Online Signature Biometrics Lab – Report

Baglan Aitu, Sanjay Kumar & Sergio Romero Tapiador

Outline

1. Introduction
2. Data
3. Feature Extraction
4. Performance Evaluation

1. Introduction

The objective of this session is to DEVELOP and EVALUATE an online signature recognition algorithm. According to the theory sessions, signature recognition systems can be divided into two categories:

- **Off-line**: the input is a static image of the signature.
- **On-line**: the signature is acquired using a specific digital sensor which includes the static image and dynamic signals related with the way the signature was done: x,y coordinates and pressure as a function of time.

Figure 1 shows a block diagram of a typical online signature recognition algorithm where [x,y,p] are the captured signals by the sensor (Cartesian coordinates and pressure), f_t is the feature vector of the query signature to be compared with the f_c feature vector of the signature stored in the database (claimed identity).

In this session we will assume that the data is available (previously acquired) and we will focus on the development of two modules:

- Feature Extraction Module.
- Matcher.

You must complete the tasks proposed in this document and answer the questions included.

Figure 1. Block Diagram of a typical online signature recognition system

2. Data

For the practice we will use 50 users from the BiosecurID database. Each of the users have 28 signatures acquired in 4 sessions with a time lapse of 2 months. From the 28 signatures, 16 are genuine (4 per session) and 12 are forgers (3 per session). In this practice we will only consider the genuine signatures.

Each of the signatures is stored in .mat file which contains three vectors of same length with the x, y coordinates and the pressure as functions of time.

The formatting of the files is uXXXXsYYYY_sgZZZZ.mat:

- XXXX: user number
- YYYY: session number
- ZZZZ: signature number

The GENUINE signatures of each session are those with ZZZZ=[0001,0002,0006,0007].

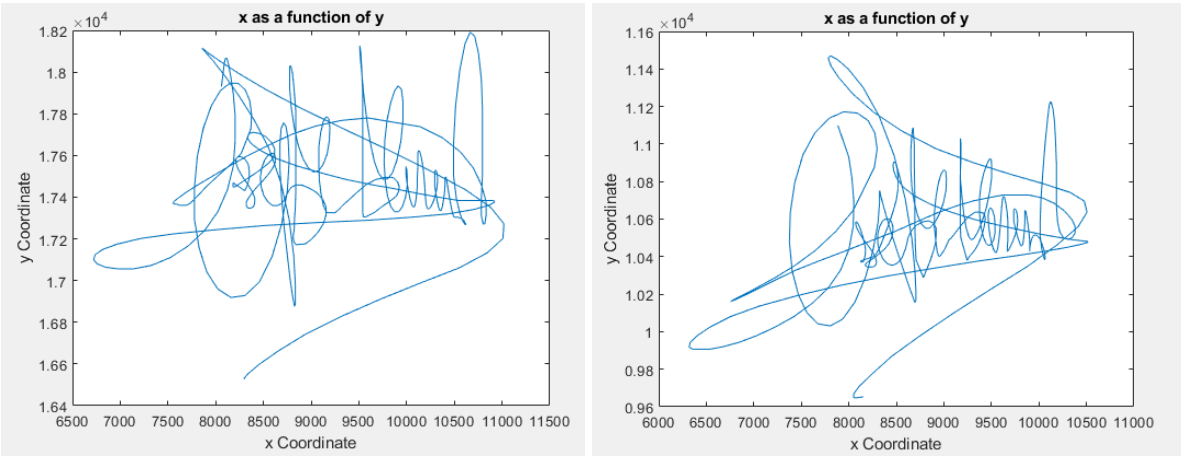
The signatures with ZZZZ=[0003,0004,0005] are the FORGERS and they will NOT be used in this practice.

QUESTION. Choose a signature (from a random user) and show (assuming that the sensor has a 200 samples/second acquisition rate):

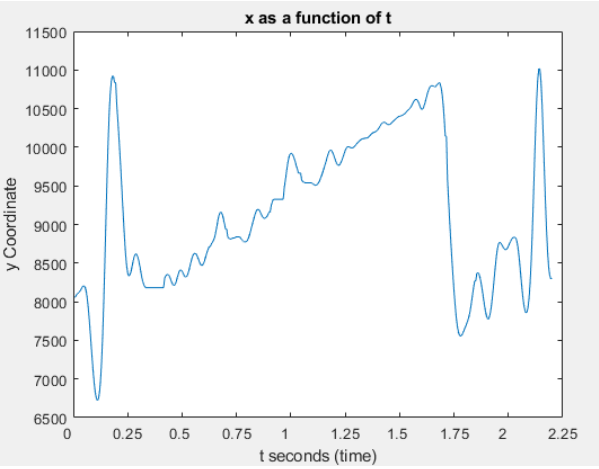
- Signal x as a function of signal y.
- Signal x as a function of time.
- Signal y as a function of time.
- Signal p as a function of time.

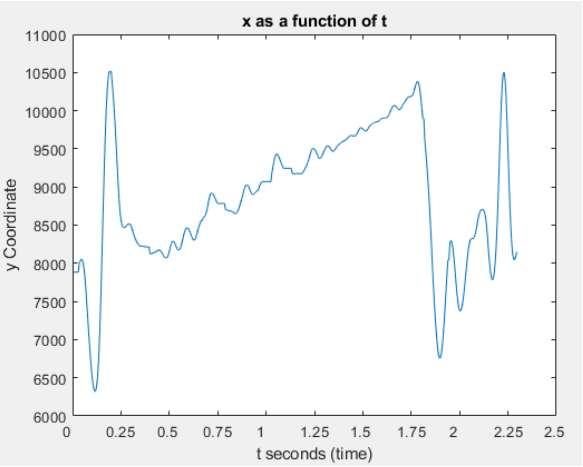
We select a random user 1048, the session number 0003 and the signature number 0001 for the first signature (left ones) and the session number 0004 and the signature number 0007 for the second signature (right ones).

x as function of y

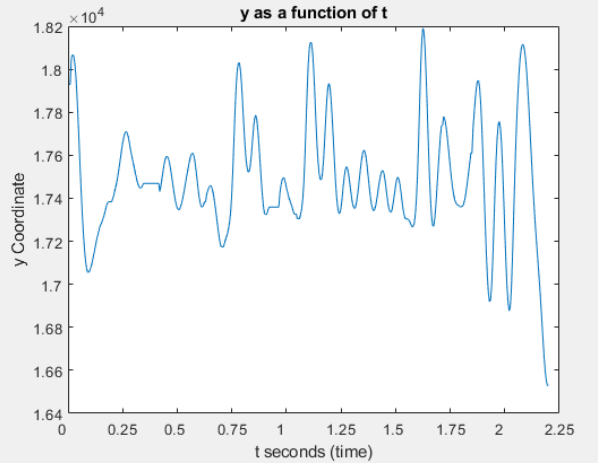


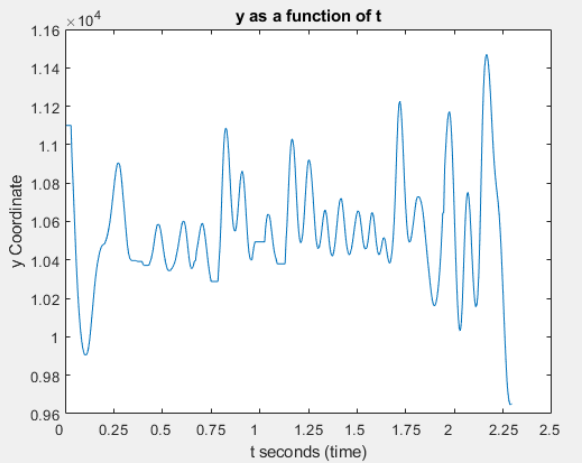
x as function of t



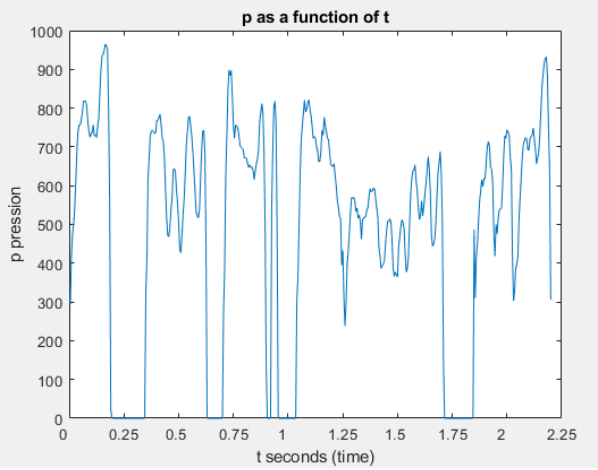


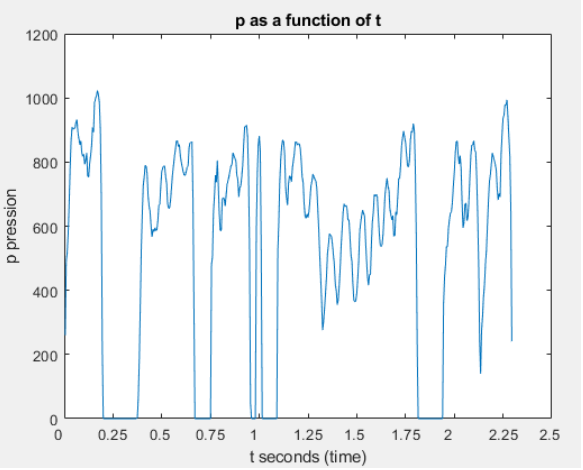
y as function of t





p as function of t





QUESTION. Are the different signals reasonable? Do they have the same length? Why?

Signals look reasonable and both of them have a similar length (around 2.25 - 2.5 seconds). Moreover, both signatures (y as a function of x) look authentic and quite similar between them. The reason for that is they belong to the same person who should have the same features for her/his signatures. Nevertheless, there is some variability in all variables, but the curves describe homogeneous traces.

3. Feature Extraction

The comparison of signals with different lengths is not trivial. Therefore, we will extract 4 global parameters of each of the signatures. So, each signature will be represented by a feature vector with fixed size equal to 4. These parameters are:

- Total duration of the signature: T
- Number of *pen-up* (number of times the pen was lifted). It means the number of times (not the number of samples) that p is equal to 0.
- Duration of *pen-down* (signal p is different to 0) T_d divided by the total duration T: T_d/T
- Average pressure in *pen-down* (signal p is different to 0).

You have to develop 4 functions to extract each of the parameters:

- T=Ttotal(x)
- Npu=Npenups(p)
- Tpd=Tpendown(p)
- Ppd=Ppendown(p).

According to those functions, we will develop a new function with input data (x,y,p) of a given signature and output data the feature vector containing the 4 parameters (FeatVect=featureExtractor(x,y,p)).

Based on your function featureExtractor you have to develop a program (ProcessBiosecurID.m) to extract all the feature vectors from the database and store it in a matrix with 3 dimensions:

- Dimension 1: number of user (1:50)
- Dimension 2: number of signature (1:16)
- Dimension 3: number of parameter (1:4)

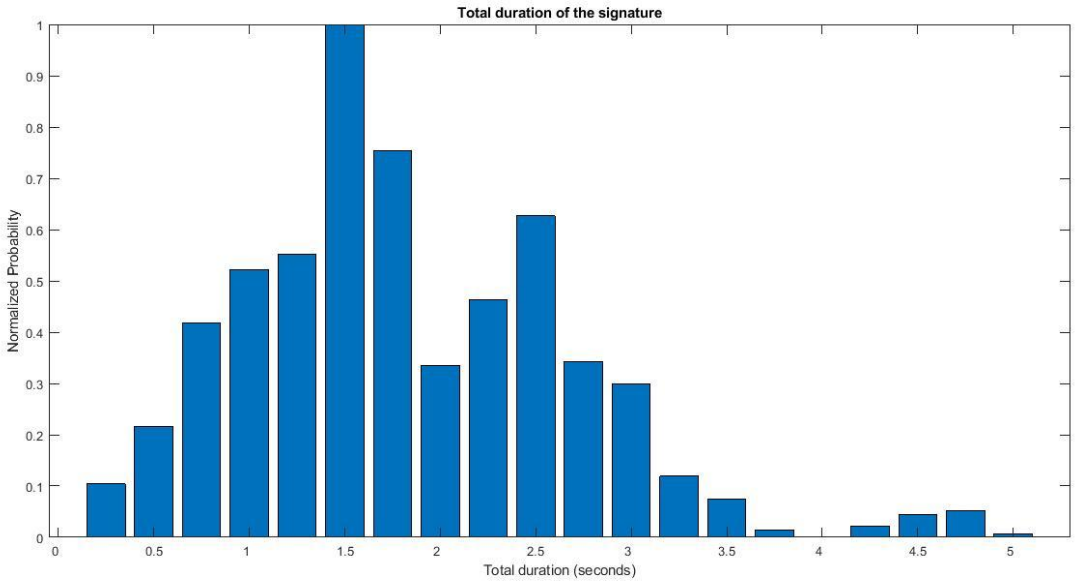
You have to save this matrix into the file BiosecurIDparameters.mat

Once you have the file BiosecurIDparameters.mat, you have to plot the distributions normalized between 0 and 1 (dividing by the total number of points of the distribution) for each of the 4 parameters.

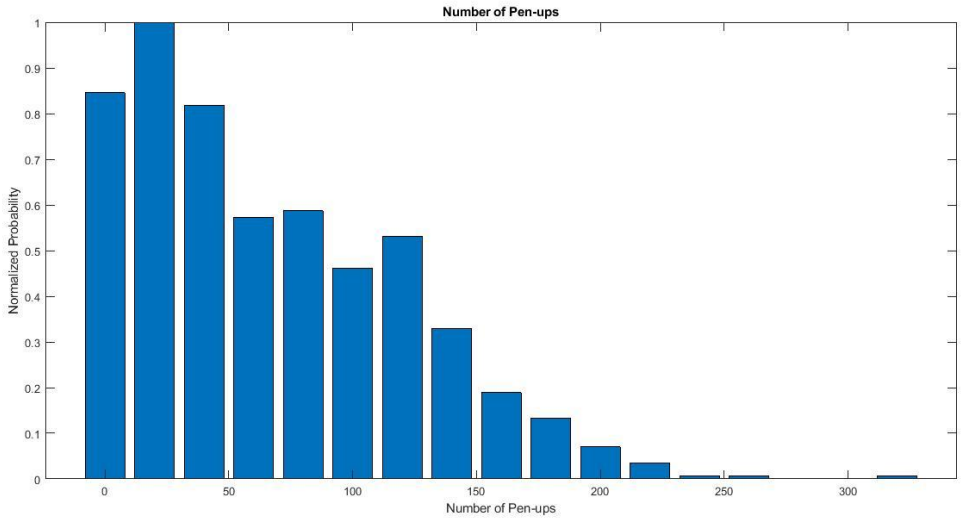
You can use the Matlab functions hist.m and histc.m

QUESTION: Plot the 4 distributions.

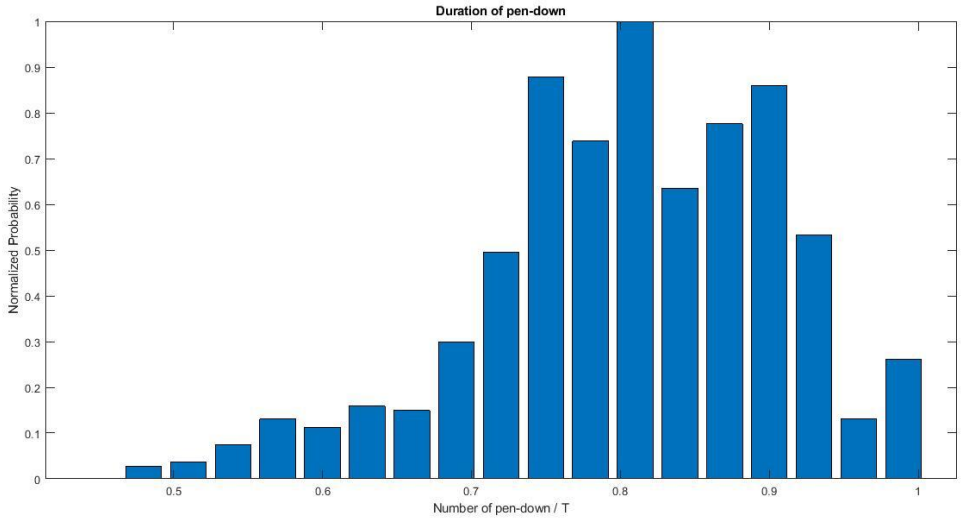
Total duration



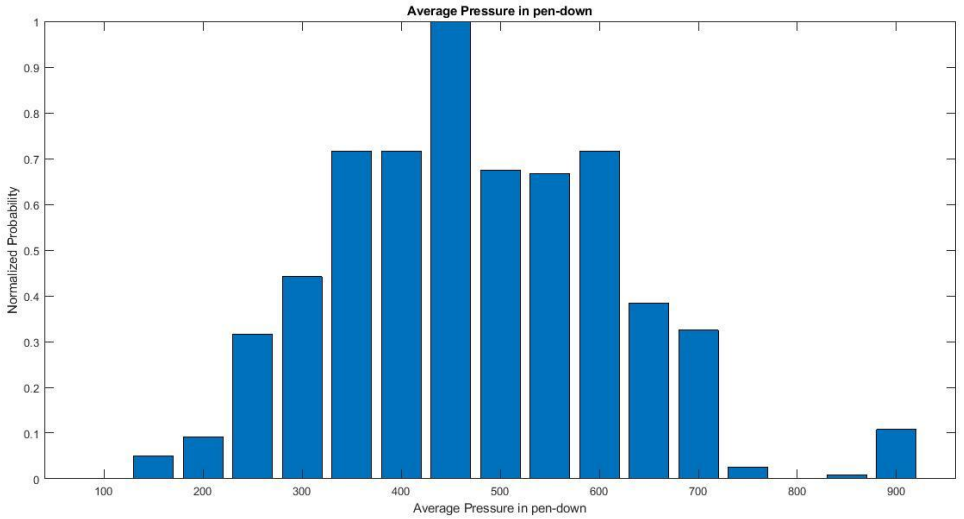
N pen-ups



T pen-down / T



Average P in pen-down



4. Performance Evaluation

We will evaluate the performance of our system according to the number of signatures N in the enrollment set (N=1, N=4 and N=12).

The similarity score between a query/test signature and the enrollment signatures (signatures in the database) will be the Euclidean distance between feature vectors (vectors with 4 parameters). The final score will be the average score of the N comparisons (comparison between the query/test sample and the N enrollment samples).

You have to develop the function Score=Matcher(test,Model) where:

- Score: is the final score of the comparison.
- test: is the feature vector of the query/test signature (1x4)
- Model: is a matrix containing the feature vectors of the signatures enrolled in the database. Therefore, this matrix contains Nx4 values in which N is the number of signatures enrolled for the claimed identity.

There are two cases to be analyzed:

Genuine Scores: scores obtained when you compare a signature with his real enrolled identity (claimed identity = enrolled identity). So these users should be accepted by the system. For each user you will use N signatures as enrolled samples and the rest for testing:

- For N=1 we will have SG=15 genuine scores.
- For N=4 we will have SG=12 genuine scores.
- For N=12 we will have SG=4 genuine scores.

For each of the scenarios (N=1,4,12) you have to save all the genuine scores into a matrix (with dimension 50xSG). Each of the three matrices will be stored into a .mat file with name: GenuineScores_N.mat.

Impostor Scores: scores obtained when you compare a signature with the enrolled samples of other users (claimed identity ≠ enrolled identity). So these users should be rejected by the system. In this case, we will compare one signature of each user (the first one) with the models of the rest of the users (excluding the genuine case). Therefore, we will obtain SI=49 impostor scores for each user and each scenario (N=1,4,12).

For each scenario (N=1,4,12) these impostor scores will be saved into a matrix with dimensions 50xSI (50x49). Each of the three matrices will be stored into a .mat file with name: ImpostorScores_N.mat.

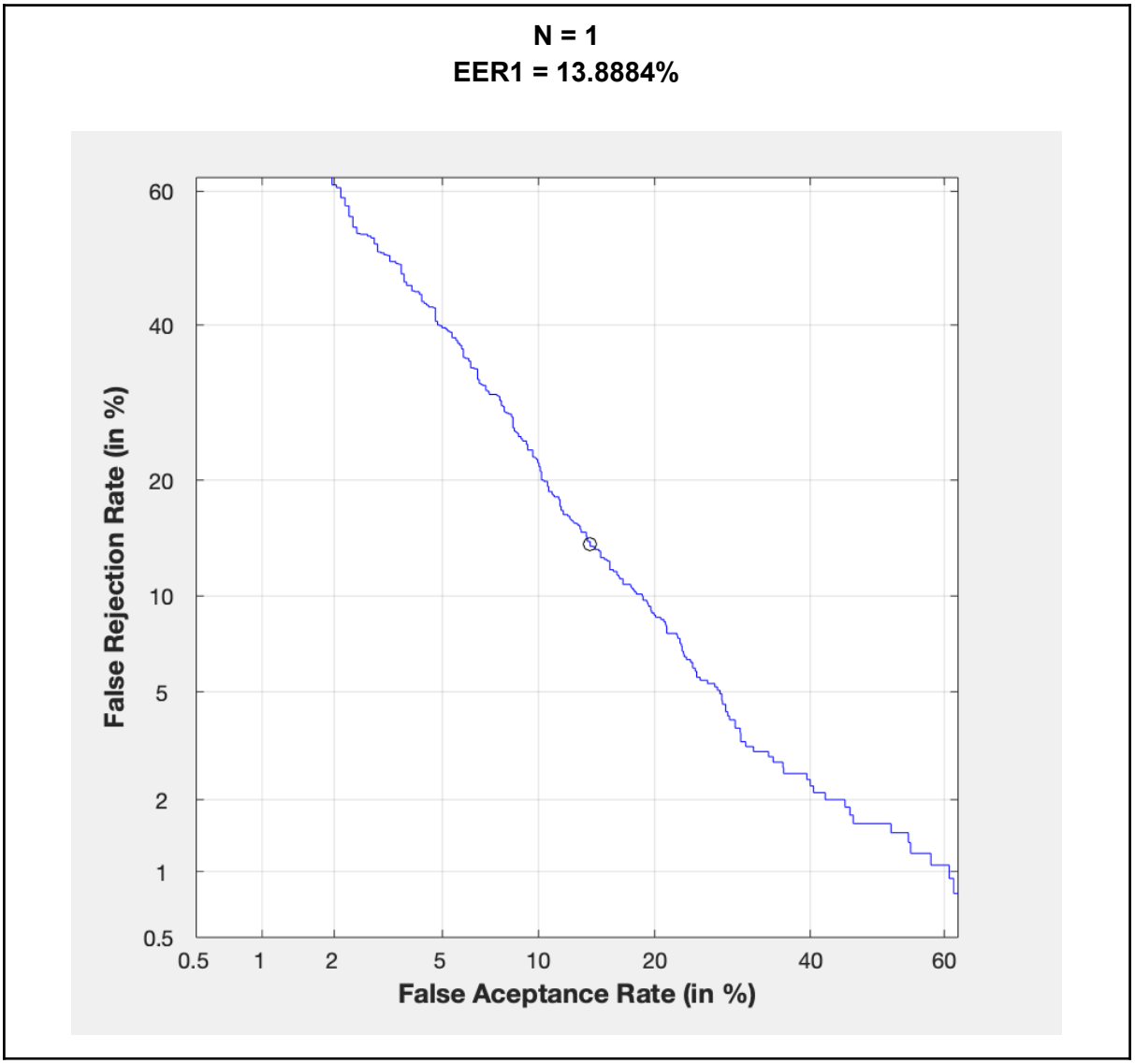
Once we obtain the genuine and impostor scores, we will evaluate the performance of our system for each of the three scenarios (N=1,4,12) as a function of: FAR/FRR, EER and DET curves.

To obtain these performance metrics you will have available the next functions:

[EER]=Eval_Det(GenuineScores, ImpostorScores, 'b')

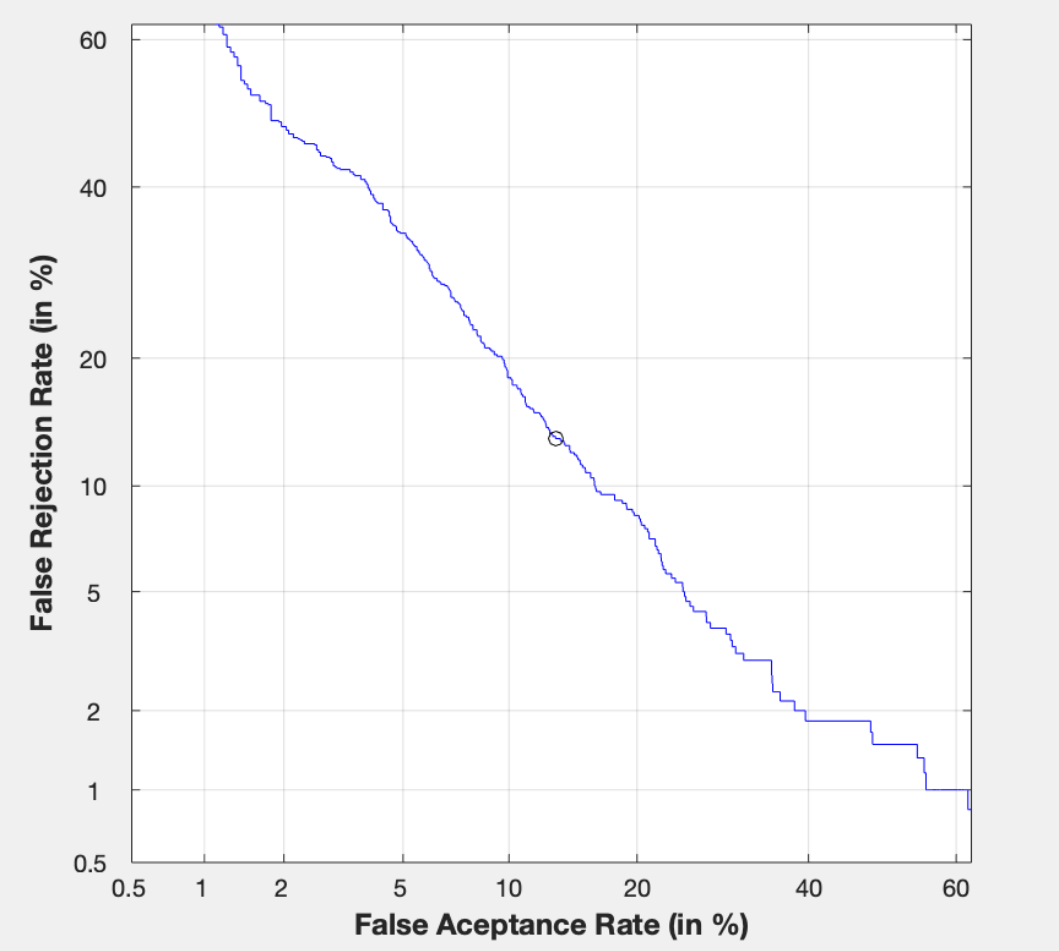
- EER: value of the Equal Error Rate (error when FAR and FRR are equal)
- GenuineScores: the scores from target or genuine comparisons. These scores are obtained after applying the following normalization: GenuineScores = 1./(GenuineScores_N+0.00000001)
- ImpostorScores: the scores form non target or impostor comparisons. These scores are obtained after applying the following normalization: ImpostorScores =1./(ImpostorScores_N+0.00000001)

QUESTION. Plot the performance graphics (DET curves) using the genuine and impostors score stored in their respective matrices (for each of the scenarios N=1,4,12). Indicate the EER value.

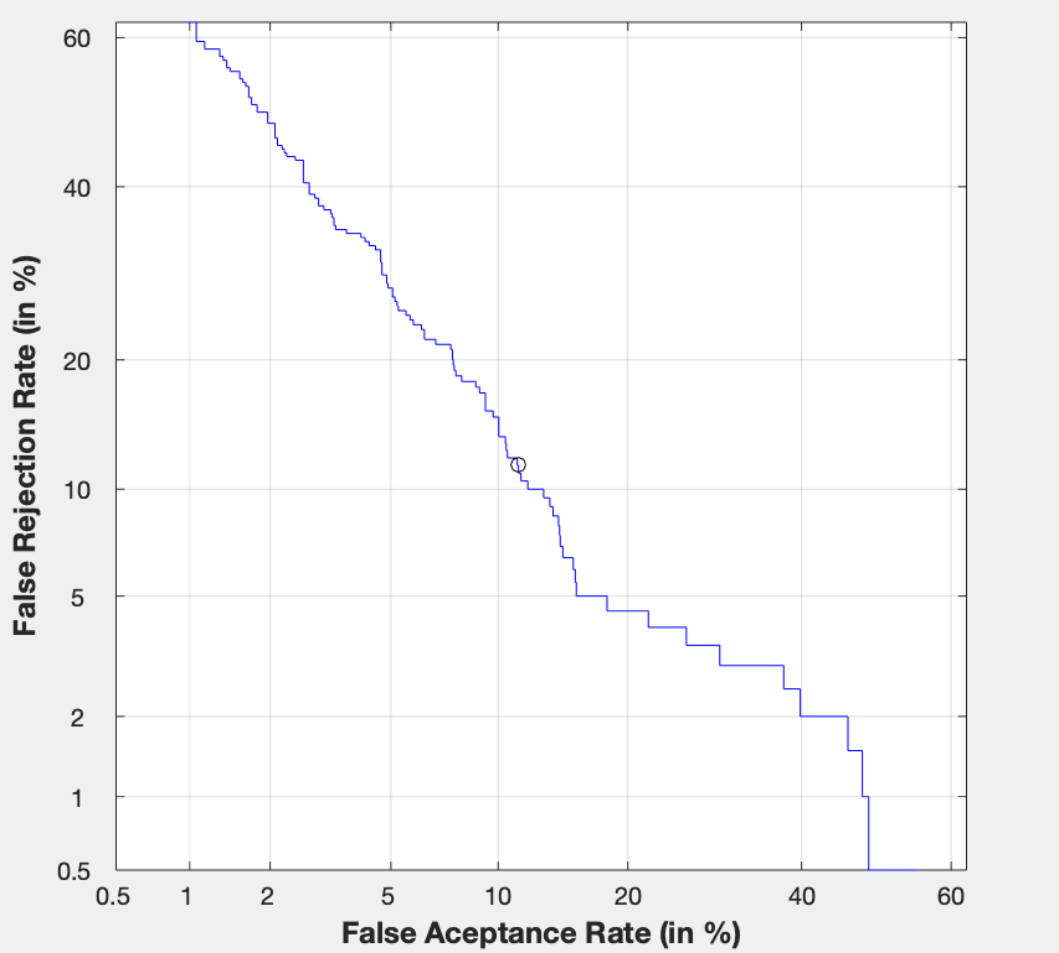


N=4

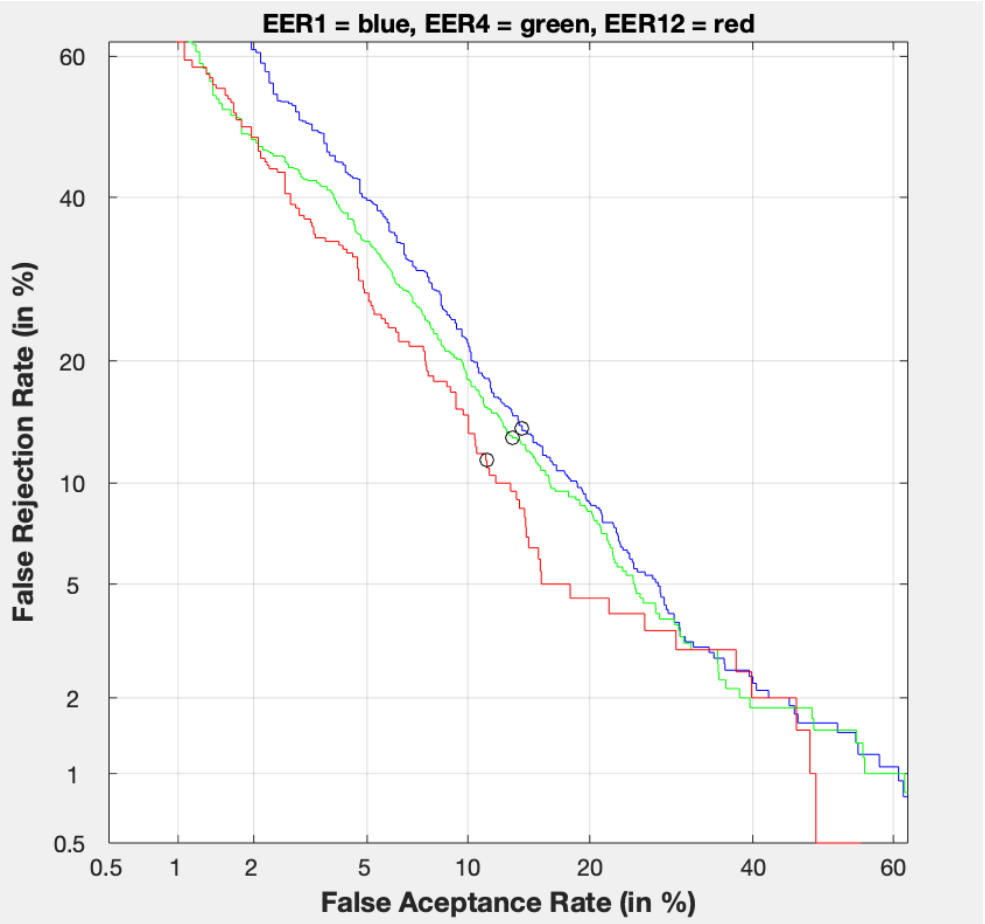
EER4 = 13.1667%



N=12
EER12 = 11.5%



QUESTION. According to the results, are they reasonable? What metrics are more illustrative? When do you obtain the best performance?



Signals look more or less reasonable. By the definition of DET, the ideal performance case is straight line. According to this 3 curves fit linearity which shows that the system is normal and the performance increases when the number of signatures increases too. Also the 4 user signature mode (green curve) is more illustrative as it has the lowest threshold value. Nevertheless, the difference between threshold values of other curves is not big which shows there is no prominent security system. The 12 user curve has the higher threshold which is the system with high security: therefore, it has the lowest EER value with a **11.5%** and the lowest false acceptance rate.

The graph shows that the number of signatures in the enrollment is directly proportional to the decreasing of ERR. Hence, the 12 user signature curve is the most robust and the closest model to the left corner which demonstrates the model with best performance.

Extra work 2

For this extra work, an online signature recognition system has been developed. In this case, the system is based on local features associated with time functions, concretely: the **x** coordinate, the **y** coordinate and the **pression.** Moreover, the Dynamic Time Warping (DTW) has been used in order to evaluate and match the signatures and 9 final features has been taken into account for the final process: the **x** coordinate, the **y** coordinate and the **pression** (as we mentioned before) and their first and second derivatives.

The DTW system developed, as the previous one, has been divided into 2 different parts: the **pre-processing** part and the **evaluation** part.

1. **Pre-processing** part: this part corresponds with the stored process, where all the 3 time functions or features of each signature are stored in a matrix called **BiosecurIDparametersDTW**. This process has been included in the ProcessBiosecurID function.

2. **Evaluation** part: this second step corresponds with the online signature recognition algorithm. Using a similar experimental protocol as the previous one (see **Evaluation_DTW** function), the evaluation process is computed through 3 main functions (each function has the same name as the name of the file):

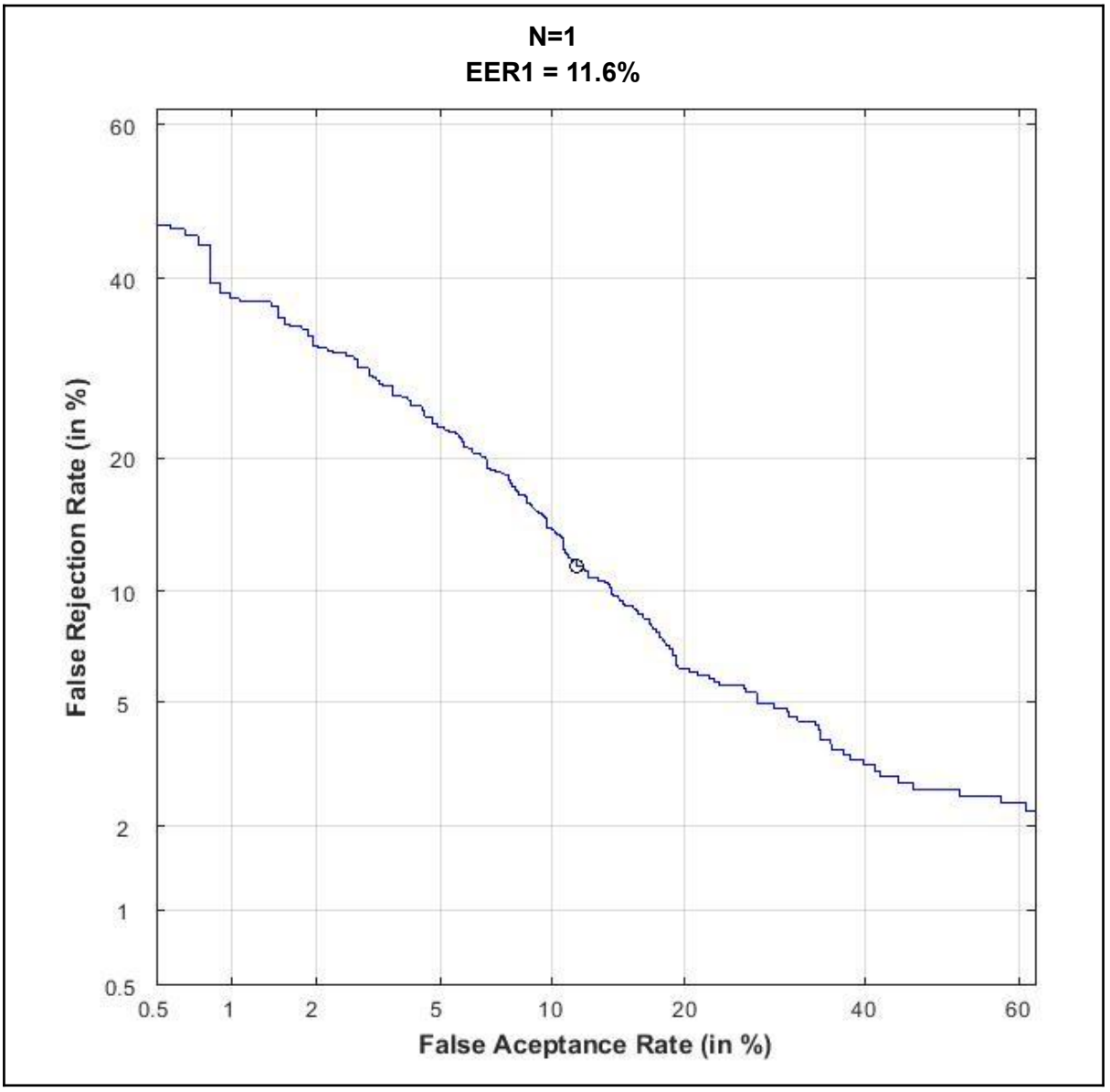
    a. **Matcher_DTW:** computes the score between the signatures and calls the functions below in order to obtain the final result. This function calculates the average score among all the model signatures with the test signature.

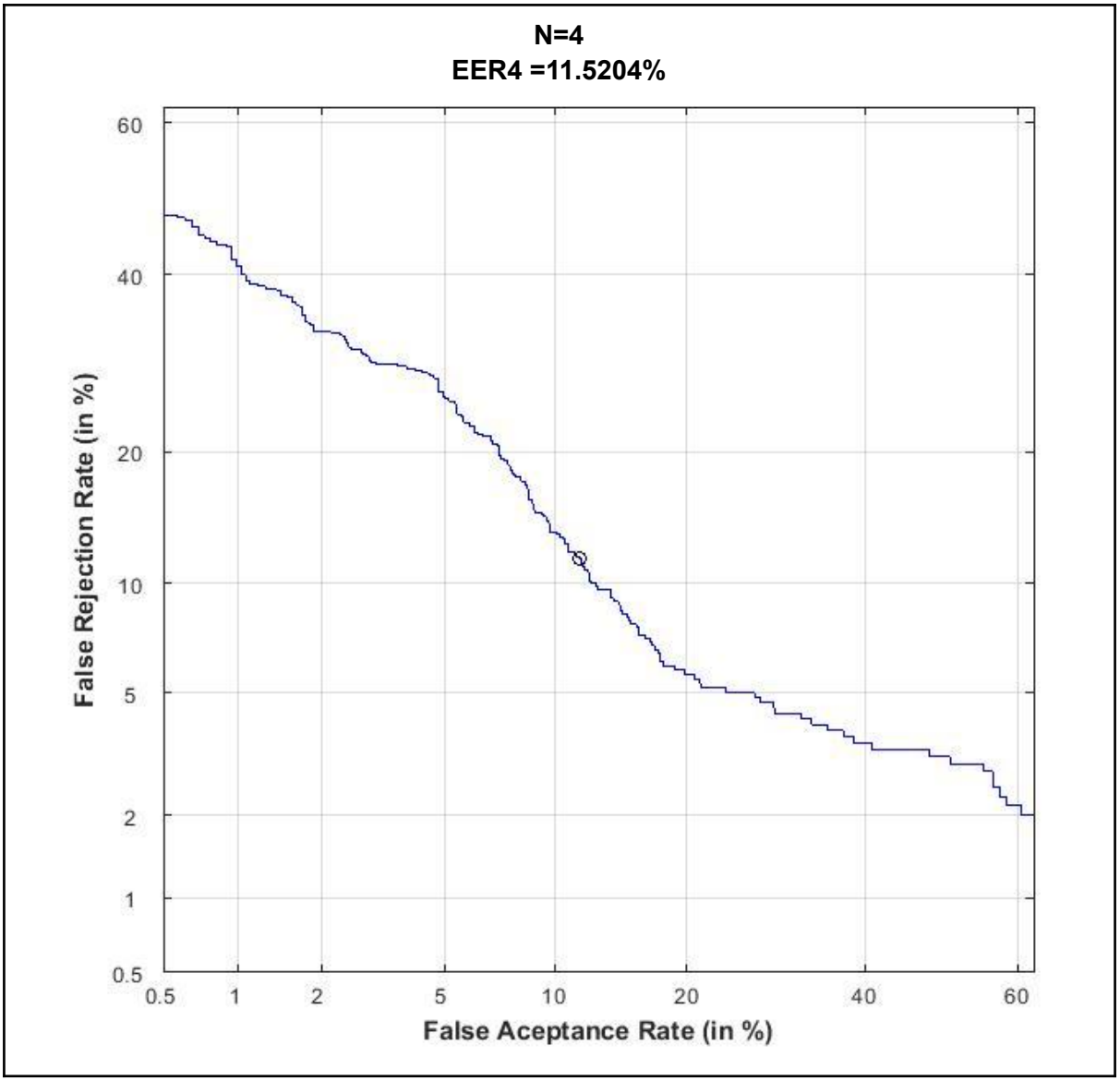
    b. **extract_dtw_features:** this function is in charge of calculating the corresponding derivatives of the features mentioned before. Moreover, all 9 features are stored in a struct.

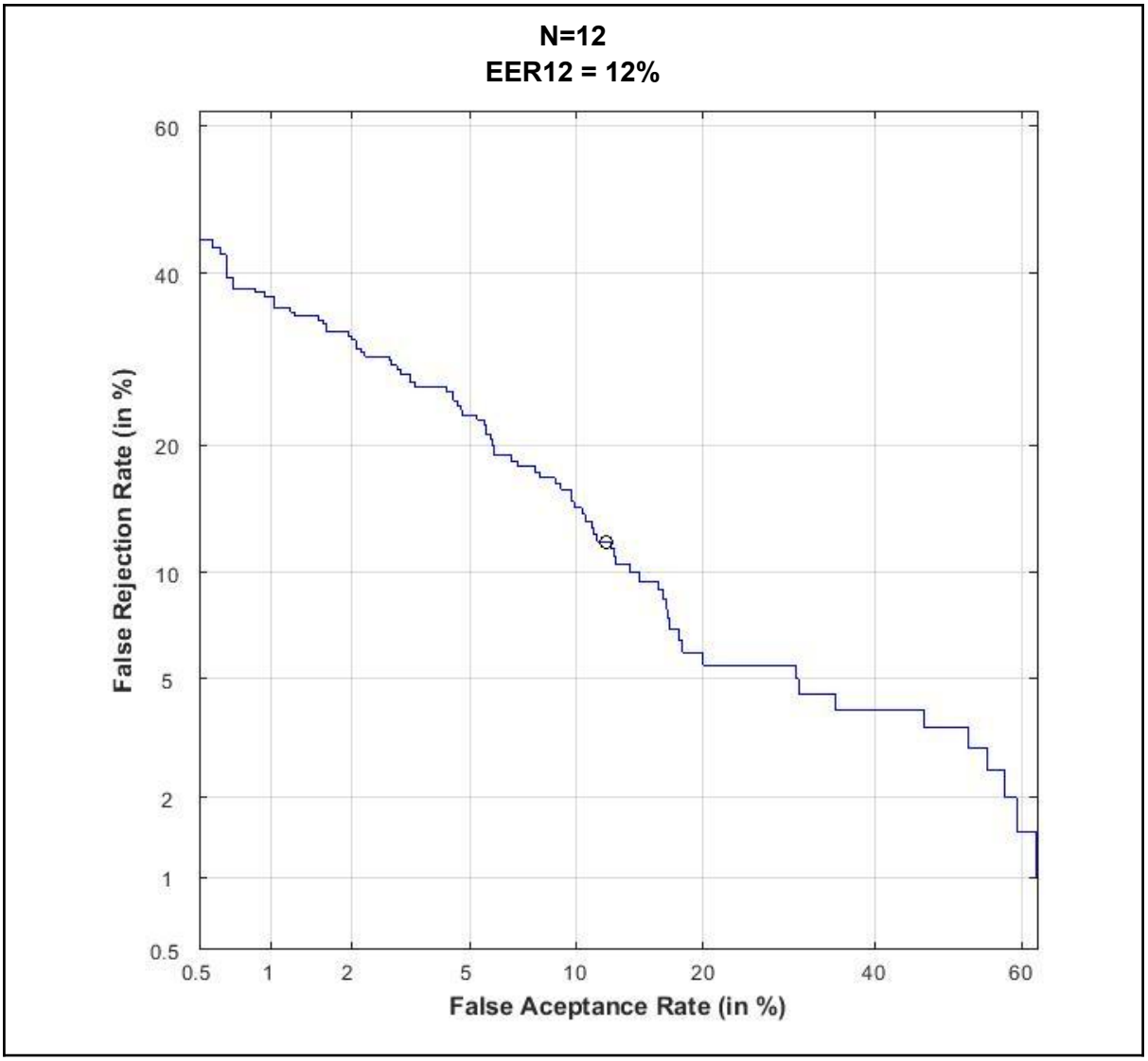
    c. **DTW_Score:** computes the dtw score by calling the dtw matlab function. Moreover, the score of each feature is calculated as mentioned in the lab assignment report: the equation to obtain the score of the 1vs1 time function comparison is $score=e^{-D/k}$, where D is the minimum accumulative distance obtain)after using DTW), and K is the number of aligned time samples. In this case, all 9 features from both signatures are processed and the final score is determined by calculating the average score from all of them.

Finally, the DTW function is executed via Evaluation_DTW.m in order to evaluate this new algorithm and to compare the results obtained with the previous experiment. As we can see in the images below, the EER performances are quite similar in comparison to the euclidean distance algorithm, achieving a **11.52% EER** for the N=4 experiment in the best scenario. Although the performances from both systems are between 11%-14% EER, we have implemented two different ways to evaluate and recognise online verification signatures using local features and time functions. Nevertheless, though both systems have similar results, the DTW system is quite slower in terms of computational cost due to the calculation

of the derivatives of the 3 features. Also, in this case the number of signatures in the train dataset doesn't play a crucial role because the performance achieved with N = 12 isn't the best one.




N=4
EER4 =11.5204%
False Rejection Rate (in %)
60
40
20
10
5
2
1
0.5
0.5
1
2
5
10
20
40
60
False Aceptance Rate (in %)


N=12
EER12 = 12%
False Rejection Rate (in %)
60
40
20
10
5
2
1
0.5
0.5
1
2
5
10
20
40
60
False Aceptance Rate (in %)