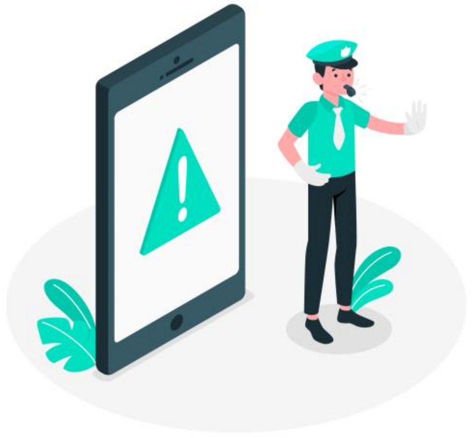
Dipartimento di Informatica
Sapienza Università di Roma

Biometric Project: BioPhone

Dario Aragona

Luca Podo

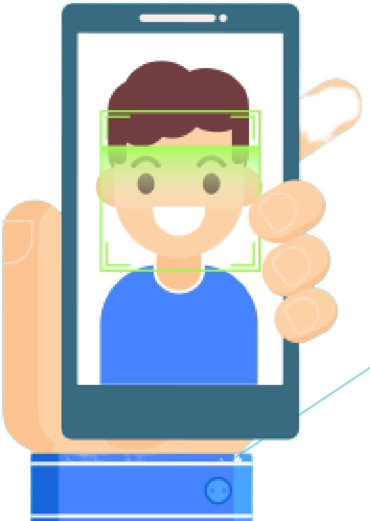


Index

- Problem & Solution
- How?
- Technological Overview
- Model Overview
- Feature selection and Model
- Back-End
- Font-end
- Data
- Evaluation
- Limits & Future developments

Problem

Smartphone pin once has been detected, can be exploited by some impostors to violate the system. This is because it is based on something the owner knows, the pin. Nowadays, biometric secure systems have been added to smartphones, for example: face recognition, fingerprint and so on. But sometimes these systems encounter some problems in some contexts, for example face recognition encounters illumination problems.

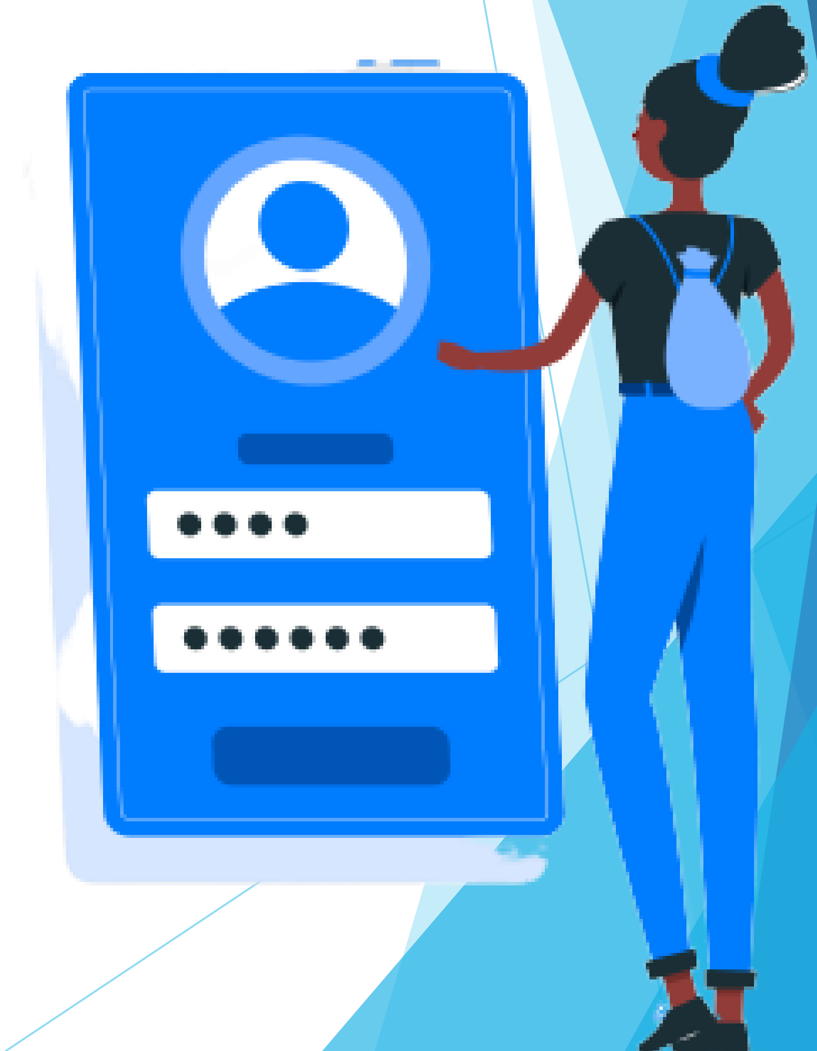


Solution

Our system tries to provide a very easy and quick method to verify the user using a double security check by using something that the user knows and something that the user is:

password + biometric data

The smartphone became the sensor to get biometric data based on how the user behaves while inserting the pin



How?

To reach our goal the system extracts information from accelerometer and from how fast the user clicks each number, getting the time between one number and the next

Then a model based on Novelty Detection try to predict if who inserted the password was the owner or not

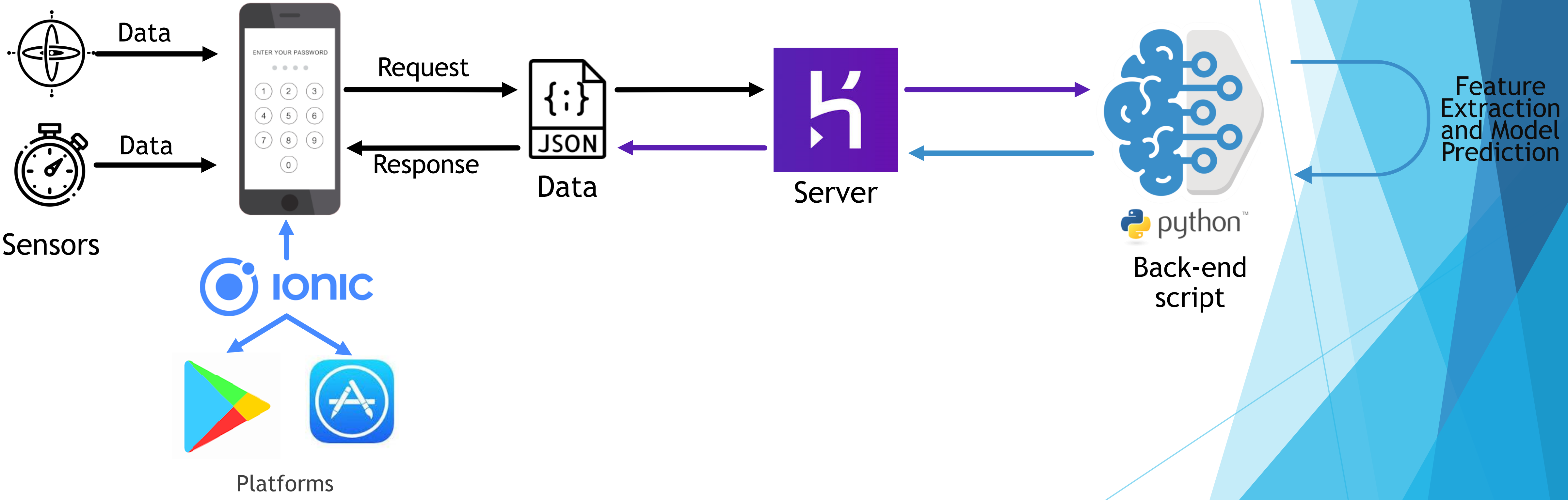
Technological Overview

Heroku is a cloud platform as a service that allows us to create a cloud server to process the data and make the prediction

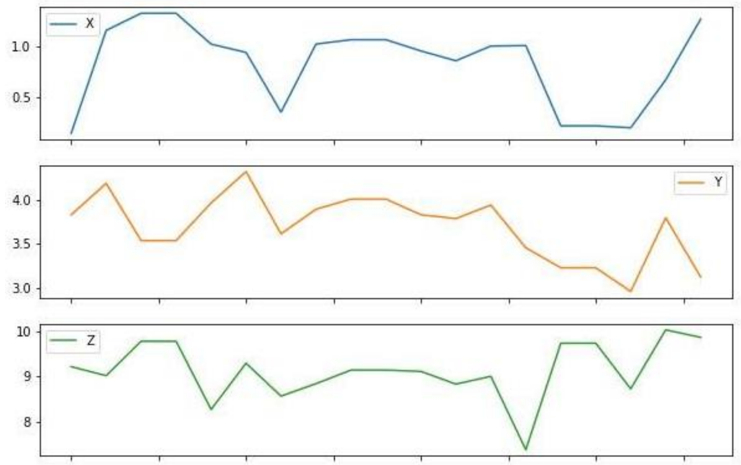
Python is the language used to write the backend script to make all the operations on the data received from the front-end

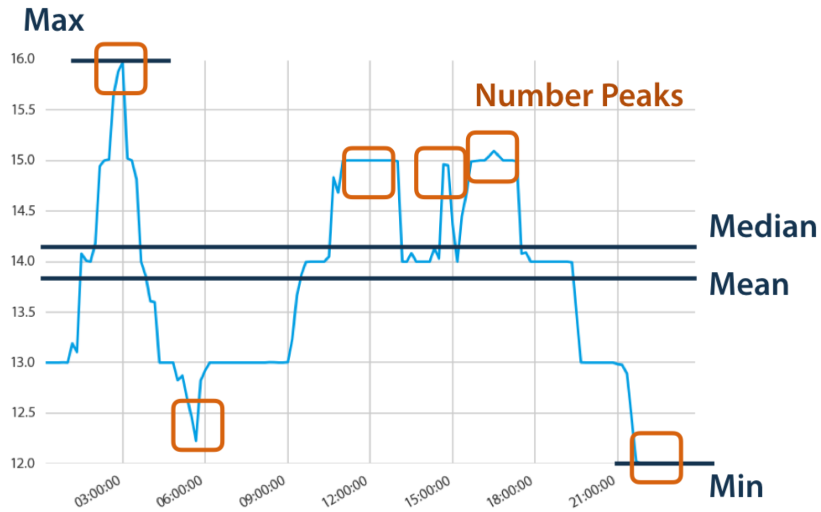
Ionic is a framework that allows to write hybrid app. We used Ionic to design the app and to interact with the sensors on the smartphone

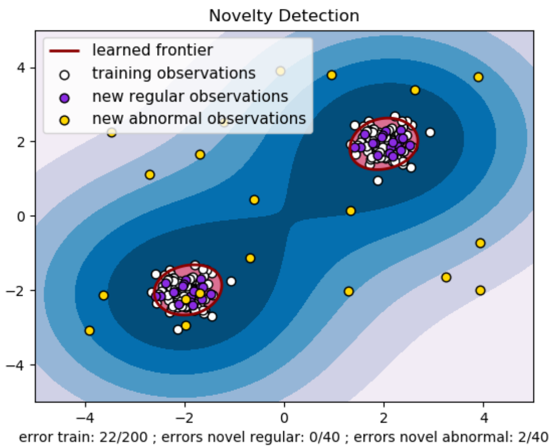
System blocks schema



Model Overview







Data:

In our system we use time series data retrieved by the accelerometer and from the typing speed of the user

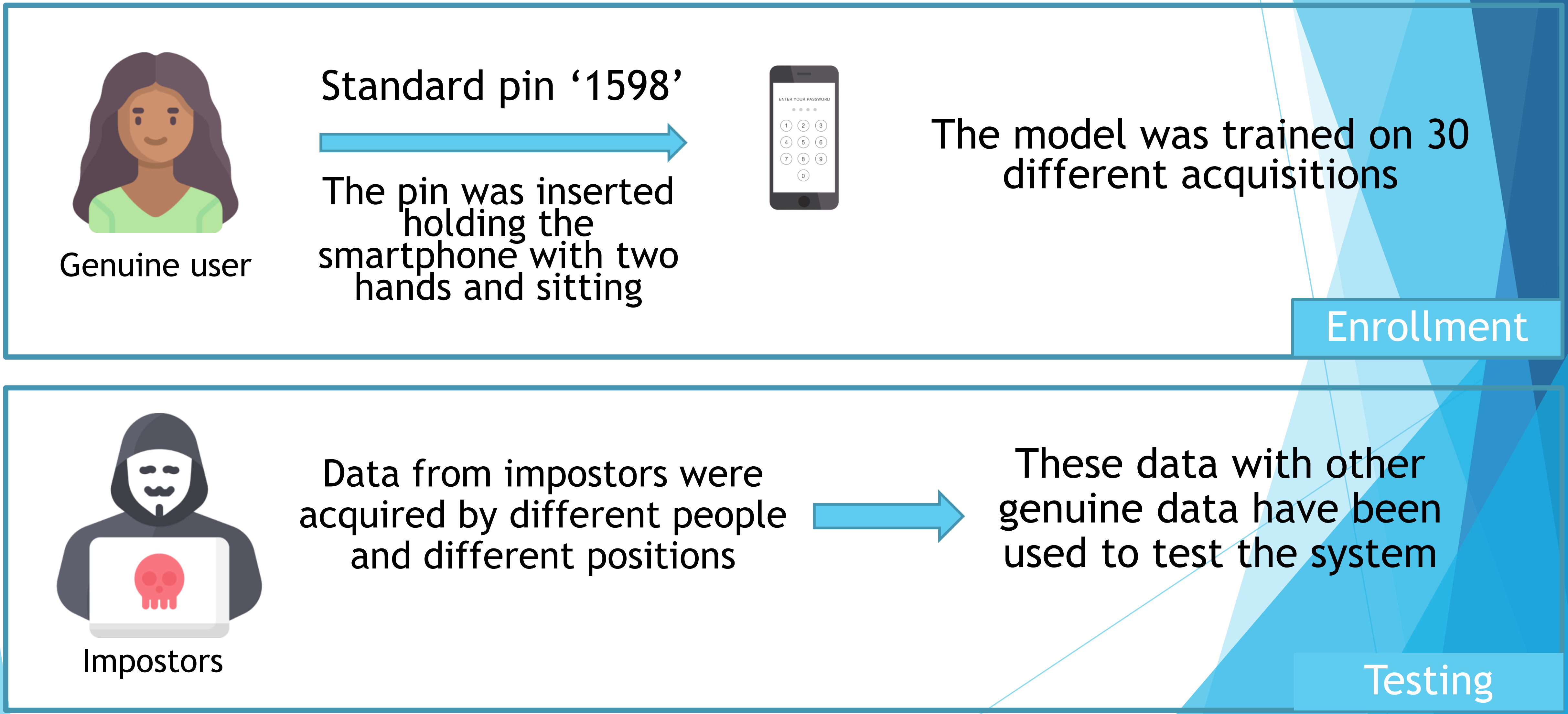
Feature extractions:

For the features extraction of the time series data, we used the library Tsfresh

Model:

To detect if who types is the right user, we use a novelty detector

Model training and testing schema



Genuine user

Standard pin ‘1598’

The pin was inserted holding the smartphone with two hands and sitting

The model was trained on 30 different acquisitions

Enrollment

Impostors

Data from impostors were acquired by different people and different positions

These data with other genuine data have been used to test the system

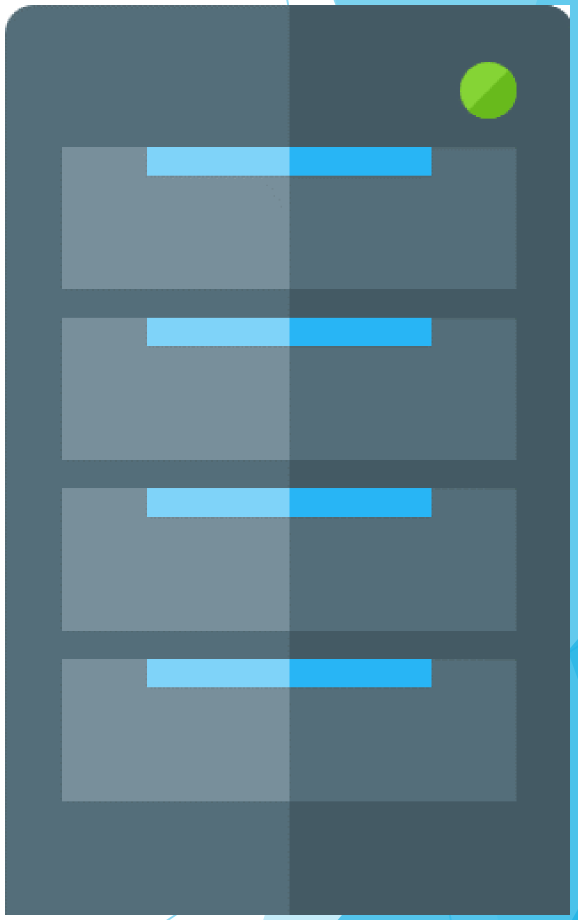
Testing

Back-end

The back-end of the system was developed by using python. The main functions of the server are:

- Receiving the request of authentication
- Process the data
- Send data to the model
- Send the result of the prediction

The script was listening for POST requests. When a new request comes from the front-end, it extracts, pre-processes the data and performs feature extraction.



Feature extraction and Model

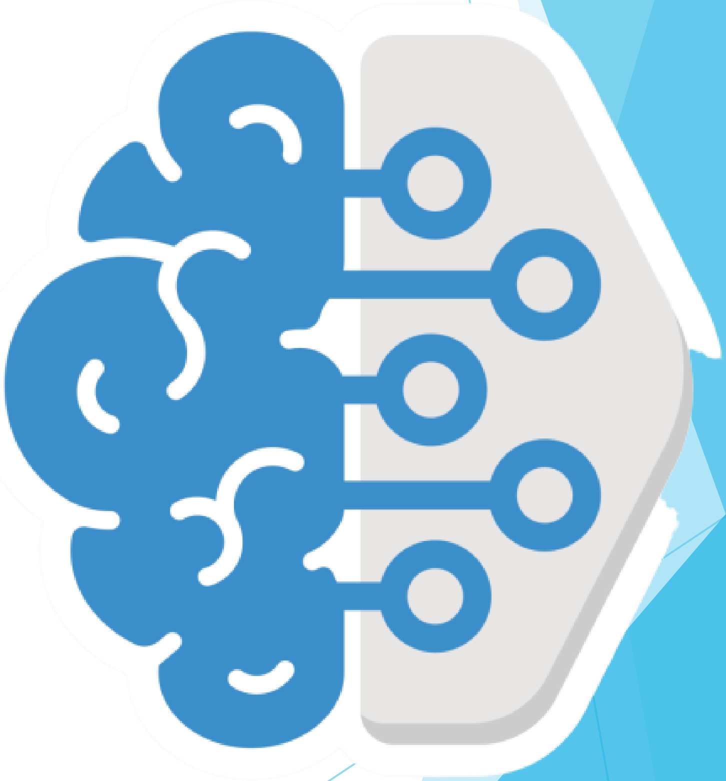
The data pre-processed by the script are processed by Tsfresh. It is a package that allows to extract 100s different features from the time series. TsFresh returns us the features vectors extracted

The features vectors are passed to the model based on LocalOutliersFactors, an unsupervised Outlier Detection. It measures how a sample is isolated with respect to the surrounding neighbourhood.

It returns [1] if it is not an anomaly, otherwise [-1]

In the Training phase we used the data collected during the enrollment to train the model to a specific data configuration. Than we used the test dataset to predict if the samples belong to the right user or not.

The online phase takes the data from the incoming request to predict.



Front-end

The front-end module was developed using hybrid to make it easy to export on android and ios.

The main functions of the front-end are:

- Retrieves data from accelerometer
- Retrieves data from typing speed
- Send the request to the server
- Show the results from the prediction

How we collected data

The data used to train the model were:

- Accelerometer data
- Typing speed of the user

Data about the accelerometer were collected every 50 milliseconds by starting at the first number pressed until the last fourth number, because the pin used was 4 digits

Data about the typing speed were calculated as the time between a number an the next pressed. So we got three different times for the three intervals

Example of how the time interval data are collected during PIN typing

CODE 1598

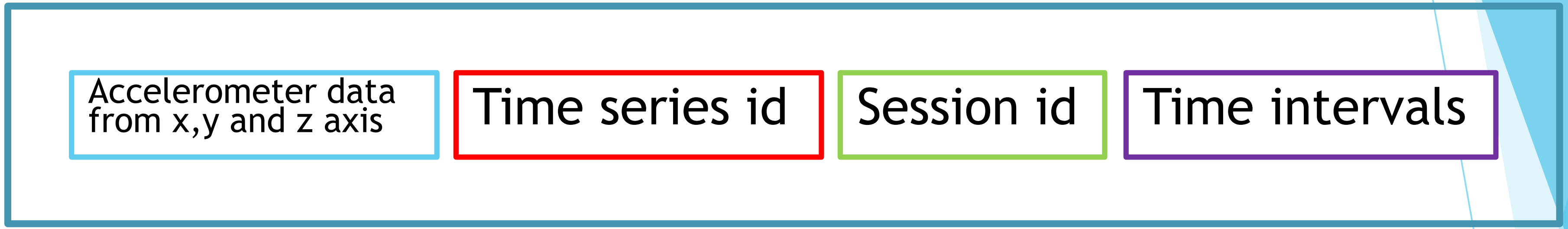
1 Accelerometer's data start to be registered

5 First time interval saved (Time1)

9 Second time interval saved (Time2)

8 Third time interval saved (Time3) and stop accelerometer acquisition

Data structure



Accelerometer data from x,y and z axis

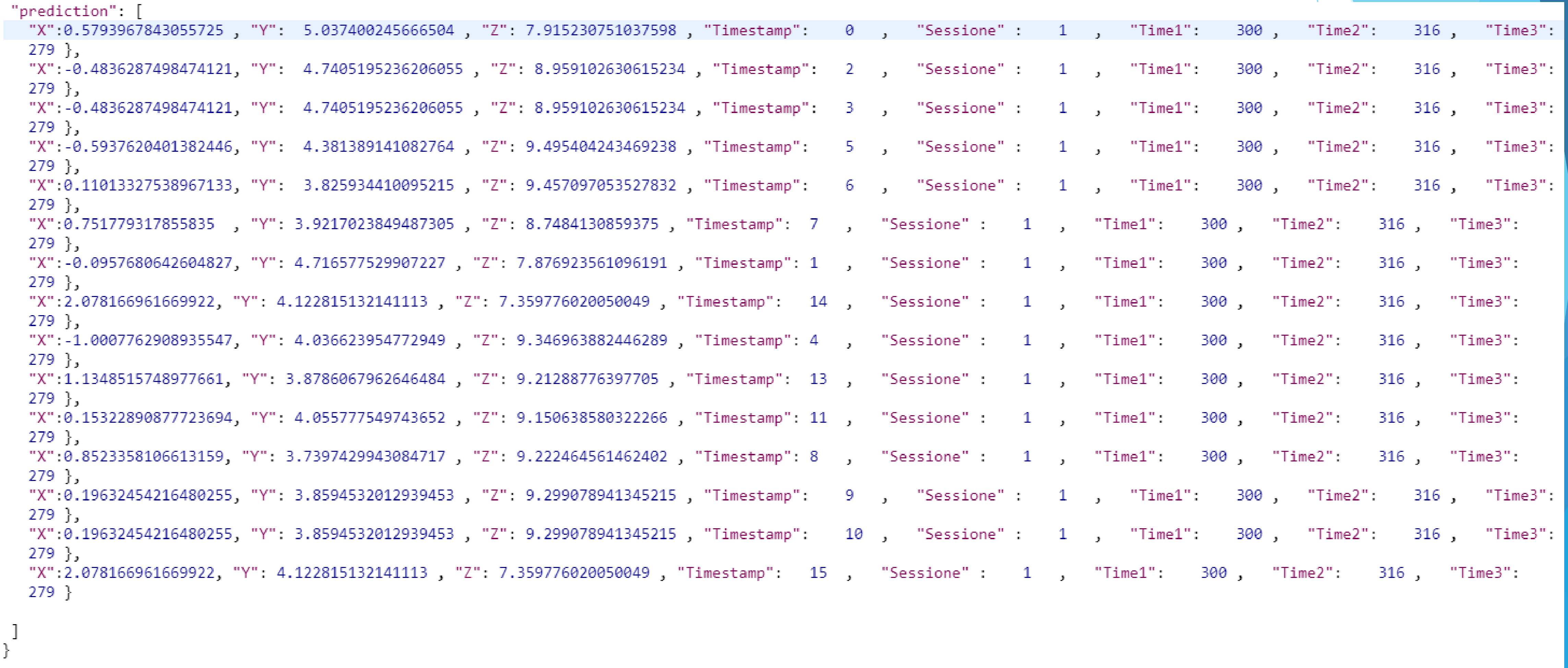
Time series id

Session id

Time intervals

1. *Accelerometer data from x,y and z axis* are the data values returned by the smartphone sensors
2. *Times series id* is the value used to identify the temporal order of the acquisition in a session
3. *Session id* is used to aggregate the data belonging the same insertion session
4. *Time intervals* are the intervals between each number pressed

Data example

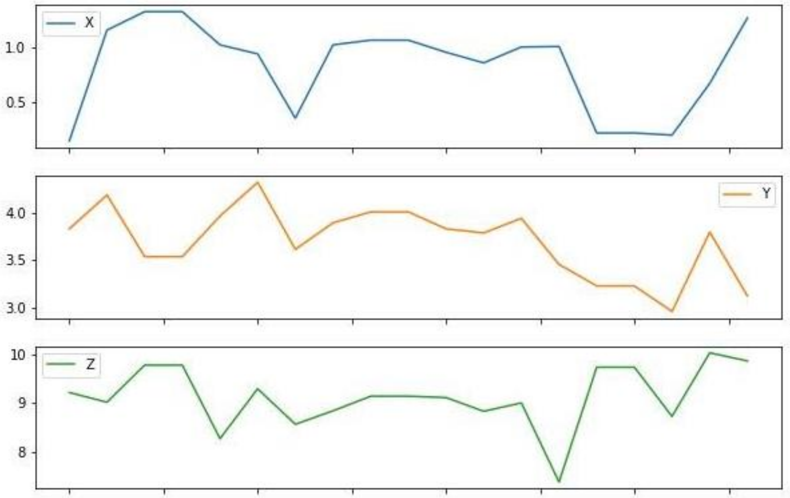


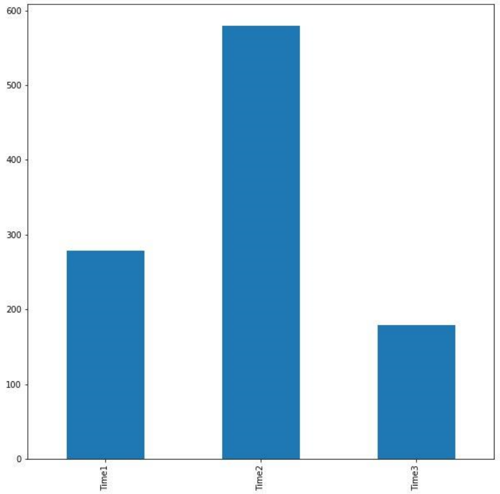
```
 "prediction": [
   "X":0.5793967843055725 , "Y":  5.037400245666504 , "Z": 7.915230751037598 , "Timestamp":    0   ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":-0.4836287498474121, "Y":  4.7405195236206055 , "Z": 8.959102630615234 , "Timestamp":  2   ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":-0.4836287498474121, "Y":  4.7405195236206055 , "Z": 8.959102630615234 , "Timestamp":  3   ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":-0.5937620401382446, "Y":  4.381389141082764 , "Z": 9.495404243469238 , "Timestamp":   5   ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":0.11013327538967133, "Y":  3.825934410095215 , "Z": 9.457097053527832 , "Timestamp":   6   ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":0.751779317855835  , "Y": 3.9217023849487305 , "Z": 8.7484130859375 , "Timestamp":  7   ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":-0.0957680642604827, "Y": 4.716577529907227 , "Z": 7.876923561096191 , "Timestamp": 1   ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":2.078166961669922, "Y": 4.122815132141113 , "Z": 7.359776020050049 , "Timestamp":   14  ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":-1.0007762908935547, "Y": 4.036623954772949 , "Z": 9.346963882446289 , "Timestamp": 4   ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":1.1348515748977661, "Y": 3.8786067962646484 , "Z": 9.21288776397705 , "Timestamp":  13  ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":0.15322890877723694, "Y": 4.055777549743652 , "Z": 9.150638580322266 , "Timestamp": 11  ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":0.8523358106613159, "Y": 3.7397429943084717 , "Z": 9.222464561462402 , "Timestamp": 8   ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":0.19632454216480255, "Y": 3.859453201293945З , "Z": 9.299078941345215 , "Timestamp":   9   ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":0.19632454216480255, "Y": 3.8594532012939453 , "Z": 9.299078941345215 , "Timestamp":   10  ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 },
   "X":2.078166961669922, "Y": 4.122815132141113 , "Z": 7.359776020050049 , "Timestamp":   15  ,   "Sessione" :   1   ,   "Time1":    300 ,   "Time2":    316 ,   "Time3":
   279 }

 ]
}
```

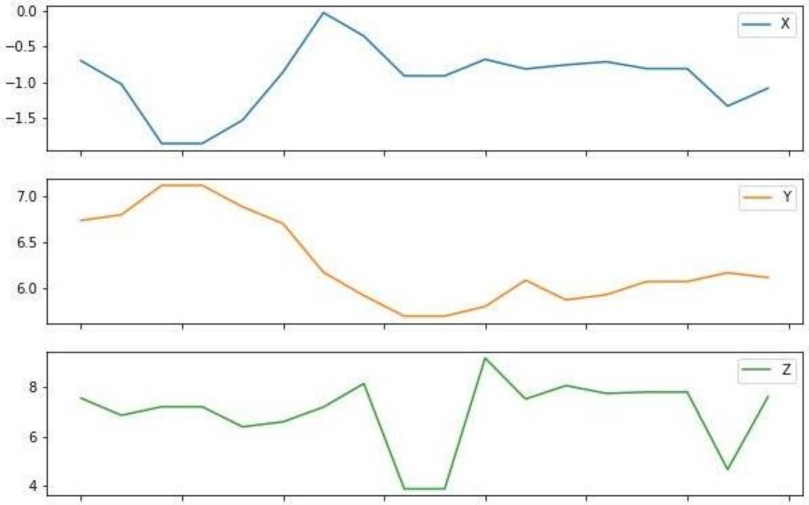
Data visualization

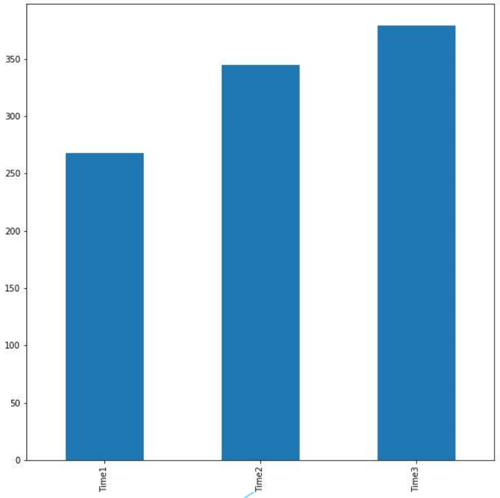
Genuine Data





Impostor Data

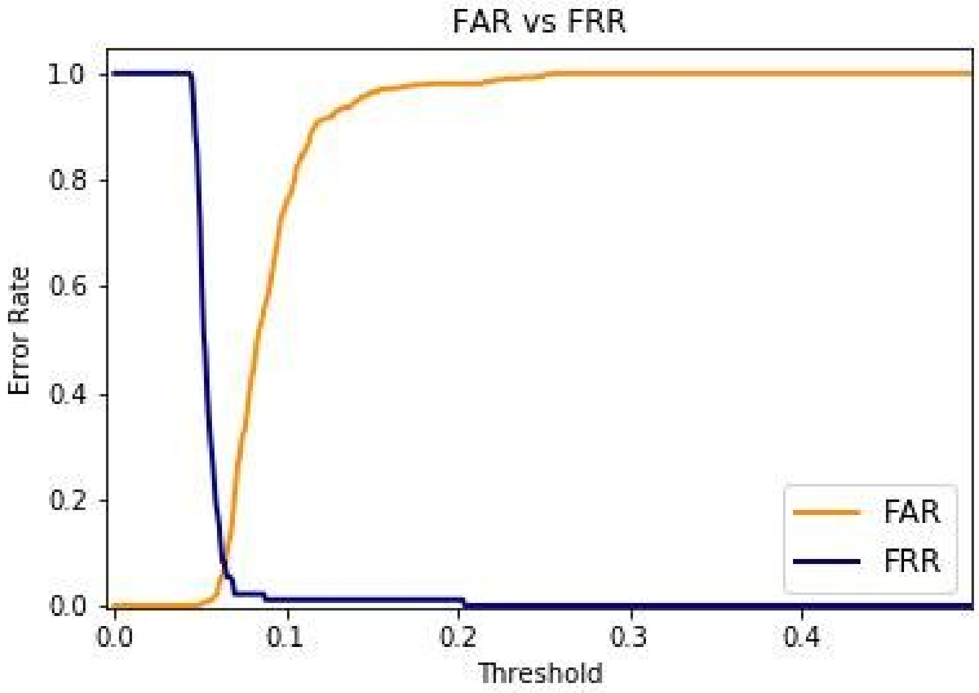




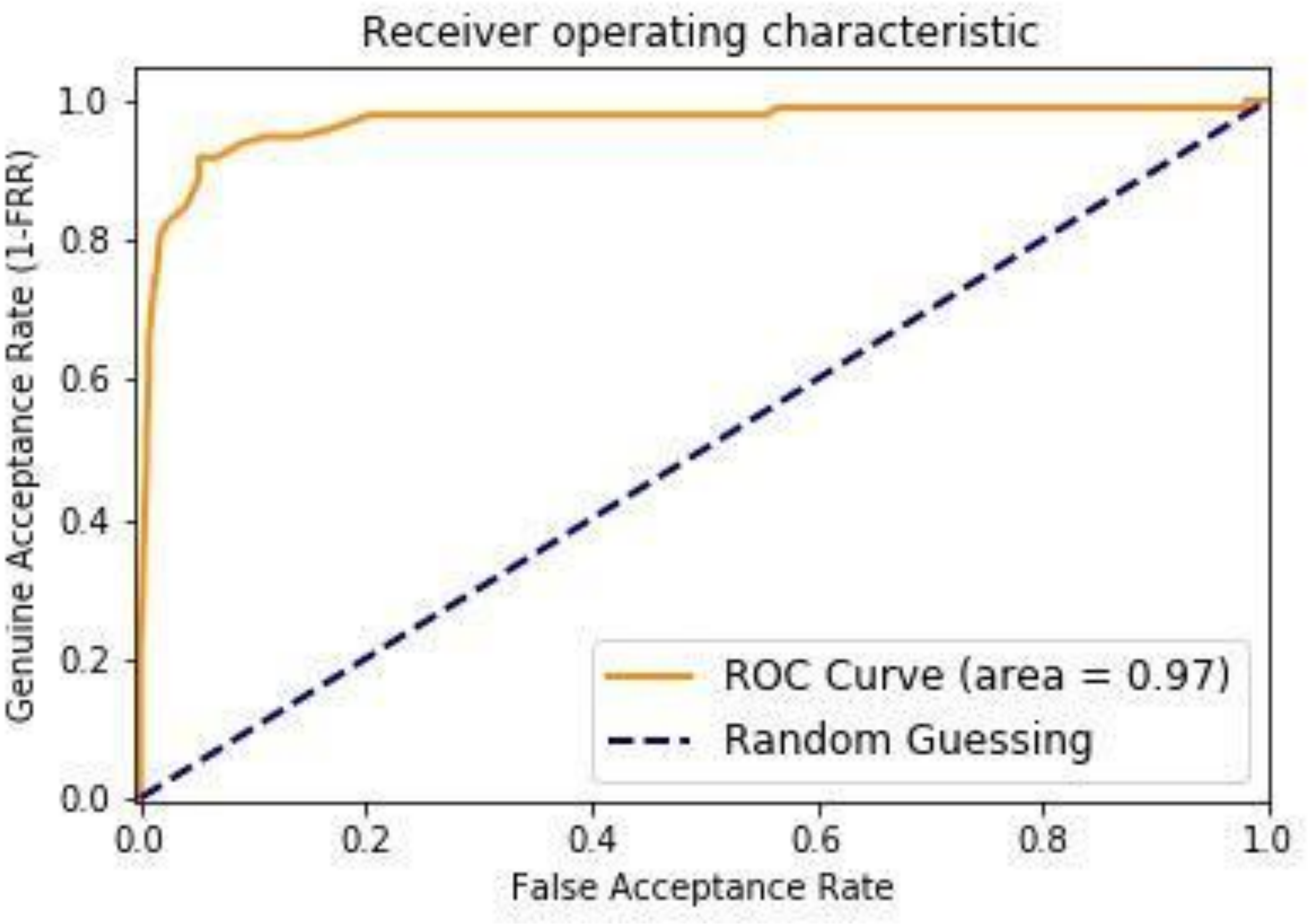
Evaluation - DET

During the testing phase, we have used K-Fold on a range of different acceptance threshold to detect the one associated with EER.

In our case, the value of threshold was 0.064



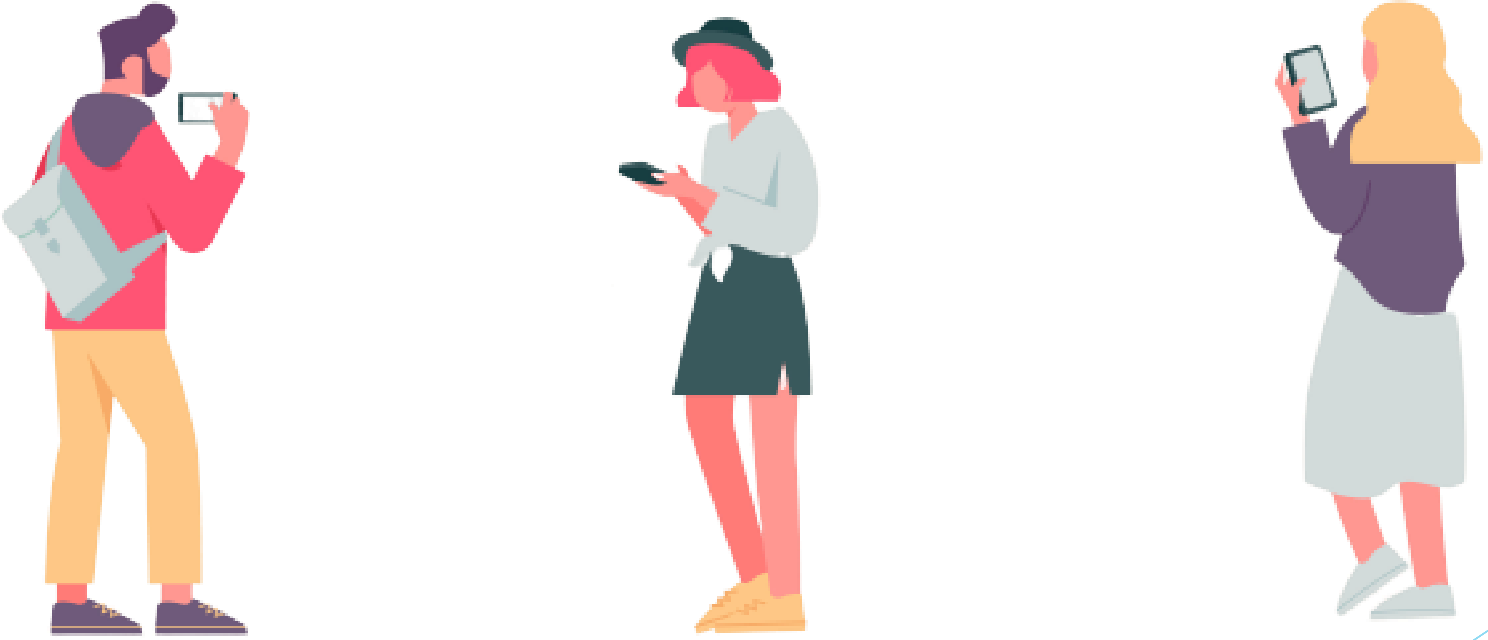
Evaluation - ROC



Limits

Possible limits of the system in this state of art are :

- The way a user holds the smartphone, if with one or two hands
- The position of the user can create different values if he is sitting or not
- Screen orientation
- The system is on cloud



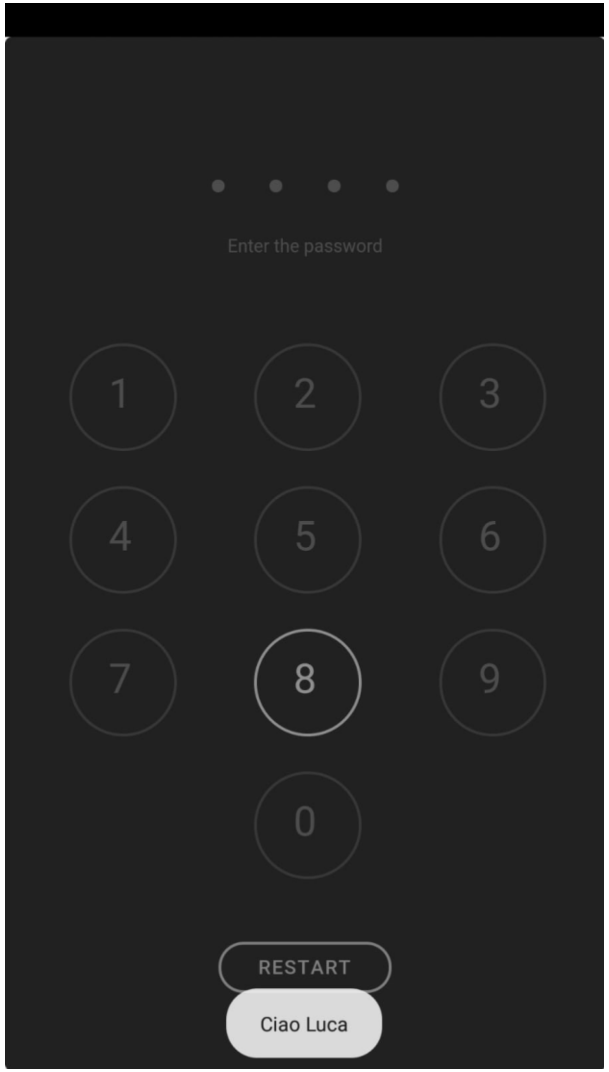
Future developments

The system in this state of art was tested without considering any differences in these different contexts.

Future developments could include:

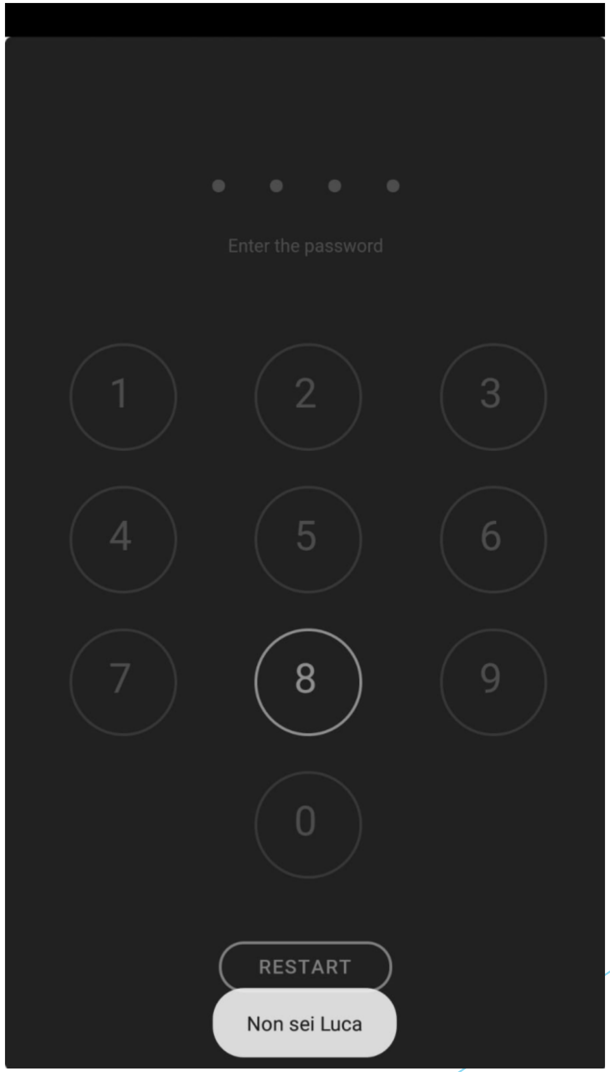
- A calibration system for one and two hands and for different positions
- Adding new sensors data
- Prediction on device
- New considerations about the system used while a person is moving

Screenshots



Ciao Luca

Genuine attempt



Non sei Luca

Impostor attempt