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 **detecting and monitoring 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**

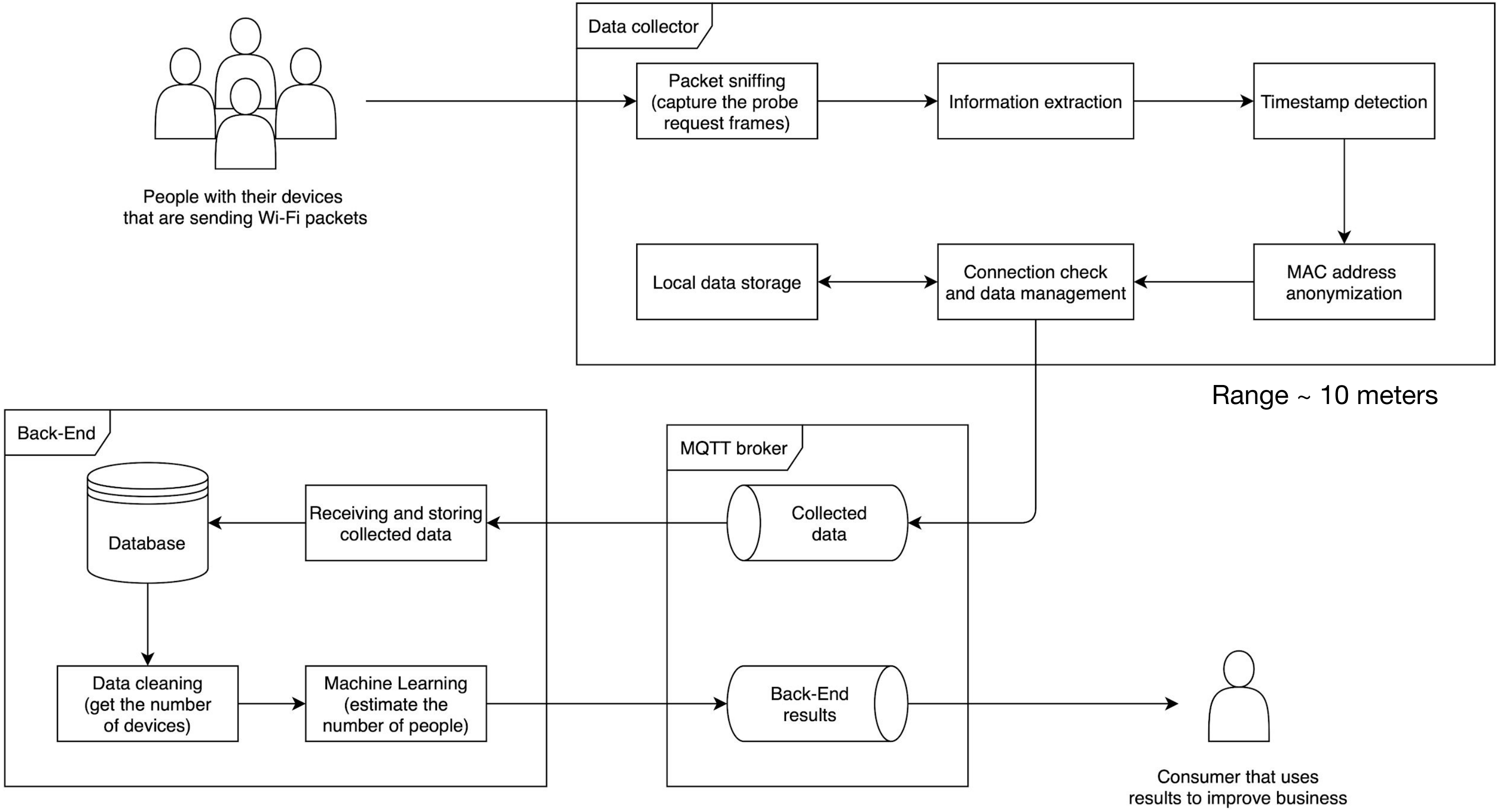
in a place of interest based on the Wi-Fi probe request frames?

Achievements

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

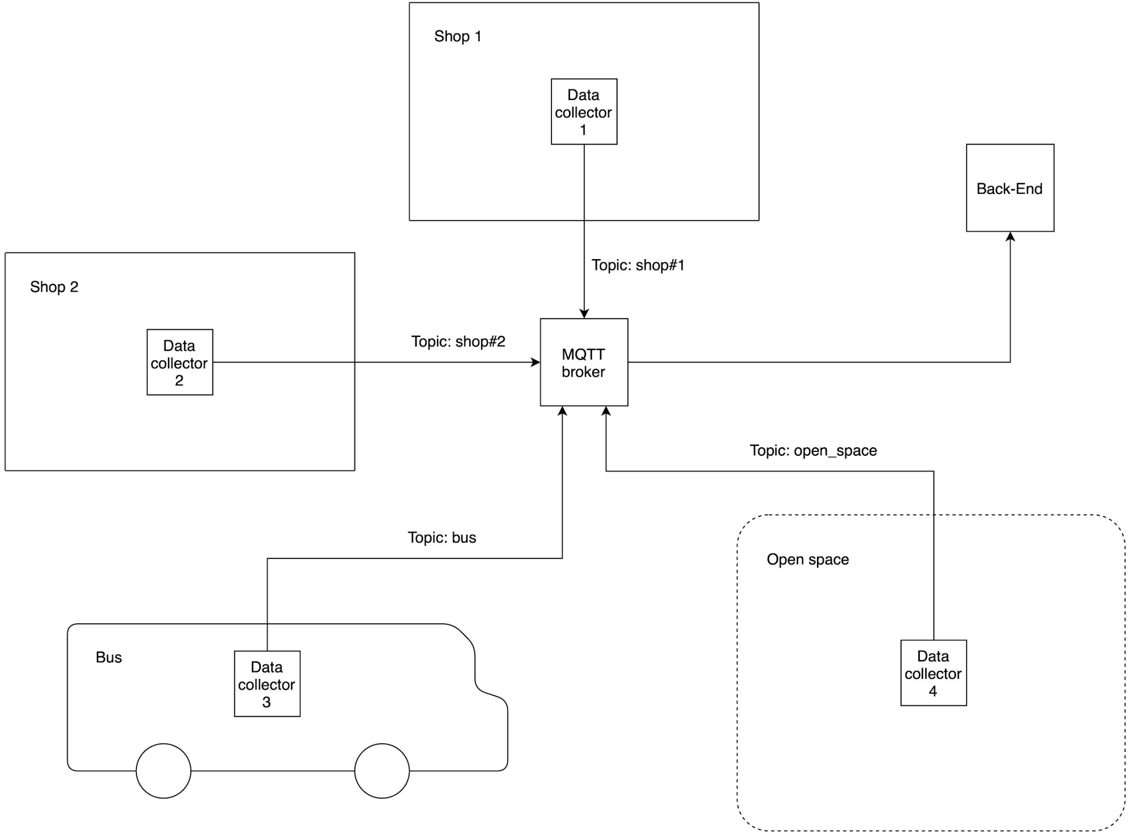
Tested the system in a Cafe and collected **4 weeks of data** and **manually-annotated ground truth** (the actual number of people)

System Architecture

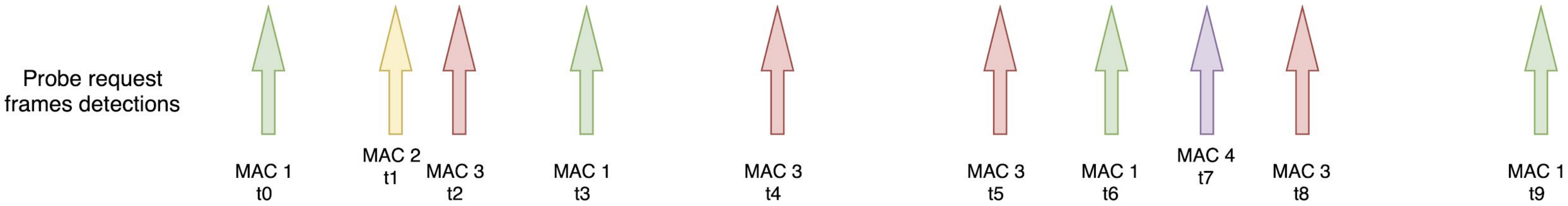

Data collector
Packet sniffing (capture the probe request frames)
Information extraction
Timestamp detection
MAC address anonymization
Connection check and data management
Local data storage
People with their devices that are sending Wi-Fi packets
Back-End
Database
Receiving and storing collected data
Data cleaning (get the number of devices)
Machine Learning (estimate the number of people)
MQTT broker
Collected data
Back-End results
Consumer that uses results to improve business

Range ~ 10 meters

Scalable Architecture



Presence of Devices



Probe request frames detections

MAC 1 t0

MAC 2 t1

MAC 3 t2

MAC 1 t3

MAC 3 t4

MAC 3 t5

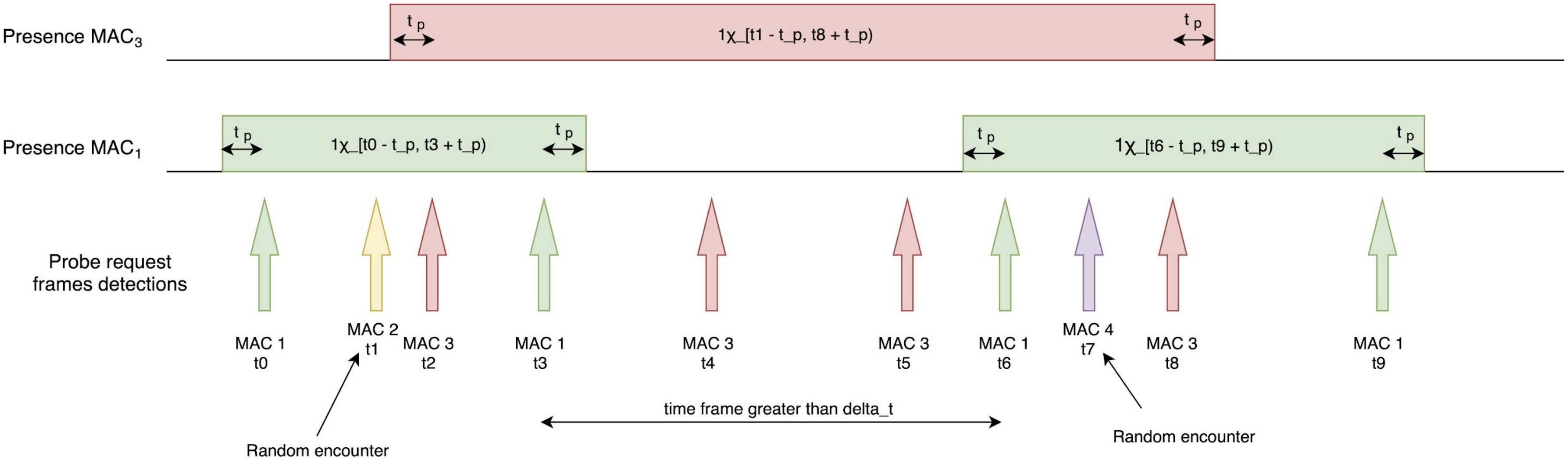
MAC 1 t6

MAC 4 t7

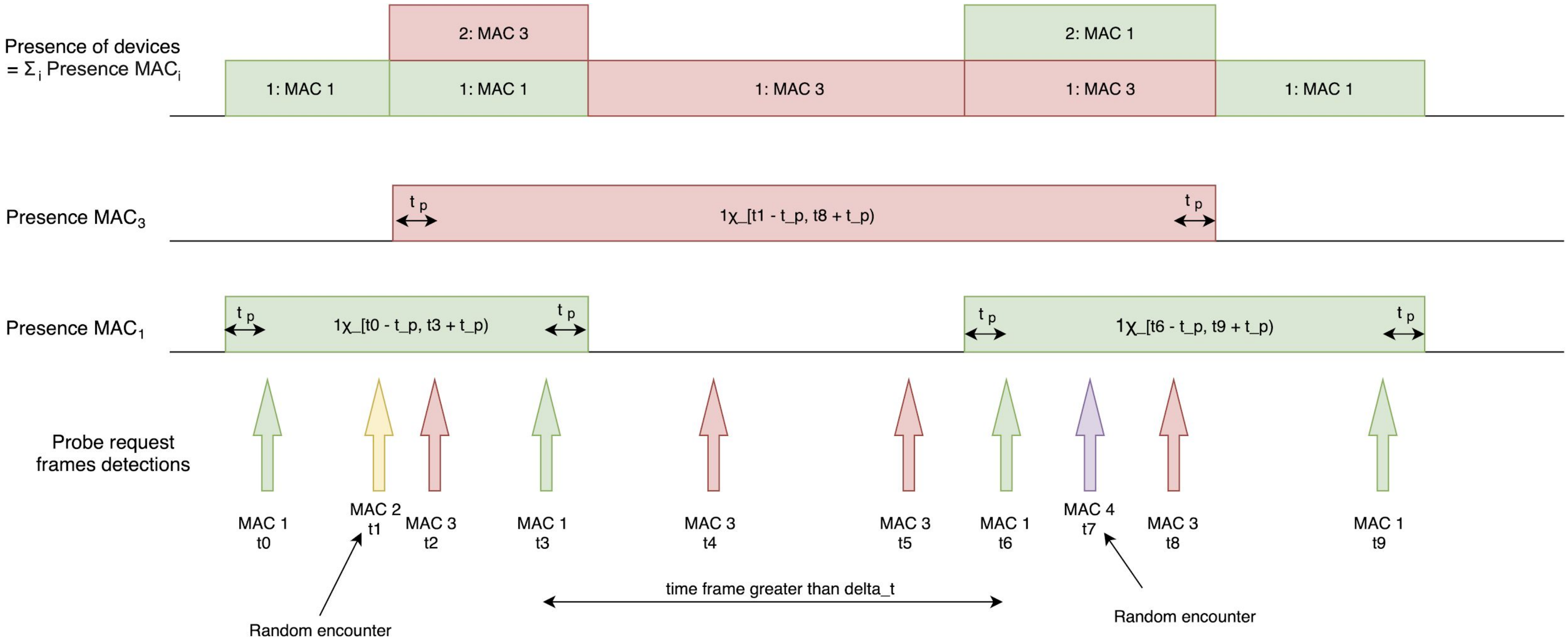
MAC 3 t8

MAC 1 t9

Presence of Devices



Presence of Devices



Feasibility Test at Home

3 days of data collection

65928 probe request frames

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

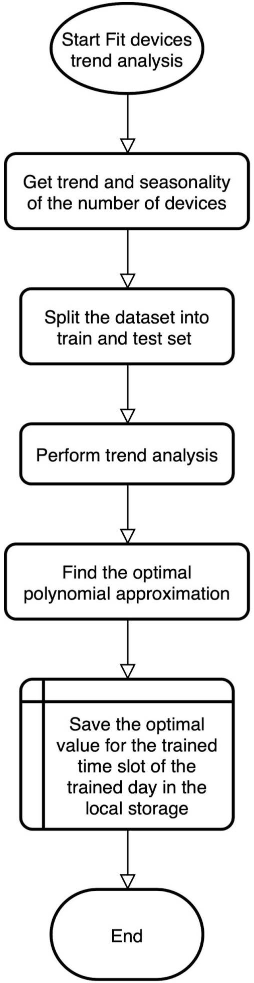
RaspberryPi in a Cafe where I annotate manually the ground truth

mosquitto broker MQTT of U-Hopper cloud infrastructure

MQTT.ORG receiver and mongoDB on U-Hopper cloud infrastructure

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

Ground Truth Annotation



Start Fit devices trend analysis

Get trend and seasonality of the number of devices

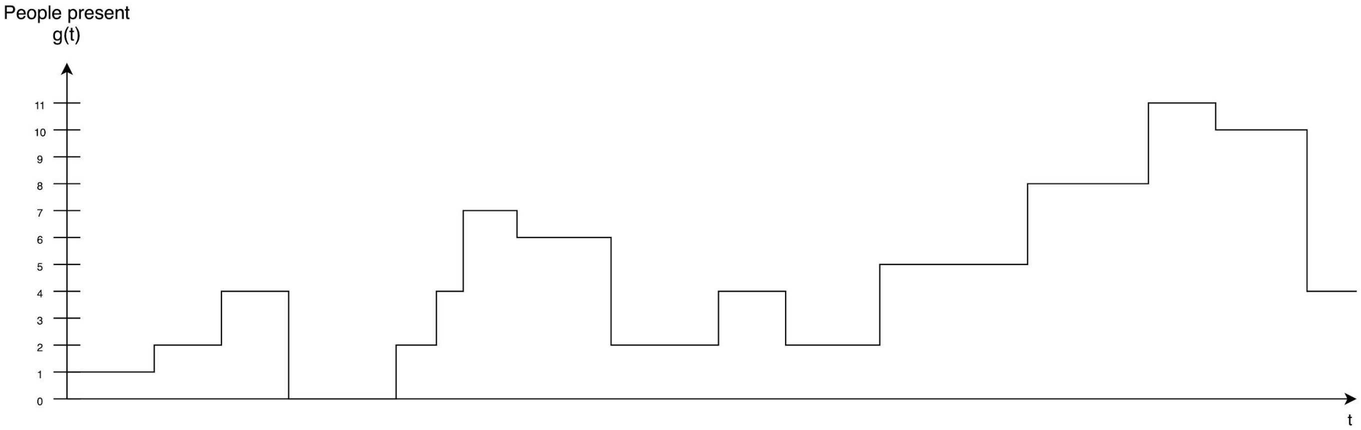
Split the dataset into train and test set

Perform trend analysis

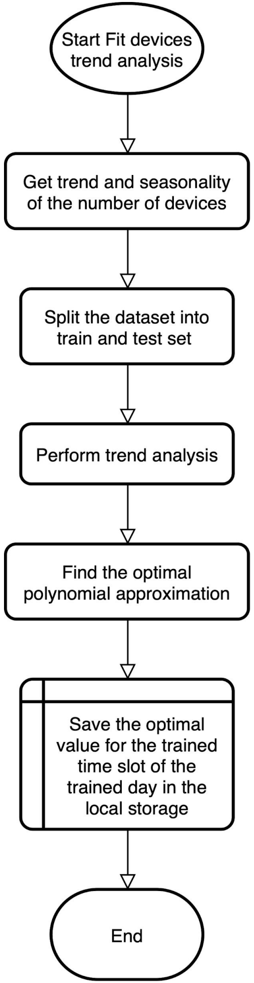
Find the optimal polynomial approximation

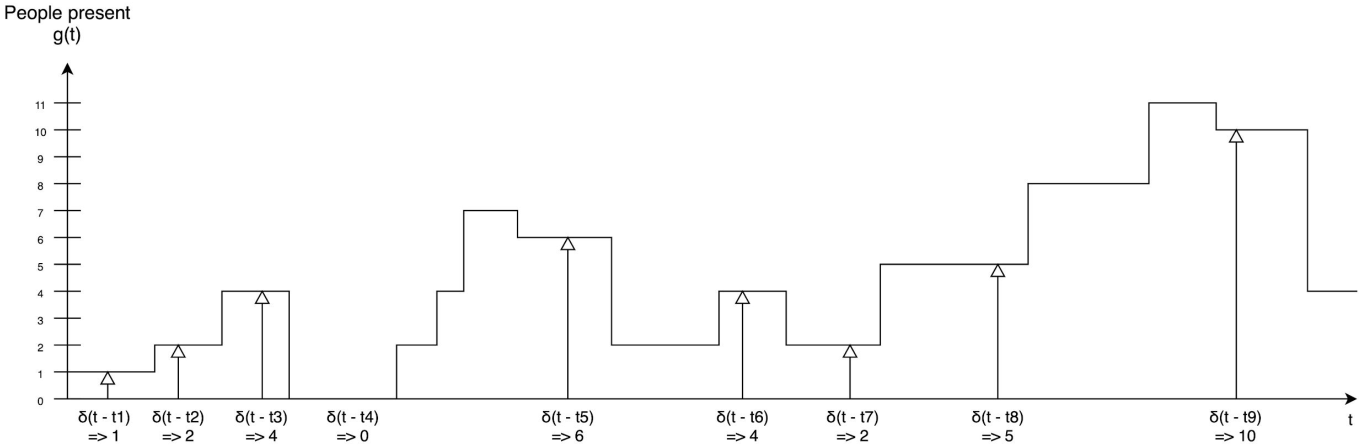
Save the optimal value for the trained time slot of the trained day in the local storage

End

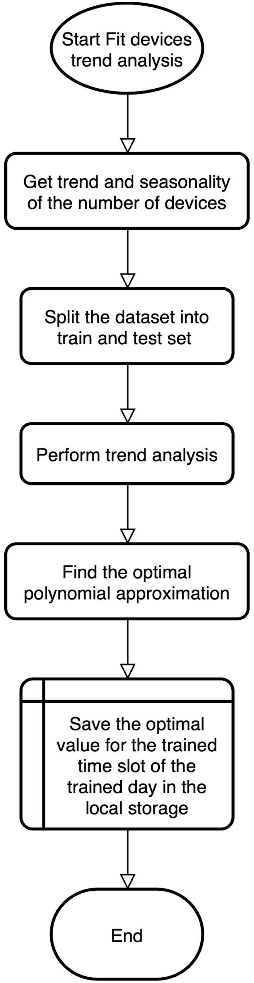


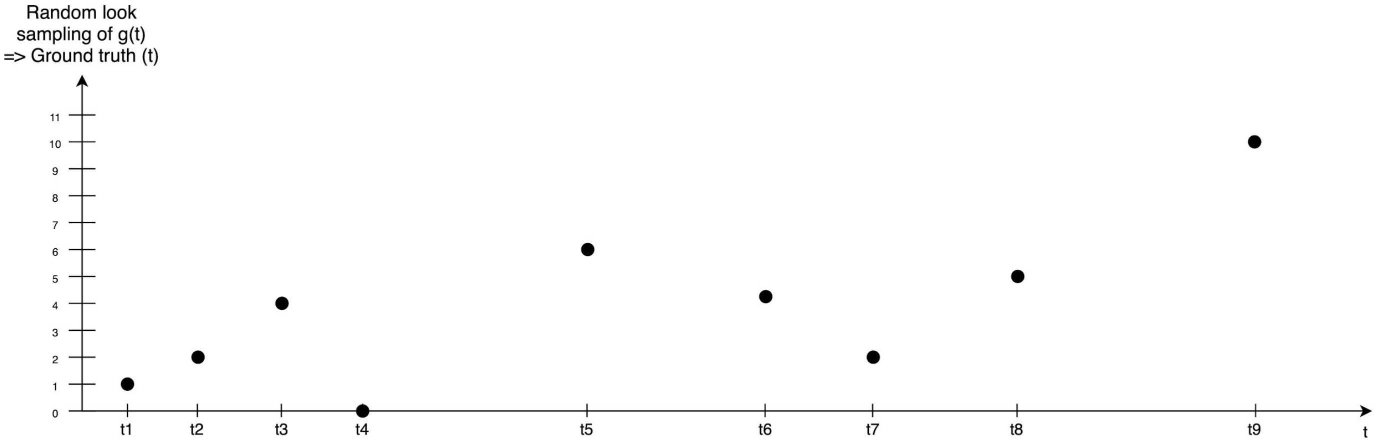
Ground Truth Annotation


Start Fit devices trend analysis
Get trend and seasonality of the number of devices
Split the dataset into train and test set
Perform trend analysis
Find the optimal polynomial approximation
Save the optimal value for the trained time slot of the trained day in the local storage
End



Ground Truth Annotation


Start Fit devices trend analysis
Get trend and seasonality of the number of devices
Split the dataset into train and test set
Perform trend analysis
Find the optimal polynomial approximation
Save the optimal value for the trained time slot of the trained day in the local storage
End



Test Results at Cafe

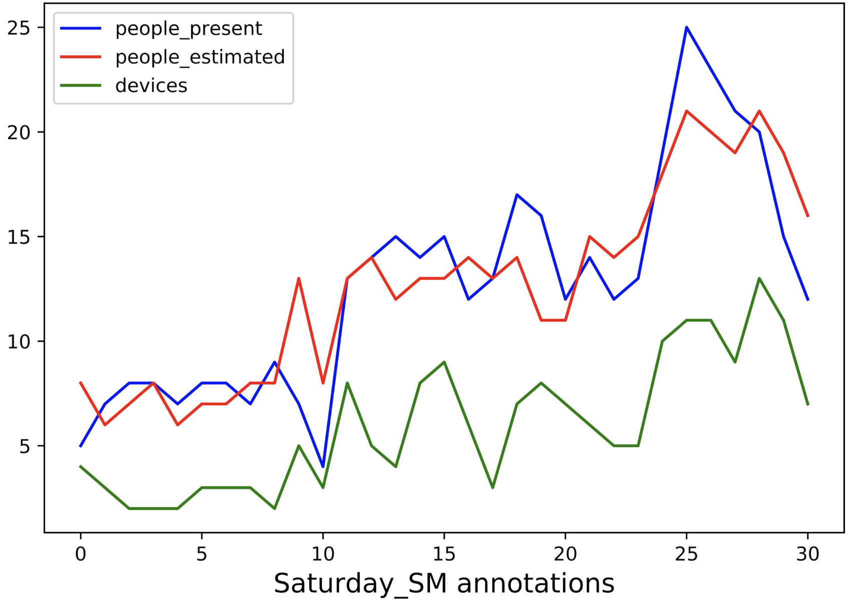
4 weeks of data collection
(24 days)

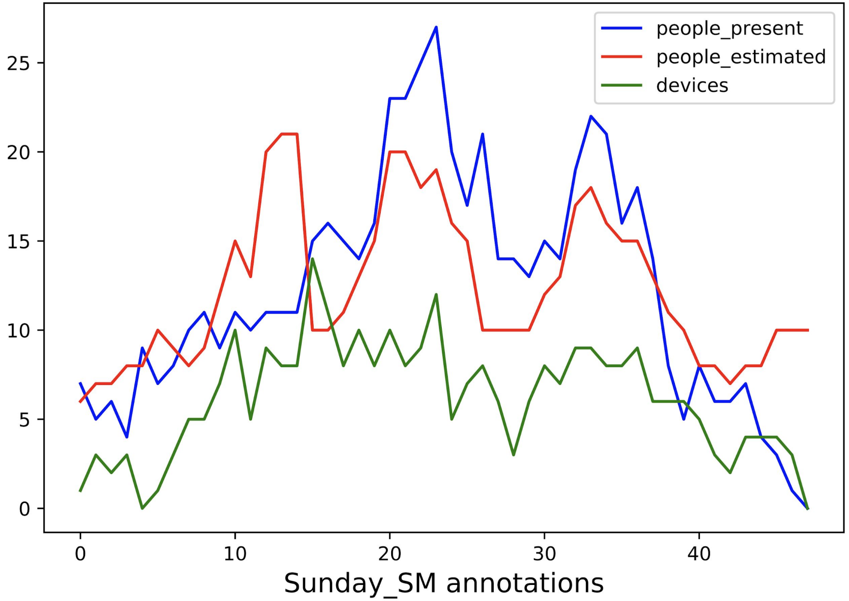
1270 manual annotation of ground truth

861979 probe request frames
(~ 560 MB of data)

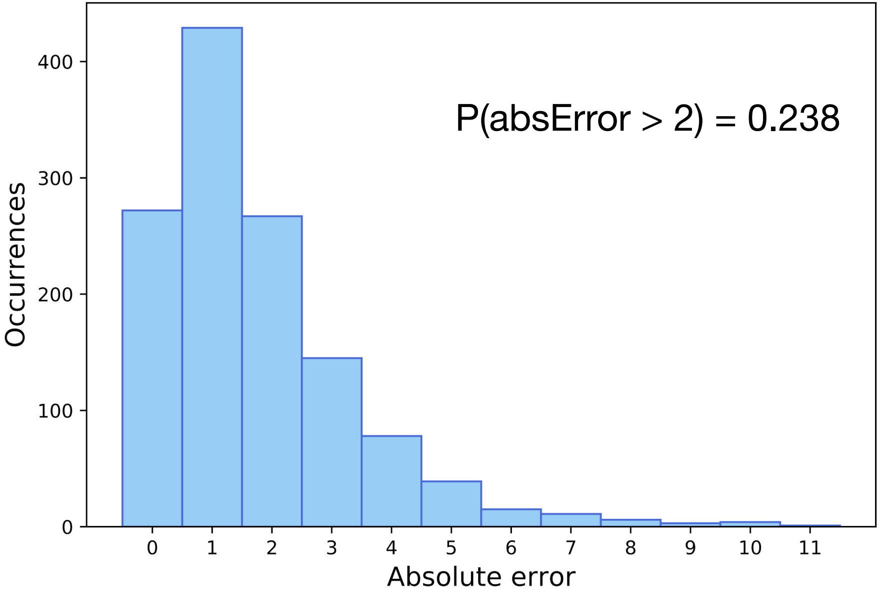
38771 unique MAC addresses detected

Test Results at Cafe



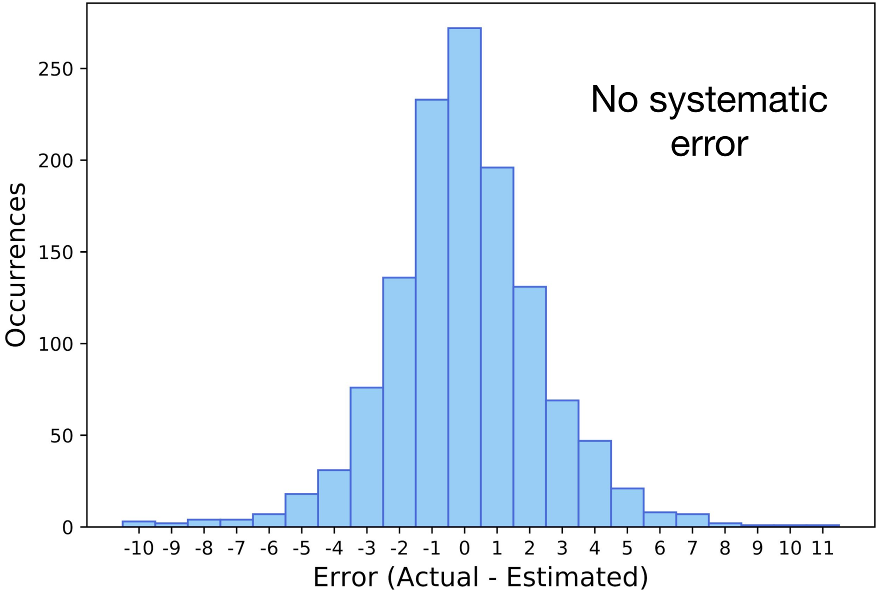


Error Distribution



Mean Absolute Error = 1.731

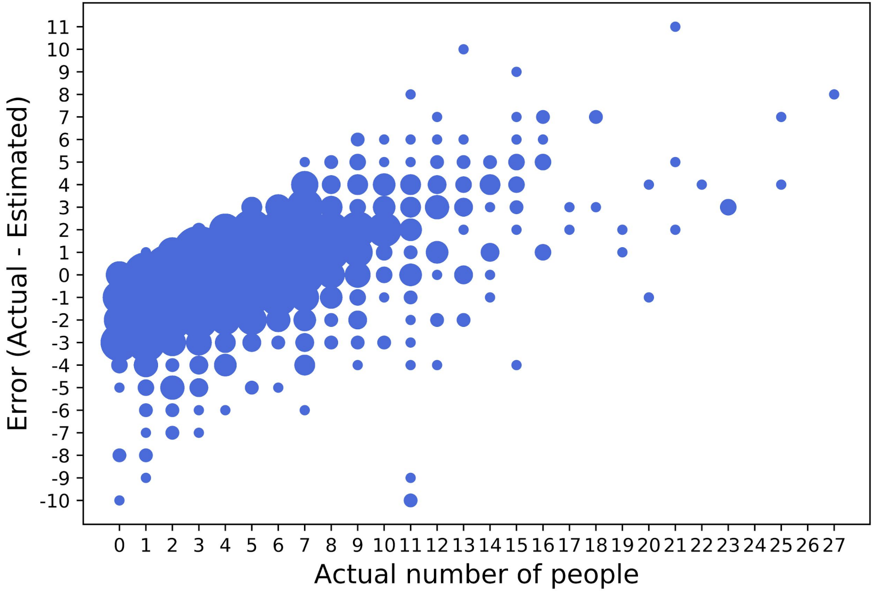
P(absError > 2) = 0.238

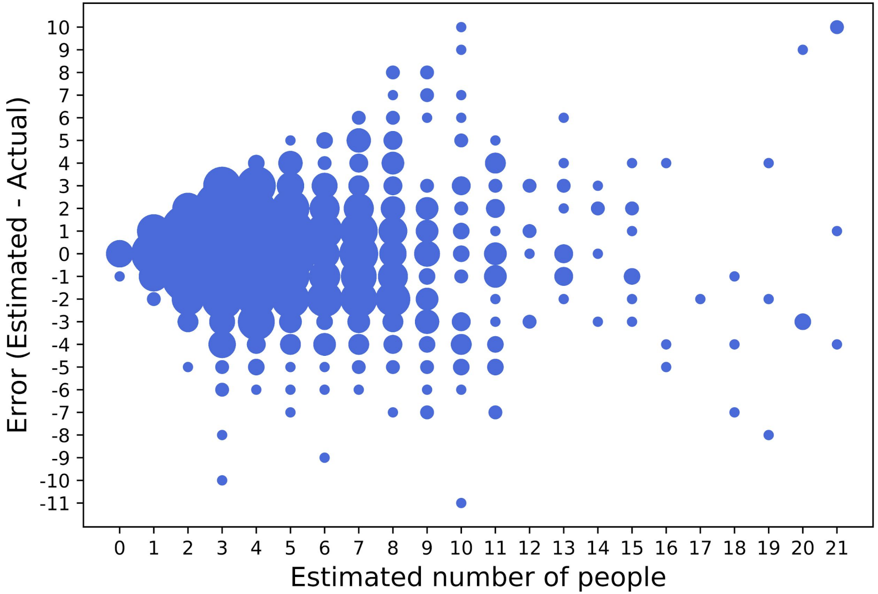


Mean Squared Error = 5.710

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 do 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

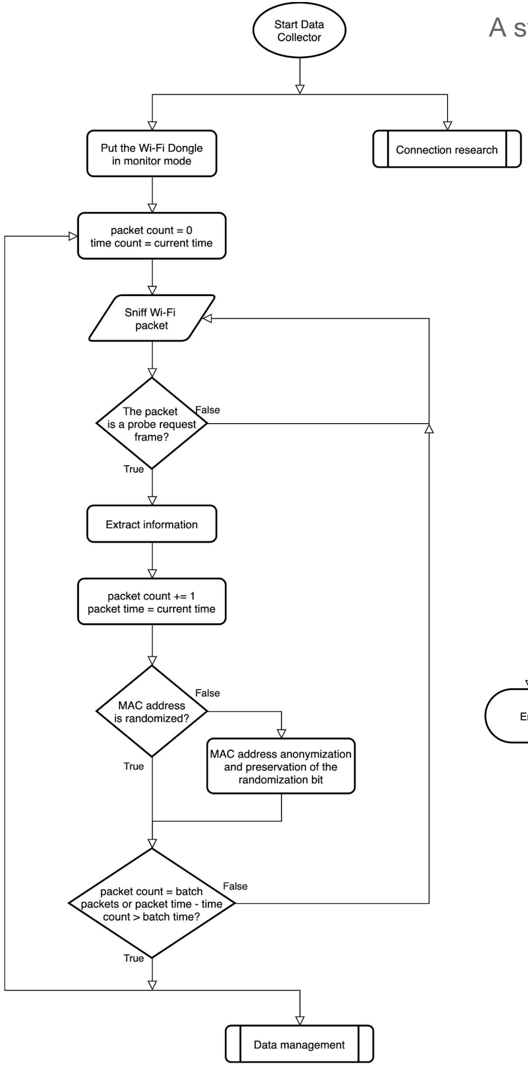
UNIVERSITÀ
DI TRENTO

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

Thank you for your attention

Samuel Bortolin

Data Collector Logic



Start Data Collector

Put the Wi-Fi Dongle in monitor mode

Connection research

packet count = 0
time count = current time

Sniff Wi-Fi packet

The packet is a probe request frame?

False

True

Extract information

packet count += 1
packet time = current time

MAC address is randomized?

False

MAC address anonymization and preservation of the randomization bit

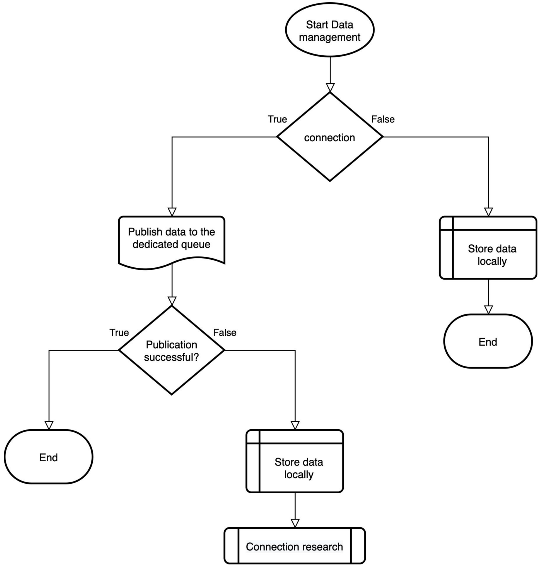
True

packet count = batch packets or packet time - time count > batch time?

False

True

Data management



Start Data management

connection

True

False

Publish data to the dedicated queue

Store data locally

End

Publication successful?

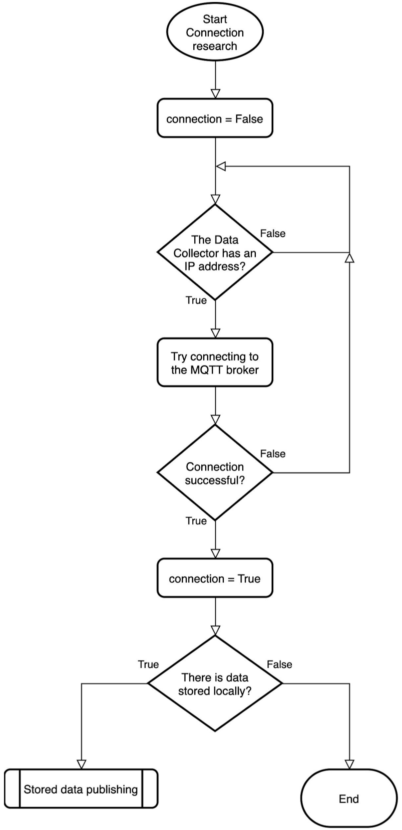
True

False

End

Store data locally

Connection research



Start Connection research

connection = False

The Data Collector has an IP address?

False

True

Try connecting to the MQTT broker

Connection successful?

False

True

connection = True

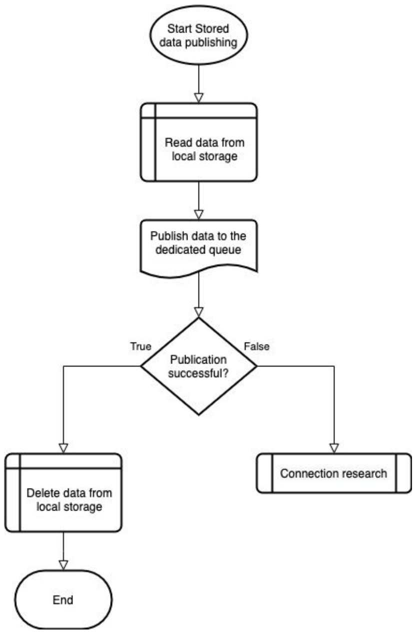
There is data stored locally?

True

False

Stored data publishing

End



Start Stored data publishing

Read data from local storage

Publish data to the dedicated queue

Publication successful?

True

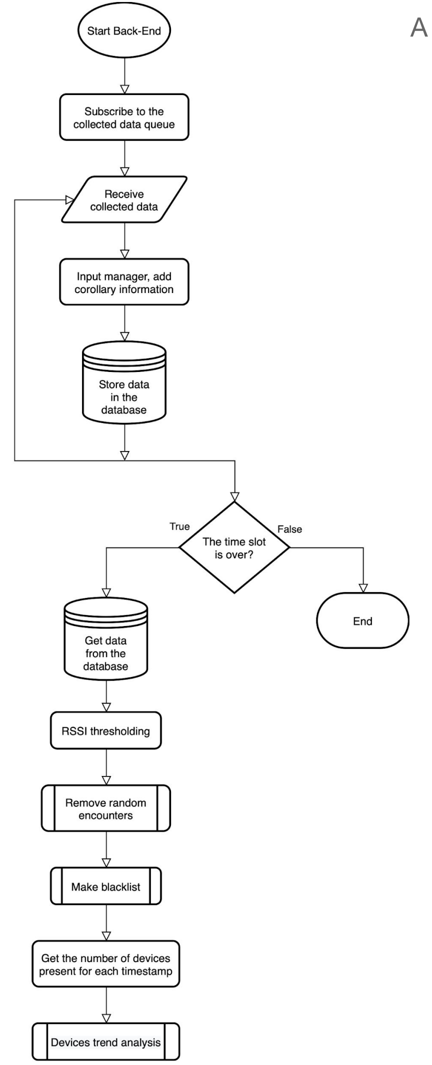
False

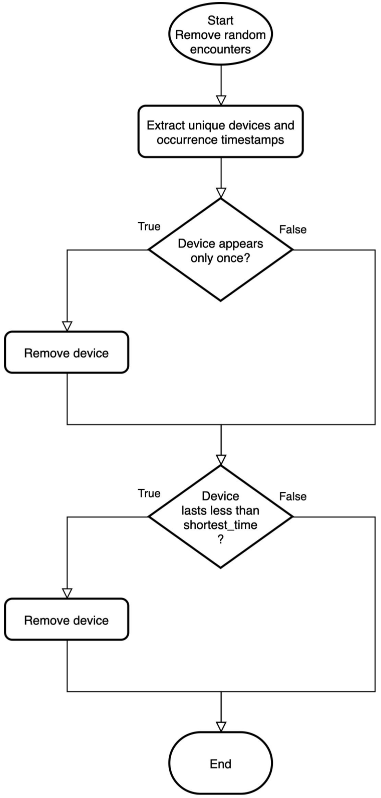
Delete data from local storage

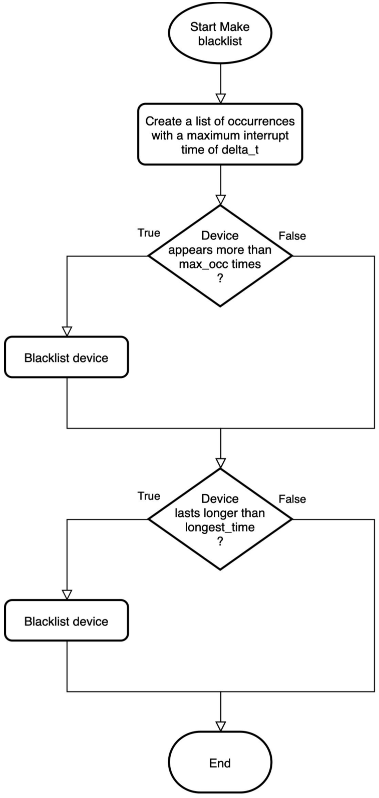
Connection research

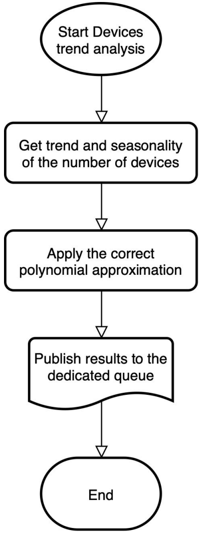
End

Back-End Logic


Start Back-End
Subscribe to the collected data queue
Receive collected data
Input manager, add corollary information
Store data in the database
The time slot is over?
True
False
End
Get data from the database
RSSI thresholding
Remove random encounters
Make blacklist
Get the number of devices present for each timestamp
Devices trend analysis


Start Remove random encounters
Extract unique devices and occurrence timestamps
True
Device appears only once?
False
Remove device
True
Device lasts less than shortest_time ?
False
Remove device
End


Start Make blacklist
Create a list of occurrences with a maximum interrupt time of delta_t
True
Device appears more than max_occ times ?
False
Blacklist device
True
Device lasts longer than longest_time ?
False
Blacklist device
End


Start Devices trend analysis
Get trend and seasonality of the number of devices
Apply the correct polynomial approximation
Publish results to the dedicated queue
End