResNet module. Below figure shows the utilized InceptionResNet modules.

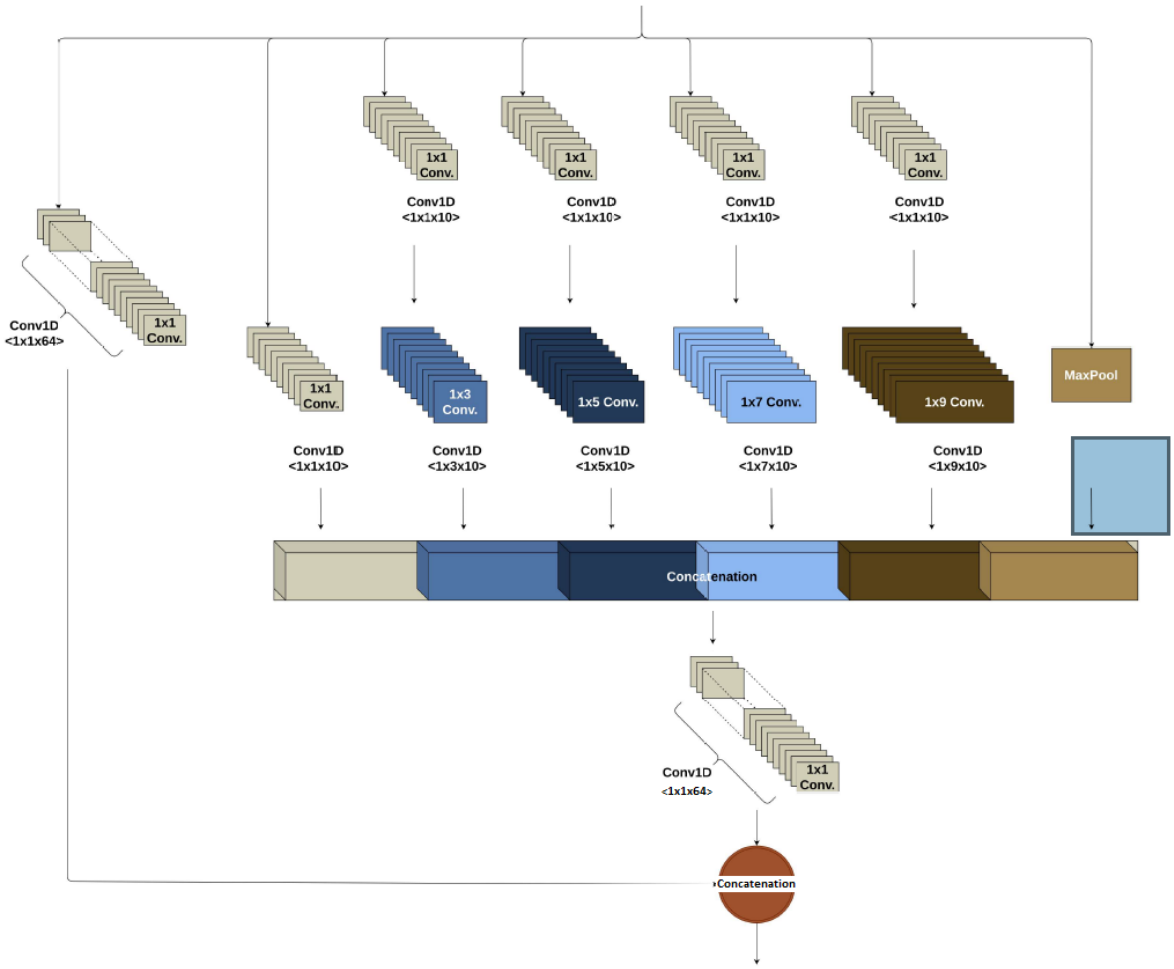


Figure 3.5: Architecture of CNN from the Proposed Deep Learning Network

The output from second InceptionResNet module also goes through a dropout layer and a batch normalization layer and is then concatenated with the raw input features. Here too, we make use of 64, 1x1 sized kernel convolutions to ensure consistent shapes for concatenation stage.

The concatenated feature maps are then passed through a bi-directional GRU with 128 units, which has powerful modeling capabilities for long-term dependencies. The benefit of using GRU as our RNN is that it is able to deal with the vanishing-exploding gradient problem and is also much more efficient than the other recurrent neural networks like LSTM. Further we have used a bi-directional GRU, which enables us to capture temporal relations in both forward and backward direction. GRU has two gates, and update gate (determines how much past information needs to be retained) and a reset gate (determines how much past information is to be forgotten), where both takes the current input x_t along with previous state h_{t-1} , as input, and uses sigmoid function to output the gate's value. The output of these gates and of the GRU unit is calculated as follows

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right)$$
$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right)$$
$$\tilde{h}_t = \tanh\left[W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right]$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

Figure 3.6: Mathematical Representation of a GRU [52]

Here W_r, U_r, W_z, U_z, W_h and U_h represents the weights, b_r, b_z, b_h are the bias vectors for input x_t and previous state h_{t-1}, σ is the logistic sigmoid function, *tanh* is the hyperbolic tangent activation function, and $\odot$ denotes the Hadamard product. A simple GRU unit is shown below.

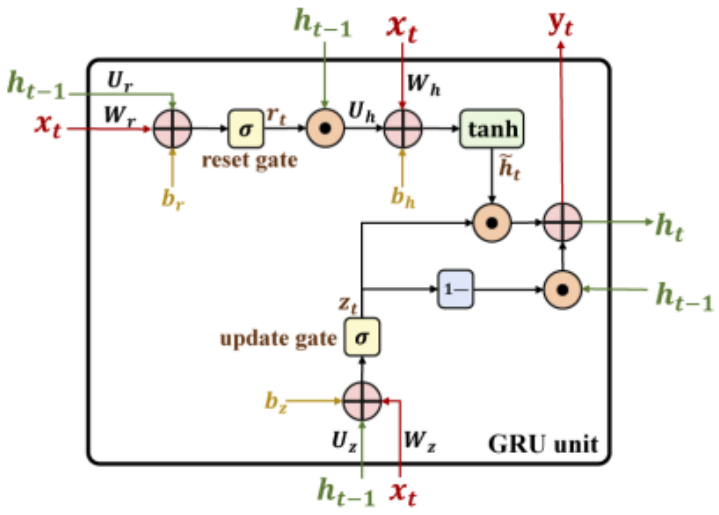


Figure 3.7: A GRU Cell [52]

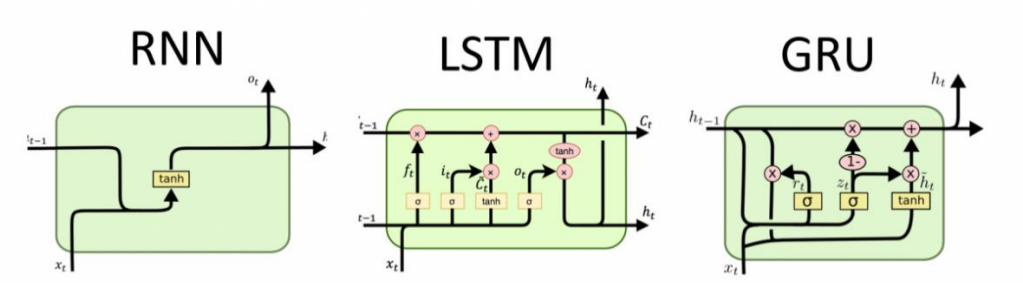


Figure 3.8: Cell Unit Design of Various Recurrent Neural Nets

Finally, the output from GRU first passes through batch normalization, and is then flattened. It then passes through a dropout layer, and then through 512 unit's dense layer. The output then passes through a dropout layer and is used to make final predictions. For the first stage which involves binary classification, we use a single output node, with sigmoid activation function and binary cross entropy loss function. For the second stage involving multi-label

classification, we use three output nodes, with sigmoid activation function and binary cross entropy loss function. This enables binary classification for each of the three anomalies (tremor, dyskinesia, bradykinesia) i.e., anomaly exists or not. The final network architecture is shown below

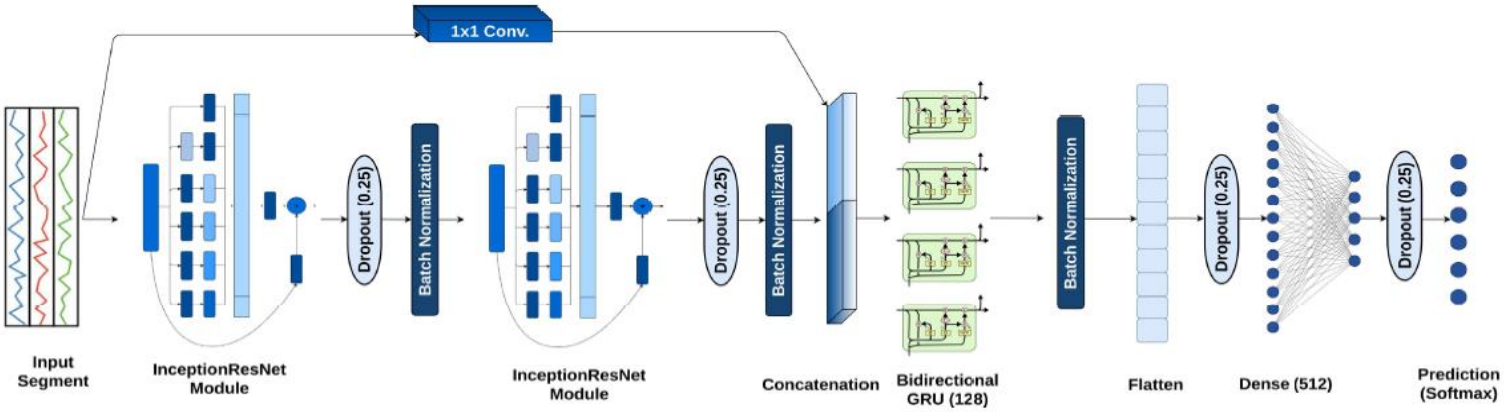


Figure 3.9: Proposed Deep Learning Network Architecture

We train our proposed model using magnitude of raw 3D acceleration values. The raw magnitude is reshaped in accordance with different segment sizes, resulting in a 3D matrix of shape (n, f, 1), where *n* is the number of records, and *f* is the segment size. For the first stage of binary classification, the model had 753,035 trainable parameters and 896 non-trainable parameters. While for the second stage of multi-label classification, the model had 754,061 trainable parameters and 896 non-trainable parameters.

Deep learning model training was done on Google Colab Pro with a GPU (Tesla P100), with the help of TensorFlow 2.4 and Keras 2.4.3 for model construction and training. Model was trained for 100 epochs using a batch size of 128. The number of epochs were selected empirically by monitoring the plot of training and validation loss and accuracy values. We also used early stopping with 'max' option for validation accuracy with a patience of 10 epochs. We also used sklearn for splitting data into train and test sets, and seaborn for making visualization.

Chapter 4

Results

In this section we present the achieved results from 72 different models. 36 different models are trained for anomaly detection (first stage of proposed pipeline) and 36 different models are trained for anomaly classification (second stage of proposed pipeline). These 36 models are driven from 3 different datasets i.e., captured from GeneActiv IMU (inertial measurement unit), or Pebble smartwatch, or Samsung S2 smartphone. These 3 datasets are then further divided into 6 pre-processing configurations, where we have 3 segment sizes (50, 150, 250) and 2 segment labeling methodologies (using mode as segment label, annotating segment as anomalous if it contains any anomalous portion in it). Then each of these 6 configurations, are used to make two different models, one machine learning based model (Random Forest) and one deep learning-based model (Newly proposed deep learning architecture). Here a separate model is made for each of the 2 stages of the proposed pipeline, i.e., anomaly detection and anomaly classification, as discussed in detail in section 3. So that makes, 2 machine learning models (one for each stage) and 2 deep learning models (one for each stage), making a total of 4 models for each pre-processing configuration. On a whole,6 pre-processing configuration, and 4 models for each configuration, makes 24 models per sensor. As we have 3 sensors, this makes a total of 72 models that are evaluated and presented in our work.

Table 4.1: Breakdown of the 72 models that are evaluated in our work. Here S1 represents annotating the segment as anomalous if it contains any anomalous portion, else labeling it as a normal one. S2 represents using segment's mode as the label for segment

Sensor	Segment Size	Segment Labeling	Stage	Model Used
Phone / GeneActiv / Pebble	250	S1	Anomaly Detection	RandomForest
				HARDenseRNN
			Anomaly Classification	RandomForest
				HARDenseRNN
		S2	Anomaly Detection	RandomForest
				HARDenseRNN
			Anomaly Classification	RandomForest
				HARDenseRNN
	150	S1	Anomaly Detection	RandomForest
				HARDenseRNN
			Anomaly Classification	RandomForest
				HARDenseRNN
		S2	Anomaly Detection	RandomForest
				HARDenseRNN
			Anomaly Classification	RandomForest
				HARDenseRNN
	50	S1	Anomaly Detection	RandomForest
				HARDenseRNN
			Anomaly Classification	RandomForest
				HARDenseRNN
		S2	Anomaly Detection	RandomForest
				HARDenseRNN
			Anomaly Classification	RandomForest
				HARDenseRNN

4.1. Anomaly detection (Binary Classification)

The first stage of our proposed pipeline is a binary classifier, which classifies the input signal segment as a normal signal or as an anomalous one. Below we present and evaluation of different anomaly detection models, using different datasets, different pre-processing configurations, and using two different modelling approaches i.e., Random Forest and our proposed deep learning network.

4.1.1. Smart Phone Data

A Samsung S2 phone is attached to the participants waist for inertial data collection. Our experiments show that models trained on this dataset outperforms the models trained on other datasets (GeneActiv dataset, Pebble dataset), for both anomaly detection and anomaly classification. This is consistent with the fact that human waist exhibits more clearly distinctive patterns and features as compared to upper limbs (wrists), while performing gait tasks[53].

4.1.1.1. Anomaly detection using Random Forest

For the anomaly detection stage, using machine learning model (Random Forest), we have achieved highest train accuracy of 93.04% with a segment size of 150 and with S1 as the

segment label. Here we achieved a recall of 92.27% and a precision of 96.49%. The same pre-processing configuration produces the highest test accuracy as well, i.e., test accuracy of 93.30%, with a recall of 92.72% and a precision of 96.48%. Results on rest of the pre-processing configurations are as shown below in Table 4.2**.**

Table 4.2: Results achieved for anomaly detection using Random Forest on phone's dataset, with 6 different preprocessing configurations

Phone							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	92.30%	91.40%	96.37%	92.69%	91.77%	96.56%
150	S2	92.84%	92.06%	96.55%	93.27%	92.58%	96.73%
50	S2	92.37%	91.68%	96.18%	92.51%	91.83%	96.28%
250	S1	92.50%	91.70%	96.11%	93.25%	92.63%	96.35%
150	S1	93.04%	92.27%	96.49%	93.30%	92.72%	96.48%
50	S1	92.41%	91.68%	96.18%	92.67%	92.04%	96.27%

It can be seen in Table 4.2 that for anomaly detection using Random Forest, the results are almost the same, irrespective of the configuration being used, but best results are being achieved from a middle level segment size (150), with S1 as the segment labelling methodology.

Following are the top 10 features, found using the best-found pre-processing configurations (segment size of 150 with S1 as labeling methodology), along with the achieved confusion matrixes (here a 1 represents anomalous class and 0 represents a normal class), and precision recall curves.

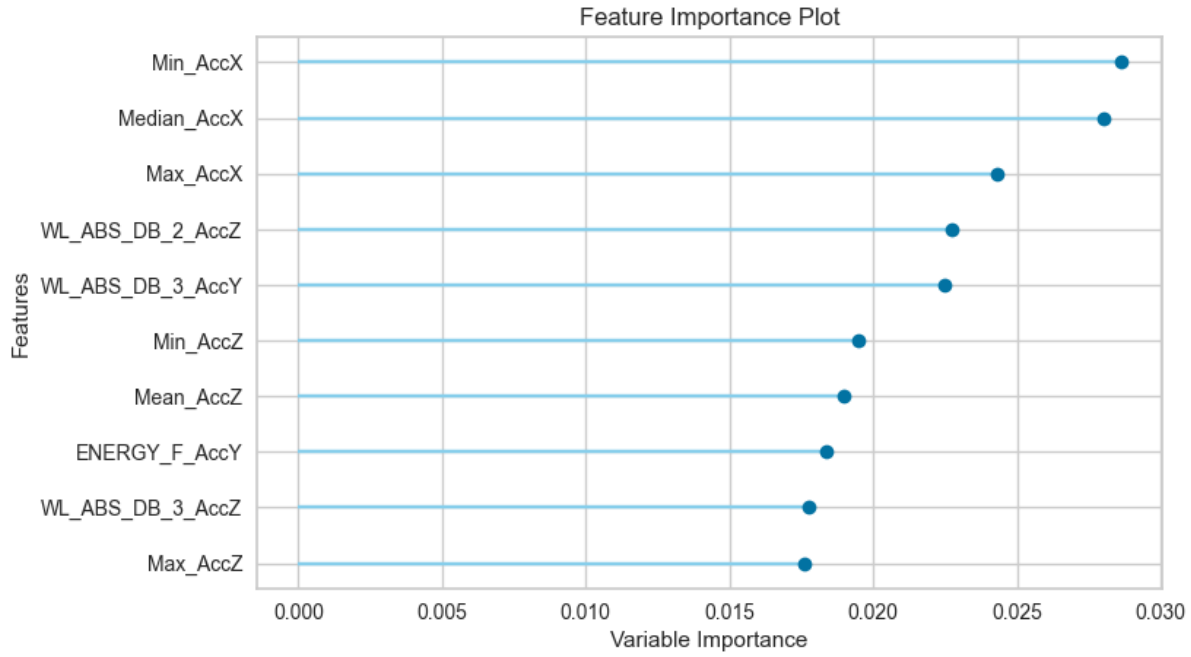


Figure 4.1: Top 10 features

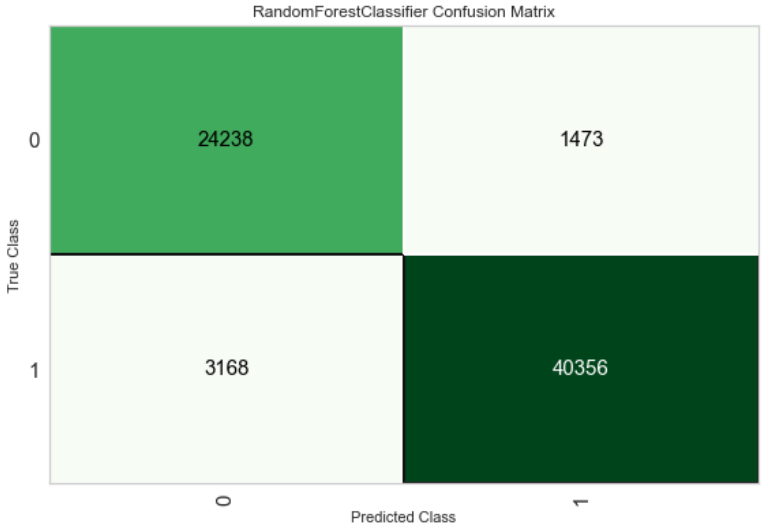


Figure 4.2: confusion matrix. Here 0 represents a normal class and 1 represents an anomalous class.

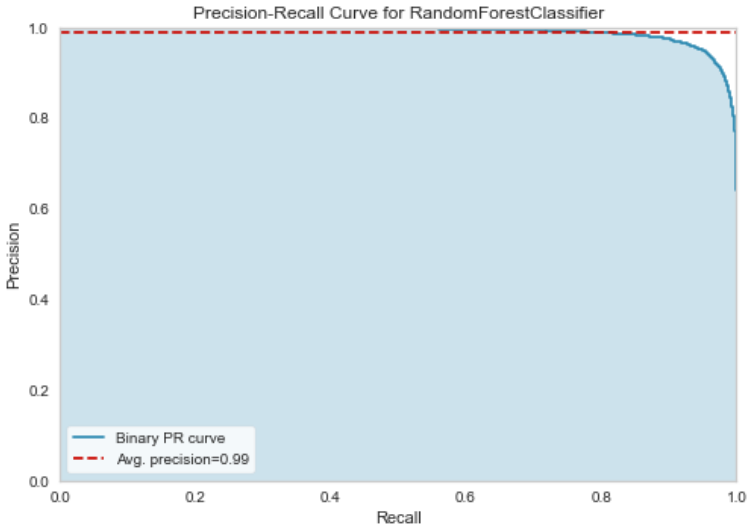


Figure 4.3: precision recall curve

4.1.1.2. Anomaly detection using HARDenseRNN

While using deep learning model (HARDenseRNN) for anomaly detection, we achieve highest train and test accuracy using a segment size of 250 and S2 segment labeling methodology. On train set, we achieve an accuracy of 96.43%, with a recall of 93.57% and a precision of 96.49%. While on the test set, we achieve an accuracy of 90.85%, with a recall and precision of 84.70% and 89.47% respectively. Results on rest of the pre-processing configurations are as shown below in Table 4.3**.** Here we have also observed that the deep learning model (HARDenseRNN)

outperforms the traditional machine learning model (Random Forest) on the train dataset, but RandomForest outperforms HARDenseRNN on the test dataset. Further we can see that the impact of changing segment size is more prominently visible on the deep learning-based model as compared with machine learning models.

Table 4.3: Results achieved for anomaly detection using HARDenseRNN on phone's dataset, with 6 different preprocessing configurations

Phone							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	96.43%	93.57%	96.49%	90.85%	84.70%	89.47%
150	S2	94.10%	93.60%	90.45%	88.47%	86.65%	82.47%
50	S2	79.39%	50.56%	87.10%	77.47%	47.28%	83.36%
250	S1	94.43%	88.91%	96.13%	89.29%	80.58%	90.05%
150	S1	95.28%	93.87%	93.46%	89.60%	86.38%	85.84%
50	S1	80.51%	80.90%	70.27%	77.90%	77.73%	67.06%

It can be seen in Table 4.3, that for anomaly detection using HARDenseRNN, larger segment sizes outperform the smaller ones, with a significant drop in accuracies on the smallest segments (size = 50). Further S2 labeling methodology outperform S1.

Following shown are the training and validation accuracy and loss curves

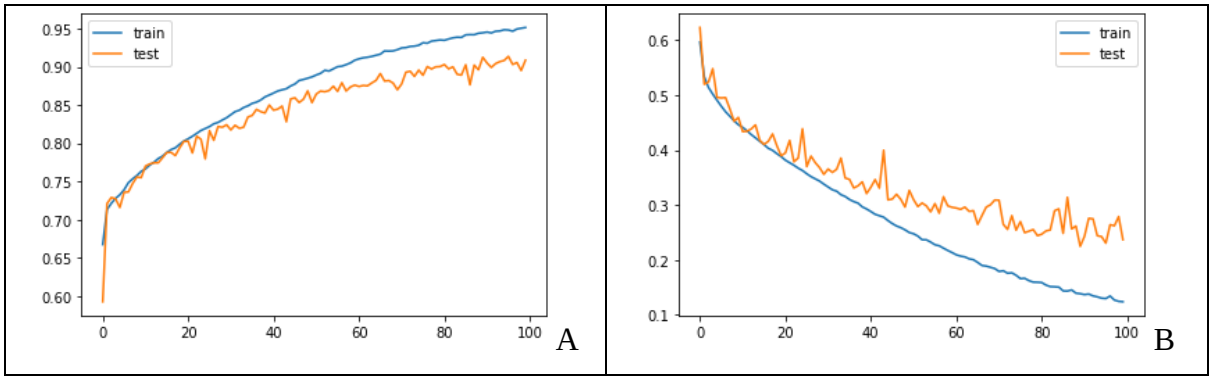


Figure 4.4: Accuracy curve (A) and Loss Curve (B)

Following is the confusion matrix on the best-found pre-processing configurations (segment size of 250 with S2 as labeling methodology), along with the precision recall curves.

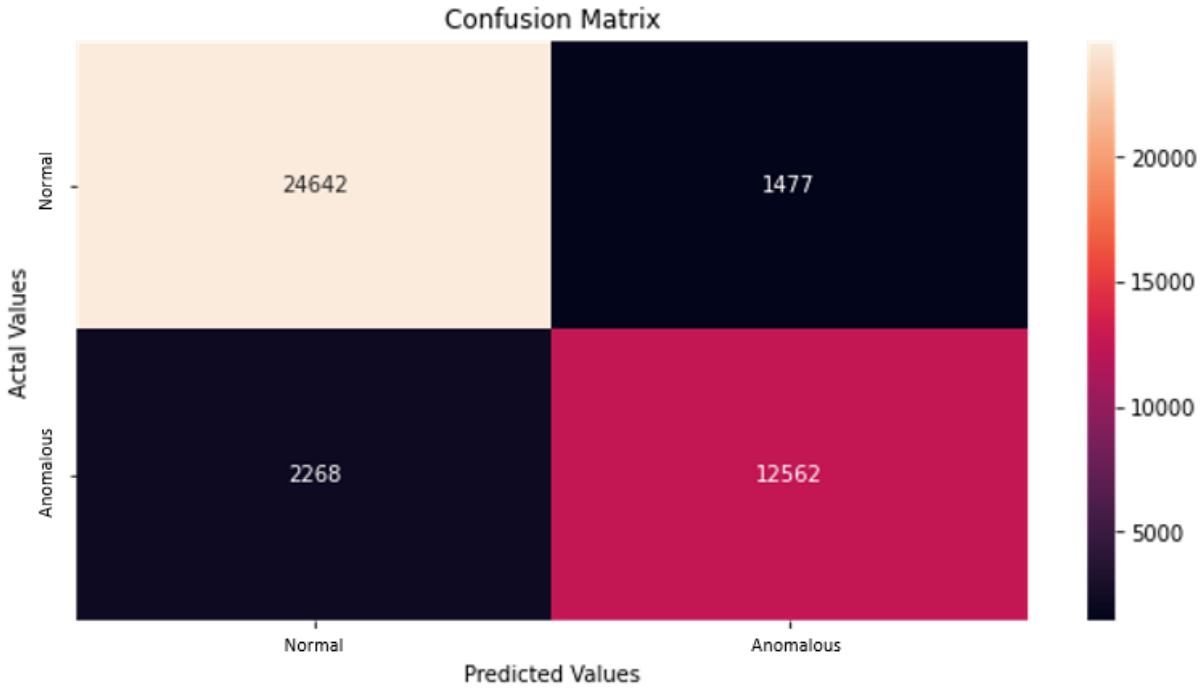


Figure 4.5: Confusion Matrix

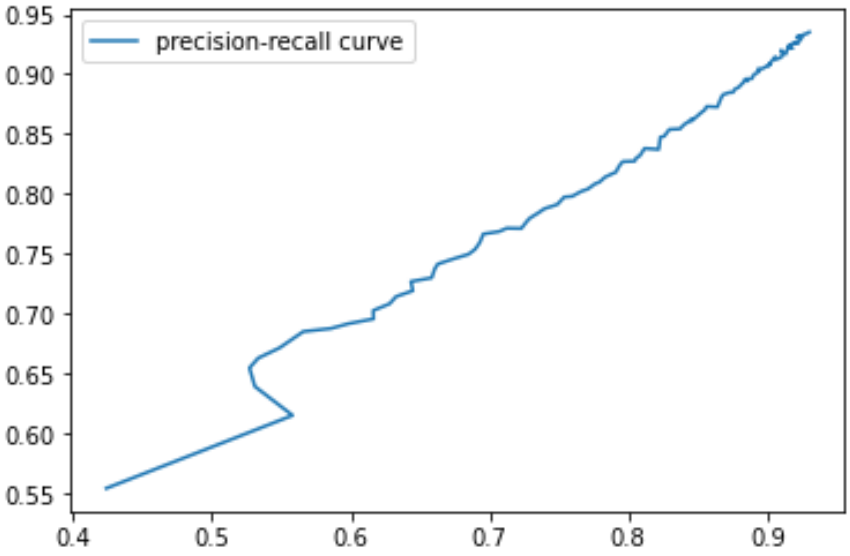


Figure 4.6: Precision Recall curve

4.1.2. Inertial Measurement Unit (IMU) Data (GeneActiv)

GeneActiv IMU is used to collect inertial data from a subjects most affected upper limb (wrist). After Phone, inertial data collected from GeneActiv smartwatch, produces the most accurate predictions. This means that data from most affected upper limb, as compared to other upper limb, is more useful for anomaly detection and classification.

4.1.2.1. Anomaly detection using Random Forest

For the anomaly detection stage, using machine learning model (Random Forest), we have achieved the most accurate results with a segment size of 250 and with S2 as the segment label. Here we achieved a train accuracy of 88.11%, a recall of 88.05% and a precision of 87.35%. The same pre-processing configuration produces the highest test accuracy as well, i.e., test accuracy of 88.63%, with a recall of 89.10% and a precision of 88.27%. Results on rest of the pre-processing configurations are as shown below in Error! Reference source not found.

Table 4.4: Results achieved for anomaly detection using Random Forest on GeneActiv dataset, with 6 different preprocessing configurations

GeneActiv							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	88.11%	88.57%	87.63%	88.63%	89.10%	88.27%
150	S2	87.90%	88.23%	87.54%	88.34%	88.56%	88.16%
50	S2	85.39%	85.57%	85.14%	85.82%	85.92%	85.69%
250	S1	88.22%	88.05%	87.35%	88.67%	88.65%	87.82%
150	S1	87.88%	87.63%	87.36%	88.43%	88.10%	88.09%
50	S1	85.38%	85.38%	85.06%	85.80%	85.71%	85.61%

It can be seen above, that for anomaly detection using Random Forest, the best results are achieved using larger segment size, with a slight drop in performance as we move towards smaller segment sizes. This holds true for both segment labeling methodologies, with higher performance using S2 labeling methodology.

Following are the top 10 features on the best-found pre-processing configurations (segment size of 250 with S2 as labeling methodology), along with the achieved confusion matrixes (here a 1 represents anomalous class and 0 represents a normal class), and precision recall curves.

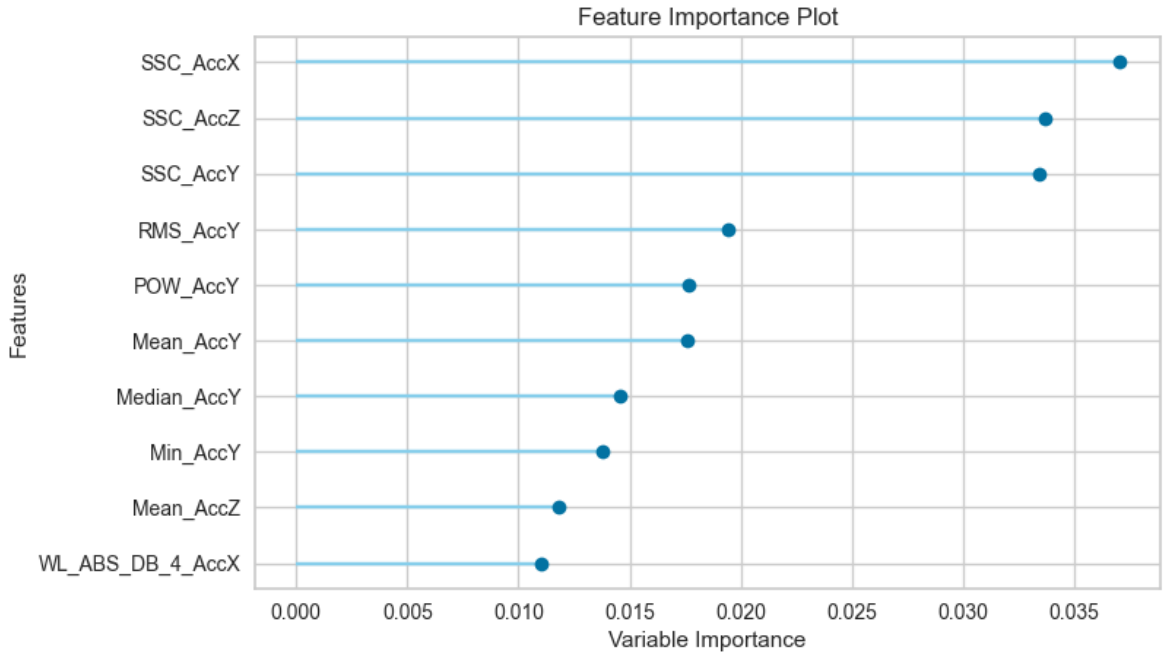


Figure 4.7: Top 10 Features

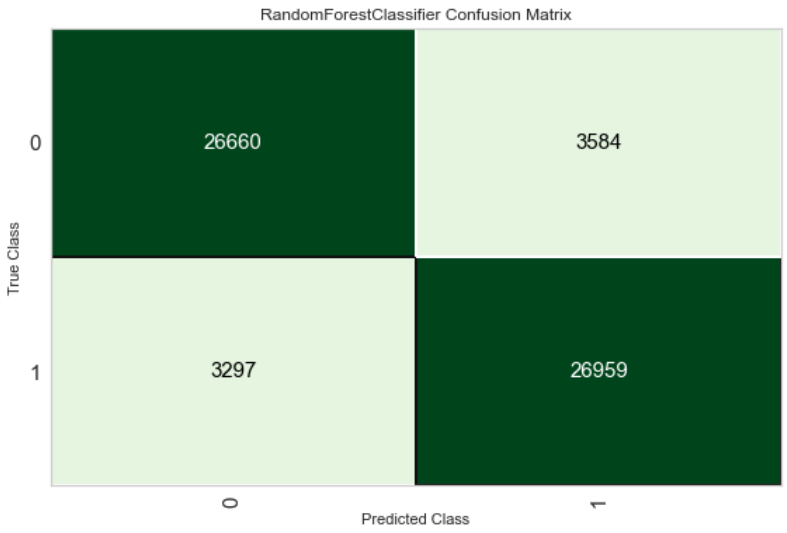


Figure 4.8: Confusion Matrix (Here a 0 represents a normal class, while a 1 represent anomalous).

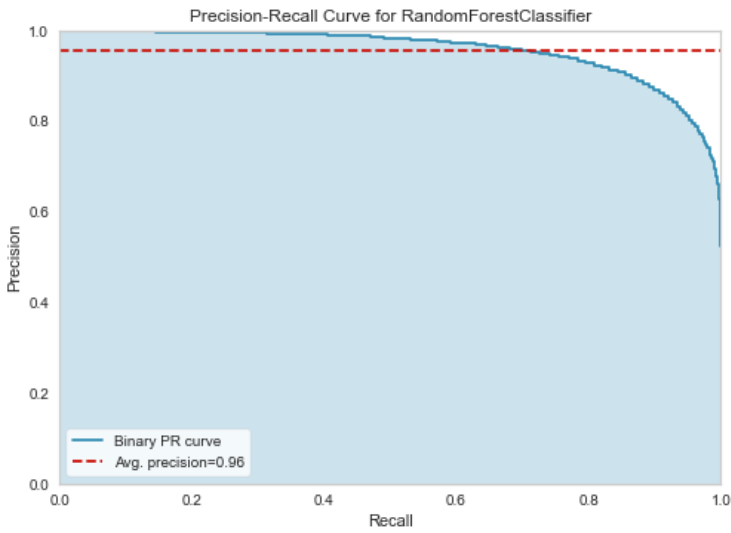


Figure 4.9: Precision Recall Curve

4.1.2.2. Anomaly detection using HARDenseRNN

For anomaly detection using deep learning model (HARDenseRNN), we achieve highest train and test accuracy using a segment size of 250 and S1 segment labeling methodology. On train set, we achieve an accuracy of 94.40%, with a recall of 96.70% and a precision of 92.89%. While on the test set, we achieve an accuracy of 88.58%, with a recall and precision of 93.03%

and 86.13% respectively. Results on rest of the pre-processing configurations are as shown below in Table 4.5**.** Here we have also observed that HARDenseRNN model performs almost in a similar manner as compared with Random Forest, with increase recall, and slightly decreased precision.

Table 4.5: Results achieved for anomaly detection using HARDenseRNN on GeneActiv's dataset, with 6 different preprocessing configurations

GeneActiv							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	93.53%	86.45%	98.32%	87.26%	78.04%	95.68%
150	S2	94.03%	93.44%	94.61%	87.44%	87.92%	87.09%
50	S2	81.66%	85.37%	79.58%	75.96%	80.30%	74.01%
250	S1	94.40%	96.70%	92.89%	88.58%	93.03%	86.13%
150	S1	92.19%	87.30%	97.28%	85.94%	79.59%	91.98%
50	S1	79.66%	78.15%	80.99%	75.13%	73.81%	76.27%

Just like in anomaly detection using Random Forest, here too the best results are achieved using larger segment size, with a drop in performance as we move towards smaller segment sizes. This holds true for both segment labeling methodologies, with higher performance (especially recall) using S1 labeling methodology.

Following shown are the training and validation accuracy and loss curves

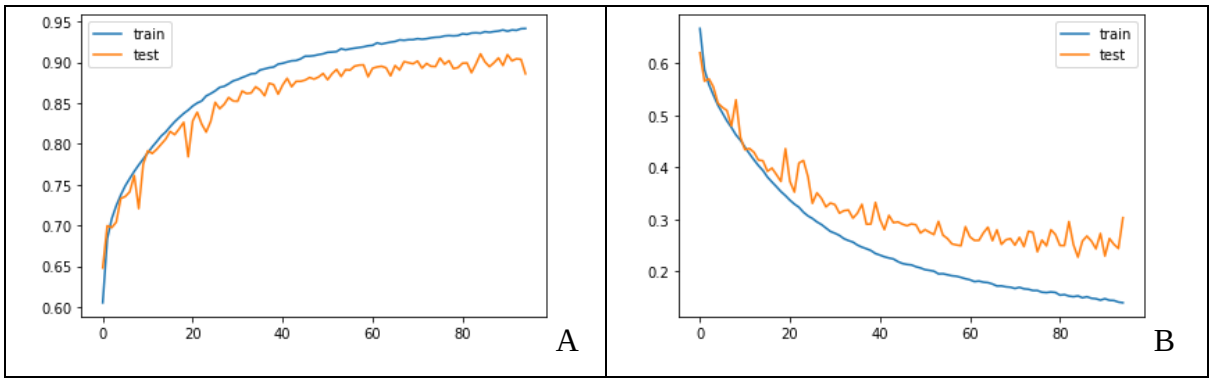


Figure 4.10: Accuracy curve (A) and Loss Curve (B)

Following is the confusion matrix on the best-found pre-processing configurations (segment size of 250 with S1 as labeling methodology), along with the precision recall curves.

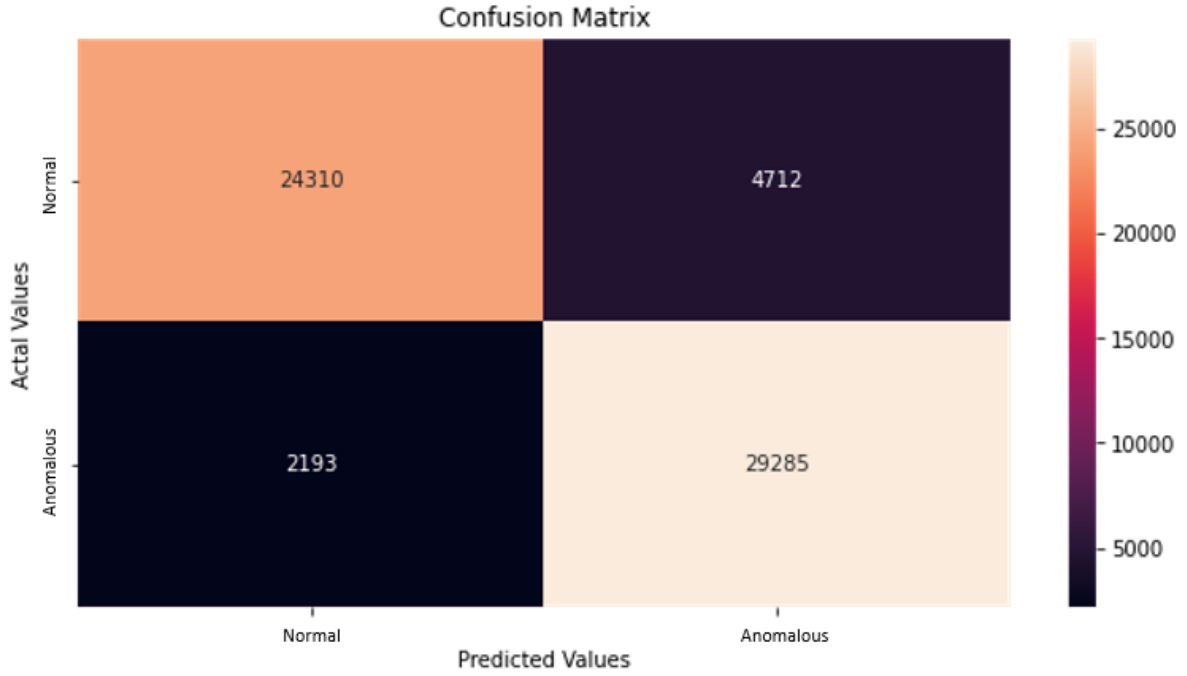


Figure 4.11: Confusion Matrix

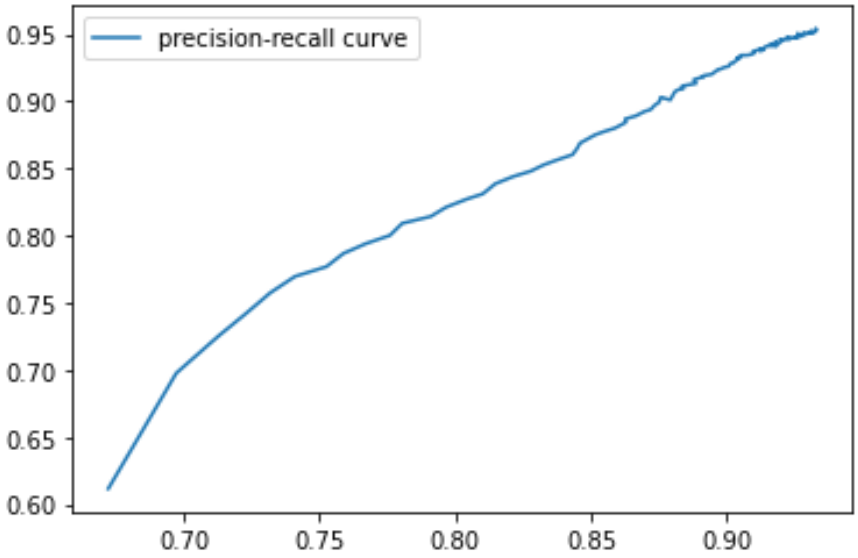


Figure 4.12: Precision Recall curve

4.1.3. Smartwatch Data (Pebble)

In the Levodopa response study, a Pebble smartwatch is used to collect inertial data from a subjects lesser affected upper limb (wrist). Results achieved from this dataset are the least accurate as compared to the other datasets, from the same Levodopa response study.

4.1.3.1. Anomaly detection using Random Forest

For the anomaly detection stage, using machine learning model (Random Forest), we have achieved the most accurate results with a segment size of 250 and with S2 as the segment label.

Here we achieved a train accuracy of 85.99%, a recall of 86.80% and a precision of 88.91%. The same pre-processing configuration produces a test accuracy of 88.34%, with a recall of 87.10% and a precision of 89.30%. Results on rest of the pre-processing configurations are as shown below in Table 4.6

Table 4.6: Results achieved for anomaly detection using Random Forest on Pebble's dataset, with 6 different preprocessing configurations

Pebble							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	85.99%	86.80%	88.91%	86.34%	87.10%	89.30%
150	S2	85.73%	86.26%	88.97%	86.28%	86.69%	89.52%
50	S2	83.24%	83.54%	87.27%	83.72%	83.83%	87.77%
250	S1	85.95%	85.69%	88.93%	86.47%	86.29%	89.44%
150	S1	85.61%	85.17%	89.13%	86.04%	85.62%	89.48%
50	S1	83.24%	83.23%	87.32%	83.72%	83.59%	87.78%

It can be seen in Table 4.6**,** that for anomaly detection using Random Forest, the performance slightly decreases as we move from larger segment sizes towards smaller segment sizes. This is true irrespective of the segment labeling methodology being used.

Following are the top 10 features on the best-found pre-processing configurations (segment size of 250 with S2 as labeling methodology), along with the achieved confusion matrixes (here a 1 represents anomalous class and 0 represents a normal class), and precision recall curves.

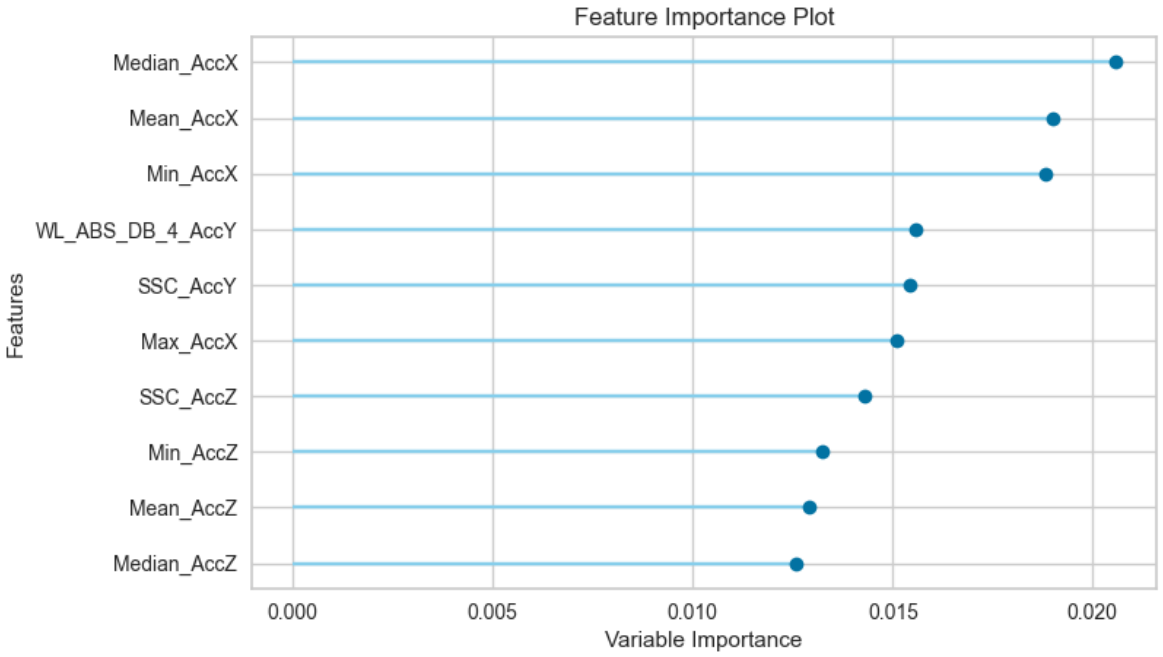


Figure 4.13: Top 10 Features

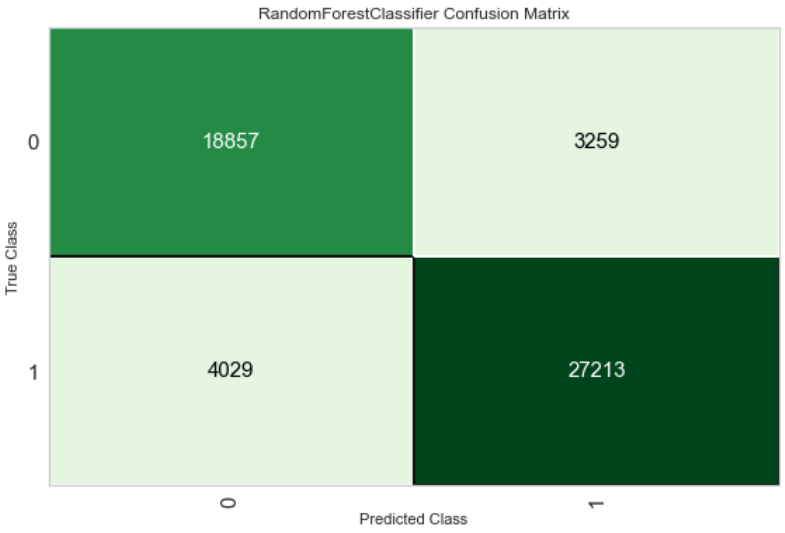


Figure 4.14: Confusion Matrix (Here a 0 represents normal class, while a 1 represents an anomalous class)

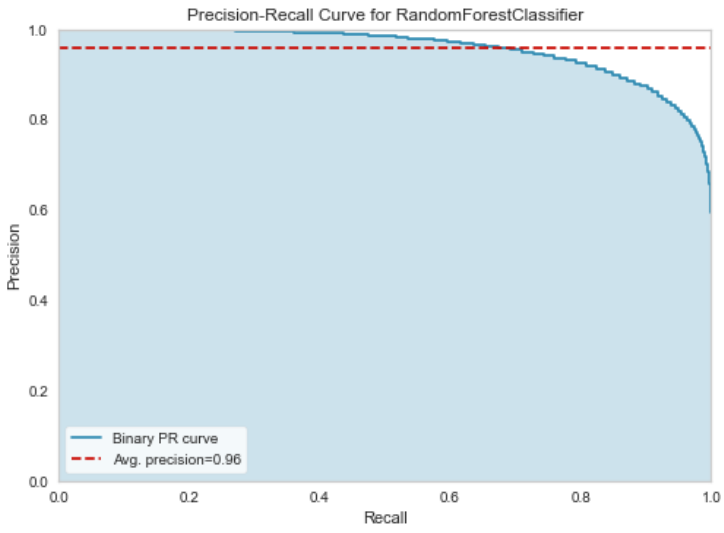


Figure 4.15: Precision Recall Curve

4.1.3.2. Anomaly detection using HARDenseRNN

For anomaly detection using deep learning model (HARDenseRNN), we achieve highest train and test accuracy using a segment size of 250 and S2 segment labeling methodology. On train set, we achieve an accuracy of 93.41%, with a recall of 87.31% and a precision of 96.55%. While on the test set, we achieve an accuracy of 87.54%, with a recall and precision of 80.10%

and 88.49% respectively. Results on rest of the pre-processing configurations are as shown below in Table 4.7**.** Here we have also observed that, although Random Forest and HARDenseRNN performs almost the same, the machine learning model (Random Forest) outperforms the deep learning model (HARDenseRNN), in terms of recall.

Table 4.7: Results achieved for anomaly detection using HARDenseRNN on Pebble's dataset, with 6 different preprocessing configurations

Pebble							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	93.41%	87.31%	96.55%	87.54%	80.10%	88.49%
150	S2	90.54%	79.46%	97.25%	84.28%	69.75%	90.03%
50	S2	77.04%	55.96%	83.23%	73.93%	51.84%	77.85%
250	S1	92.99%	87.78%	95.89%	87.39%	81.22%	88.74%
150	S1	92.67%	89.64%	93.02%	86.31%	82.22%	85.23%
50	S1	78.23%	57.54%	84.56%	74.52%	53.49%	78.84%

Here too, for anomaly detection using HARDenseRNN, better performance is observed with larger segment sizes, and with S2 segment labelling methodology.

Following shown are the training and validation accuracy and loss curves

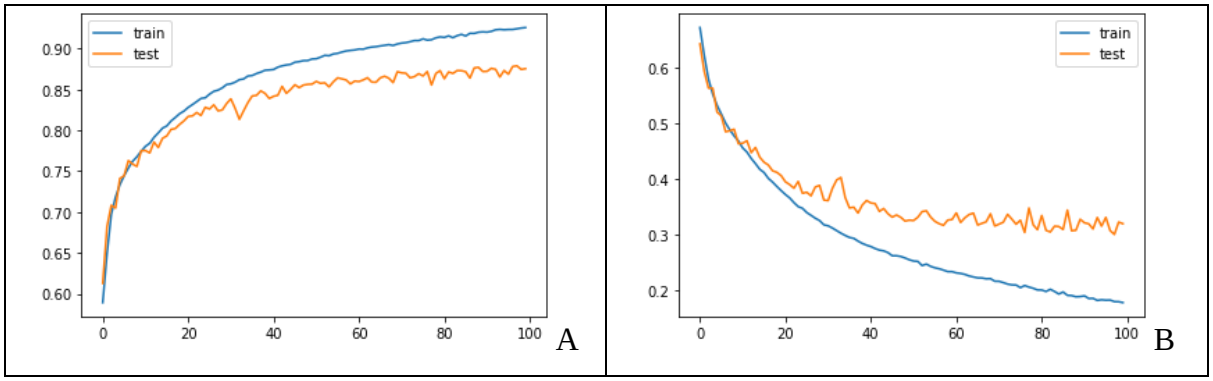


Figure 4.16: Accuracy curve (A) and Loss Curve (B)

Following is the confusion matrix on the best-found pre-processing configurations (segment size of 250 with S2 as labeling methodology), along with the precision recall curves.

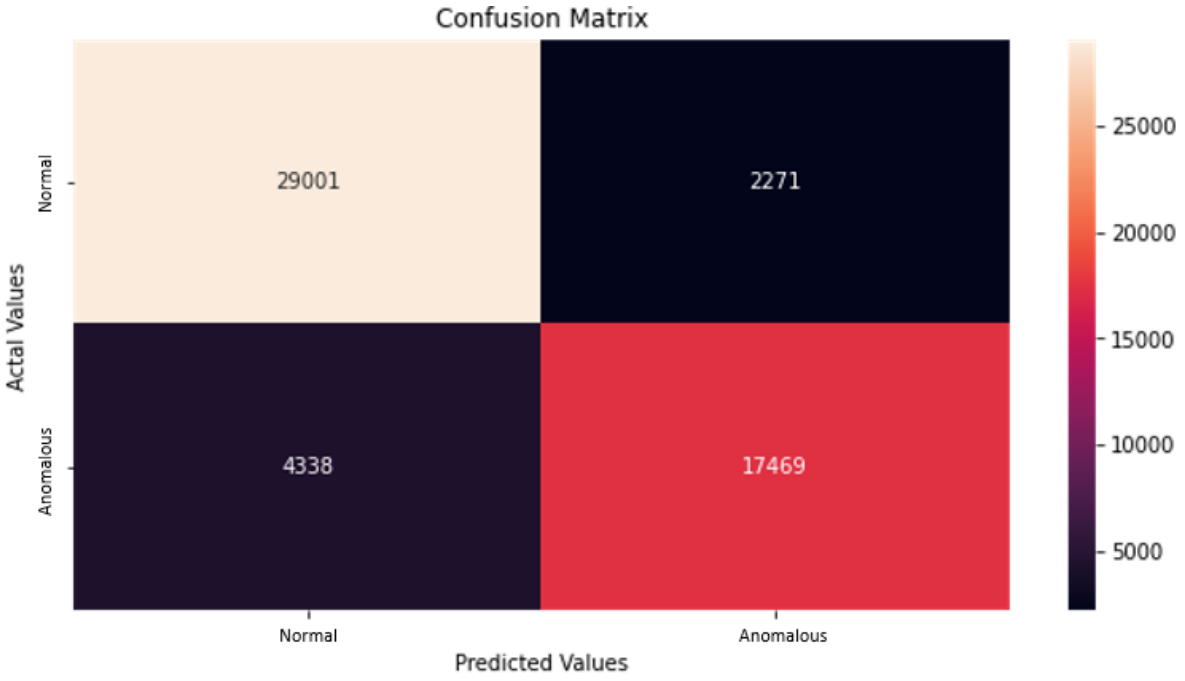


Figure 4.17: Confusion Matrix

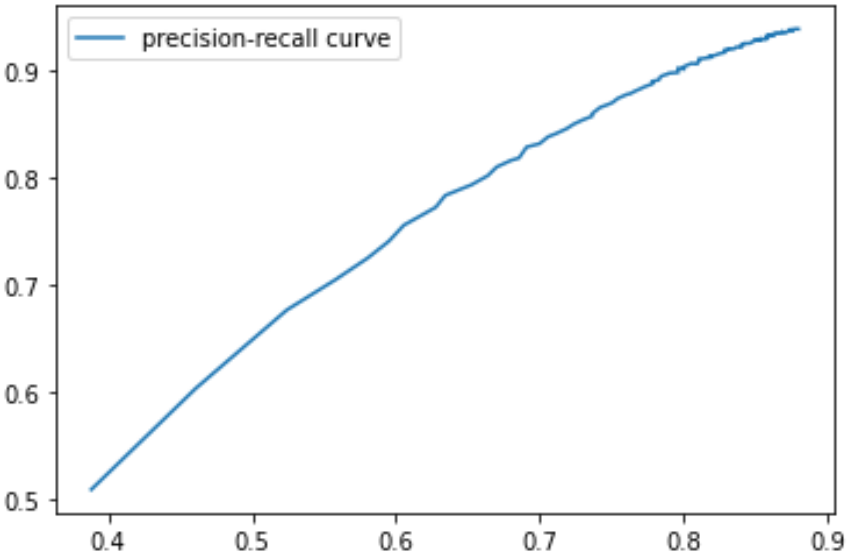


Figure 4.18: Precision & Recall

4.2. Anomaly Classification (Multi-Label Classificatio)

If the first stage classifies the signal as an anomalous one, it is then passed to the second stage of anomaly classification. This is a multi-label classification stage, where the input signal segment is tested for the type/types of anomalies present inside it. Below we present and evaluation of different anomaly detection models, using different datasets, different pre-processing configurations, and using two different modelling approaches i.e., Random Forest and our proposed deep learning network.

4.2.1. Smart Phone Data

Our experiment has shown top performance for models trained using data from the waist (Phone's dataset). This again signifies the fact that human waist exhibits more clearly distinctive patterns and features as compared to upper limbs (wrists), while performing gait tasks[53].

4.2.1.1. Anomaly classification using Random Forest

For the second stage of anomaly classification with Random Forest, we achieve the most accurate results using a segment size of 50 with S2 as the labeling methodology. We achieved an accuracy of 93.38% and 93.84% on train and test sets respectively, with a recall of 79.65% and 79.39% on train and test sets respectively, and with a precision of 93.48% and 93.91% on train and test sets respectively. Results on rest of the pre-processing configurations are as shown below in Table 4.8

Table 4.8: Results achieved for anomaly classification, using Random Forest on phone's dataset, with 6 different preprocessing configurations

Phone							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	91.63%	72.94%	91.93%	91.99%	73.10%	92.26%
150	S2	92.33%	75.83%	92.60%	92.82%	75.85%	93.06%
50	S2	93.38%	79.65%	93.48%	93.84%	79.39%	93.91%
250	S1	90.96%	56.25%	91.29%	91.34%	50.05%	91.59%
150	S1	91.69%	55.70%	91.95%	92.40%	61.51%	92.64%
50	S1	93.41%	60.90%	93.51%	93.54%	65.26%	93.65%

It can be seen in Table 4.8**,** that for anomaly classification using Random Forest, the best results are achieved using smaller segment sizes as compared with the larger ones. Also, using S2 as the segment labeling method produces much better recall, while producing almost the same accuracy and precision.

Following are the top 10 features on the best-found pre-processing configurations (segment size of 50 with S2 as labeling methodology), along with the achieved confusion matrixes (here the class labels are from 0 to 6 depending upon which anomalies are present), and precision recall curves.

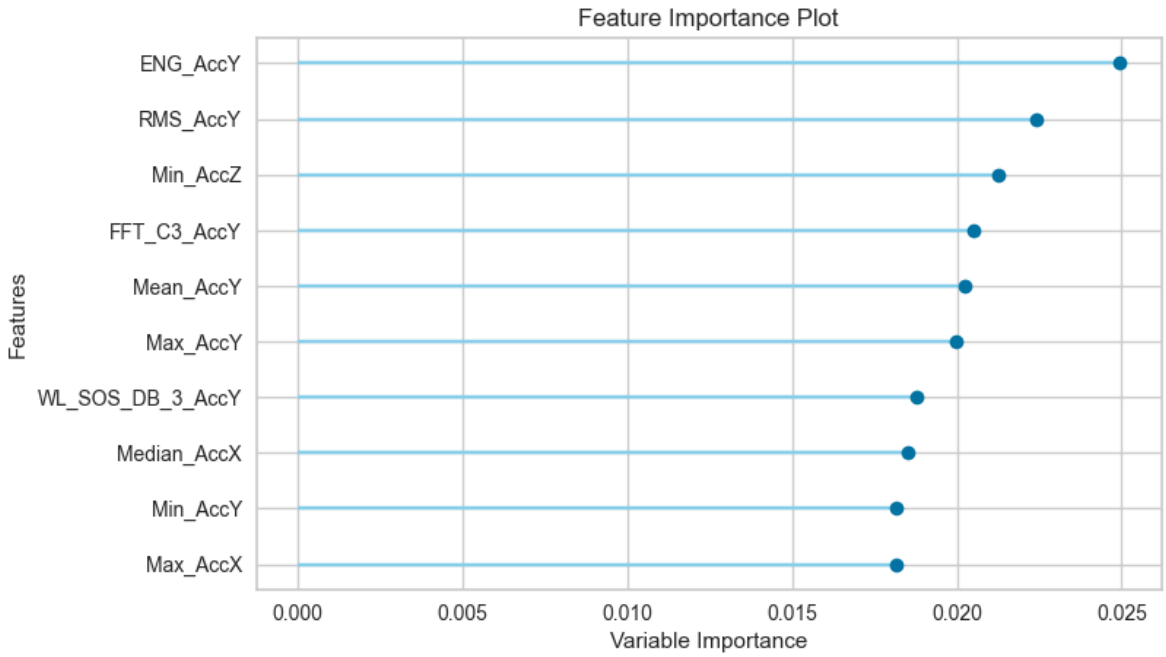


Figure 4.19: Top 10 features

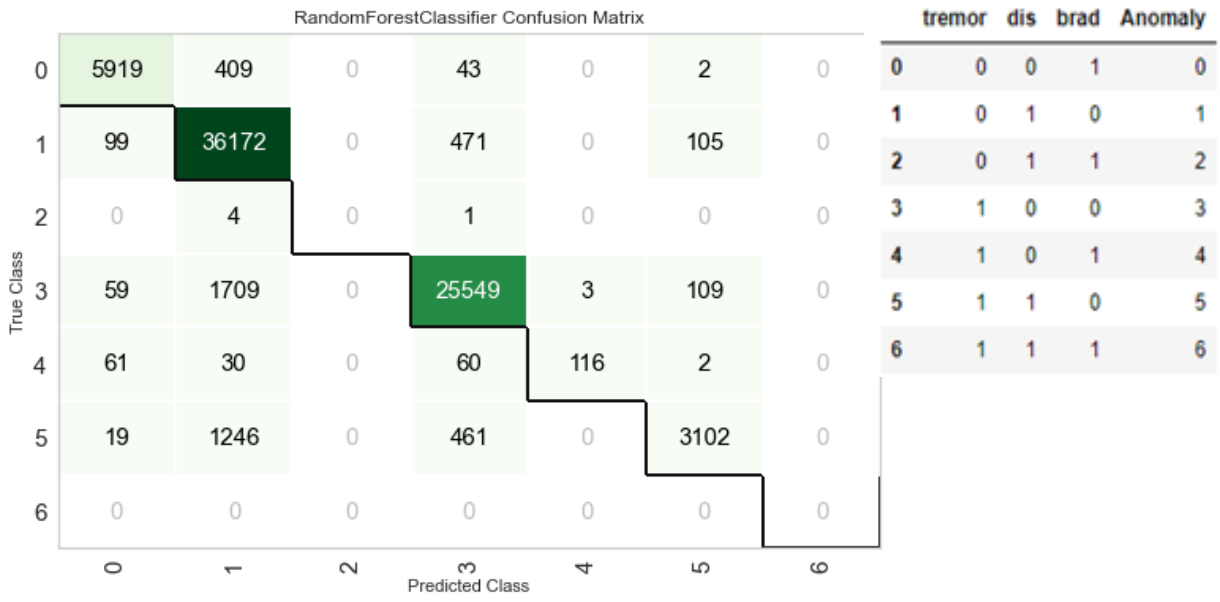


Figure 4.20: Confusion matrix for multi-label classification using label powerset approach.

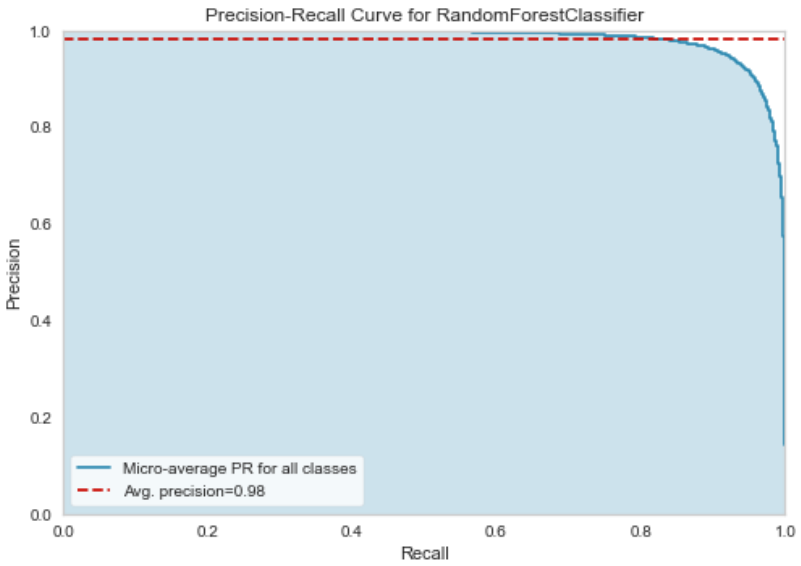


Figure 4.21: Precision Recall curve

4.2.1.2. Anomaly classification using HARDenseRNN

For anomaly classification, deep learning model (HARDenseRNN) outperforms machine learning model (Random Forest), with approximately the same accuracy and precision, but with significant improvement in recall. Here we achieve the most accurate results using a segment size of 150 with S2 as the labeling methodology. We achieved an accuracy of 96.19% and 91.37% on train and test sets respectively, with a recall of 95.97% and 95.14% on train and test sets respectively, and with a precision of 96.67% and 91.85% on train and test sets respectively. Results on rest of the pre-processing configurations are as shown below in Table 4.9

Table 4.9: Results achieved for anomaly classification using HARDenseRNN on phone's dataset, with 6 different preprocessing configurations

Phone							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	93.91%	95.54%	95.96%	90.14%	91.25%	91.62%
150	S2	96.19%	95.97%	96.67%	91.37%	91.54%	91.85%
50	S2	89.86%	91.65%	89.33%	84.30%	86.67%	84.29%
250	S1	95.87%	97.33%	96.73%	91.56%	93.11%	91.70%
150	S1	95.14%	95.94%	96.29%	90.29%	91.79%	90.56%
50	S1	88.91%	90.43%	89.42%	83.78%	85.66%	84.48%

It can be seen in Table 4.9**,** that for anomaly classification using HARDenseRNN, smaller segment size product the least accurate results, irrespective of segment labeling methodology being used.

Following shown are the training and validation accuracy and loss curves

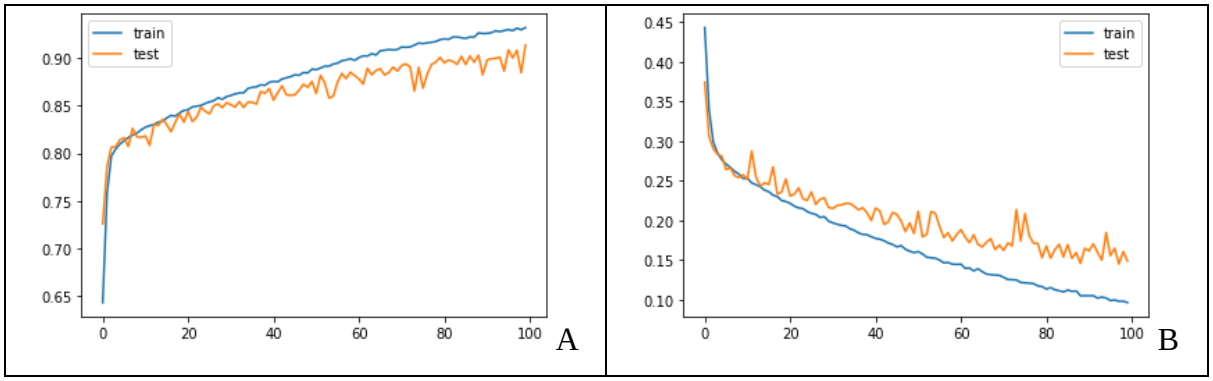


Figure 4.22: Accuracy curve (A) and Loss Curve (B)

Following are the confusion matrixes on the best-found pre-processing configurations (segment size of 150 with S2 as labeling methodology), along with the precision recall curves.

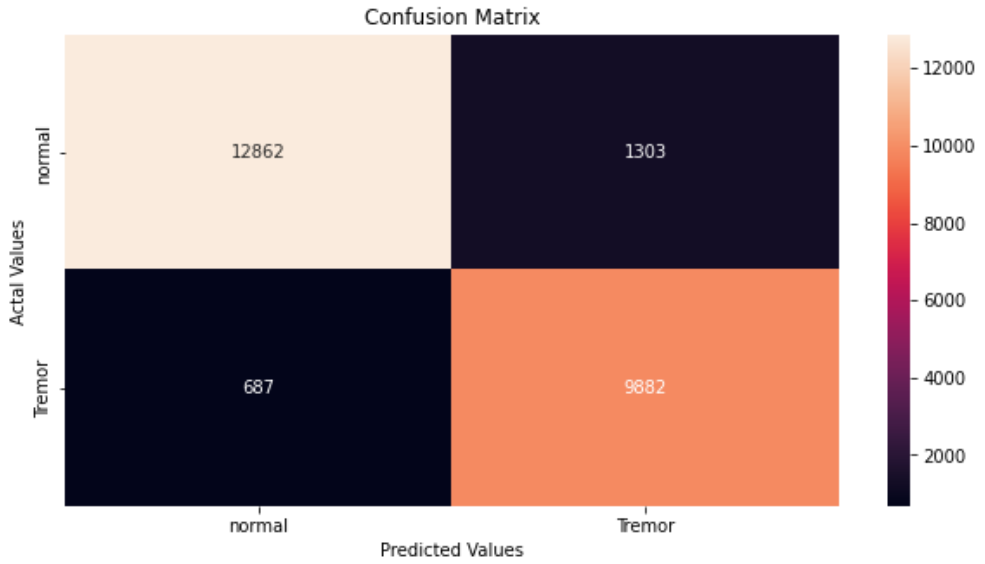


Figure 4.23: Confusion Matrix for Tremors

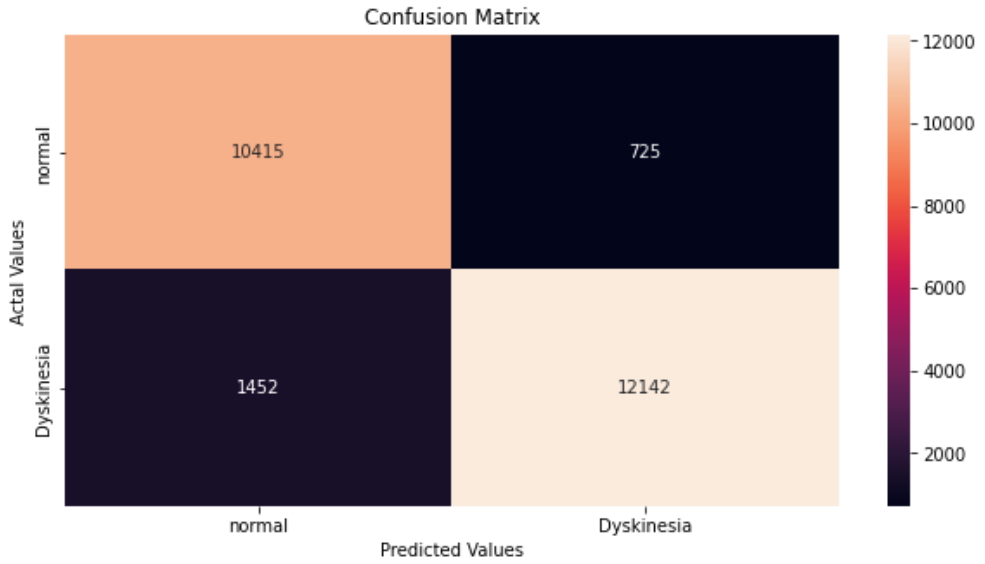


Figure 4.24: Confusion Matrix for Dyskinesia

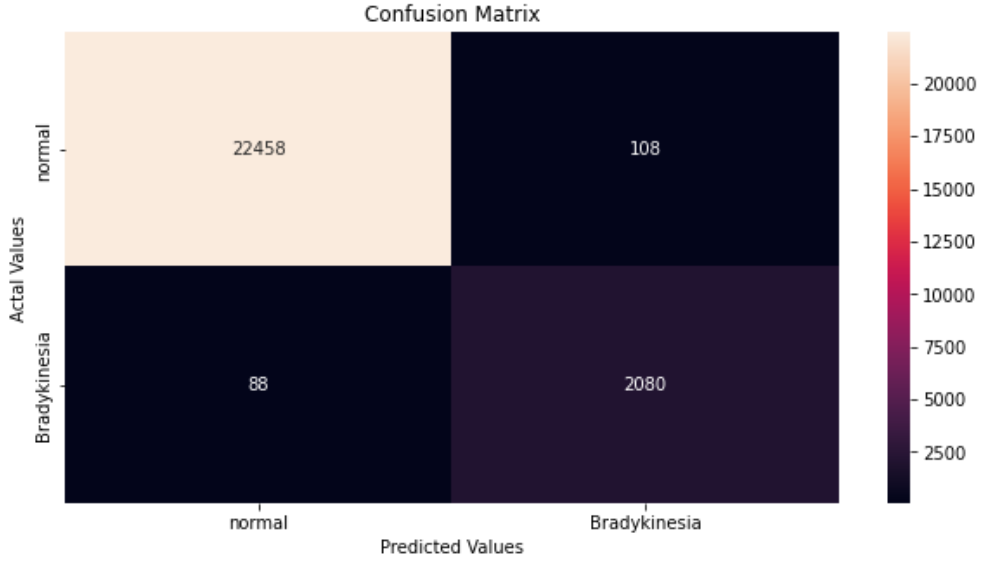


Figure 4.25: Confusion Matrix for Bradkinesia

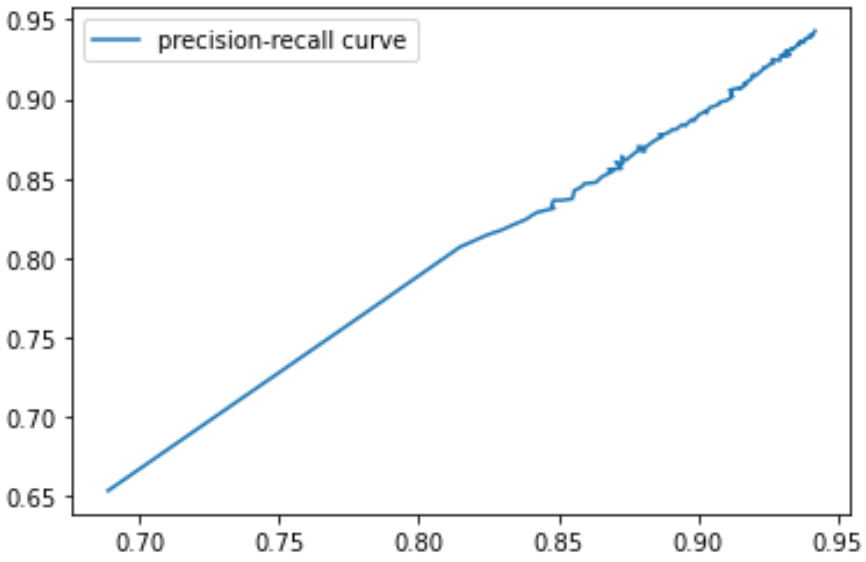


Figure 4.26: Precision Recall curve

4.2.2. Inertial Measurement Unit (IMU) Data (GeneActiv)

Our experiments show that models trained using data from most affected upper limb, outperforms models trained on the data from other upper limb.

4.2.2.1. Anomaly classification using Random Forest

For anomaly classification with Random Forest, we achieve the most accurate results using a segment size of 250 with S2 as the labeling methodology. We achieved an accuracy of 83.59% and 83.98% on train and test sets respectively, with a recall of 71.34% and 71.23% on train and test sets respectively, and with a precision of 84.60% and 85.05% on train and test sets respectively. Results on rest of the pre-processing configurations are as shown below in Table 4.10**.**

Table 4.10: Results achieved for anomaly classification using Random Forest on GeneActiv's dataset, with 6 different preprocessing configurations

GeneActiv							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	83.59%	71.34%	84.60%	83.98%	71.23%	85.05%
150	S2	82.91%	71.01%	84.00%	83.20%	71.99%	84.32%
50	S2	79.88%	65.83%	80.89%	80.31%	66.90%	81.27%
250	S1	81.83%	65.89%	83.06%	82.88%	67.72%	84.01%
150	S1	81.67%	66.86%	82.88%	82.08%	67.63%	83.21%
50	S1	79.50%	64.65%	80.56%	80.28%	65.98%	81.28%

It can be seen in Table 4.10**,** that for anomaly classification using Random Forest, the best results are achieved using larger segment size, with a significant drop in performance as we move towards smaller segment sizes. This holds true for both segment labeling methodologies, with higher performance using S2 labeling methodology.

Following are the top 10 features on the best-found pre-processing configurations (segment size of 250 with S2 as labeling methodology), along with the achieved confusion matrixes (here the class labels are from 0 to 6 depending upon which anomalies are present), and precision recall curves.

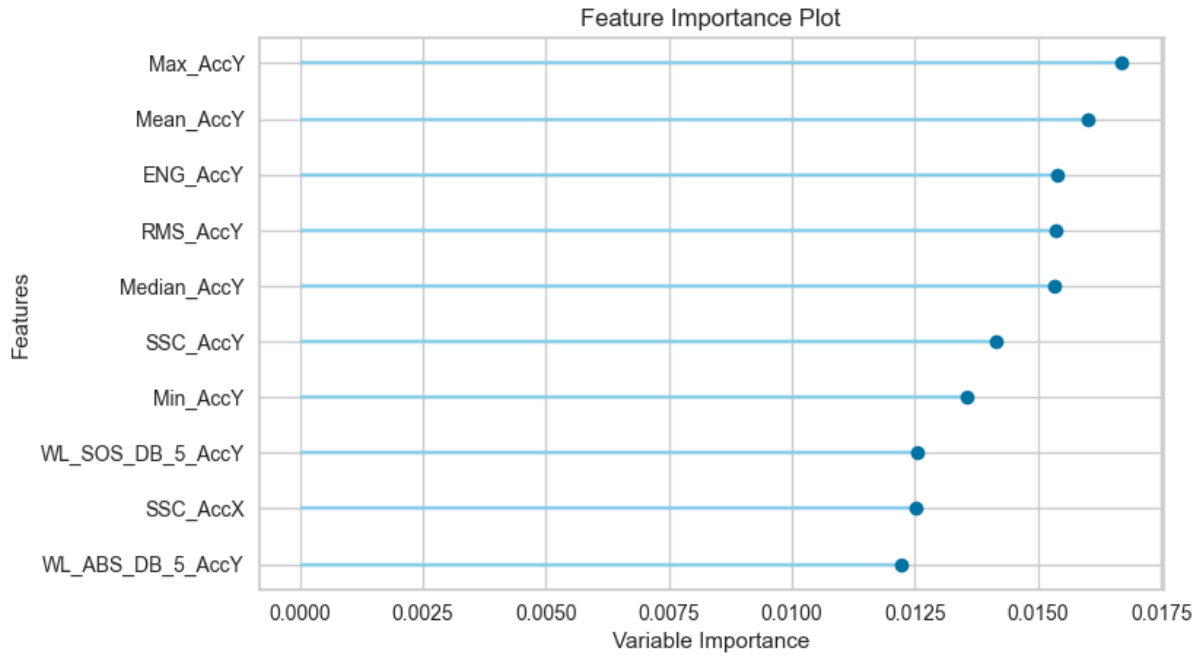
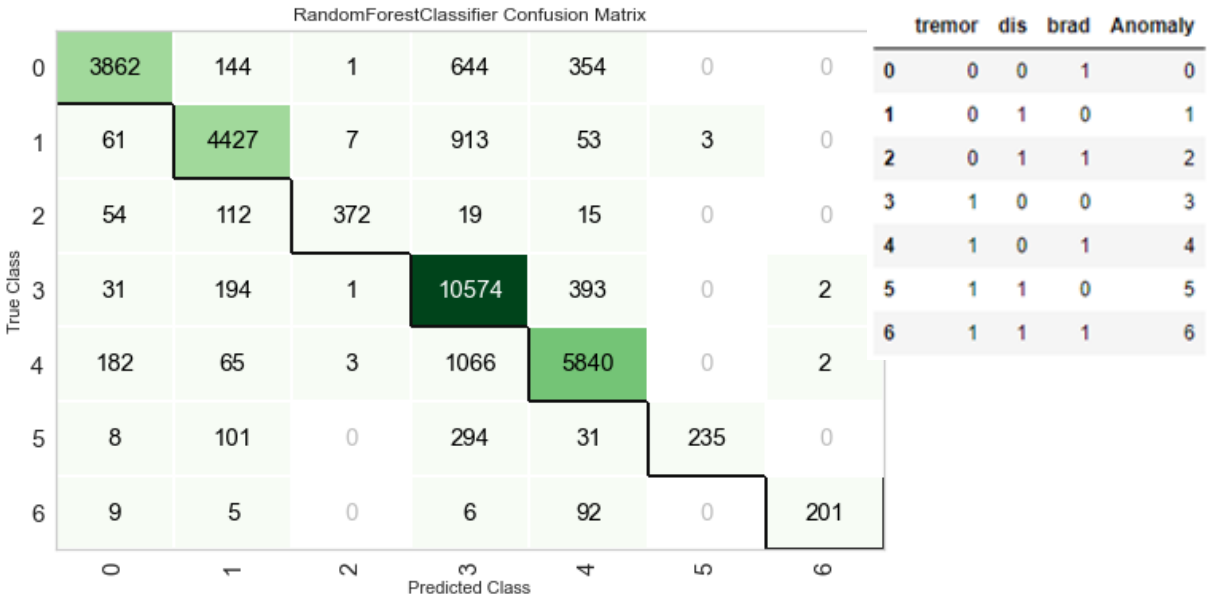


Figure 4.27: Top 10 Feature



	tremor	dis	brad	Anomaly
0	0	0	1	0
1	0	1	0	1
2	0	1	1	2
3	1	0	0	3
4	1	0	1	4
5	1	1	0	5
6	1	1	1	6

Figure 4.28: Confusion Matrix for Multi-Label classification

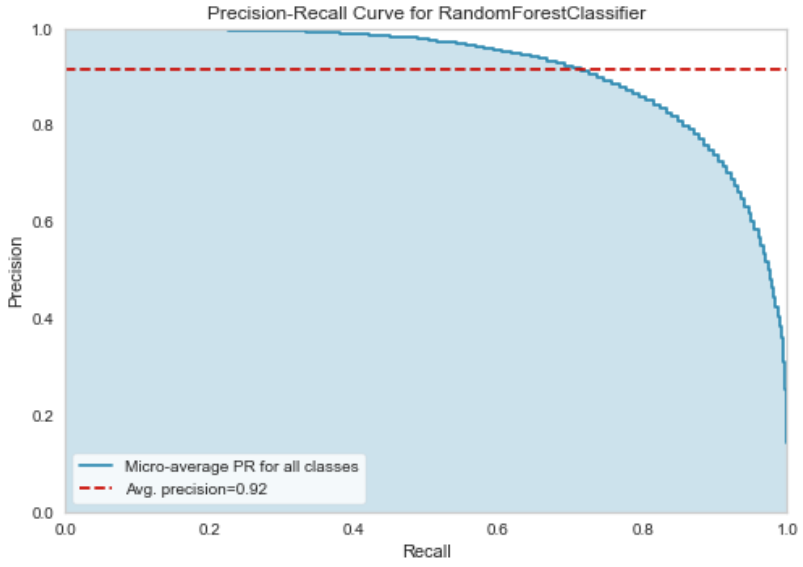


Figure 4.29: Precision Recall curve

4.2.2.2. Anomaly classification using HARDenseRNN

Deep learning model (HARDenseRNN) significantly outperforms machine learning model (Random Forest), in the task of anomaly classification. Here we achieve the most accurate results using a segment size of 250 with S2 as the labeling methodology. We achieved an accuracy of 90.57% and 85.97% on train and test sets respectively, with a recall of 95.41% and 90.54% on train and test sets respectively, and with a precision of 96.36% and 91.55% on train

and test sets respectively. Results on rest of the pre-processing configurations are as shown below in Table 4.11

Table 4.11: Results achieved for anomaly classification using HARDenseRNN on GeneActiv dataset, with 6 different preprocessing configurations

GeneActiv							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	90.57%	95.41%	96.36%	85.97%	90.54%	91.55%
150	S2	80.66%	95.29%	94.88%	76.51%	89.82%	89.02%
50	S2	79.97%	85.05%	87.43%	74.26%	79.13%	81.72%
250	S1	82.65%	96.90%	97.57%	79.71%	91.54%	91.95%
150	S1	85.02%	93.97%	94.95%	80.14%	88.41%	89.05%
50	S1	78.65%	86.00%	87.14%	73.03%	80.33%	81.58%

It can be seen in Table 4.11**,** that for anomaly classification using HARDenseRNN, the best results are achieved using larger segment size (250) with S2 segment labeling methodology. A signification drop is observed while using smaller segment sizes and with S1 labeling methodology.

Following shown are the training and validation accuracy and loss curves

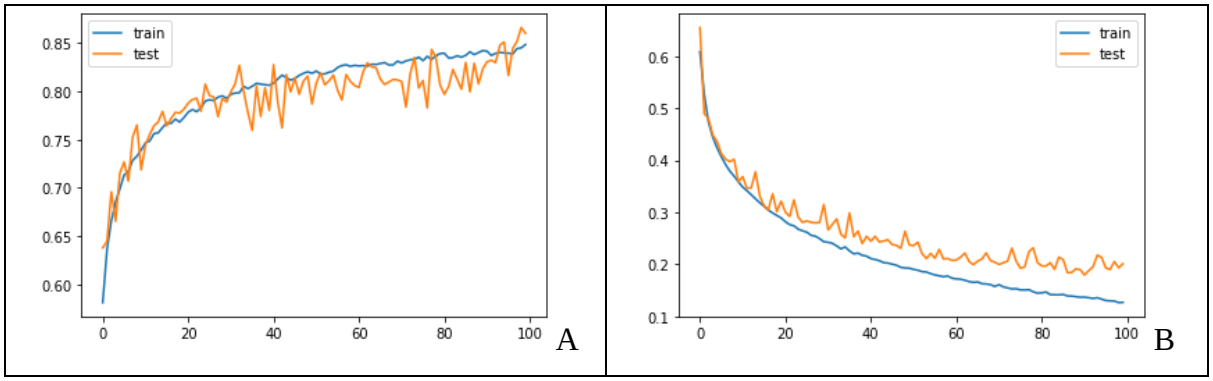


Figure 4.30: Accuracy curve (A) and Loss Curve (B)

Following is the confusion matrix on the best-found pre-processing configurations (segment size of 250 with S2 as labeling methodology), along with the precision recall curves.

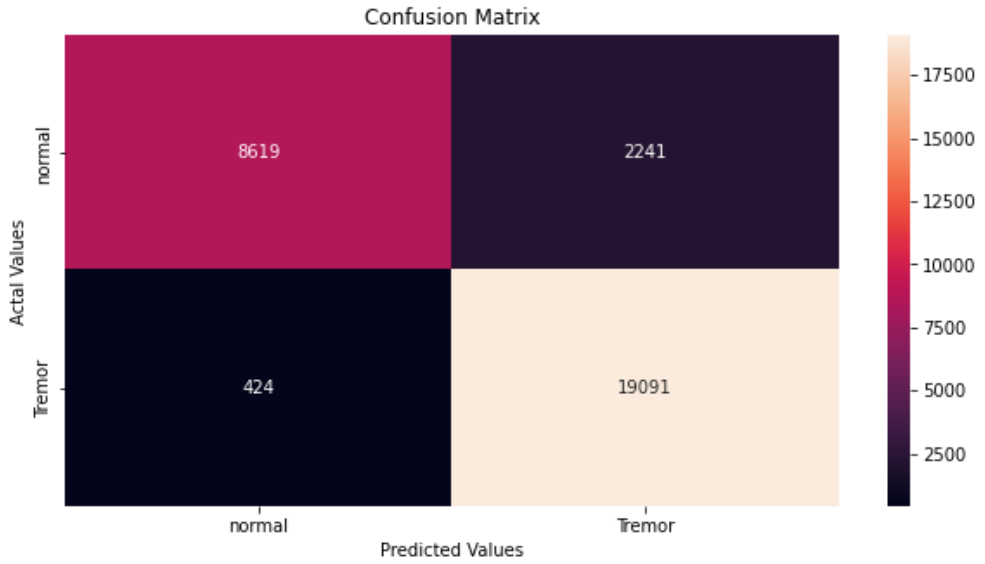


Figure 4.31: Confusion matrix for Tremors

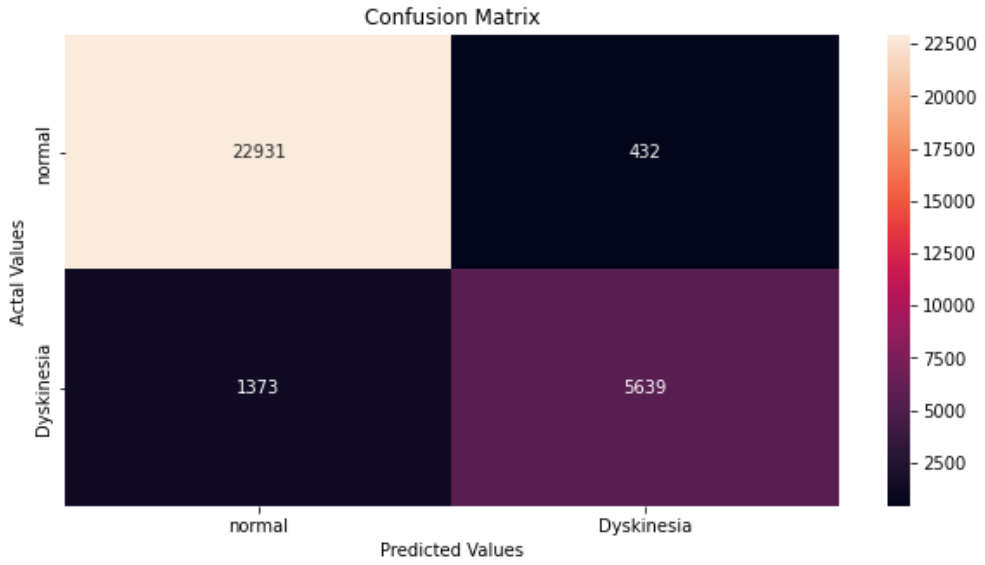


Figure 4.32: Confusion matrix for Dyskinesia.

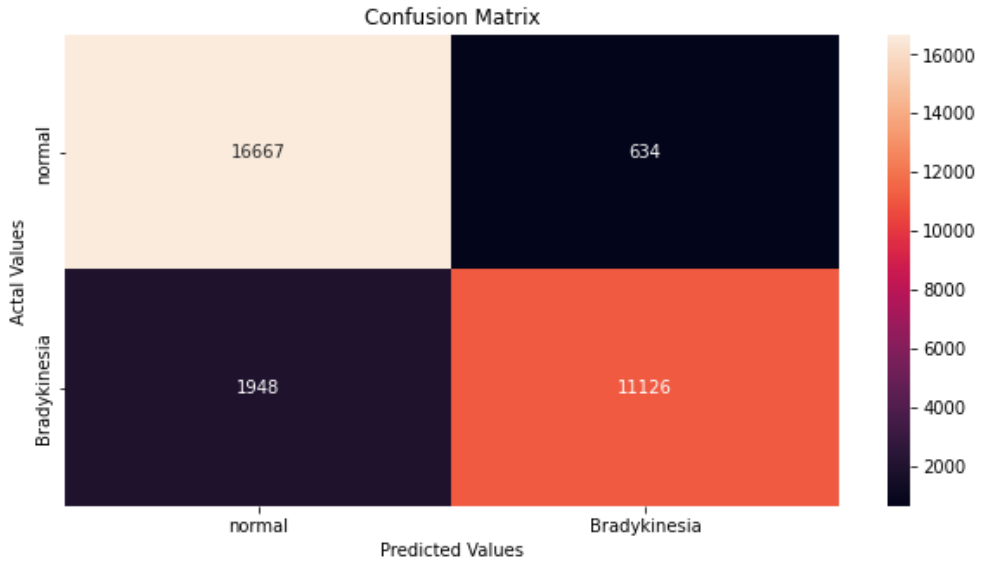


Figure 4.33: Confusion Matrix for Bradykinesia

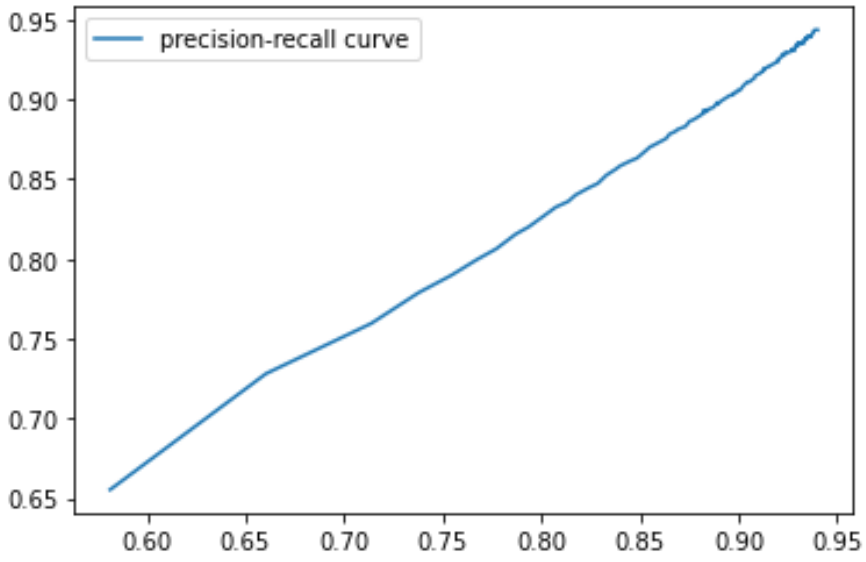


Figure 4.34: Precision Recall curve

4.2.3. Smartwatch Data (Pebble)

Our experiments shows that models trained on data from the least affected upper limb, exhibits the lowest performance.

4.2.3.1. Anomaly classification using Random Forest

For anomaly classification with Random Forest, we achieve the most accurate results using a segment size of 250 with S2 as the labeling methodology. We achieved an accuracy of 83.92% and 85.33% on train and test sets respectively, with a recall of 62.23% and 64.08% on train and test sets respectively, and with a precision of 84.98% and 86.25% on train and test sets

respectively. Results on rest of the pre-processing configurations are as shown below in Table 4.12

Table 4.12: Results achieved for anomaly classification using Random Forest on Pebble's dataset, with 6 different preprocessing configurations

Pebble							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	83.92%	62.23%	84.98%	85.33%	64.08%	86.25%
150	S2	83.65%	61.57%	84.74%	84.48%	63.31%	85.40%
50	S2	81.36%	56.71%	82.23%	81.91%	57.35%	82.71%
250	S1	82.13%	57.63%	83.08%	83.09%	59.19%	83.94%
150	S1	82.18%	57.87%	83.27%	83.04%	58.47%	84.11%
50	S1	80.75%	55.21%	81.64%	81.46%	56.51%	82.26%

As we can see, the highest performance for anomaly classification using machine learning (Random Forest) is achieved with the largest segment size and S2 as the segment labeling methodology.

Following are the top 10 features on the best-found pre-processing configurations (segment size of 250 with S2 as labeling methodology), along with the achieved confusion matrixes (here the class labels are from 0 to 6 depending upon which anomalies are present), and precision recall curves.

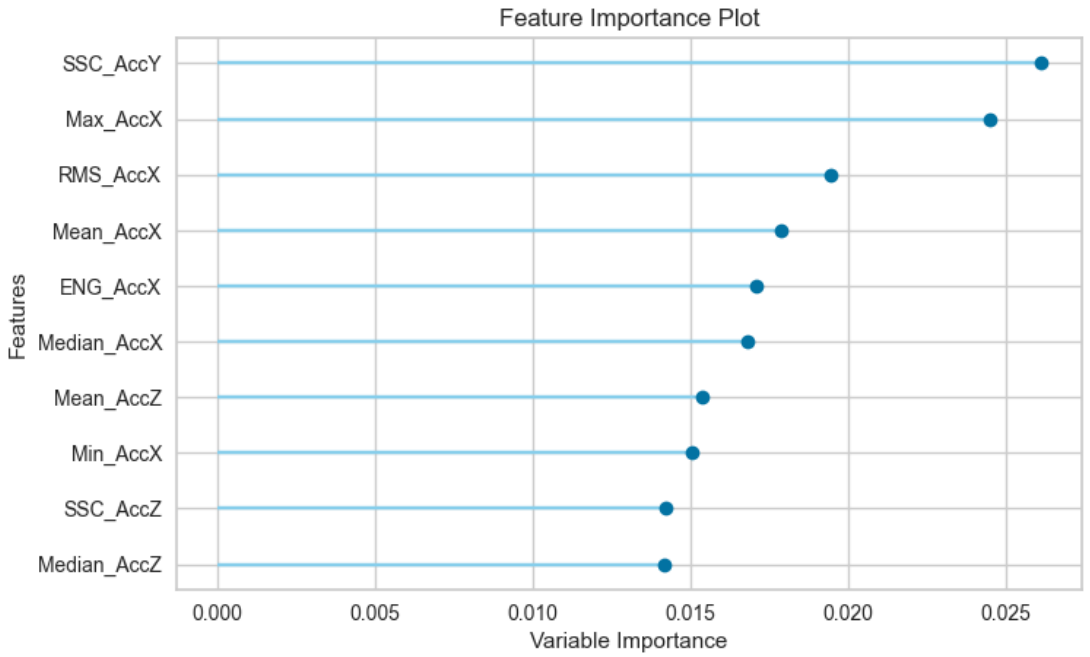
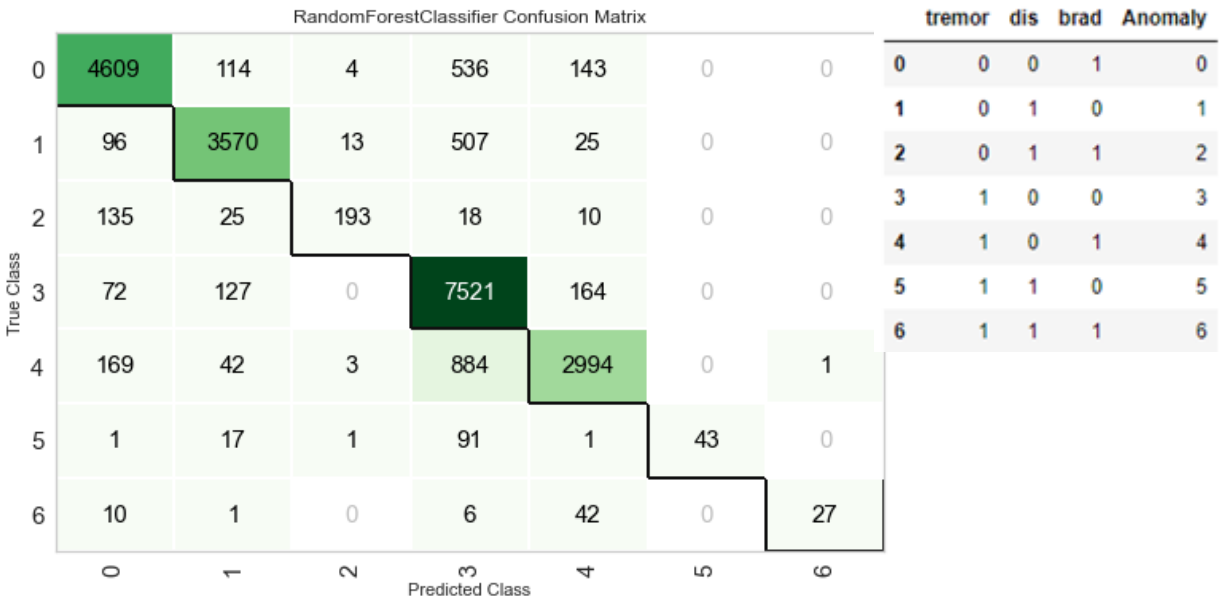


Figure 4.35: Top 10 features



	tremor	dis	brad	Anomaly
0	0	0	1	0
1	0	1	0	1
2	0	1	1	2
3	1	0	0	3
4	1	0	1	4
5	1	1	0	5
6	1	1	1	6

Figure 4.36: Multi-Label classification confusion matrix

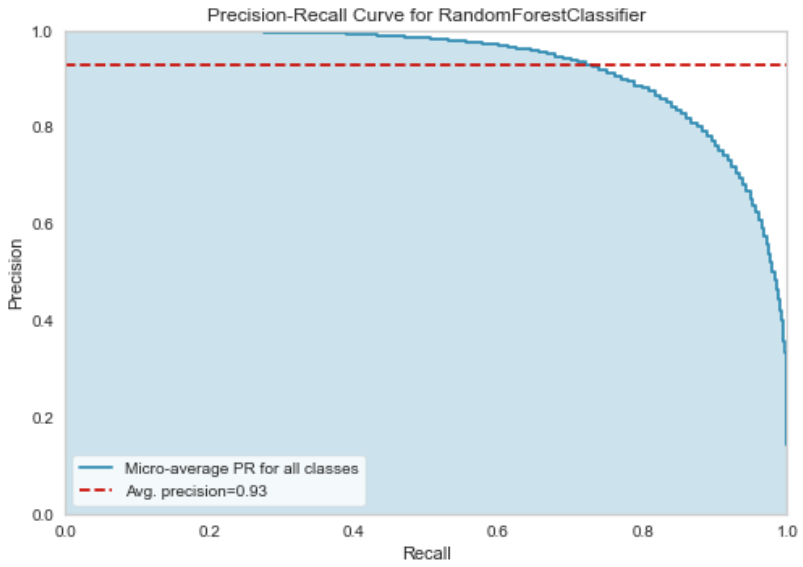


Figure 4.37: Precision and Recall curve

4.2.3.2. Anomaly classification using HARDenseRNN

For Anomaly classification, deep learning model (HARDenseRNN) significantly outperforms Random Forest in anomaly classification. Here we achieve the most accurate results using a segment size of 250 with S2 as the labeling methodology. We achieved an accuracy of 89.59% and 84.49 % on train and test sets respectively, with a recall of 97.71% and 91.90% on train and test sets respectively, and with a precision of 98.22% and 91.93% on train and test sets

respectively. Results on rest of the pre-processing configurations are as shown below in Table 4.13

Table 4.13: Results achieved for anomaly classification using HARDenseRNN on Pebble's dataset, with 6 different preprocessing configurations.

Pebble							
Segment Size	Segment Labeling	On Train Dataset			On Test Dataset		
		Accuracy	Recall	Precision	Accuracy	Recall	Precision
250	S2	89.59%	97.71%	98.22%	84.49%	91.90%	91.93%
150	S2	84.86%	96.91%	95.80%	80.04%	90.79%	88.51%
50	S2	80.36%	83.41%	90.29%	73.73%	76.29%	83.13%
250	S1	84.01%	96.48%	98.09%	79.76%	89.71%	92.54%
150	S1	85.29%	95.61%	95.96%	79.82%	89.29%	89.07%
50	S1	77.10%	80.21%	88.18%	71.32%	73.82%	81.70%

As we can see, the highest performance for anomaly classification using deep learning (HARDenseRNN) is achieved with the largest segment size and S2 as the segment labeling methodology. A drop in performance is observed as we move towards smaller window sizes and towards S1.

Following shown are the training and validation accuracy and loss curves

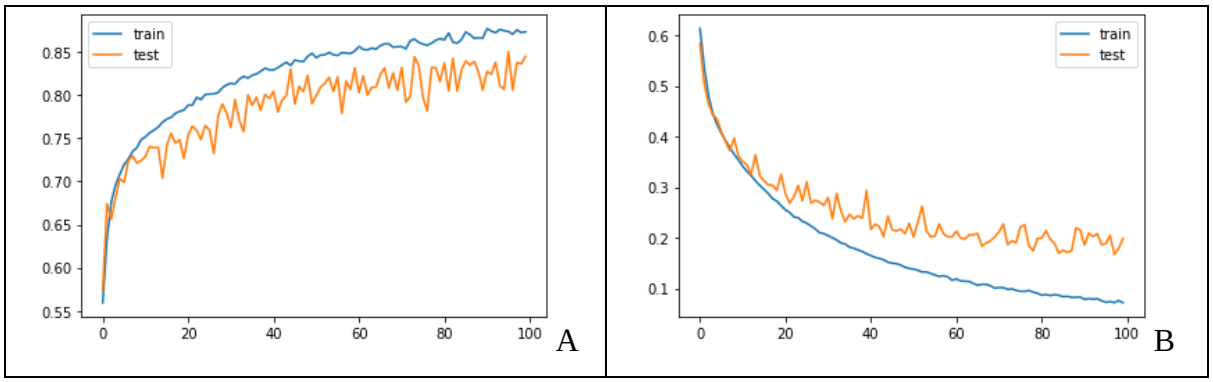


Figure 4.38: Accuracy curve (A) and Loss Curve (B)

Following is the confusion matrix on the best-found pre-processing configurations (segment size of 250 with S2 as labeling methodology), along with the precision recall curves.

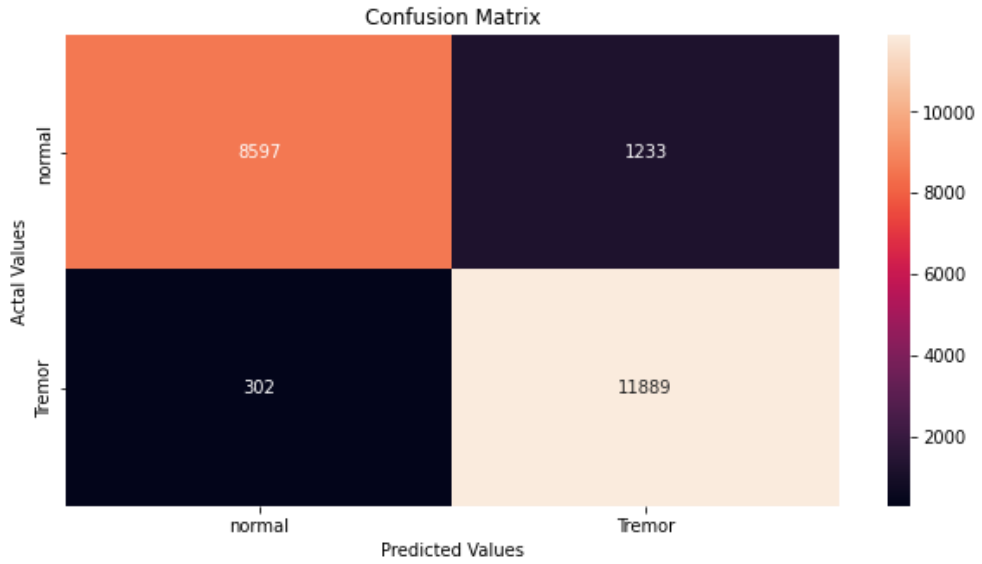


Figure 4.39: Confusion Matrix for Tremor

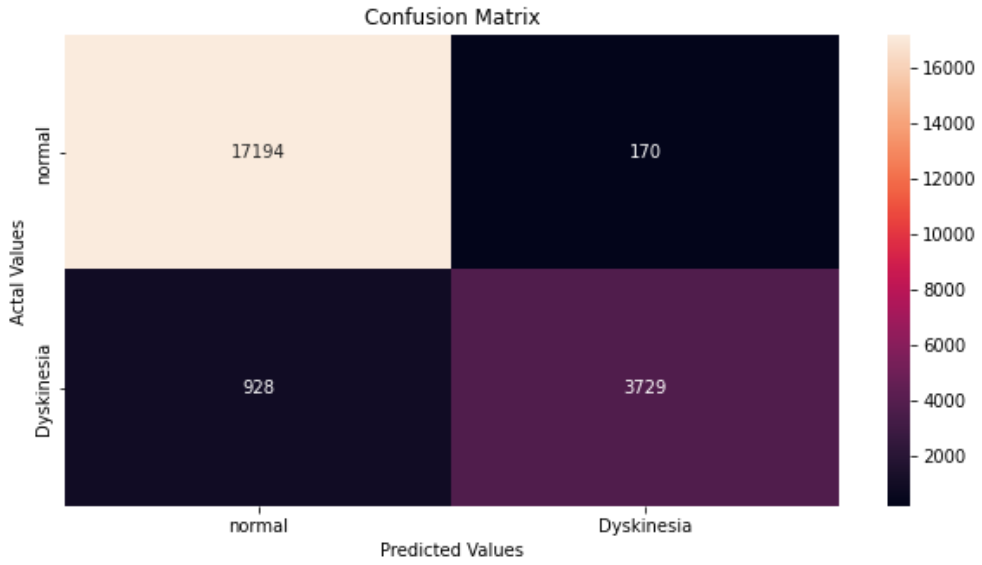


Figure 4.40: Confusion Matrix for Dyskinesia

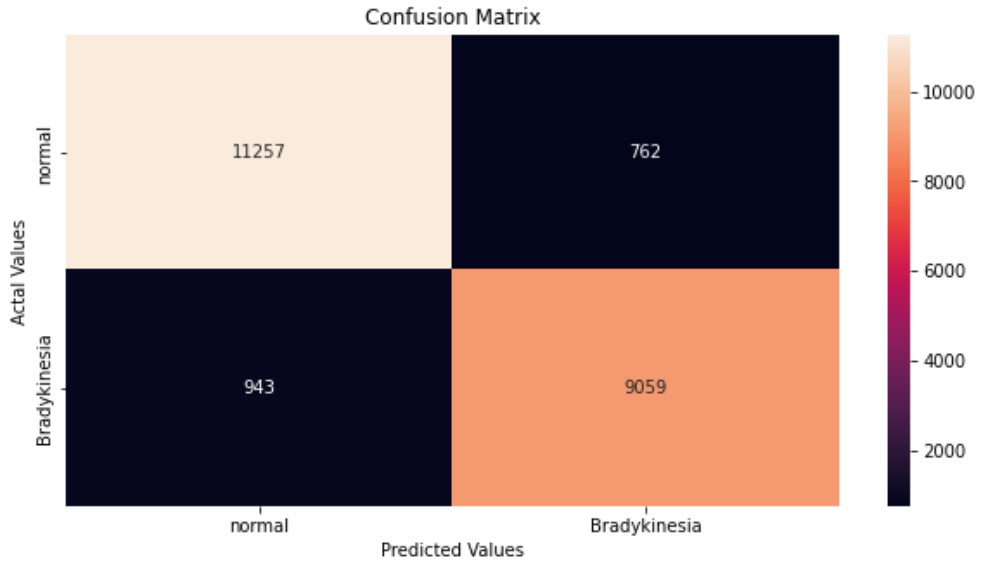


Figure 4.41: Confusion Matrix for Bradykinesia

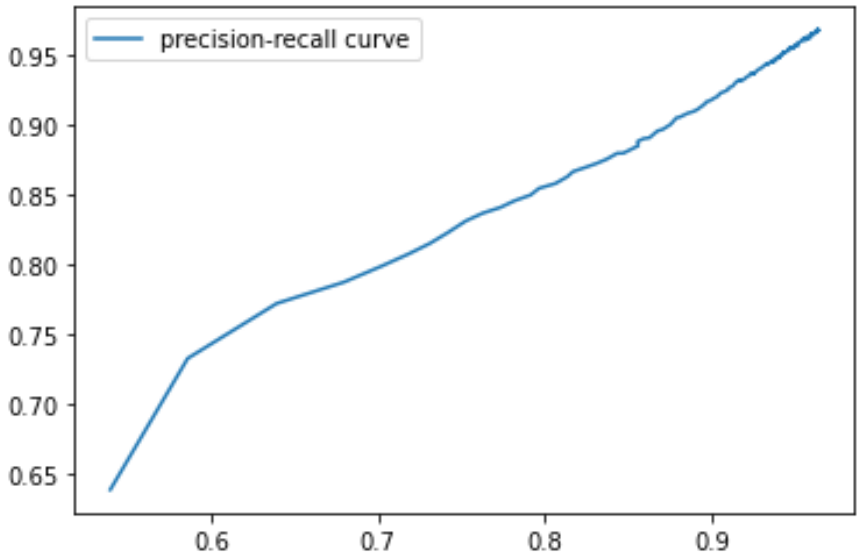


Figure 4.42: Precision and Recall curve

4.3. Computation Cost

We evaluated our models for performance (as shared in above sections) and also for computational cost i.e., complete execution time for a single input segment, and found that deep learning models outperforms the machine learning models in terms of execution time as well. As the input segment size is different for different pre-processing configuration, we have evaluated our models for all 6 preprocessing configurations. Following table shows execution time (seconds) required for various stages of both machine learning and deep learning-based pipelines.

Table 4.14: Execution time of various stages of the Machine Learning Based Pipeline

Machine Learning model (Random Forest) Based Pipeline								
PreProcessing Configuration	Feature Computation		Anomaly Detection		Anomaly Classifier		Total Time	
	All Features	Top 10 Features	All Features	Top 10 Features	All Features	Top 10 Features	All Features	Top 10 Features
F250-S1	0.07	0.01	0.18	0.09	0.19	0.11	0.44	0.21
F150-S1	0.07	0.01					0.44	0.21
F50-S1	0.07	0.01					0.44	0.21
F250-S2	0.07	0.01					0.44	0.21
F150-S2	0.08	0.01					0.45	0.21
F50-S2	0.07	0.01					0.44	0.21

Table 4.15: Execution time of various stages of the Proposed Deep Learning Model Based Pipeline

Proposed Deep Leaning model based pipeline			
PreProcessing Configuration	Model Inference		Total Time
	Anomaly Detection	Anomaly Classifier	
F250-S1	0.09	0.08	0.17
F150-S1	0.09	0.09	0.18
F50-S1	0.08	0.08	0.16
F250-S2	0.09	0.09	0.18
F150-S2	0.07	0.08	0.15
F50-S2	0.08	0.08	0.15

As can be seen, Random Forest based pipeline takes on average 0.44 seconds for complete execution when using all 150 features and takes 0.21 seconds on average for complete execution using top 10 features only. Time used along various stages of pipeline is also mentioned in above table. On the other hand, our proposed deep learning model takes 0.16 seconds, on average, for complete end to end execution. One clear time gap between the two pipelines is of feature computation, but even if we take that out, our proposed model inference time is still much less than that of Random Forest. For Anomaly detection Random Forest takes 0.18 seconds (for the whole feature set), on average, while our proposed deep learning model takes on 0.08 seconds, on average i.e., our proposed model takes 55% lesser time as compared with a machine learning based approach i.e., Random Forest. Even with top 10 features only, Random Forest take 0.14 seconds, on average, which is 42% more than what our proposed deep learning model takes.

For anomaly classification, Random Forest takes 0.19 seconds while using full feature set and takes 0.15 seconds while using only top 10 features, which is 57% and 46% more then what our proposed deep learning model takes, respectively.

Chapter 5

Discussions

This chapter majorly focusses on the findings that we have discovered over the course of this research. In addition, we point out the current limitations of our work.

5.1. Summary of Findings

The target of this research was to develop an optimized methodology to detect and classify anomalies that occurs in patients suffering with Parkinson's.

For our target, we proposed a 2 staged pipeline, where in the first stage we perform anomaly detection and in the second stage we perform anomaly classification. We explore three different datasets, that are acquired from three different sensors i.e., GeneActiv smartwatch, Pebble smartwatch and Samsung S2 smartphone. For each of these sensors we have explored 6 different preprocessing configurations. For each of the sensor, and for each preprocessing configuration we evaluate 2 different models, where one is a machine learning based model (Random Forest) and the other is a deep learning model (HARDenseRNN). On a whole, we have evaluated 72 different models.

We have presented a set of 150 features (time domain, frequency domain, and wavelet domain) for machine learning based pipeline. Top 10 performing features for top performing pre-processing configuration, are also shown in the results section. Depending upon the input sensor being used, and upon the point of body being used for data collection, these top 10 features may vary. However, we found statistical features, wavelet features, and frequency domain features as the common top performing features. This signifies two things, 1) with change in the body point being used for data collection, nature of information captured is changed and thus the top performing features change as well. 2) features from time, frequency, and wavelet domain all are commonly found amongst the top 10 features, which means that we cannot simply rely on a particular domain for feature extraction, when using machine learning based pipelines.

We observed that the best results for both anomaly detection and anomaly classification is achieved on dataset acquired using Samsung S2 smartphone. This signifies using waist as the point of data collection. For anomaly detection using Random Forest, we achieved the highest performance using a segment size of 150 with S1 as the segment labeling methodology. For anomaly detection using HARDenseRNN, we achieved the highest performance using a

segment size of 250 with S2 as the segment labeling methodology. For anomaly detection, we found that although Random Forest outperforms HARDenseRNN on the test dataset, but it takes more time for complete execution. Even with only top 10 features, Random Forest was found to be slower than our proposed deep learning model (as shown in the previous section). Even the size of our proposed deep learning model is only 9MB which is comparatively much smaller than that of a Random Forest, whose size depends upon the number of trees and depth of each tree. These factors make our proposed deep learning model the optimal choice for real time applications and deployment on low-end devices.

Table 5.1: Best performing model for anomaly detection, along with best preprocessing configuration, for Phone's dataset

Phone (Anomaly Detection)								
Model	Configuration		On Train Dataset			On Test Dataset		
	Segment Size	Segment Labeling	Accuracy	Recall	Precision	Accuracy	Recall	Precision
Random Forest	150	S1	93.04%	92.27%	96.49%	93.30%	92.72%	96.48%
HARDenseRNN	250	S2	96.43%	93.57%	96.49%	90.85%	84.70%	89.47%

For anomaly classification using Random Forest, we achieved the highest performance using a segment size of 50 with S2 as the segment labeling methodology. For anomaly classification using HARDenseRNN, we achieved the highest performance using a segment size of 150 with S2 as the segment labeling methodology. For anomaly classification, we found that HARDenseRNN outperforms Random Forest, with approximately the same accuracy and precision, but with significant improvement in recall, with much faster execution time and smaller model size.

Table 5.2: Best performing model for anomaly classification, along with best preprocessing configuration, for Phone's dataset

Phone (Anomaly Classification)								
Model	Configuration		On Train Dataset			On Test Dataset		
	Segment Size	Segment Labeling	Accuracy	Recall	Precision	Accuracy	Recall	Precision
Random Forest	50	S2	93.38%	79.65%	93.48%	93.84%	79.39%	93.91%
HARDenseRNN	150	S2	96.19%	95.97%	96.67%	91.37%	91.54%	91.85%

After phone the best performance is observed with **GeneActiv's** dataset. This signifies the use of more affected upper limb, as compared to the less affected one, for data collection. For anomaly detection using Random Forest, we achieved the highest performance using a segment size of 250 with S2 as the segment labeling methodology. For anomaly detection using HARDenseRNN, we achieved the highest performance using a segment size of 250 with S1 as

the segment labeling methodology. For anomaly detection, we found that HARDenseRNN model performs almost in a similar manner as compared with Random Forest, with increase recall, and slightly decreased precision.

Table 5.3: Best performing model for anomaly detection, along with best preprocessing configuration, for GeneActiv's dataset

GeneActiv (Anomaly Detection)								
Model	Configuration		On Train Dataset			On Test Dataset		
	Segment Size	Segment Labeling	Accuracy	Recall	Precision	Accuracy	Recall	Precision
Random Forest	250	S2	88.11%	88.57%	87.63%	88.63%	89.10%	88.27%
HARDenseRNN	250	S1	94.40%	96.70%	92.89%	88.58%	93.03%	86.13%

For anomaly classification using Random Forest, we achieved the highest performance using a segment size of 250 with S2 as the segment labeling methodology. For anomaly classification using HARDenseRNN, we achieved the highest performance using a segment size of 250 with S2 as the segment labeling methodology. For anomaly classification, we found that Deep learning model (HARDenseRNN) significantly outperforms machine learning model (Random Forest), in the task of anomaly classification.

Table 5.4: Best performing model for anomaly classification, along with best preprocessing configuration, for GeneActiv's dataset

GeneActiv (Anomaly Classification)								
Model	Configuration		On Train Dataset			On Test Dataset		
	Segment Size	Segment Labeling	Accuracy	Recall	Precision	Accuracy	Recall	Precision
Random Forest	250	S2	83.59%	71.34%	84.60%	83.98%	71.23%	85.05%
HARDenseRNN	250	S2	90.57%	95.41%	96.36%	85.97%	90.54%	91.55%

Lastly, we have the Pebble smartwatch dataset. For anomaly detection using Random Forest, we achieved the highest performance using a segment size of 250 with S2 as the segment labeling methodology. For anomaly detection using HARDenseRNN, we achieved the highest performance using a segment size of 250 with S2 as the segment labeling methodology. For anomaly detection, we found that HARDenseRNN model gives better accuracy, but Random Forest produces better recall and precision.

Table 5.5: Best performing model for anomaly detection, along with best preprocessing configuration, for Pebble's dataset

Pebble (Anomaly Detection)								
Model	Configuration		On Train Dataset			On Test Dataset		
	Segment Size	Segment Labeling	Accuracy	Recall	Precision	Accuracy	Recall	Precision
Random Forest	250	S2	85.99%	86.80%	88.91%	86.34%	87.10%	89.30%
HARDenseRNN	250	S2	93.41%	87.31%	96.55%	87.54%	80.10%	88.49%

For anomaly classification using Random Forest, we achieved the highest performance using a segment size of 250 with S2 as the segment labeling methodology. For anomaly classification using HARDenseRNN, we achieved the highest performance using a segment size of 250 with S2 as the segment labeling methodology. For anomaly classification, we found that HARDenseRNN produces almost the same accuracy as compared with the Random Forest mode but show a significant improvement in recall and precision.

Table 5.6: Best performing model for anomaly classification, along with best preprocessing configuration, for Pebble's dataset

Pebble (Anomaly Classification)								
Model	Configuration		On Train Dataset			On Test Dataset		
	Segment Size	Segment Labeling	Accuracy	Recall	Precision	Accuracy	Recall	Precision
Random Forest	250	S2	83.92%	62.23%	84.98%	85.33%	64.08%	86.25%
HARDenseRNN	250	S2	89.59%	97.71%	98.22%	84.49%	91.90%	91.93%

5.2. Comparison With Existing Approaches

Parkinson's detection and the detection of its symptoms has been a hot topic in research community, and a lot of uncovered ground is still there. Past studies have tried different input modalities like vision, speech, EEG, EMG, and inertial sensors. [26] used convolutional neural networks for hand tremor detection and achieved an accuracy of 97%. [22] uses multi-layer perceptron for detection of bradykinesia and achieves an accuracy of 85%. Our work on the other hand, deals with tremor detection irrespective of the body part they occur on. Also, our work is not only specific to tremors, but also covers dyskinesia and bradykinesia. Even with multiple anomalies, our model achieves 96.67% accuracy for anomaly identification (multi-label classification). Other existing works are also focused toward early detection of Parkinson's, or towards detection of a single anomaly. Modalities like vision and speech have their cons associated with them. Challenges like occlusion may affect vision-based techniques. Speech data can prove to be good for detection of anomalies like dysarthria (inability to control

muscles used in speech), but might prove ineffective for cases where we want to detect a tremor in the arm. To the best of our knowledge, this is the first time a pipeline has been presented which deals with anomaly detection, and anomaly severity quantification using multi-label classification, with dataset collected from IMU's from devices like smartwatches, smartphones, which are common with everyone. We have also presented optimal pre-processing configuration, along with top features for machine learning based pipeline. We have also presented a new deep neural net which is also suitable for deployment in real world applications, even with low end devices.

5.3. Limitations

Currently our work only focuses only on anomaly detection (Binary Classification) and symptoms quantification (Multi-Label Anomaly Classification), and not on severity prediction of a specific anomaly. So, we could quantify a subject's symptom with the highest severity if it contains all 3 anomalies, but currently we have not dealt with quantifying severity of an individual anomaly e.g., how severe was the tremor attack?

This is because in the current dataset, severity quantification of only tremors is given. Even the data distribution across these different severities is greatly skewed. Also, very few numbers of records are available for higher severity tremor classes. This creates a huge class imbalance for tremor severity prediction, and thus we have targeted upon creation of a more suitable dataset, as a part of our future work, and incorporate individual anomaly severity quantification in our proposed pipeline as well.

Chapter 6

Conclusions

This chapter concludes all the research work done under this thesis, by describing the contributions and indicating the direction for future work in the domain of "Parkinson's Anomalous event detection, classification, and severity predictions". First, we will talk about the core contributions of our research and then highlight the possible dimensions of the future work.

6.1. Summary of Research Contributions

In our presented study, we have proposed an optimized approach towards anomalous event detection and classification from inertial data of humans suffering with Parkinson's. To reach the optimal pipeline, we have evaluated 72 different models, across multiple different dimensions. Following are our major research contributions, that have helped us reach an optimized pipeline.

1- We have evaluation 6 different pre-processing configurations and have present the optimal ones. We explored 3 different segment sizes i.e., 50, 150 and 250. Additionally, we explored 2 different segment labeling methodologies i.e., (i) Using segment's mode as the segment label, (ii) treating the segment as an anomalous one if it contains any anomalous portion in it, else treating it as a normal segment.

2- We have presented top 10 features, from 150 total features, for Random Forest model. Using only these top 10 features, instead of using all 150 features, we can move closer to real time predictions, and achieve approximately same accuracies at the same time.

3- We present a newly designed, InceptionResNet and RNN inspired, deep learning network, for anomalous event detection and classification, using only a single raw feature i.e., magnitude of acceleration.

4- We present a performance wise comparative analysis of a famous machine learning model (Random Forest) with deep learning model (HARDenseRNN), for the tasks of anomaly detection and classification.

6.2. Future Work

The possible dimensions that we foresee as the future work of this research, are as follows:

1- The dataset that we have used for our study, is not suitable for individual anomaly severity prediction, as the data is highly skewed, with very low records of high severity anomalous events, creating a huge class imbalance. Identifying the severity of anomaly is very important to generate appropriate alarms and cues. Thus, creation of better datasets, i.e., ones with better class distribution, is very important to take this research a step further.
2- Enhancing this work from only anomaly detection and classification to anomaly detection, classification, and severity prediction.
3- The research can be taken one step further by predicting anomalous events instead of just detecting anomalous event. This can then be further extended towards prediction anomalous event along with their classification and severity prediction.

References

[1] N. Anantrasirichai, J. Burn, and D. Bull, “Terrain Classification from Body-Mounted Cameras during Human Locomotion,” *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2249–2260, Oct. 2015, doi: 10.1109/TCYB.2014.2368353.

[2] C. H. Lin, F. C. Wang, T. Y. Kuo, P. W. Huang, S. F. Chen, and L. C. Fu, “Early Detection of Parkinson’s Disease by Neural Network Models,” *IEEE Access*, vol. 10, pp. 19033–19044, 2022, doi: 10.1109/ACCESS.2022.3150774.

[3] I. el Maachi, G.-A. Bilodeau, and W. Bouachir, “Deep 1D-Convnet for accurate Parkinson disease detection and severity prediction from gait,” Oct. 2019, doi: 10.1016/j.eswa.2019.113075.

[4] L. Jeancolas *et al.*, “Automatic Detection of Early Stages of Parkinson’s Disease through Acoustic Voice Analysis with Mel-Frequency Cepstral Coefficients.” [Online]. Available: https://hal.archives-ouvertes.fr/hal-02474494

[5] J. F. Daneault *et al.*, “Accelerometer data collected with a minimum set of wearable sensors from subjects with Parkinson’s disease,” *Scientific Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1038/s41597-021-00830-0.

[6] G. Rodríguez, F. E. Casado, R. Iglesias, C. v. Regueiro, and A. Nieto, “Robust step counting for inertial navigation with mobile phones,” *Sensors (Switzerland)*, vol. 18, no. 9, Sep. 2018, doi: 10.3390/s18093157.

[7] A. C. Y. Lim, P. Natarajan, R. D. Fonseka, M. Maharaj, and R. J. Mobbs, “The application of artificial intelligence and custom algorithms with inertial wearable devices for gait analysis and detection of gait-altering pathologies in adults: A scoping review of literature,” *Digital Health*, vol. 8. SAGE Publications Inc., Jan. 01, 2022. doi: 10.1177/20552076221074128.

[8] M. Z. U. H. Hashmi, Q. Riaz, M. Hussain, and M. Shahzad, “What lies beneath one’s feet? Terrain classification using inertial data of human walk,” *Applied Sciences (Switzerland)*, vol. 9, no. 15, Aug. 2019, doi: 10.3390/app9153099.

[9] Q. Riaz, A. Vögele, B. Krüger, and A. Weber, “One small step for a man: Estimation of gender, age and height from recordings of one step by a single inertial sensor,” *Sensors (Switzerland)*, vol. 15, no. 12, pp. 31999–32019, Dec. 2015, doi: 10.3390/s151229907.

[10] Q. Riaz, M. Z. U. H. Hashmi, M. A. Hashmi, M. Shahzad, H. Errami, and A. Weber, “Move your body: Age estimation based on chest movement during normal walk,” *IEEE Access*, vol. 7, pp. 28510–28524, 2019, doi: 10.1109/ACCESS.2019.2901959.

[11] M. A. Hashmi, Q. Riaz, M. Zeeshan, M. Shahzad, and M. M. Fraz, “Motion Reveal Emotions: Identifying Emotions from Human Walk Using Chest Mounted Smartphone,” *IEEE Sensors Journal*, vol. 20, no. 22, pp. 13511–13522, Nov. 2020, doi: 10.1109/JSEN.2020.3004399.

[12] S. Pardoel, G. Shalin, E. D. Lemaire, J. Kofman, and J. Nantel, “Grouping successive freezing of gait episodes has neutral to detrimental effect on freeze detection and prediction in Parkinson’s disease,” *PLoS ONE*, vol. 16, no. 10 October, Oct. 2021, doi: 10.1371/journal.pone.0258544.

[13] D. Yang *et al.*, “Detection of Mild Cognitive Impairment Using Convolutional Neural Network: Temporal-Feature Maps of Functional Near-Infrared Spectroscopy,” *Frontiers in Aging Neuroscience*, vol. 12, May 2020, doi: 10.3389/fnagi.2020.00141.

[14] I. Gohar, Q. Riaz, M. Shahzad, M. Z. U. H. Hashmi, H. Tahir, and M. E. Ul Haq, “Person re-identification using deep modeling of temporally correlated inertial motion patterns,” *Sensors (Switzerland)*, vol. 20, no. 3, Feb. 2020, doi: 10.3390/s20030949.

[15] R. T. Shahar and M. Agmon, “Gait analysis using accelerometry data from a single smartphone: Agreement and consistency between a smartphone application and gold-standard gait analysis system,” *Sensors*, vol. 21, no. 22, Nov. 2021, doi: 10.3390/s21227497.

[16] K. H. Chu, X. Jiang, and C. Menon, “Wearable step counting using a force myography-based ankle strap,” *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 4, p. 205566831774630, Jan. 2017, doi: 10.1177/2055668317746307.

[17] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, “Deep Learning-Based Gait Recognition Using Smartphones in the Wild,” Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.00338

[18] K. Zhang *et al.*, “Age Group and Gender Estimation in the Wild with Deep RoR Architecture,” *IEEE Access*, vol. 5, pp. 22492–22503, Oct. 2017, doi: 10.1109/ACCESS.2017.2761849.

[19] V. von Tscharner, H. Enders, and C. Maurer, “Subspace Identification and Classification of Healthy Human Gait,” *PLoS ONE*, vol. 8, no. 7, Jul. 2013, doi: 10.1371/journal.pone.0065063.

[20] X. Li, Y. Makihara, C. Xu, Y. Yagi, and M. Ren, “Gait-based human age estimation using age group-dependent manifold learning and regression,” *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28333–28354, Nov. 2018, doi: 10.1007/s11042-018-6049-7.

[21] Q. Wang, L. Ye, H. Luo, A. Men, F. Zhao, and Y. Huang, “Pedestrian stride-length estimation based on LSTM and denoising autoencoders,” *Sensors (Switzerland)*, vol. 19, no. 4, Feb. 2019, doi: 10.3390/s19040840.

[22] H. Juanjuan, Y. Zhiming, W. Jianguo, L. Bochen, and Y. Xianjun, “An Automatic Detection Method for Bradykinesia in Parkinson’s Disease Based on Inertial Sensor,” in *2020 IEEE 3rd International Conference on Electronics Technology, ICET 2020*, May 2020, pp. 166–169. doi: 10.1109/ICET49382.2020.9119604.

[23] A. Dvorani *et al.*, “Inertial Sensor Based Detection of Freezing of Gait for On-Demand Cueing in Parkinson’s Disease,” in *IFAC-PapersOnLine*, 2020, vol. 53, no. 2, pp. 16004–16009. doi: 10.1016/j.ifacol.2020.12.400.

[24] L. Borzì, I. Mazzetta, A. Zampogna, A. Suppa, F. Irrera, and G. Olmo, “Predicting Axial Impairment in Parkinson’s Disease through a Single Inertial Sensor,” *Sensors*, vol. 22, no. 2, Jan. 2022, doi: 10.3390/s22020412.

[25] V. Jakob *et al.*, “Validation of a sensor-based gait analysis system with a gold-standard motion capture system in patients with parkinson’s disease,” *Sensors*, vol. 21, no. 22, Nov. 2021, doi: 10.3390/s21227680.

[26] L. Tong, J. He, and L. Peng, “CNN-Based PD Hand Tremor Detection Using Inertial Sensors,” *IEEE Sensors Letters*, vol. 5, no. 7, Jul. 2021, doi: 10.1109/LSENS.2021.3074958.

[27] M. Son *et al.*, “The effect of levodopa on bilateral coordination and gait asymmetry in Parkinson’s disease using inertial sensor,” *npj Parkinson’s Disease*, vol. 7, no. 1, Dec. 2021, doi: 10.1038/s41531-021-00186-7.

[28] N. Roth *et al.*, “Hidden Markov Model based stride segmentation on unsupervised free-living gait data in Parkinson’s disease patients,” *Journal of NeuroEngineering and Rehabilitation*, vol. 18, no. 1, Dec. 2021, doi: 10.1186/s12984-021-00883-7.

[29] S. Alberto *et al.*, “Validation of quantitative gait analysis systems for Parkinson’s disease for use in supervised and unsupervised environments,” *BMC Neurology*, vol. 21, no. 1, Dec. 2021, doi: 10.1186/s12883-021-02354-x.

[30] L. B. Peres *et al.*, “Discrimination between healthy and patients with Parkinson’s disease from hand resting activity using inertial measurement unit,” *BioMedical Engineering Online*, vol. 20, no. 1, Dec. 2021, doi: 10.1186/s12938-021-00888-2.

[31] S. Lee, R. Hussein, R. Ward, Z. Jane Wang, and M. J. McKeown, “A convolutional-recurrent neural network approach to resting-state EEG classification in Parkinson’s disease,” *Journal of Neuroscience Methods*, vol. 361, Sep. 2021, doi: 10.1016/j.jneumeth.2021.109282.

[32] M. Shaban, “Automated screening of Parkinson’s disease using deep learning based electroencephalography,” in *International IEEE/EMBS Conference on Neural Engineering, NER*, May 2021, vol. 2021-May, pp. 158–161. doi: 10.1109/NER49283.2021.9441065.

[33] E. E. Asher *et al.*, “Connectivity of EEG synchronization networks increases for Parkinson’s disease patients with freezing of gait,” *Communications Biology*, vol. 4, no. 1, Dec. 2021, doi: 10.1038/s42003-021-02544-w.

[34] H. W. Loh *et al.*, “Gaborpdnet: Gabor transformation and deep neural network for parkinson’s disease detection using eeg signals,” *Electronics (Switzerland)*, vol. 10, no. 14, Jul. 2021, doi: 10.3390/electronics10141740.

[35] S. K. Khare, V. Bajaj, and U. R. Acharya, “PDCNNet: An Automatic Framework for the Detection of Parkinson’s Disease Using EEG Signals,” *IEEE Sensors Journal*, vol. 21, no. 15, pp. 17017–17024, Aug. 2021, doi: 10.1109/JSEN.2021.3080135.

[36] K. Rezaee, S. Savarkar, X. Yu, and J. Zhang, “A hybrid deep transfer learning-based approach for Parkinson’s disease classification in surface electromyography signals,” *Biomedical Signal Processing and Control*, vol. 71, Jan. 2022, doi: 10.1016/j.bspc.2021.103161.

[37] S. M. Rissanen, M. Koivu, P. Hartikainen, and E. Pekkonen, “Ambulatory surface electromyography with accelerometry for evaluating daily motor fluctuations in Parkinson’s disease,” *Clinical Neurophysiology*, vol. 132, no. 2, pp. 469–479, Feb. 2021, doi: 10.1016/j.clinph.2020.11.039.

[38] H. M. Adem, A. W. Tessema, and G. L. Simegn, “Classification of Parkinson’s Disease Using EMG Signals from Different Upper Limb Movements Based on Multiclass Support Vector Machine,” *International Journal Bioautomation*, vol. 26, no. 1, pp. 109–125, 2021, doi: 10.7546/ijba.2022.26.1.000849.

[39] U. Kleinholdermann, M. Wullstein, and D. Pedrosa, “Prediction of motor Unified Parkinson’s Disease Rating Scale scores in patients with Parkinson’s disease using surface electromyography,” *Clinical Neurophysiology*, vol. 132, no. 7, pp. 1708–1713, Jul. 2021, doi: 10.1016/j.clinph.2021.01.031.

[40] H. Bengacemi, A. Hacine Gharbi, P. Ravier, K. Abed-Meraim, and O. Buttelli, “Surface EMG signal segmentation based on HMM modelling: Application on Parkinson’s disease,” *ENP Engineering Science Journal*, vol. 1, no. 1, pp. 63–74, Jul. 2021, doi: 10.53907/enpesj.v1i1.27.

[41] C. Quan, K. Ren, and Z. Luo, “A Deep Learning Based Method for Parkinson’s Disease Detection Using Dynamic Features of Speech,” *IEEE Access*, vol. 9, pp. 10239–10252, 2021, doi: 10.1109/ACCESS.2021.3051432.

[42] M. Hireš, M. Gazda, P. Drotár, N. D. Pah, M. A. Motin, and D. K. Kumar, “Convolutional neural network ensemble for Parkinson’s disease detection from voice recordings,” *Computers in Biology and Medicine*, vol. 141, Feb. 2022, doi: 10.1016/j.compbiomed.2021.105021.

[43] M. Masud *et al.*, “CROWD: Crow search and deep learning based feature extractor for classification of parkinson’s disease,” *ACM Transactions on Internet Technology*, vol. 21, no. 3, Aug. 2021, doi: 10.1145/3418500.

[44] A. Ouhmida, O. Terrada, A. Raihani, B. Cherradi, and S. Hamida, “Voice-Based Deep Learning Medical Diagnosis System for Parkinson’s Disease Prediction,” Jul. 2021. doi: 10.1109/ICOTEN52080.2021.9493456.

[45] G. Morinan *et al.*, “Computer-vision based method for quantifying rising from chair in Parkinson’s disease patients,” *Intelligence-Based Medicine*, vol. 6, Jan. 2022, doi: 10.1016/j.ibmed.2021.100046.

[46] S. Rupprechter *et al.*, “A clinically interpretable computer-vision based method for quantifying gait in parkinson’s disease,” *Sensors*, vol. 21, no. 16, Aug. 2021, doi: 10.3390/s21165437.

[47] J. Archila, A. Manzanera, and F. Martínez, “A multimodal Parkinson quantification by fusing eye and gait motion patterns, using covariance descriptors, from non-invasive computer vision,” *Computer Methods and Programs in Biomedicine*, vol. 215. Elsevier Ireland Ltd, Mar. 01, 2022. doi: 10.1016/j.cmpb.2021.106607.

[48] K. Jahn, A. Zwergal, and R. Schniepp, “Gangstörungen im alter - Klassifikation, diagnostik und therapie aus neurologischer sicht,” *Deutsches Arzteblatt*, vol. 107, no. 17. Deutscher Arzte-Verlag GmbH, pp. 306–316, Apr. 30, 2010. doi: 10.3238/arztebl.2010.0306.

[49] W. Tao, T. Liu, R. Zheng, and H. Feng, “Gait analysis using wearable sensors,” *Sensors*, vol. 12, no. 2. pp. 2255–2283, Feb. 2012. doi: 10.3390/s120202255.

[50] E. W. Watt and A. A. T. Bui, “Evaluation of a Dynamic Bayesian Belief Network to Predict Osteoarthritic Knee Pain Using Data from the Osteoarthritis Initiative.” [Online]. Available: http://www.-

[51] C. Schüldt, I. Laptev, and B. Caputo, “Recognizing Human Actions: A Local SVM Approach *.”

[52] X. Liu, Y. Wang, X. Wang, H. Xu, C. Li, and X. Xin, “Bi-directional gated recurrent unit neural network based nonlinear equalizer for coherent optical communication system,” *Optics Express*, vol. 29, no. 4, p. 5923, Feb. 2021, doi: 10.1364/oe.416672.

[53] J. Lu, G. Wang, and P. Moulin, “Human identity and gender recognition from gait sequences with arbitrary walking directions,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 1, pp. 51–61, Jan. 2014, doi: 10.1109/TIFS.2013.2291969.