AUTOMATED PROCESSING AND VISUALIZATION OF **WIFI RTT USING THE INTEL OPEN DATASET**

NEXT GENERATION NETWORKS

CONTRIBUTORS

- ROY ALIA ASIKU 229759
- EMIRU EYAEL SOLOMON
- ZHANG BO
- PHIRI MACDONALD

OUTLINE

Introduction

Dataset

Objectives

Design and implementation

Results

Conclusion

Introduction

- **Wi-Fi systems** are becoming **ubiquitous**
- Use of **Wi-Fi based positioning systems** is growing popular
- The 2016 release of IEEE 802.11 WLAN standard [1], defined a protocol for **fine timing measurement** (FTM)/round-trip time (RTT).
- This protocol **enables two WLAN stations to measure their distance** (range) with respect to one another.
- **Experiments** conducted in **office environment**
- **Time delay is usually estimated** using maximum-likelihood and super-resolution methods

FTM transactions

?STA- Station

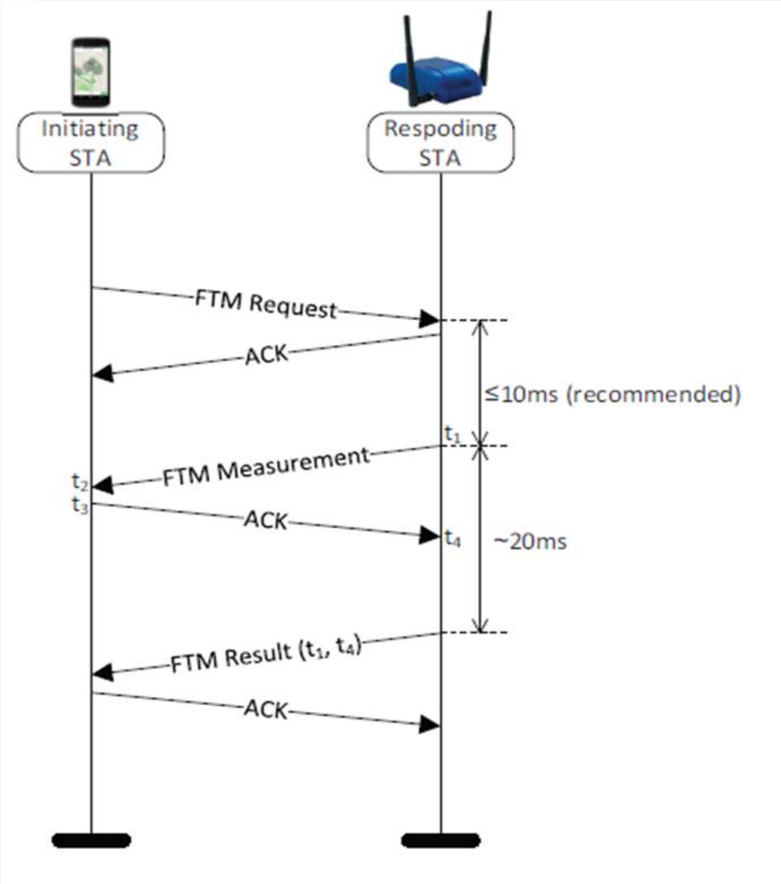
The range between the two stations is calculated using

Range =$c\,\frac{(t_4-t_1)-(t_{3-}t_2)}{2}$

t_1 - the time of departure (ToD) measured by the RSTA,

t4 -the time of arrival (ToA), which is estimated by the RSTA and

c = 3 x 108m/s is the electromagnetic propagation speed.



Dataset

We use the intel open dataset [1]

29851 observations and 467 attributes (columns)

Size 105,207 KB

Project Goal

- **Import** the Intel Open Wi-Fi RTT **dataset**, and **extract useful statistical information** from it,
- **Characterize** the dataset, **extract models** of the data, and
- **automate** the process of data **analysis**.

Objectives

- Understand (analyze) dataset from a **pure data science perspective** (mode)
- Analyze dataset from a **communications engineering mode**
- Extract a **statistical model** of the data to assign a distribution to it
- **Visualize the data** in both modes
- Extract a **model** of the data for the purpose of prediction on interesting applications such as ranging, TOA estimation, navigation.
- **Automate** the above processes

Design – operational modes

- **Data science** mode
- **Verbose Data science** mode
- **Communications engineering** mode
- **Verbose Communications engineering** mode
- Default- **verbose data science + communications theory**

Design choices

- Selection of **sample data points** – **Random**
- Selection of **data science libraries** – Include the most **useful and most common**
- Other parameters:
  - **Inferred from the data**
  - **Calculated** most optimal parameters for the given dataset
  - Inherit from the **paper** [1]

Implementation

Programming Language: **Python**

Inputs:

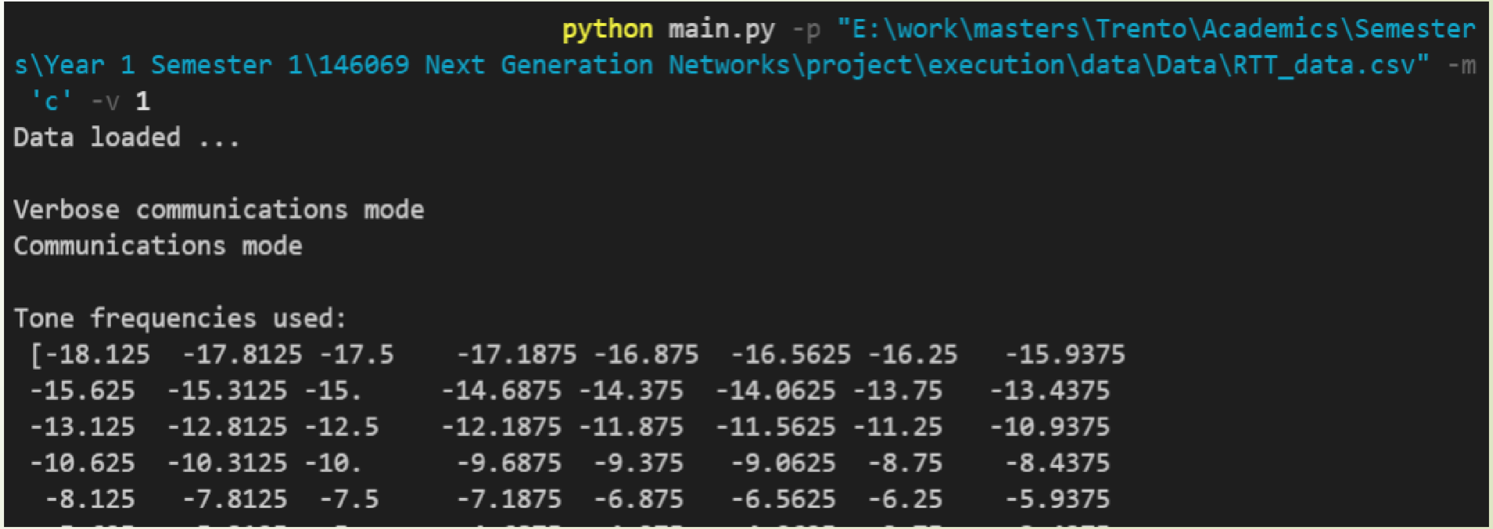
- ❑ RTT intel **open dataset** [‘*RTT_dataset_sample.csv*’]
- ❑ Operation modes:
  - Pure Data Science (***DS***)
  - Pure Communications Engineering (***CE***)
  - ***DS + Verbose***
  - ***CE + Verbose***
  - Default (***DS +CE+ Verbose***)

Outputs: **CLI** text data, graphs, models, GUI text

GitHub: https://github.com/Asiku-Roy-Alia/intel_wifi_dataset_auto_analysis

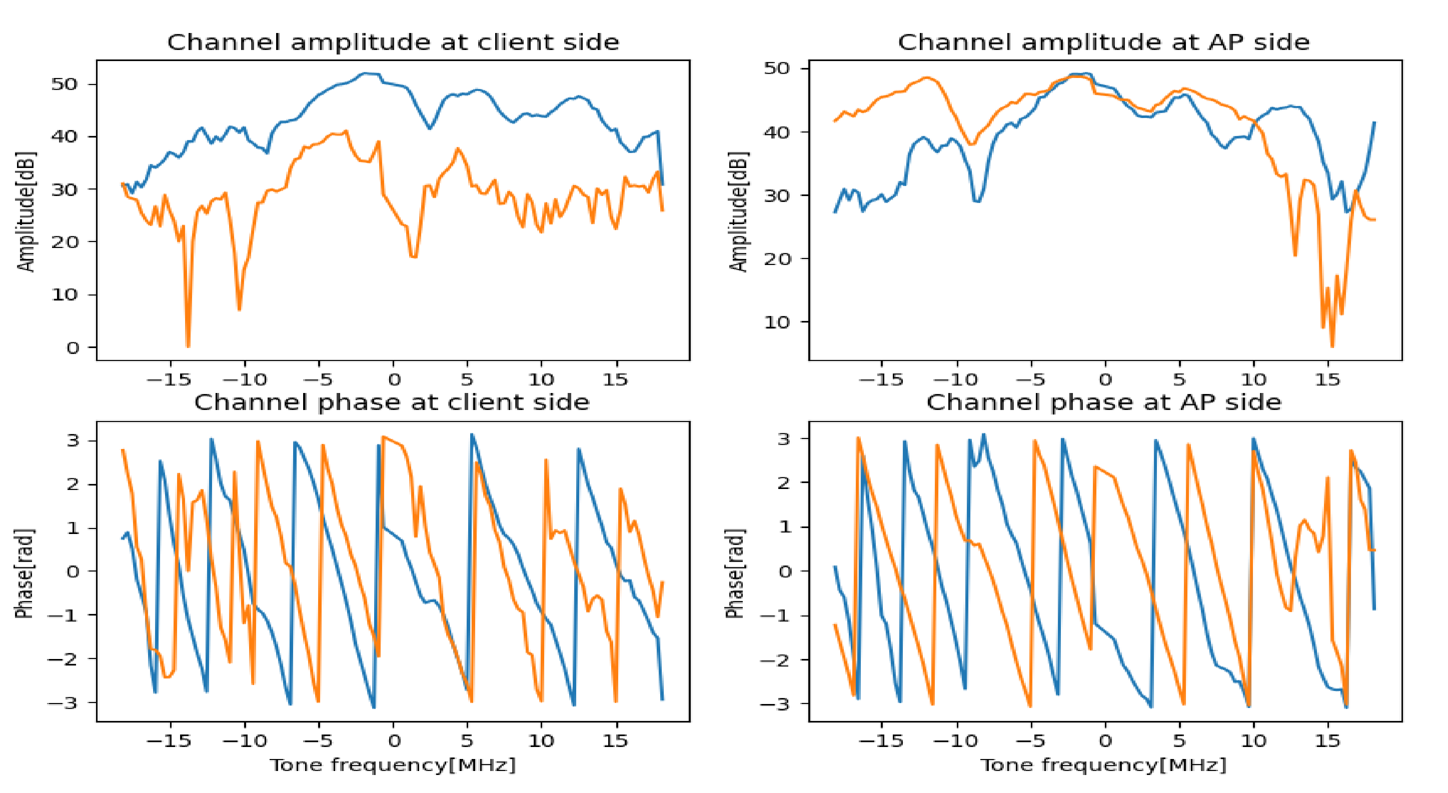
Implementation – CLI (CE mode)

Execution: ***python.exe main.py –p ‘RTT_data.csv’ –m ‘c’ –v 1*** for doing a **verbose** analysis in the **communications engineering mode** with the **input RTT_data.csv** in the current directory.

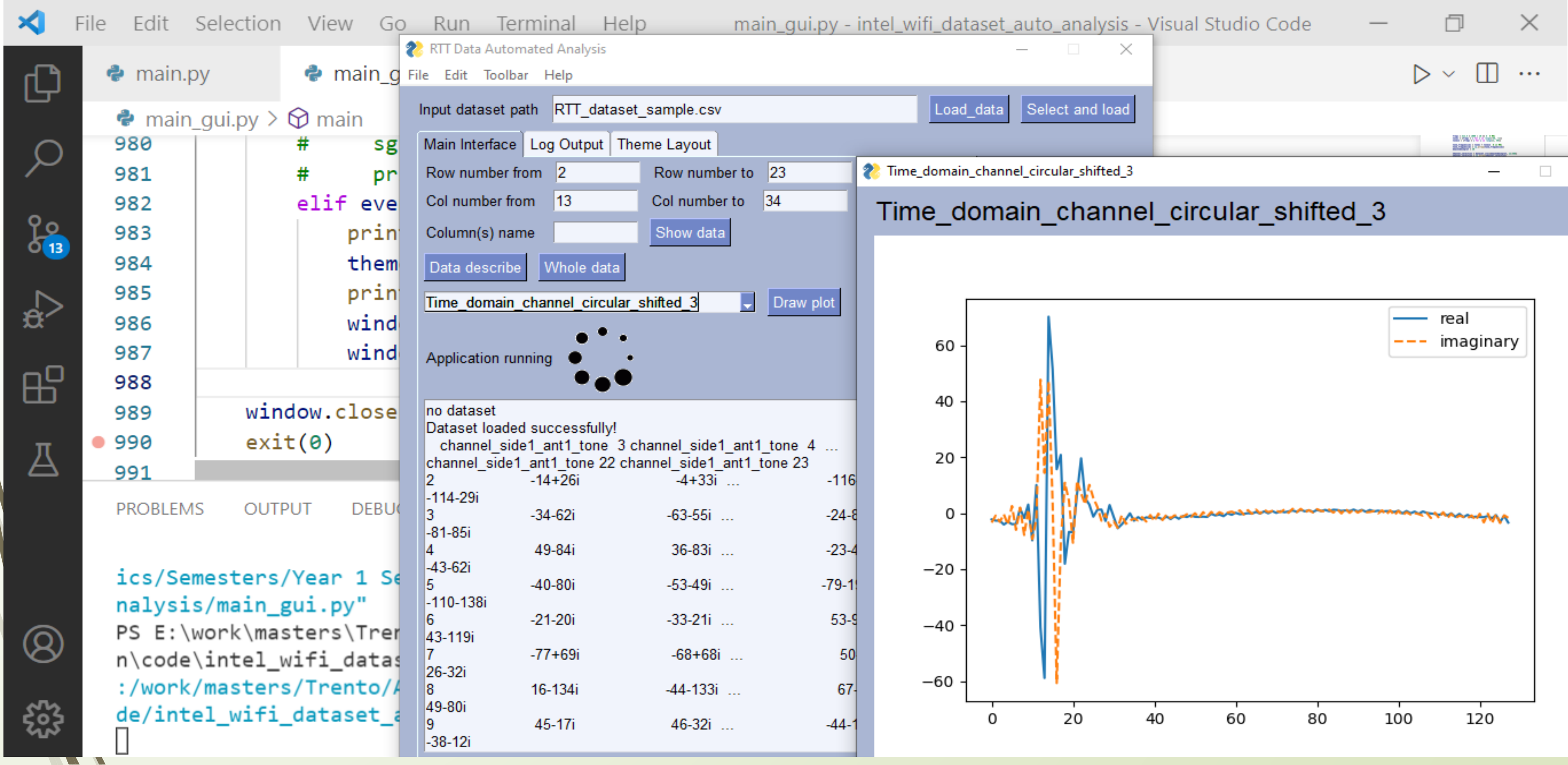


```
                                    python main.py -p "E:\work\masters\Trento\Academics\Semester
s\Year 1 Semester 1\146069 Next Generation Networks\project\execution\data\Data\RTT_data.csv" -m
 'c' -v 1
Data loaded ...

Verbose communications mode
Communications mode

Tone frequencies used:
 [-18.125  -17.8125 -17.5    -17.1875 -16.875  -16.5625 -16.25   -15.9375
 -15.625  -15.3125 -15.     -14.6875 -14.375  -14.0625 -13.75   -13.4375
 -13.125  -12.8125 -12.5    -12.1875 -11.875  -11.5625 -11.25   -10.9375
 -10.625  -10.3125 -10.      -9.6875  -9.375   -9.0625  -8.75    -8.4375
  -8.125   -7.8125  -7.5     -7.1875  -6.875   -6.5625  -6.25    -5.9375
```

Implementation – Graphs (CE mode)



GUI Mode



```
        #       sg
        #       pr
        elif eve
            prin
            them
            prin
            wind
            wind

    window.close
    exit(0)
```

```
ics/Semesters/Year 1 Se
nalysis/main_gui.py"
PS E:\work\masters\Tren
n\code\intel_wifi_datas
:/work/masters/Trento/A
de/intel_wifi_dataset_a
```

Input dataset path: RTT_dataset_sample.csv

Row number from: 2 Row number to: 23

Col number from: 13 Col number to: 34

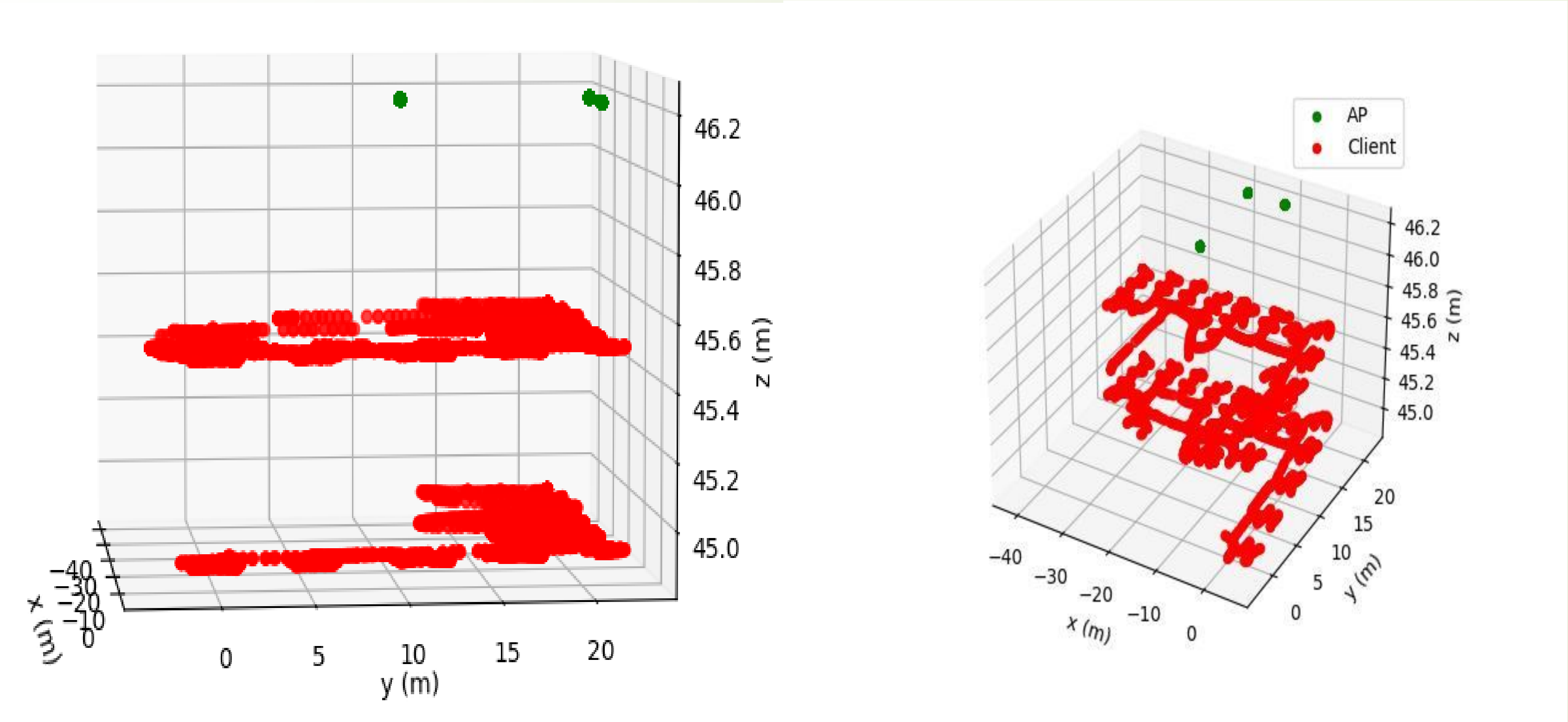
Column(s) name:

Time_domain_channel_circular_shifted_3

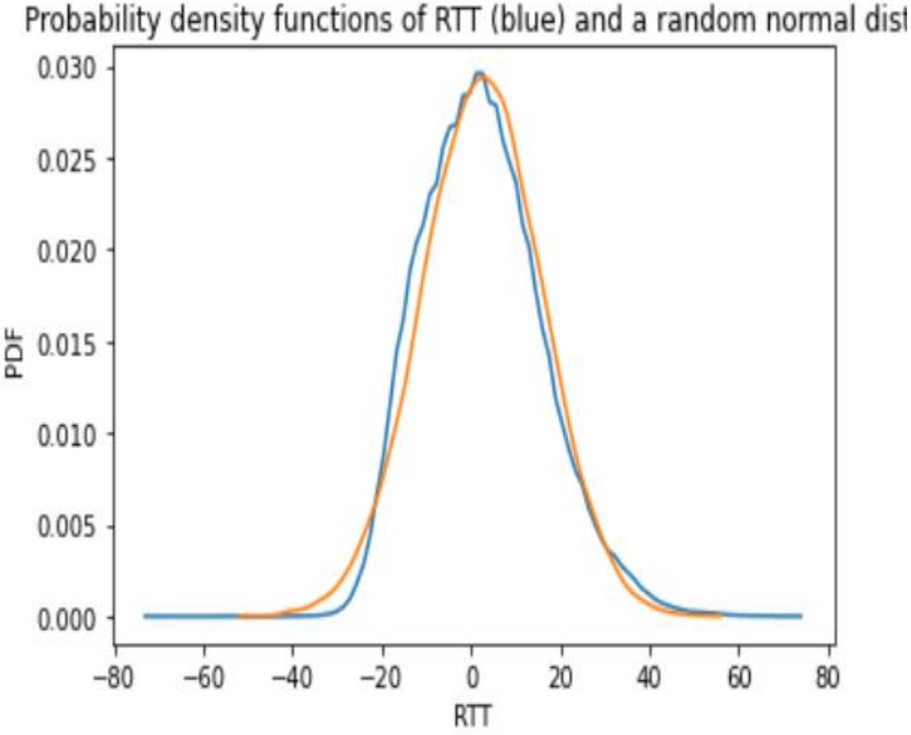
Application running

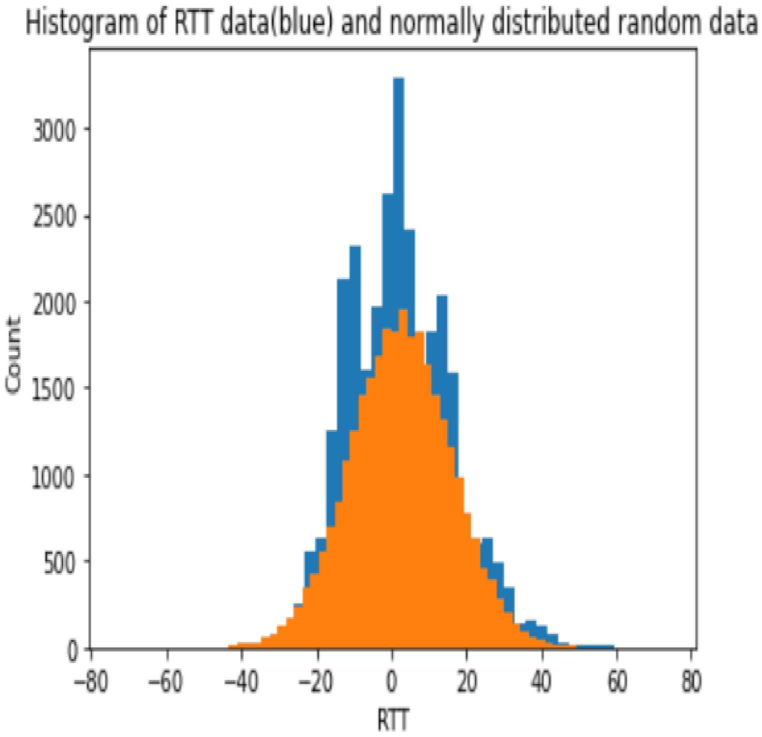
```
no dataset
Dataset loaded successfully!
   channel_side1_ant1_tone  3 channel_side1_ant1_tone  4  ...
channel_side1_ant1_tone 22 channel_side1_ant1_tone 23
2                  -14+26i                  -4+33i  ...                -116
-114-29i
3                  -34-62i                 -63-55i  ...                -24-8
-81-85i
4                   49-84i                  36-83i  ...                -23-4
-43-62i
5                  -40-80i                 -53-49i  ...                -79-1
-110-138i
6                  -21-20i                 -33-21i  ...                 53-9
43-119i
7                  -77+69i                 -68+68i  ...                  50
26-32i
8                  16-134i                -44-133i  ...                 67-
49-80i
9                   45-17i                  46-32i  ...                -44-1
-38-12i
```

Data Visualization



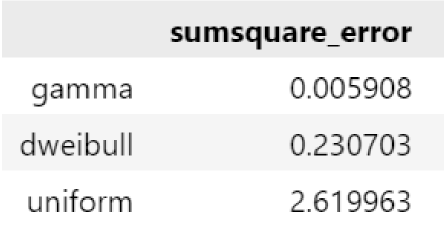
Results-Statistical Model


Probability density functions of RTT (blue) and a random normal dist

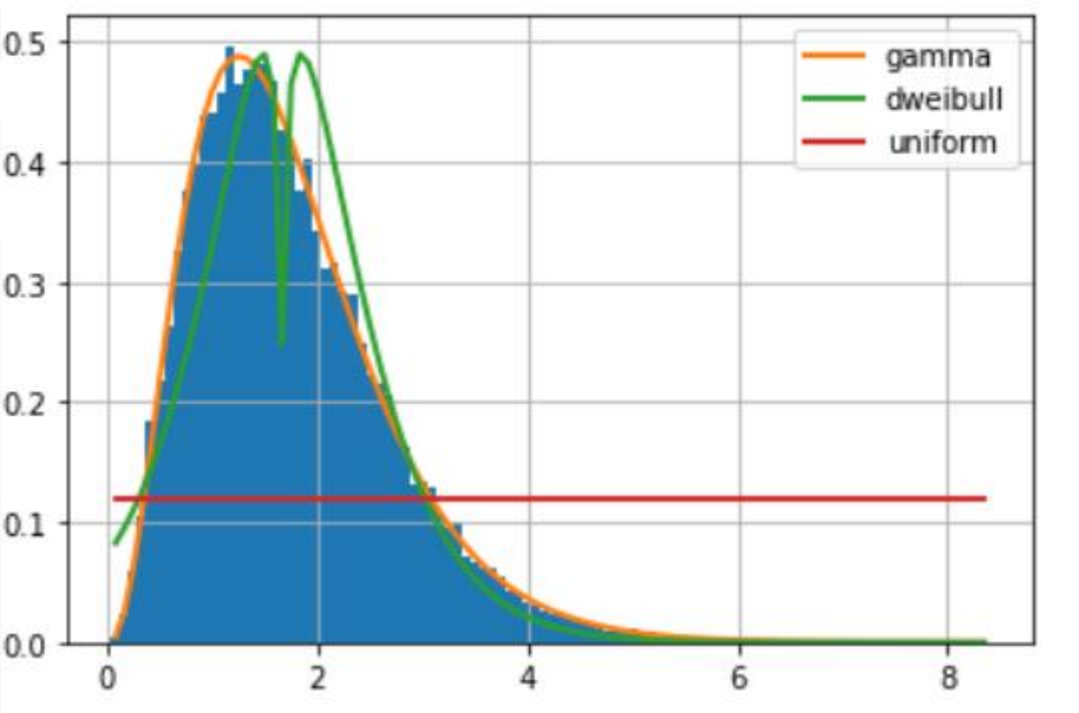

Histogram of RTT data(blue) and normally distributed random data

Results-Distribution fitting

Multiple distributions. MSE performance benchmark.



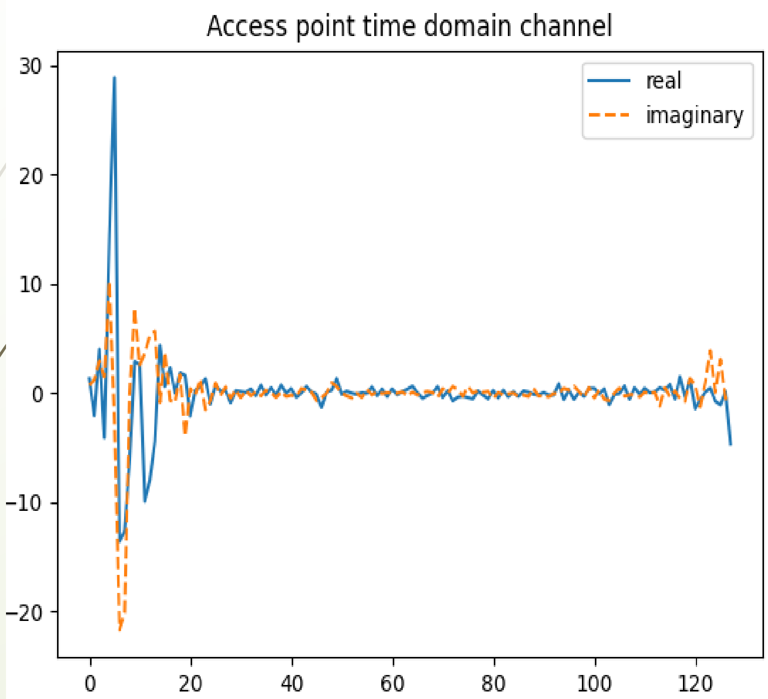
	sumsquare_error
gamma	0.005908
dweibull	0.230703
uniform	2.619963

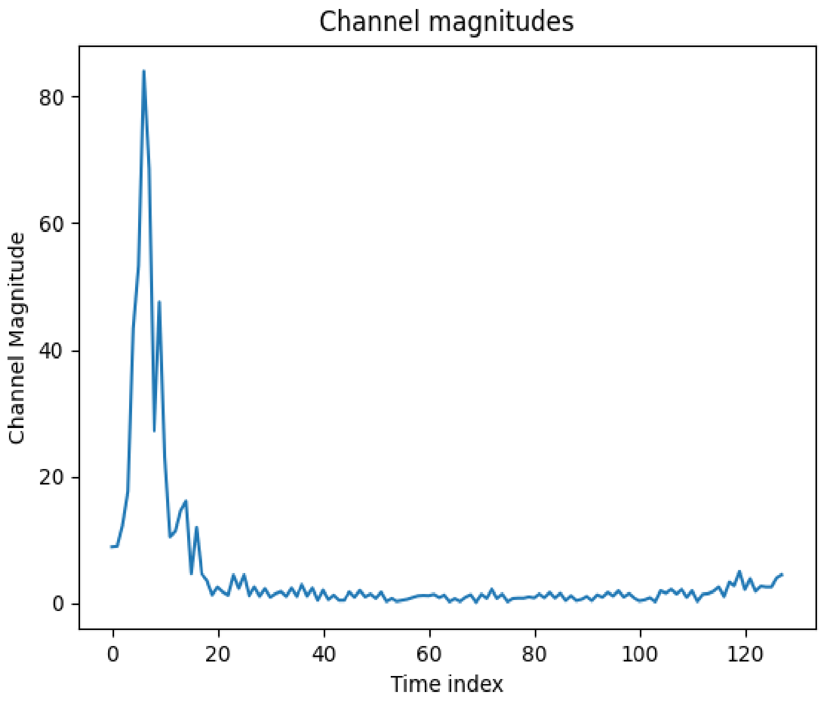


Range&Time of Arrival(ToA) estimation

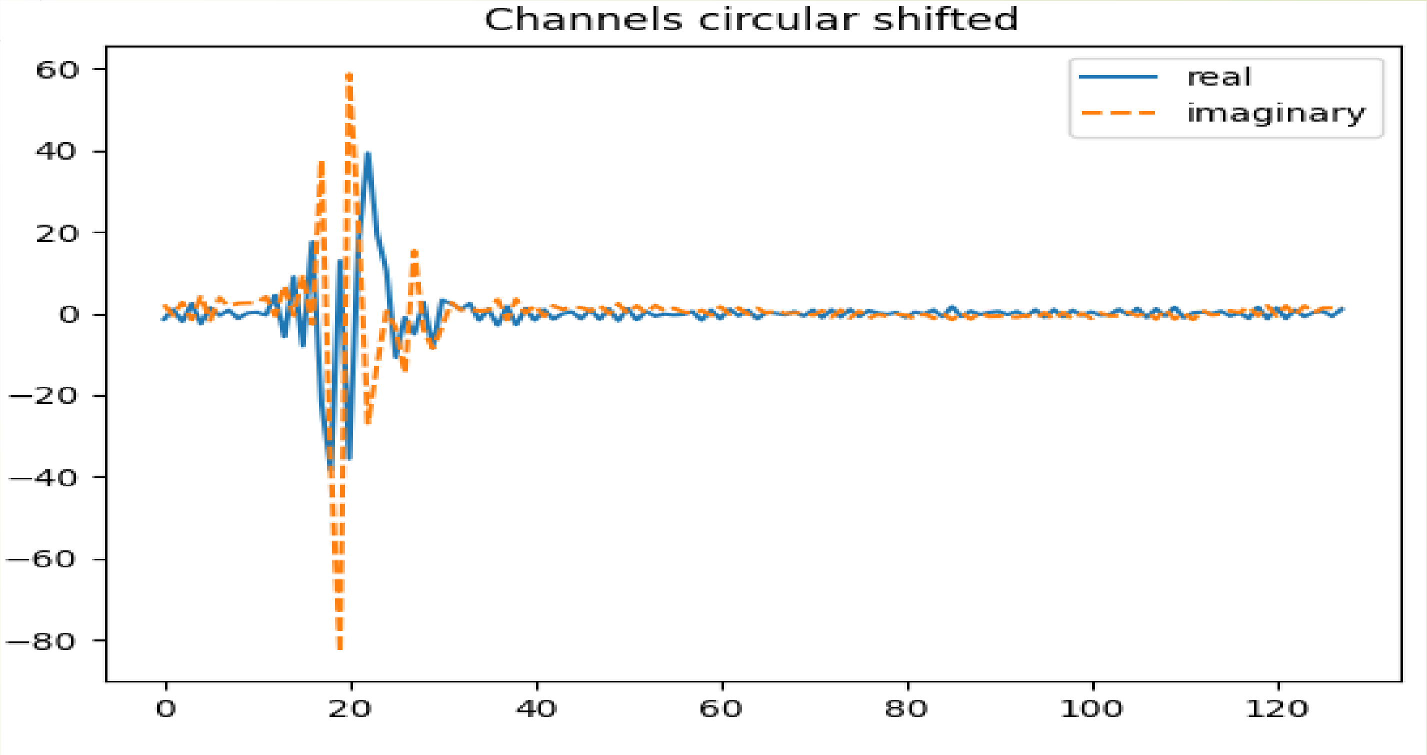
- ❑ **Extract channels** from the observation
- ❑ **Reshape** extracted channels [N,114, 2,2]
- ❑ **Zero pad** channels
- ❑ **Typecasting** from string to complex
- ❑ **Add** DC **tones** and guard subcarriers
- ❑ **Time domain conversion** using inverse fast Fourier Transform Shift (for index correction) and the Inverse Fast Fourier Transform

Time domain channel responses


Access point time domain channel


Channel magnitudes

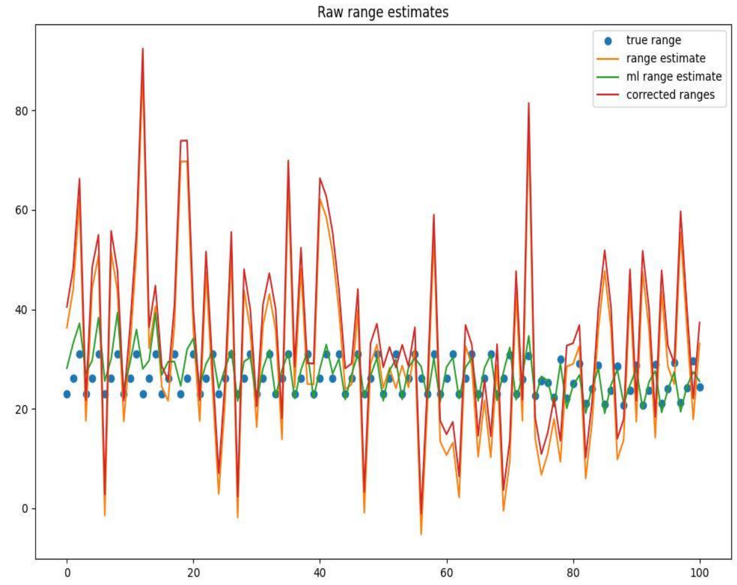
Circular shifted channels

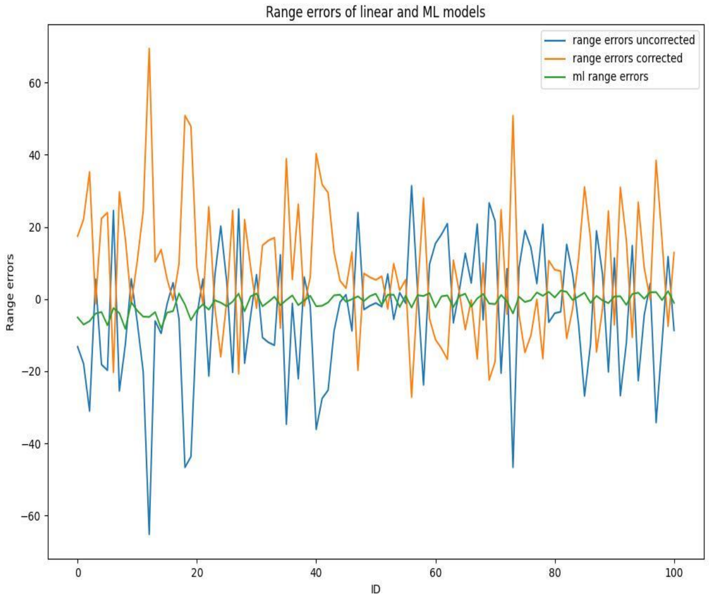


Range and ToA estimation cont’

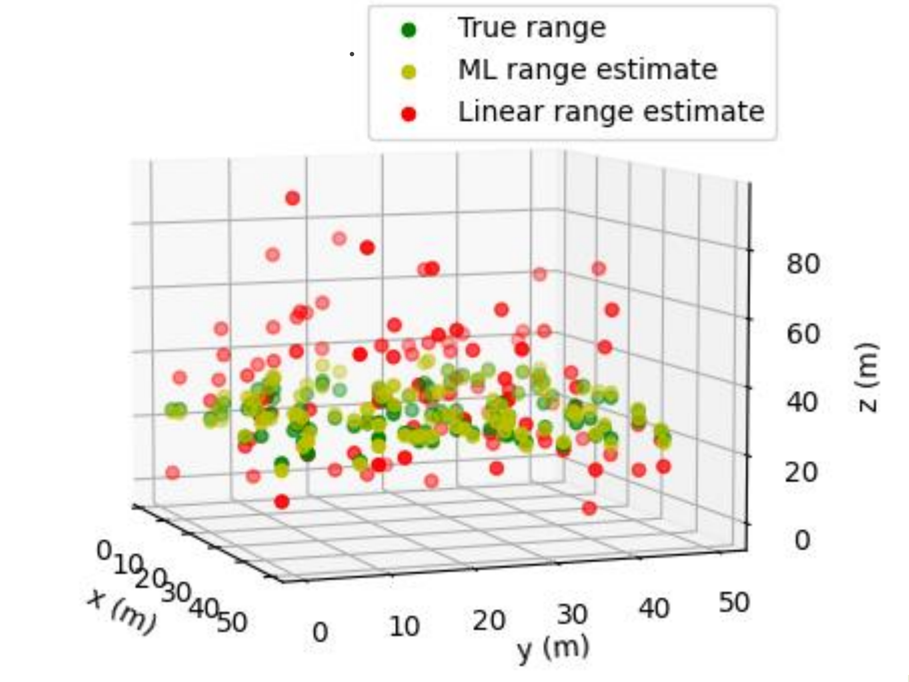
- **Channel circular** shift for better estimation, N= 13
- **Arg Max** of Time Domain channel
- **Correct circular shift** using the inverse operation
- Get time **delta per sample**, delta_t=1/bandwidth, bandwidth = 40MHz
- **ToA estimate** = ToA index*delta_t
- **Range estimate**= ToD_factor + (est_ToA_client + est_ToA_AP)*c/2, c =3x10^8m/s

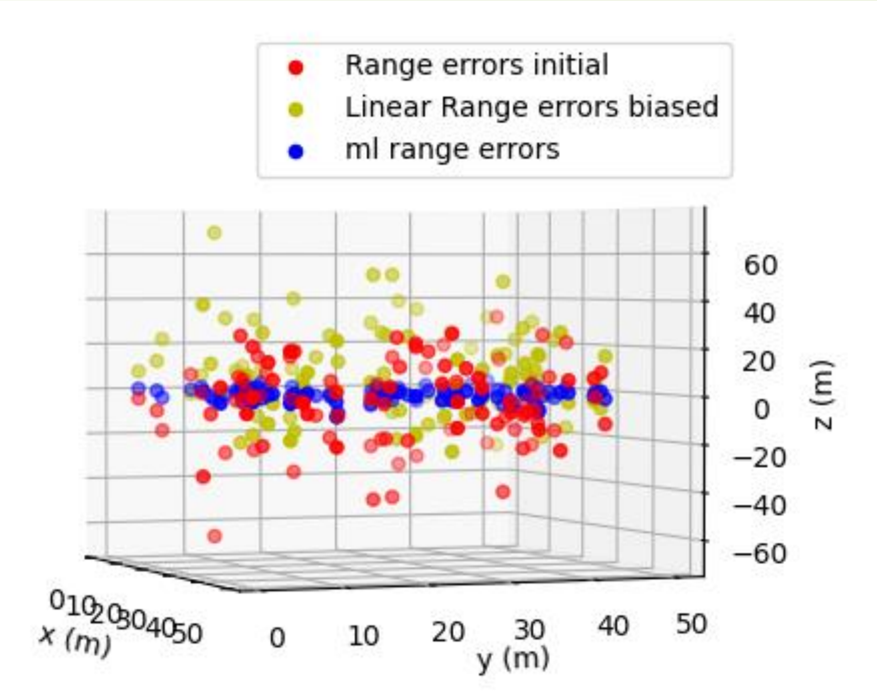
Range estimates and errors1D





Range estimates and errors 3D





Conclusion

- We used the Intel Open RTT Dataset
- A CLI-based software for automated analysis in 2 functional modes
- Statistical and ML model of RTT distributions

References

[1] N. Dvorecki, O. Bar-Shalom, L. Banin and Y. Amizur 'A Machine Learning Approach for Wi-Fi RTT Ranging' 2019 ITM of the ION, January 28-31, Reston, Virginia. Intel Communication & Device Group
Intel Corp.

THANK YOU