A Report on

Multi-Device Authentication of Users based on Heart-Rate Monitoring Data

By

Shambwani Rohit Pradeep 2021H1030055G

Under the Guidance of
Prof. Sougata Sen

Birla Institute of Technology & Science, Pilani
K K Birla Goa Campus
2021-22

ABSTRACT

Authentication is one of the important security services that gives the assurance that the communicating entity is the one claimed to be.There are many types of authentication performed based on different needs like User authentication,Device Authentication,etc.Authentication can be done through different methods like passwords,OTPs,security codes,etc which serve as a strong security foundation.The aim of this project is to perform wearable-based authentication of a user using his/her heart rates and whether all the devices are worn by user at an instant of time by correlating the features extracted from the dataset.So,the problem statement of the project is Given a dataset of 5 participants that comprises of Heart Rate in BPM and Respiratory Rate interval measured in milliseconds which is collected from three devices namely Polar Heart Rate Monitor,Garmin-HRM Dual and VIVO ,the aim of the project is to confirm whether the three devices are being worn by the same user at any instant of time. This will be done based on the Heart-Rate data collected from the device that has been mentioned above by the help of analysis based on RR intervals and Heart Rate Variability.At first, features extraction will be done which will lead to correlating the features obtained.Correlation will be done in both ways of aligning and Non-aligning Windows.Later,False Acceptance Rate and False Rejection Rate will be computed which will give Equal Error Rate at the end to determine the threshold value.

TABLE OF CONTENTS

INTRODUCTION

When we say authentication,our main objective is to give assurance that the entity which is using the service is actually the entity which is assigned to use the service.Authentication serves as a crucial aspect in achieving the network security because if authentication is compromised the whole network can be compromised as the attacker will have the control of the services which he/she is not intended to have.This can lead to collapse of an organization if the person whose identity got compromised is the head or in a leading position.The challenges that authentication possesses is also of a large scale as it is very intricate procedure to identify and there shouldn't be any chance of errors.

Various types of authentication mechanisms exist today which are Single Factor,Two Factor,Password Based Authentication along with Single Sign-On and many more. The authentication factors are primarily classified into mainly three groups mainly passwords(Personal Identification Number (PIN)),tokens(Bank Card) and biometrics-based(voice,fingerprints).The project also focuses on a type of biometric based user-authentication which is heart rate but first,a knowledge of wearable based authentication is required.Wearable-based authentication involves a wearable to achieve the process of authentication of a user.The wearable serves as a tracker for tracking user's biometrics like audio,movements,etc.It generates the required data gathered from accelerometers,gyroscopes in it to compare to while doing the authentication.

In this project,usage of three heart rate monitors is done which are named as Polar Heart Rate Monitor,Garmin HRMDual and Vivo.All the three devices have been worn on five participants to generate their heart-beats in beats per minute and respiration rate intervals(RR) in milliseconds which is used for doing the authentication process further.The authentication process starts with extracting the features involved in Heart Rate variability. There are three types of domain measurements that are included in the Heart Rate Variablity analysis which are namely Time Domain,Frequency Domain and Non-Linear Measurements.The time domain measurements comprises of features such as Mean and Std.Deviation of Heart Beats along with Respiratory Rate (RR)intervals, generating the count for RR intervals that differ by more than 50 ms ,calculating the percentage of this count.Frequency Domain measurements involve signals mostly ECG in our case to generate it.Some of the Frequency domain measurements are ULF Power,VLF Power,LF peak,LF power,HF peak and HF power. Non linear measurements helps to quantify the unavailability of time series.Some of the non-linear measurements are S,Sd1,Sd2,ApEn,SampEn which correspond to area of ellipse which represents total HRV,Poincare plot standard deviation,approximate entropy,sample entropy,etc

The project aims to analyze the dataset containing the data to five participants wearing all the three wearables of Polar Heart Rate Monitor,HRMDual and Vivo respectively and then extracting the Time-Domain measurements as per requirement after which at further stages doing correlation between them.Correlating is done in both the aligning windows and Non-Aligning Windows for getting False Rejection Rate and False Acceptance Rate graphs respectively.Intersecting both these graphs determines Equal Error Rate giving the value of threshold required for model's accuracy.

LITERATURE SURVEY

The literature survey consisted of studying various authentication techniques that were published or researched upon before. These included authentication based on audio in which audio waves are studied to make the authentication successful.A bilateral authentication mechanism was also surveyed which provided recurring authentication instead of traditional login logout authentication that helped the user to stay signed out when he/she was not working on the terminal or computer.

A zero-effort bilateral recurring authentication also known as ZEBRA comprised of a user wearing a bracelet in his/her hand.The bracelet consisted of a built-in accelerometer,gyroscope and radio to track the wrist movements.The bracelet is worn on user's dominant wrist.Whenever the user typed something on the computer screen,the wrist movement of the user was tracked by the bracelet worn by him/her. This data that was tracked was then used to match the inputs the terminal received from the user.The terminal is responsible for comparing this tracking data with the inputs that user gives through keyboard and mouse.Since the hand that sends inputs to the terminal is the same hand that has bracelet on it,the tracking data and the input values correlate since the hand which acts as the source is the same.This authentication mechanism performed fairly well in terms of accuracy and is also fast in terms of catching the attacker performing authentication.ZEBRA's architecture consisted of five main components which were namely,Interaction Extractor that extracted interactions from user's input, a segmenter segmented the data obtained from accelerometer and gyroscope into blocks,feature extractor extracted features from the data obtained with the help of Segmenter, the interaction classifier classified it into one of the actual sequences of interaction, and finally the authenticator component did the comparison.

One more authentication mechanism included using internal body voice for authenticating wearables like who is wearing(identification) and is he/she the correct person to wear it(Verification) .An emphasis on using vocal resonance as an unobtrusive measurement for authentication is explained. Evaluation of this measurement is based on two machine learning algorithms which help determine its accuracy by testing it on volunteer subjects.The two machine learning algorithms that were used in the paper were Gaussian Mixture Model(GMM) and Deep Neural Network(DNN).Along with that,the paper focussed on Microphone locations and how it can impact the result.Distinguishing the body voice from air voice is also a factor to be considered and is required to be differentiated.To tackle this there are two sets of GMMs in GMM model and a fully dense layer contemplated in the DNN model.This method comprises of components like features extraction which extracted the features from audio or speech waves,enrollment algorithms which involved taking voice samples from user by making the user read a small passage which is then segmented into sliding windows,authentication algorithm which is GMM and DNN in this case.After that results are compared to check if the model is trained properly by testing it on bunch of volunteers.

IMPLEMENTATION

The implementation procedure comprises of 5 stages which are as follows:

A) Feature Extraction
B) Correlating Features
C) Computing False Rejection Rate(FRR) in case of Aligned Windows
D) Computing False Acceptance Rate (FAR) in case of Non-Aligned Windows
E) Thresholding with Equal Error Rate(EER)

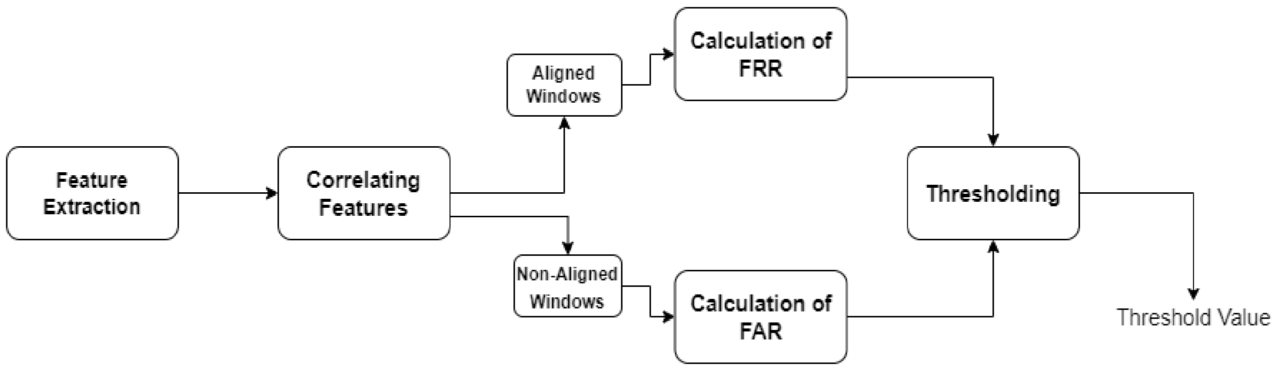


Fig. Stages of Implementation

The implementation starts with feature extraction from the dataset that includes Time Domain Features of Heart Rate Variability which are then correlated with each other based on the device into two halves.In one half, correlation is done based on aligning windows and in other half,it is done on non-aligning windows which is explained in detail later.
Correlating aligning windows leads to calculation of False Rejection Rate(FRR) which determines the number of entries that are being falsely rejected in case of windows aligning on the devices and Correlating Non-Aligning Windows leads to calculation of False Acceptance Rate (FAR) which determines the number of entries that are being falsely accepted in case of windows not aligning.
The last step involves determining the Equal Error Rate(EER) which determines the point of intersection of FRR and FAR and that point gives the threshold value above which entries are rejected and below or equal to that they are accepted.

A) Feature Extraction

From the dataset, the initial part of implementation included extracting some relevant features and hence,the first part of the solution to our problem statement comprises of Features Extraction and in this case, Time Domain Features of Heart Rate Variability(HRV).The features extraction was implemented in Python language with the help of Jupyter Notebook.These features were extracted for all the three devices which are involved in the process of authentication i.e Polar HeartRate Monitor, HRMDual and Vivo.

Following are the time domain features which were extracted:

Variable	**Unit**	**Description**
RR std	ms	Standard Deviation of all RR interval
RR mean	ms	Mean of all RR Intervals
HR mean	bpm	Mean of all Heartbeat
HR std	bpm	Standard Deviation of HeartBeat
NN50 Count	N/A	Number of pairs of adjacent RR intervals differing by more than 50 ms in all the measurements
pNN50	%	NN50 Count divided by the total number of all RR intervals

The mean and standard deviation computation also consisted of window size derived from using Boxcar function that considered some percentage overlapping of intervals to gain a deeper analysis by creating a time series that showed a progress of how mean and standard deviation were changing as the time passed by.But this method was only applicable to RR intervals and not to HeartBeats because it was not necessary.
After Feature Extraction,the next step is to define some kind of correlation between them which will lead to a conclusion as to how authentication mechanisms can be imposed.Below are some screenshots to show output of the various features extracted that are mentioned above:

Time Domain Features

1.RR Std (Standard Deviation of RR_Intervals)

```
import statistics
```

```
arPRR=df['Polar_RR'].to_numpy()
arHRMRR=df['HRMDUAL_RR'].to_numpy()
arViRR=df['Vivo_RR'].to_numpy()
print("Standard Deviation of the Polar_RR intervals is % s "% (statistics.stdev(arPRR)))
print("Standard Deviation of the HRMDual_RR intervals is % s "% (statistics.stdev(arHRMRR)))
print("Standard Deviation of the Vivo_RR intervals is % s "% (statistics.stdev(arViRR)))
```

```
Standard Deviation of the Polar_RR intervals is 155.58598908642128
Standard Deviation of the HRMDual_RR intervals is 568.8620219350207
Standard Deviation of the Vivo_RR intervals is 153.2351134694656
```

2. RR Mean (Mean of All RR_Intervals)

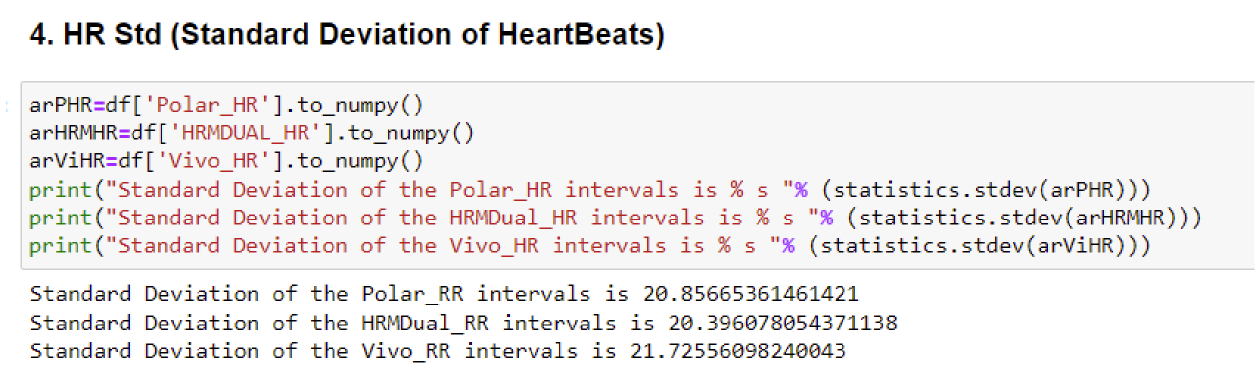
```
armPRR=df['Polar_RR'].to_numpy()
armHRMRR=df['HRMDUAL_RR'].to_numpy()
armViRR=df['Vivo_RR'].to_numpy()
print("Mean of the Polar_RR intervals is % s "% (statistics.mean(armPRR)))
print("Mean of the HRMDual_RR intervals is % s "% (statistics.mean(armHRMRR)))
print("Mean of the Vivo_RR intervals is % s "% (statistics.mean(armViRR)))
```

```
Mean of the Polar_RR intervals is 669
Mean of the HRMDual_RR intervals is 758
Mean of the Vivo_RR intervals is 675
```

3. HR Mean (Mean of all HeartBeats)

```
armPHR=df['Polar_HR'].to_numpy()
armHRMHR=df['HRMDUAL_HR'].to_numpy()
armViHR=df['Vivo_HR'].to_numpy()
print("Mean of the Polar_HR  is % s "% (statistics.mean(armPHR)))
print("Mean of the HRMDual_HR  is % s "% (statistics.mean(armHRMHR)))
print("Mean of the Vivo_HR  is % s "% (statistics.mean(armViHR)))
```

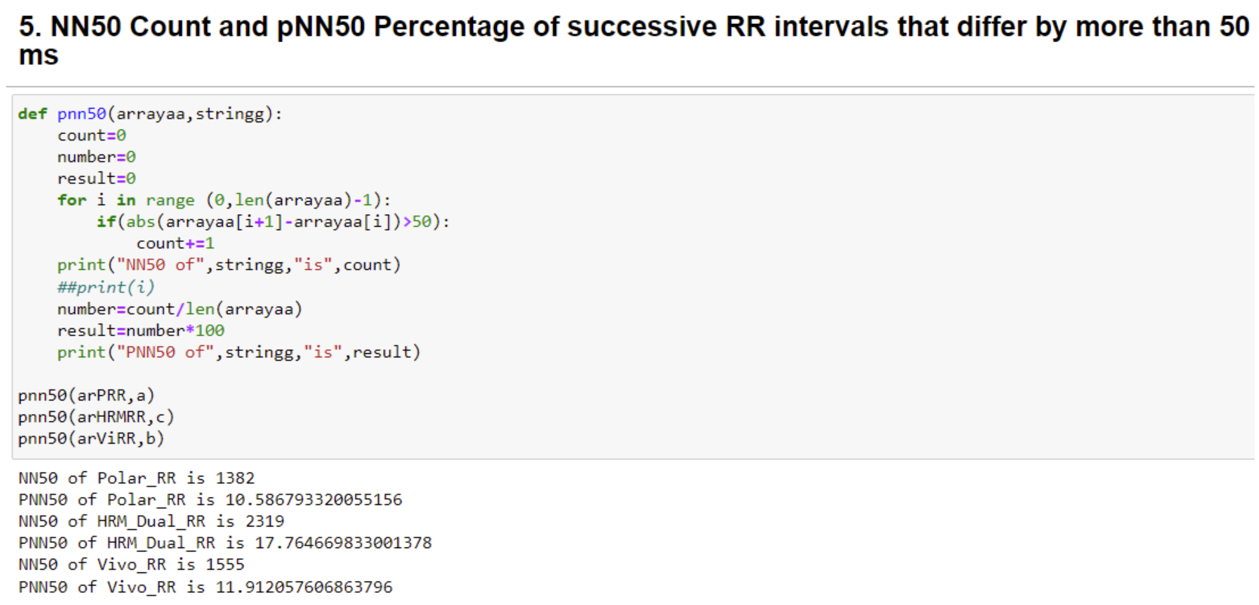
```
Mean of the Polar_HR  is 94
Mean of the HRMDual_HR  is 94
Mean of the Vivo_HR  is 93
```



4. HR Std (Standard Deviation of HeartBeats)

```
arPHR=df['Polar_HR'].to_numpy()
arHRMHR=df['HRMDUAL_HR'].to_numpy()
arViHR=df['Vivo_HR'].to_numpy()
print("Standard Deviation of the Polar_HR intervals is % s "% (statistics.stdev(arPHR)))
print("Standard Deviation of the HRMDual_HR intervals is % s "% (statistics.stdev(arHRMHR)))
print("Standard Deviation of the Vivo_HR intervals is % s "% (statistics.stdev(arViHR)))
```

```
Standard Deviation of the Polar_RR intervals is 20.85665361461421
Standard Deviation of the HRMDual_RR intervals is 20.396078054371138
Standard Deviation of the Vivo_RR intervals is 21.72556098240043
```



5. NN50 Count and pNN50 Percentage of successive RR intervals that differ by more than 50 ms

```
def pnn50(arrayaa,stringg):
    count=0
    number=0
    result=0
    for i in range (0,len(arrayaa)-1):
        if(abs(arrayaa[i+1]-arrayaa[i])>50):
            count+=1
    print("NN50 of",stringg,"is",count)
    ##print(i)
    number=count/len(arrayaa)
    result=number*100
    print("PNN50 of",stringg,"is",result)

pnn50(arPRR,a)
pnn50(arHRMRR,c)
pnn50(arViRR,b)
```

```
NN50 of Polar_RR is 1382
PNN50 of Polar_RR is 10.586793320055156
NN50 of HRM_Dual_RR is 2319
PNN50 of HRM_Dual_RR is 17.764669833001378
NN50 of Vivo_RR is 1555
PNN50 of Vivo_RR is 11.912057606863796
```

B) Correlating Features

Feature Correlation serves as a important aspect in understanding the relationship between multiple features or attributes in the dataset after feature extraction.With the help of feature correlation, understanding how one feature is dependent on another feature becomes clear.It also helps to understand how one or more features are associated with each other.Correlation helps to predict one feature or attribute from another feature or attribute.It is used as a basis for many basic modelling techniques.Correlation of features in our case is implemented using Python using numpy library’s function np.correlate()

The below diagram illustrates how correlation is done in our project using Feature Extraction and computing of Features:

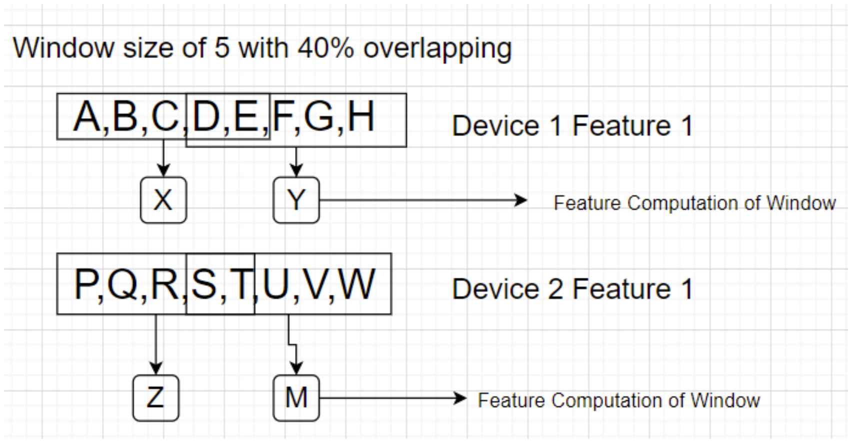

Window size of 5 with 40% overlapping
A,B,C,D,E,F,G,H
Device 1 Feature 1
X
Y
Feature Computation of Window
P,Q,R,S,T,U,V,W
Device 2 Feature 1
Z
M
Feature Computation of Window

Fig. Feature Extraction and Correlation

As shown in above diagram, Initially, the feature computation or extraction is done with the window size considered as 5 in our case with 40% considered as the percentage of overlapping. The four values obtained in the above diagram i.e X,Y,Z and M correspond to Device 1 Feature 1 values and Device 2 Feature 1 values respectively. The correlation between these two devices is done in 2 manners for their respective uses. Correlation in aligned windows gives the insight of the same units of time a user must have worn these devices and Correlation in nonaligned windows indicates the different units of time in which a particular user has not worn these devices together.

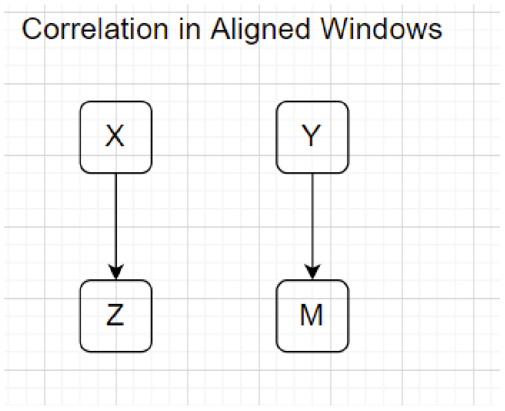


Fig. Correlation in Aligned Windows

The above figure shows correlation in case of Aligned windows in which X (Dev.1 Feature 1 value at first window) is correlated with Z (Dev.2 Feature 1 value at first window) hence aligning it and correlating it.Same is the case for Y correlating with M.\

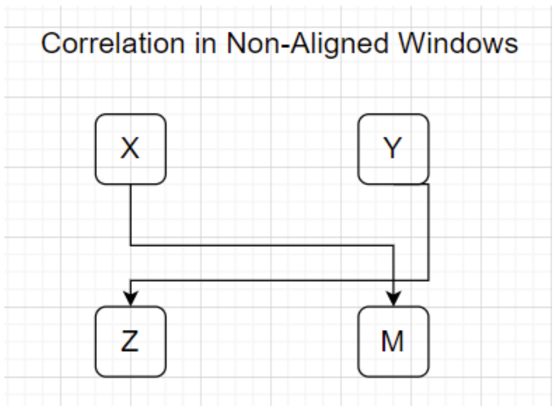


Fig. Correlation in Non-Aligned Windows

The above figure shows correlation in case of Non-Aligned windows in which X (Dev.1 Feature 1 value at first window) is correlated with M (Dev.2 Feature 1 value at second window) hence non-aligning it and correlating it.Same is the case for Y correlating with Z.

C) Computing False Rejection Rate(FRR) in case of Aligned Windows

False Rejection Rate(FRR) and False Acceptance Rate (FAR) are two important aspects of evaluation of a model in Biometric Systems.Both False Acceptance and Rejection Rates together help to determine the threshold value of the model which is used for prediction.The False Rejection Rate is defined as the "probability that a system fails to detect a match between an input pattern and the matching pattern stored in the database". It involves the measurement of valid inputs which are incorrectly rejected.

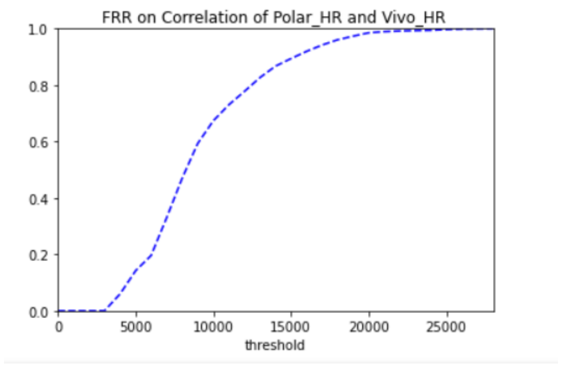
The curve of FRR represents the case where an original is treated as an impostor and gets falsely rejected.In our problem statement, we use FRR in case of aligning windows as that indicates the same instant of time where user has worn devices at once.The computation of FRR determines how many times a correct user wearing all the devices is falsely rejected or treated as an impostor.

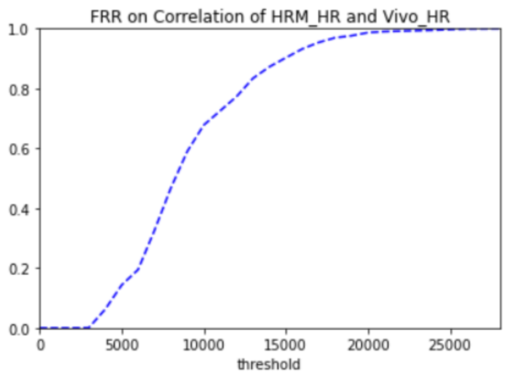
The algorithm for FRR is simple and is as follows:

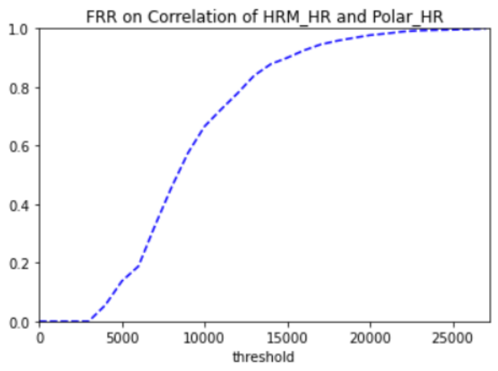
```
1.frr = []                    //List for storing FRR values
2.threshold = []              //List for storing thresholds
3.for i in range(0,(int)(max(correlationresultsarray)),incr): // correlationresultsarray is an array comprising of correlation results stored in case of aligning windows
        num = 0                     //Count

        for x in correlationresultsarray:
                if x<i:                         //If x gets falsely rejected
                        num+=1          //Increment Count
     frr(num/len(correlationresultsarray)) //Append the percentage to Frr array
```

```
threshold.append(i) //Appending the threshold value at which this Frr is obtained
```

Some of the FRR Graphs which were obtained are shown below:







The first figure shows FRR on Correlation of Polar_HR and Vivo_HR with aligning windows and Feature selected is Mean of HeartRates.At first,mean of Heart Rates is calculated which is stored in an array and this array is correlated with another array comprising of same feature but of different device.The correlation results are stored in an array on which this FRR algorithm is performed to get the results. The other two figures are also derived in the same way except they involve different device's correlation.The second figure shows FRR computation in case of HRM_HR and Vivo_HR whereas the third figure shows FRR computation in case of HRM_HR and Polar_HR.The same computation is done for other features like Standard Deviation of RR and HR also.

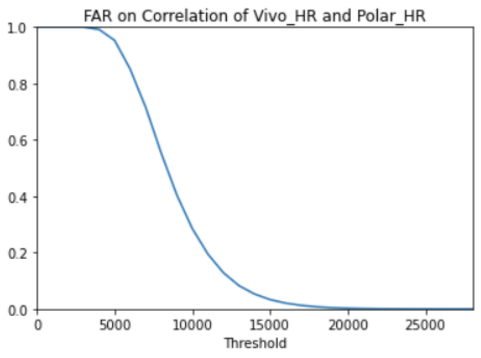
D) Computing False Acceptance Rate(FAR) in case of Non-Aligned Windows

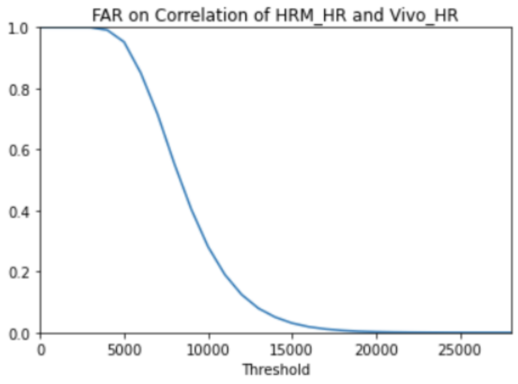
The False Acceptance Rate is defined in Wikipedia as "the probability that the system incorrectly matches the input pattern to a non-matching template in the database".It involves the measurement of invalid inputs that are incorrectly accepted. In other words, if a person is an

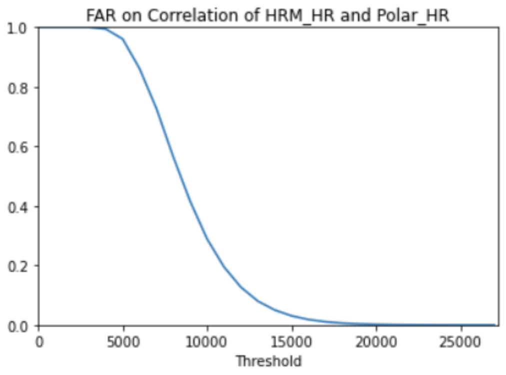
impostor in reality but gets treated as a genuine one then we say that person is falsely accepted and should be granted access privileges which shouldn't be the case in real life.
The curve of FAR represents the case where an impostor is treated as an original and gets falsely accepted.In our problem statement, we use FAR in case of non-aligning windows as that indicates the different instances of time where user has worn devices.The computation of FAR determines how many times an incorrect user wearing all the devices is falsely accepted or treated as an original.
The algorithm for FAR is simple and is as follows:

```
1.far = []                //List for storing FAR values
2.threshold = []          //List for storing thresholds
3.for i in range(0,(int)(max(correlationresultsarray)),incr): // correlationresultsarray is an array
comprising of correlation results stored in case of aligning windows
        num = 0                //Count

        for x in correlationresultsarray:
                if x>i:                //If x gets falsely accepted
                        num+=1         //Increment Count
      far(num/len(correlationresultsarray)) //Append the percentage to FAR array
      threshold.append(i) //Appending the threshold value at which this FAR is obtained
```

Some of the FAR Graphs which were obtained are shown below:



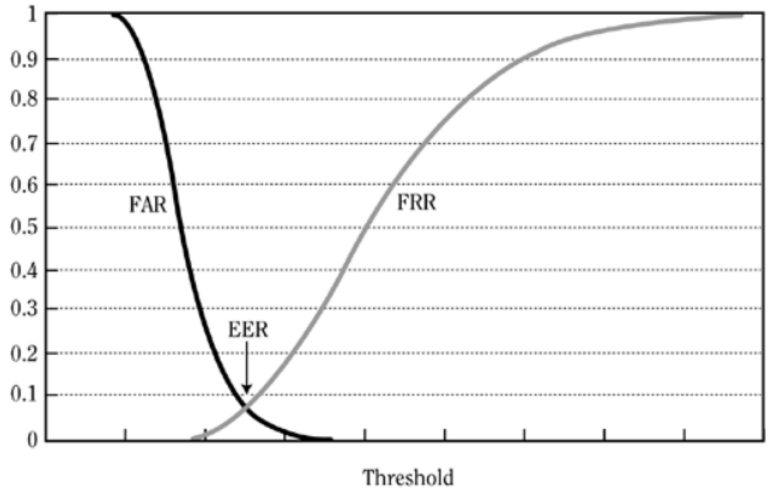




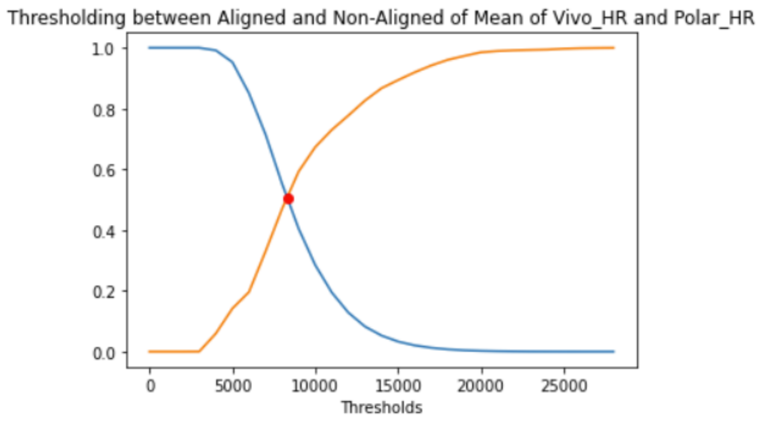
The first figure shows FAR on Correlation of Polar_HR and Vivo_HR with Non-aligning windows and Feature selected is Mean of HeartRates.At first,mean of Heart Rates is calculated which is stored in an array and this array is correlated with another array comprising of same feature but of different device.The correlation results of non-aligned windows are stored in an array on which this FAR algorithm is performed to get the results. The other two figures are also derived in the same way except they involve different device's correlation.The second figure shows FAR computation in case of HRM_HR and Vivo_HR whereas the third figure shows FAR computation in case of HRM_HR and Polar_HR.The same computation is done for other features like Standard Deviation of RR and HR also.

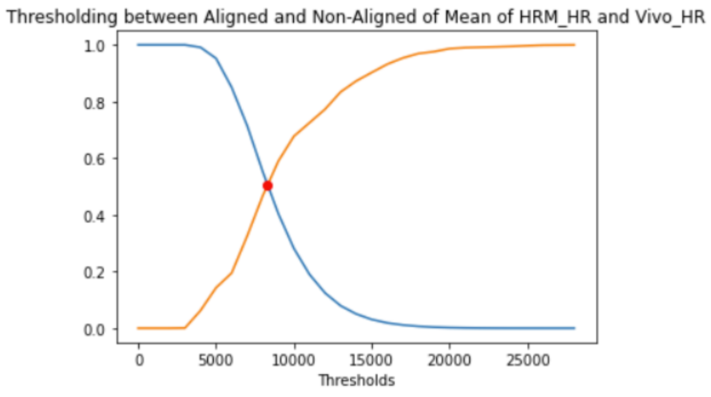
E) Thresholding with Equal Error Rate(EER)

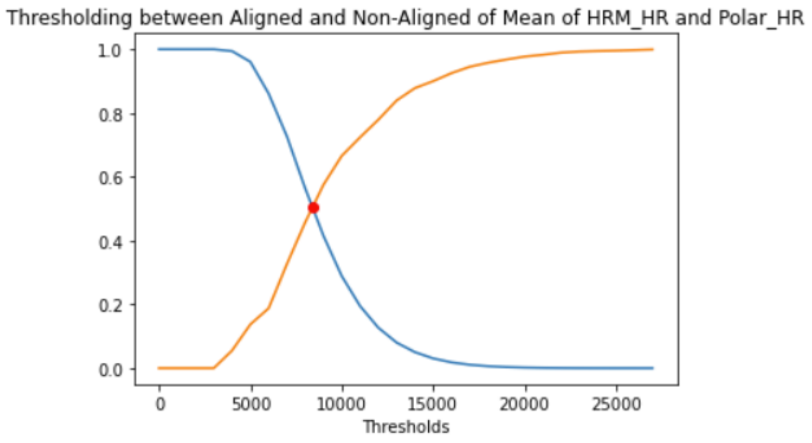
EER or Equal Error rate is defined as a point at which both False Acceptance Rate and False Rejection Rates are equal. In short, the intersection of two curves FAR and FRR gives a point which is termed as EER and based on the point we determine the ideal threshold value to be selected.The figure below shows how EER is computed:



As we can see from the figure EER lies at the intersection point of FAR and FRR curves.In practice the lower the EER value, the higher the accuracy of biometric system is there.
The figures below show the Equal Error Rate point in case of FRR of Aligned Windows and FAR of Non-Aligned Windows in case of HeartRates where the feature selected is Mean of HeartRates.







The first figure shows Equal Error Rate point (in red) between False Rejection Rate(FRR) (in orange) and False Acceptance Rate(FAR)(in blue) in case of Polar_HR and Vivo_HR where the feature selected is Mean of HeartRates.The other two figures are also derived in the same way except they involve different device's Equal Error Rate .The second figure showsEER computation in case of HRM_HR and Vivo_HR whereas the third figure shows EER computation in case of HRM_HR and Polar_HR.The same computation is done for other features like Standard Deviation of RR and HR also.

- EER in case of Polar_HR and Vivo_HR (First Fig) came out to be 0.507 at the threshold value of 8304.489
- EER in case of HRM_HR and Vivo_HR (Second Fig) came out to be 0.505 at the threshold value of 8313.087
- EER in case of HRM_HR and Polar_HR (Third Fig) came out to be 0.504 at the threshold value of 8399.194

RESULTS

- Of the 13,054 entries in the dataset, initially Time Domain Features such as Mean and Standard Deviation of Heart Rates and Respiratory Rates were extracted with window size of 5 and 40% taken as the percentage of overlapping.
- After Feature Extraction the Feature Values were stored in an array and the array size was of 4351 entries.
- On these 4351 entries, correlation of features was performed in both the cases of aligning and non-aligning windows.
- After correlation on Aligning windows the size of correlation results array was same as expected of 4351 entries whereas the size of correlation results array increased as expected in case of Non-Aligning windows i.e 4351x4351 - 4351=18926850 entries.
- FAR and FRR was computed in case of Non-Aligning and Aligning Windows respectively.
- Equal Error Rate was computed by finding the point of intersection of FAR and FRR graphs.

FUTURE WORK

The future work involves:

- Data Collection of More Participants so that the model can be trained more precisely.
- All the steps of implementation on this new dataset.
- Using the threshold value obtained from Equal Error Rate Method and applying it on the newly collected data to check the accuracy of the model based on that particular threshold value.

CONCLUSION

At the end ,all the stages of implementation were completed successfully from the start till the final deduction of Threshold Value at the end.All the feature extraction,correlation of features and computing FRR and FAR was done successfully.At the end,a proper numerical value of EER was obtained.