Hackathon challenge 2022

People counting with Radar and ML(Hackathon)

About this document

This document is about solution for this year hackathon challenge. The problem was people counting with radar and with use of machine learning algorithm.

# Scope and purpose

Scope of this problem was detecting number of person based on signal we receive from our radar(device) and using different machine learning techniques to count persons that are in the range of radar.

Purpose of this assignment is to find the cheapest and the easiest way to count number of peoples in the small room. It can be used to determine number of people and to warn them if it is getting crowded.

# Intended audience

Intended audience is jury that will grade our work.

Table of contents

[About this document 1](#_Toc104479899)

[Table of contents 1](#_Toc104479900)

[1 Collecting the data and preprocessing steps 2](#_Toc104479901)

[1.1 Machine learning model 2](#_Toc104479902)

[1.1.1 Results 3](#_Toc104479903)

[1.1.1.1 Further improvements 3](#_Toc104479904)

[Revision history 4](#_Toc104479905)

# Collecting the data and preprocessing steps

1. Only use the Add-Ins menu to format your document.
2. Please read the Getting Started guide to learn how to use the macro’s and styles in this template.

|  |
| --- |
|  |

1. Data received from radar

We did several tests and different configuration of the setup. We did recordings with different number of peoples, obstacles and interference. We obtained data where peoples were moving with different speed. We attempted different statical positions and we were careful to be in a 3m range of the radar to get meaningful data. Our recordings lasted for around 10 to 20 seconds.

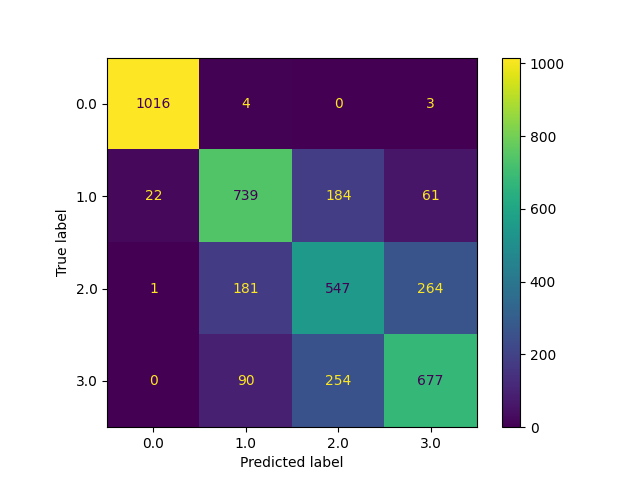
* Preprocessing steps

First we extract range-Doppler map from data. Then we were dealing with the external noise and we managed to extract it and remove it from the data. We decided to sum results that we gathered from each receiving antenna. Then we scaled the data and did the SMOTE oversampling. We got the idea for many different features such as max value of the frame, the sum of all values that are in the matrix which represents the image, we used values on y-axis, etc. We did the PCA to reduce the dimension of our data. We tried to have as many different features that don’t have big correlation rate.

## Machine learning model

We decided to use XGBoost model for our classification problem. That model is very common when we have a classification problem because of speed of the algorithm and built-in regularization. We used GridSearchCv to detetminate values of hyperparameters and we got learning rate = 0.2, max depth = 5 and number of estimators = 2000. We used f1-score to valuate our model.

### Results



**Figure 2. Confusion matrix**

We got the f1-score 0.73 on our test data.

#### Further improvements

Geather more data, improving feature selection, more testing

1. A Reference. See the code examples at [www.infineon.com](http://www.infineon.com/)

Revision history

| Document version | Date of release | Description of changes |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |