



UNIVERSITÀ
DI TRENTO

A system for estimating crowd density based on Wi-Fi probe request frames

Department of Information Engineering and Computer Science

Supervisors

Fabrizio Granelli
Daniele Miorandi

Student

Samuel Bortolin

Academic year 2019/2020



About this Project

External internship at **U-Hopper**:

- Big Data Analytics
- Business Intelligence
- IoT solutions
- Artificial Intelligence solutions





Problem Statement

Badly handled demand in company that provides services to physical customers can lead to overcrowding and inefficiency of the services

- Inefficient and badly organized service leads to **higher costs**
- It is important to **detect and monitor crowds** to reduce the risk of COVID-19 spreading during this global pandemic period



State of the Art

- Analysis of different methods for estimating crowd density
- Many fields of application and several implementations

Infrared sensors, LSE, treadle switch-based systems, Video methods,
Audio methods, Wi-Fi, Bluetooth, BLE, LTE, Radar, RFID approaches



Why Wi-Fi Solution?

- High diffusion of Wi-Fi devices
- Low-cost implementation
- Real-time data transmission
- Customer privacy ensured
(GDPR compliance)



Standard 802.11 → Management
frames → Probe request frames



Research Statement

Is it possible to **continuously estimate the density of the crowd**
in a place of interest based on the Wi-Fi probe request frames?



Achievements

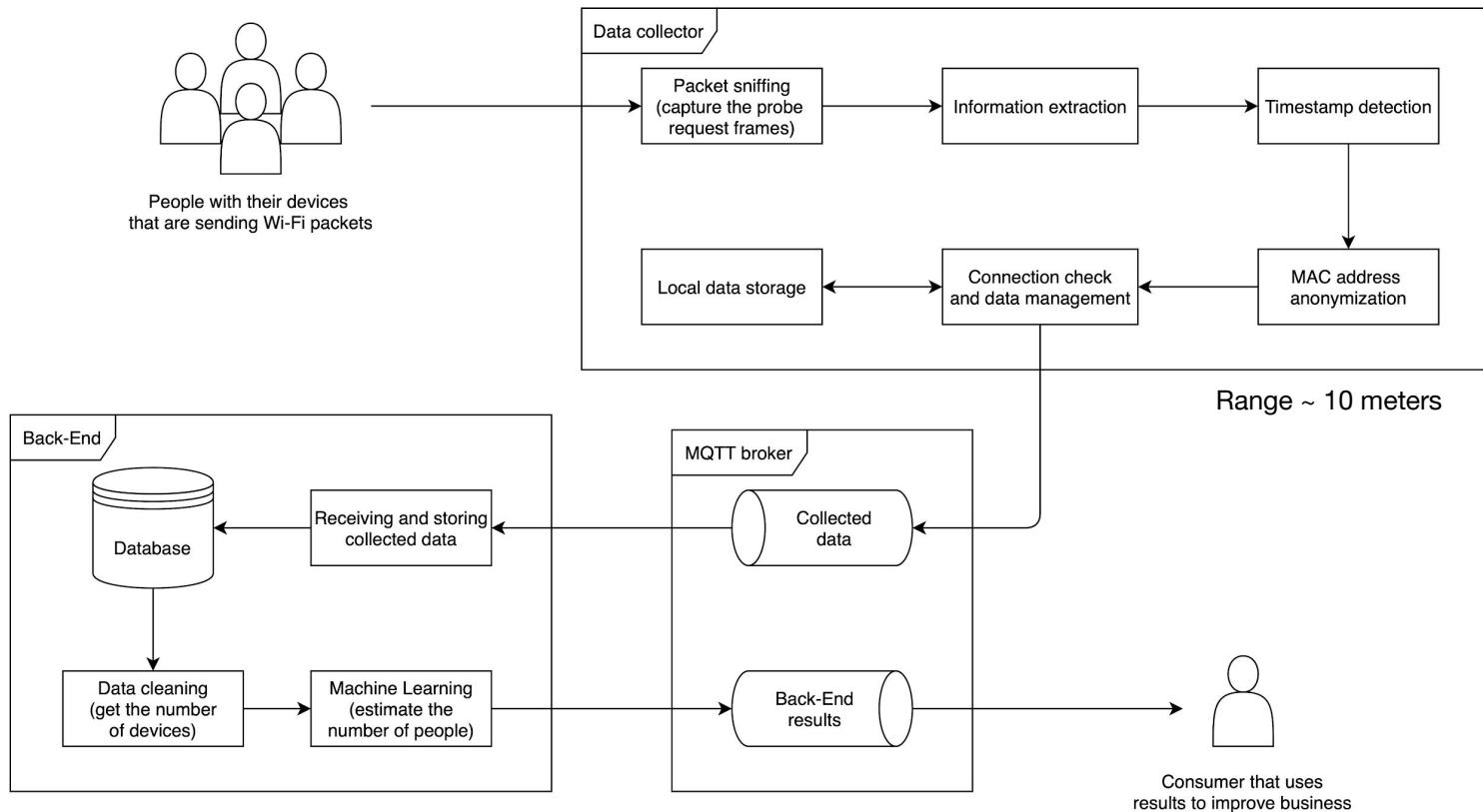


Designed and implemented a system for this problem that could work in several contexts

Tested the system in a Cafe and collected
4 weeks of data and manually-annotated ground truth (the actual number of people)

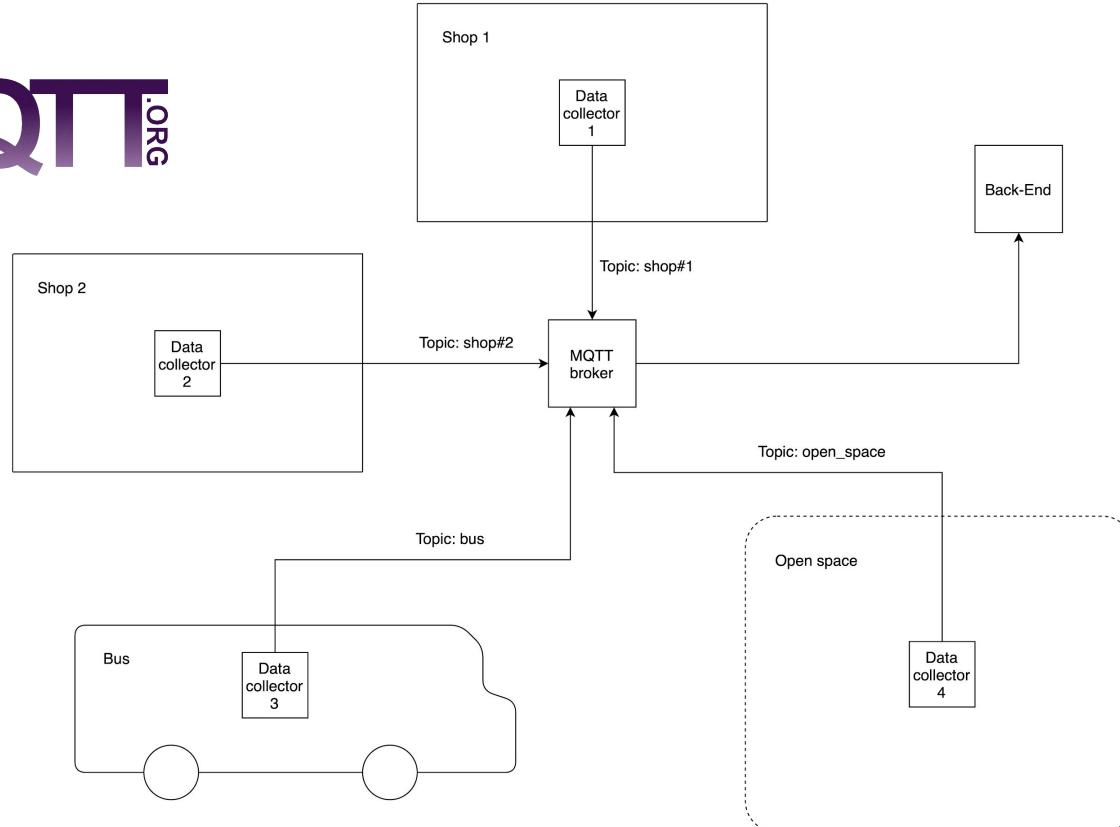


System Architecture



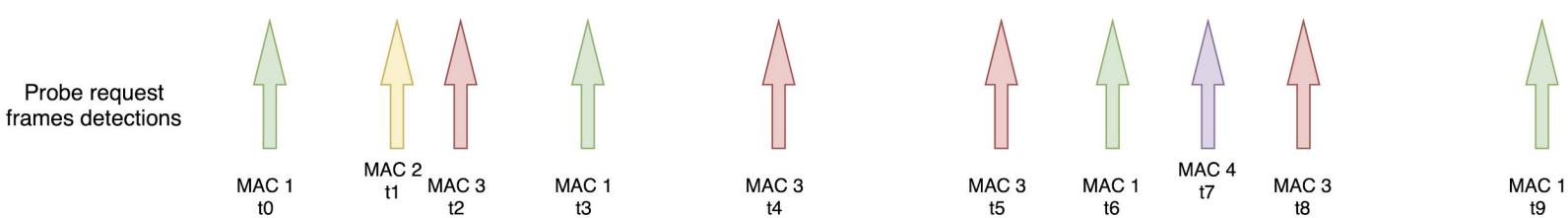


Scalable Architecture



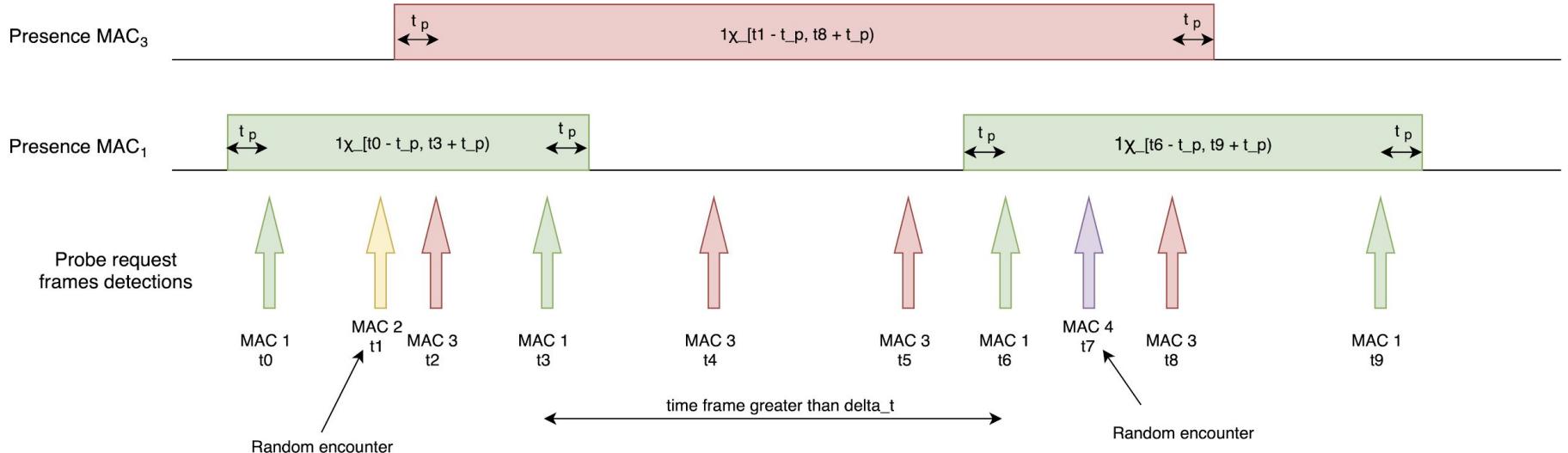


Presence of Devices





Presence of Devices



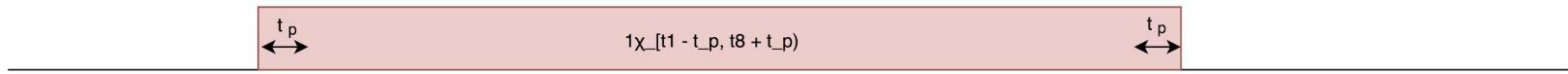


Presence of Devices

Presence of devices
 $= \sum_i \text{Presence MAC}_i$



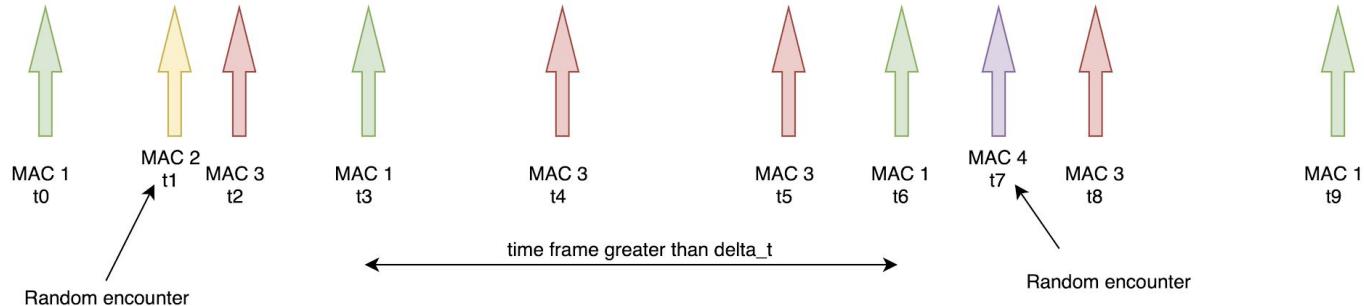
Presence MAC₃



Presence MAC₁



Probe request
frames detections





Feasibility Test at Home

3 days of data collection

65928 probe request frames captured

12 home devices detected

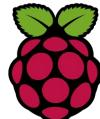
2 main range of RSSI:

- -35 ÷ -69 dBm in the kitchen
- -71 ÷ -91 dBm not in the kitchen

Feasibility of the method for
detecting devices in the area



System Validation



RaspberryPi in a Cafe where I annotate manually the ground truth

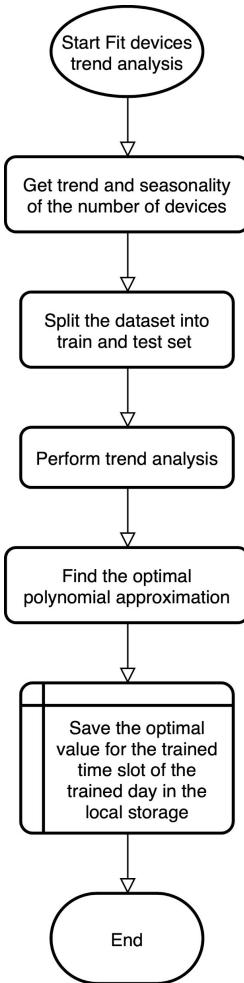


mosquitto broker of U-Hopper cloud infrastructure

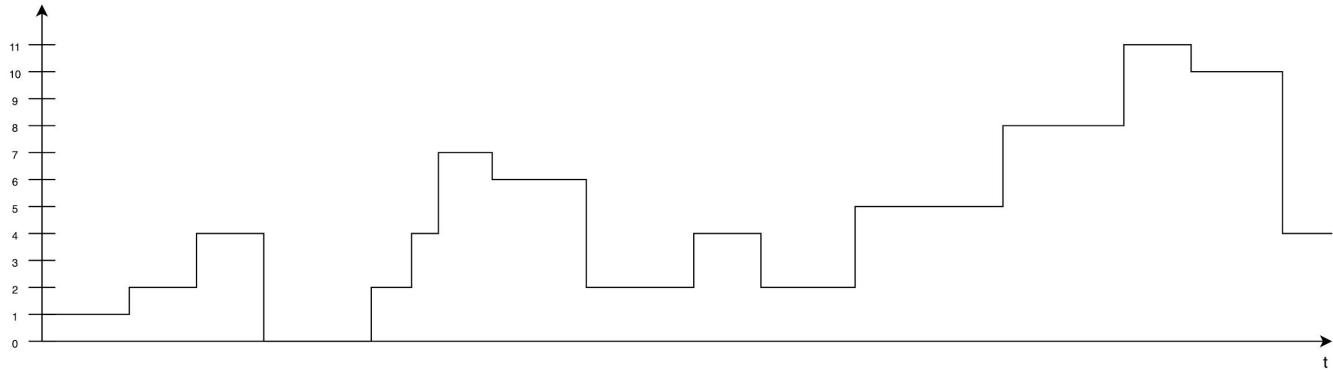


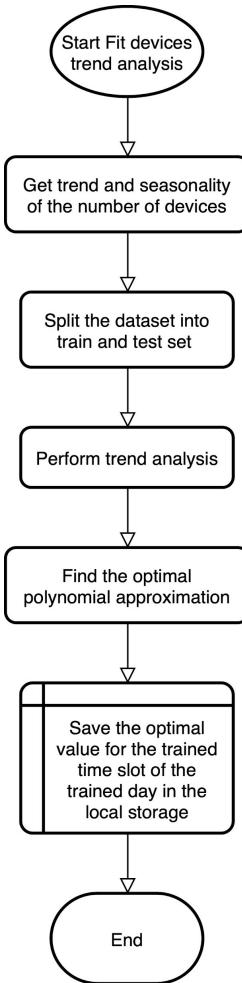
paho receiver and mongoDB on U-Hopper cloud infrastructure

Analyzer and Estimator on my PC to use on the data and the annotated ground truth to test accuracy and reliability of the proposed system

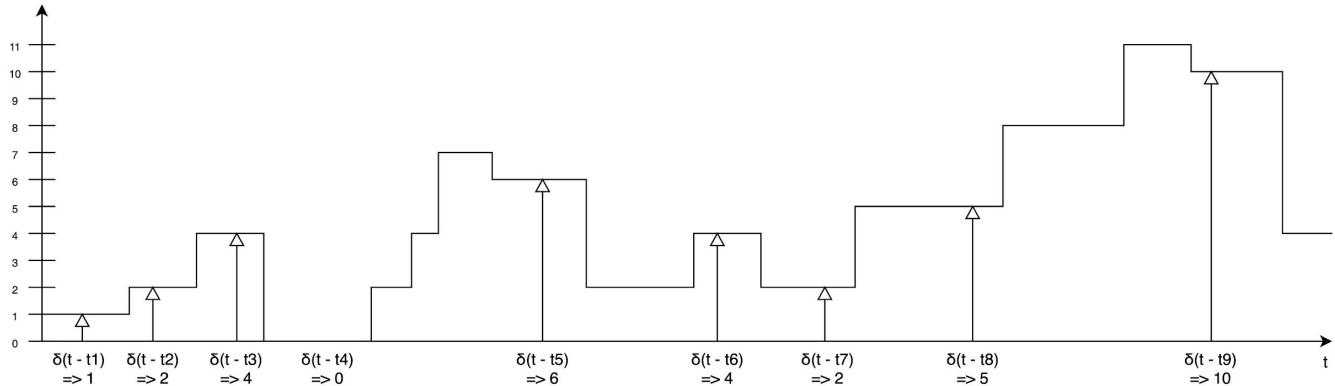


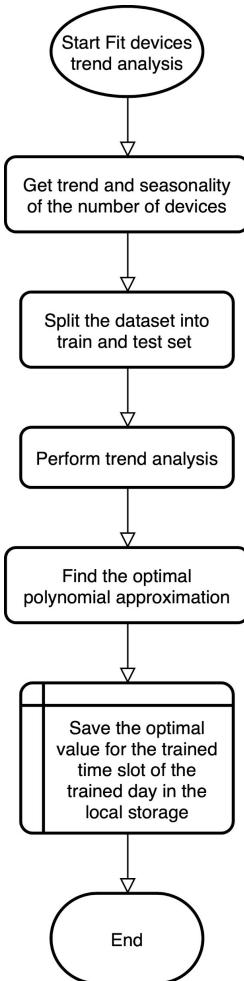
People present
 $g(t)$



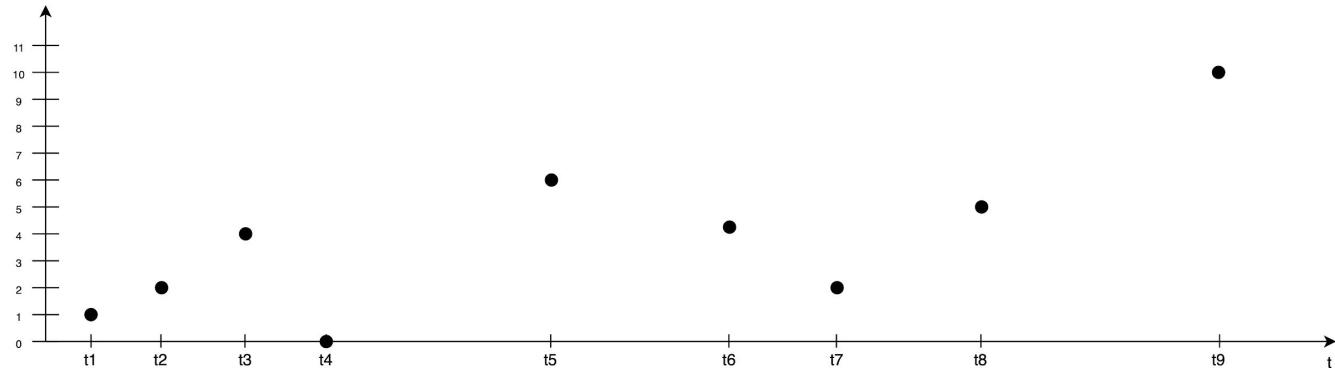


People present
 $g(t)$





Random look sampling of $g(t)$
=> Ground truth (t)





Test Results at Cafe



4 weeks of data
collection
(24 days)

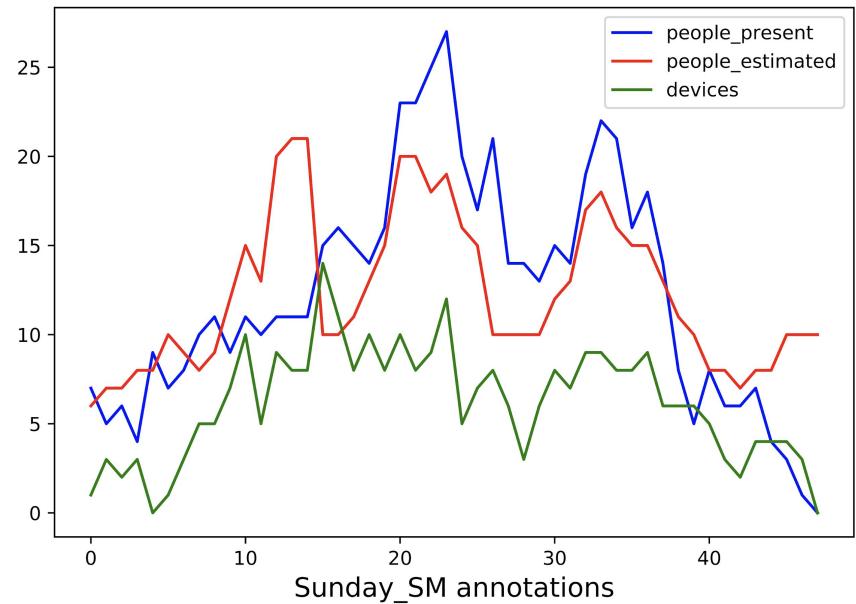
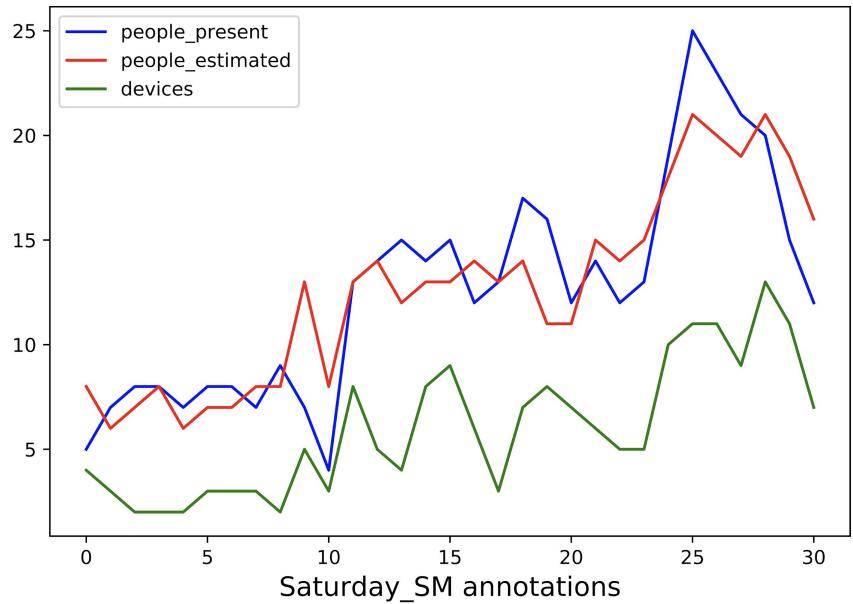
1270 manual
annotation of
ground truth

861979 probe
request frames
captured
(~ **560 MB**)

38771 unique
MAC addresses
detected



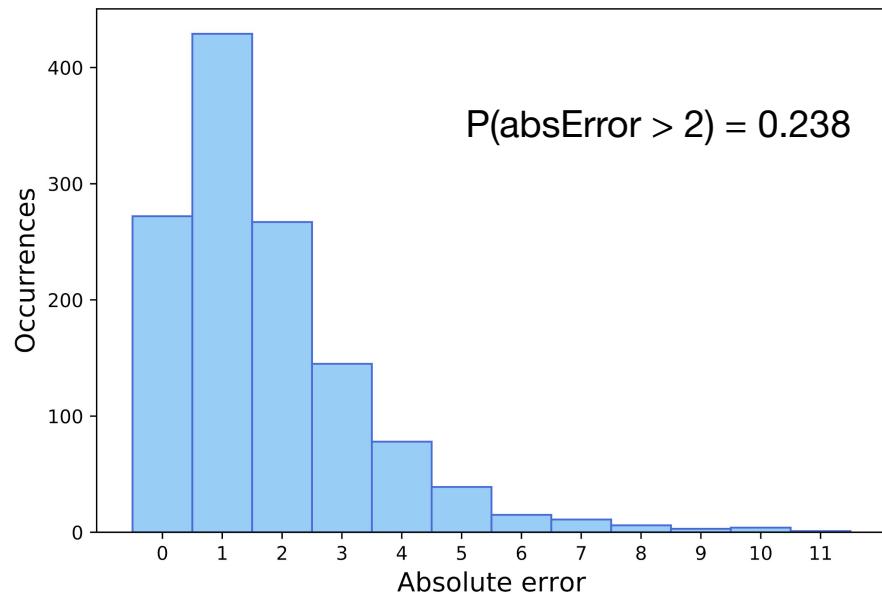
Test Results at Cafe





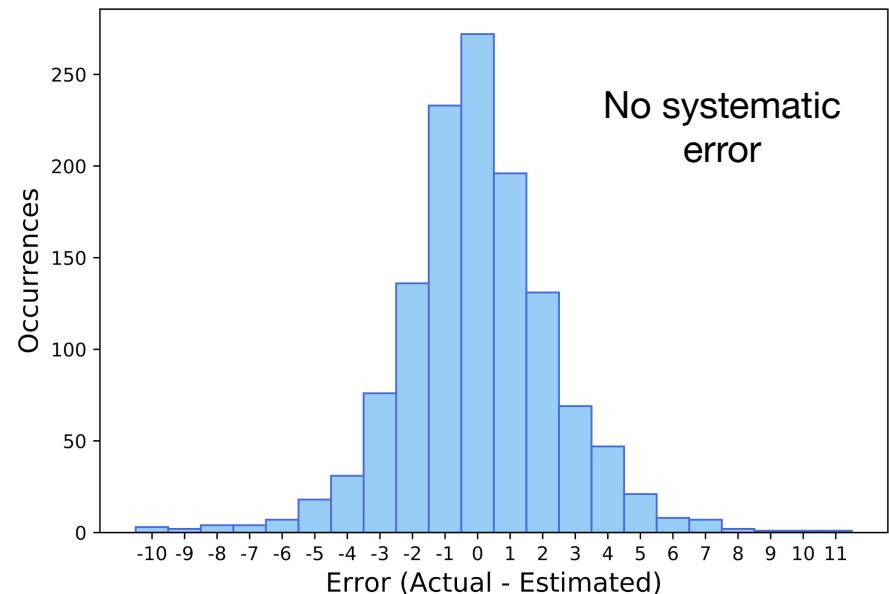
Error Distribution

Mean Absolute Error = 1.731



$P(\text{absError} > 2) = 0.238$

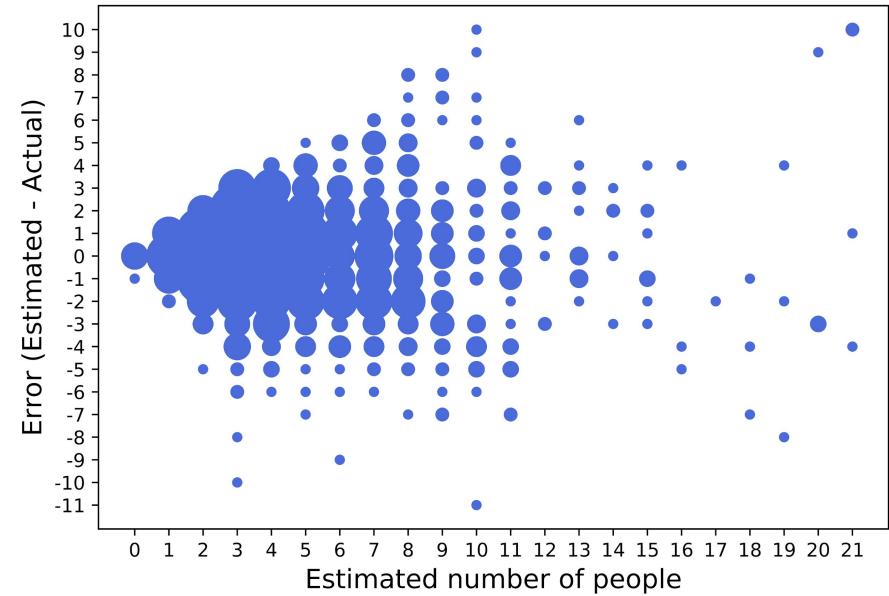
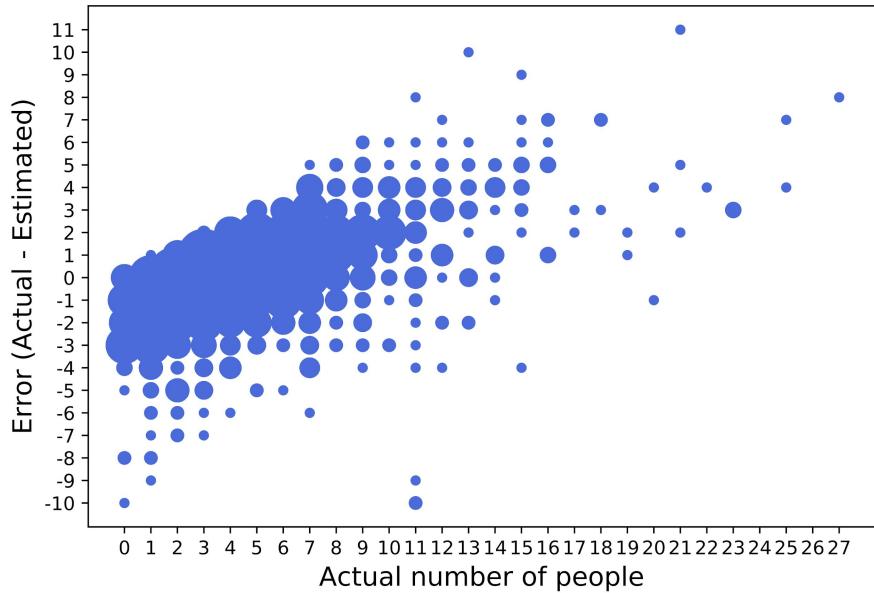
Mean Squared Error = 5.710



No systematic error



Scatter Plot of the Error





Summary

- **It is possible** to continuously estimate the density of the crowd in a place of interest based on the Wi-Fi probe request frames
- **Designed and implemented** a reliable system to achieve that
- Tested the system in a Cafe and collected 4 weeks of data and manually-annotated ground truth with an **overall MAE of 1.731**



Future Works

- **Real-time execution**
- Test of the system in **different contexts**
- Extension to multiple data collectors
- Improvement of the Machine Learning model



UNIVERSITÀ
DI TRENTO

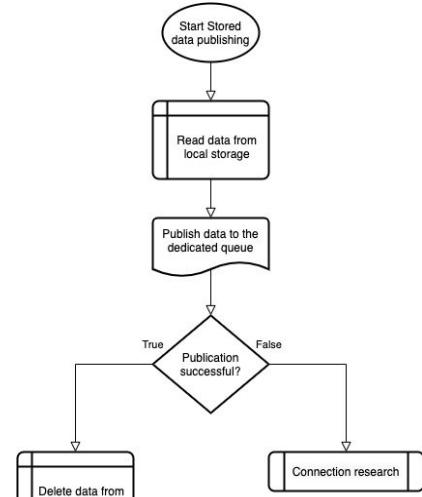
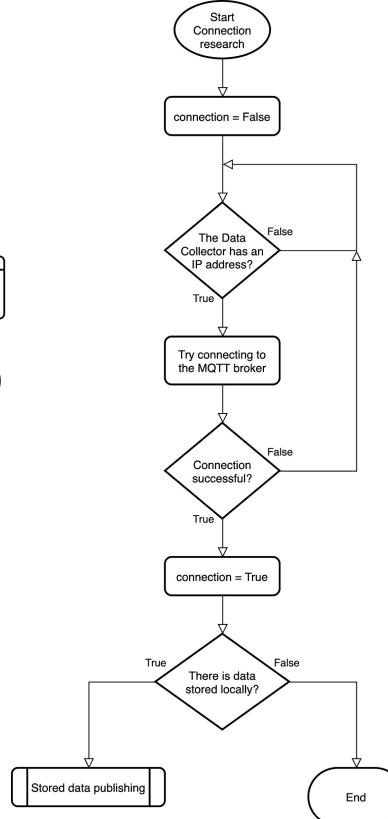
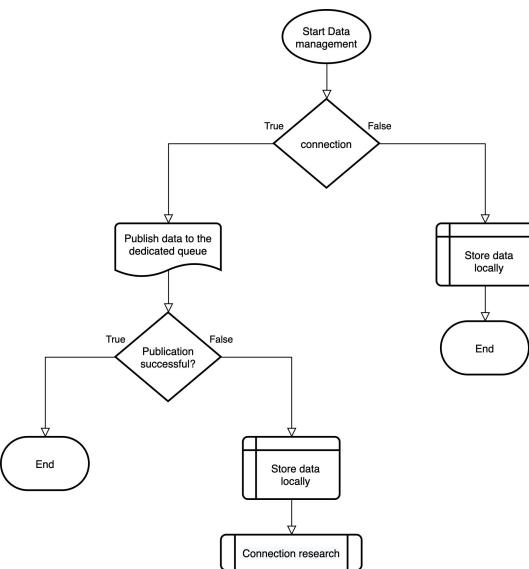
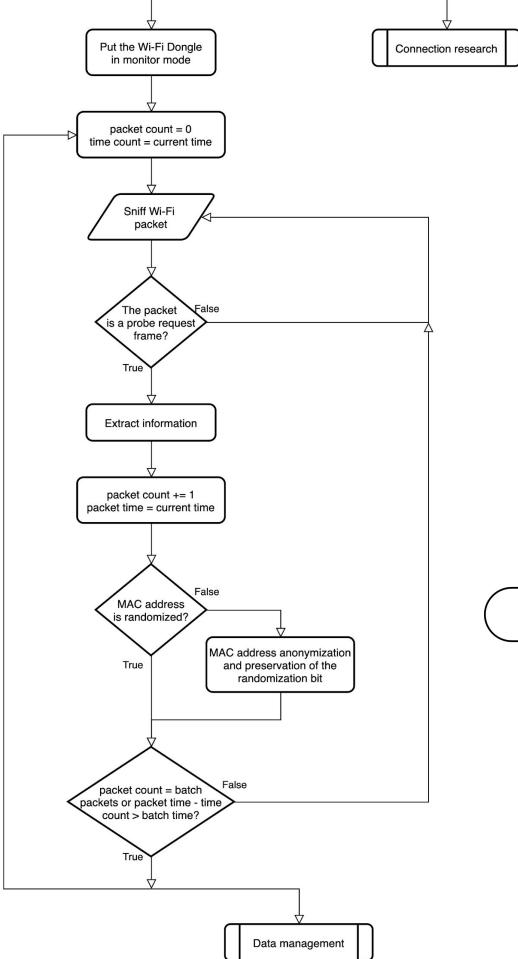
A system for estimating crowd density based on Wi-Fi probe request frames

Thank you for your attention

Samuel Bortolin



Data Collector Logic





Back-End Logic

