

# WirelessEye - Seeing over WiFi Made Accessible

Philipp H. Kindt, Cristian Turetta, Alejandro Masrur,  
Samarjit Chakraborty, Graziano Pravadelli, and **Florenc Demrozi**

*email: [florenc.demrozi@uis.no](mailto:florenc.demrozi@uis.no)*



University  
of Stavanger



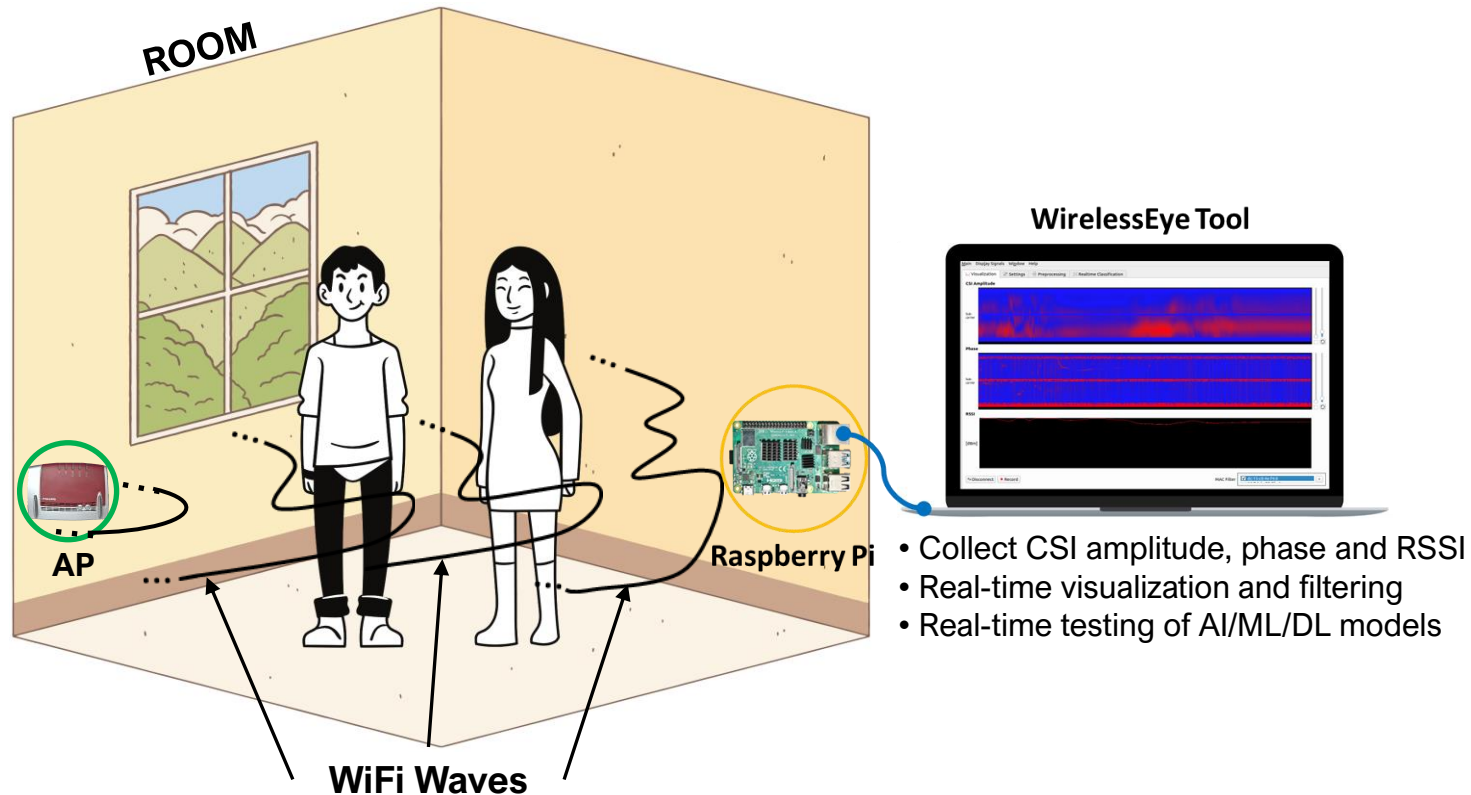
TECHNISCHE UNIVERSITÄT  
CHEMNITZ



THE UNIVERSITY  
of NORTH CAROLINA  
at CHAPEL HILL



# What is WiFi Sensing about?



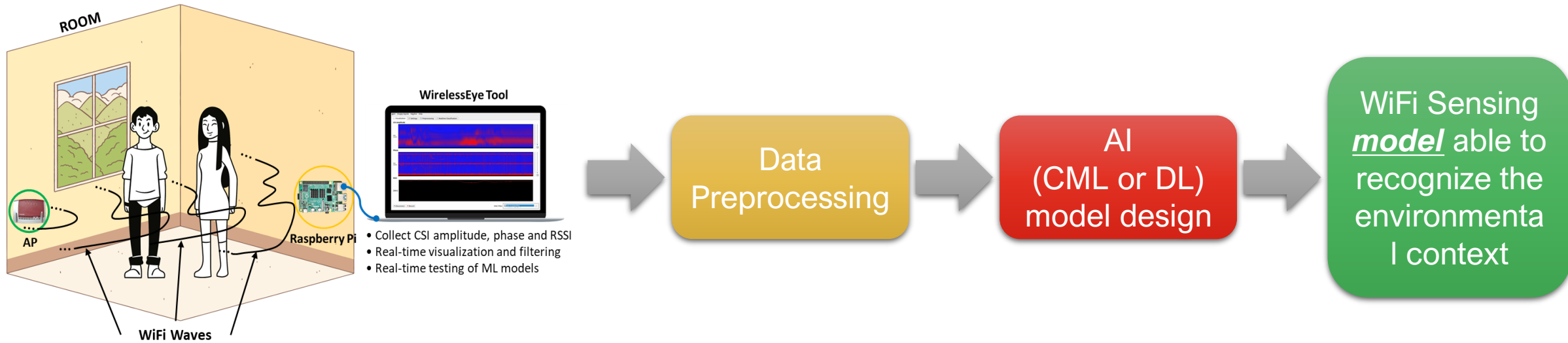
## Propagation pattern change due

- Walls
- Objects
- People
- Movements

## Used to

- Environment changes
- Objects position
- People
  - Position
  - Identity
  - Breathing phase
  - Heartbeat
  - Sleeping phase
- Movements

# WiFi Sensing pipeline from an AI perspective



Physics of  
radio signals

Artificial  
Intelligence

# What do we need to start with WiFi sensing?

Solid understanding of Wireless Communication

Programming Skills (Python, C/C++, or MATLAB)

Artificial Intelligence

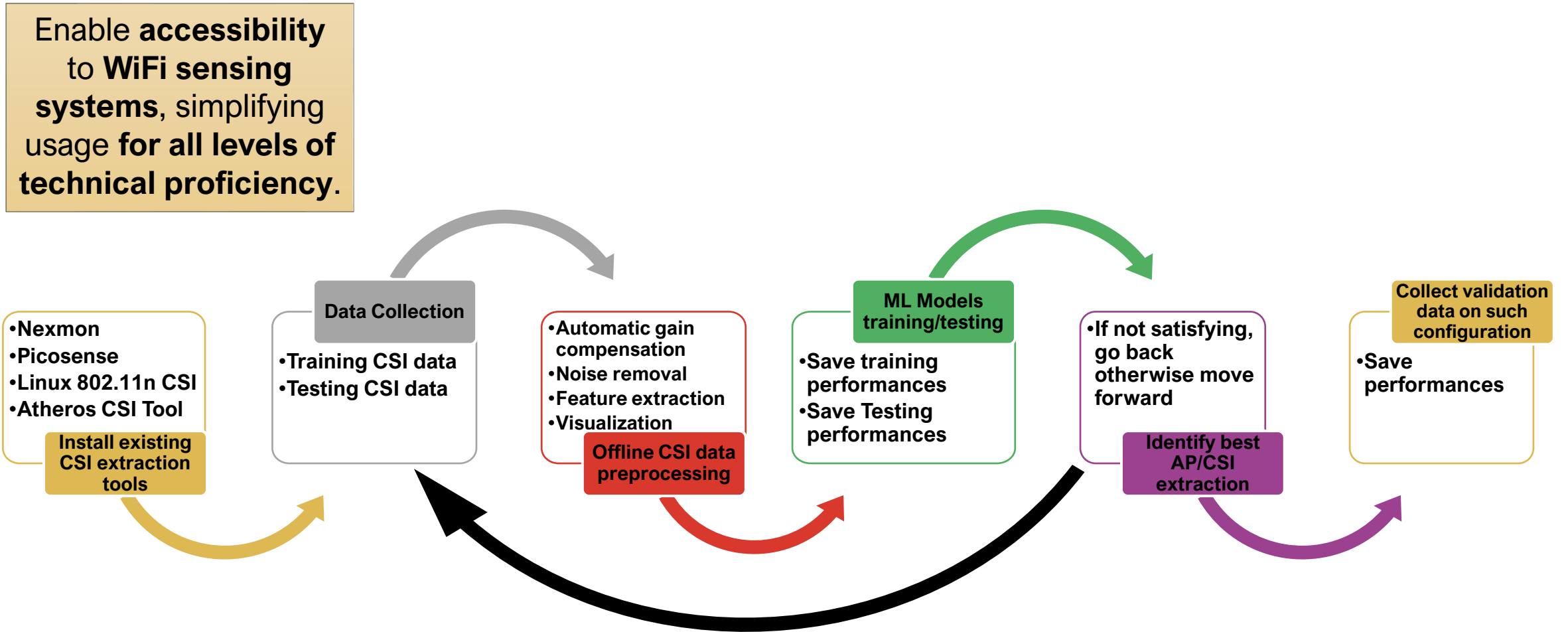


Networking Knowledge

Signal Processing

Hardware Knowledge

# Setting up a WiFi Sensing system



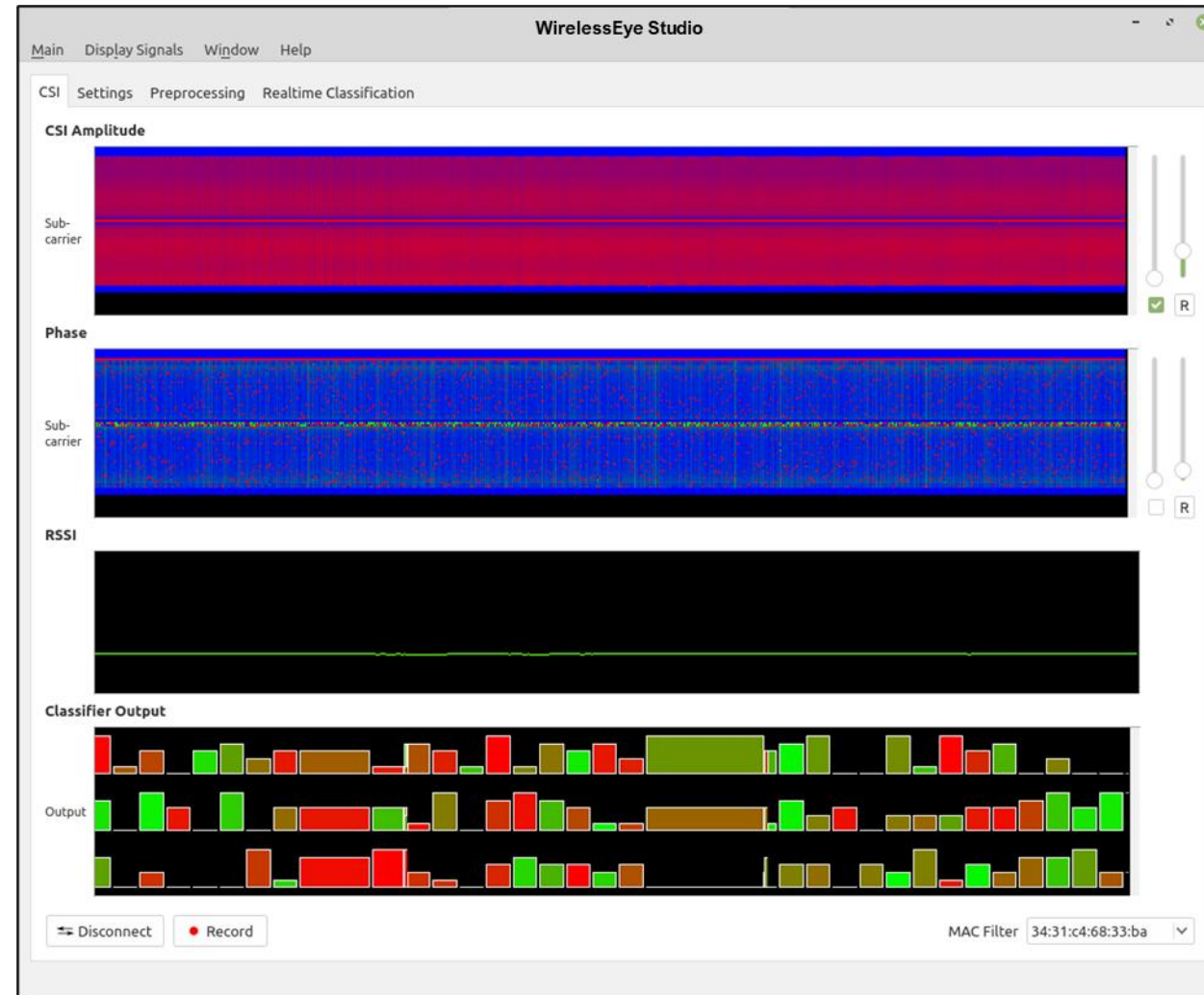


# Our Goal

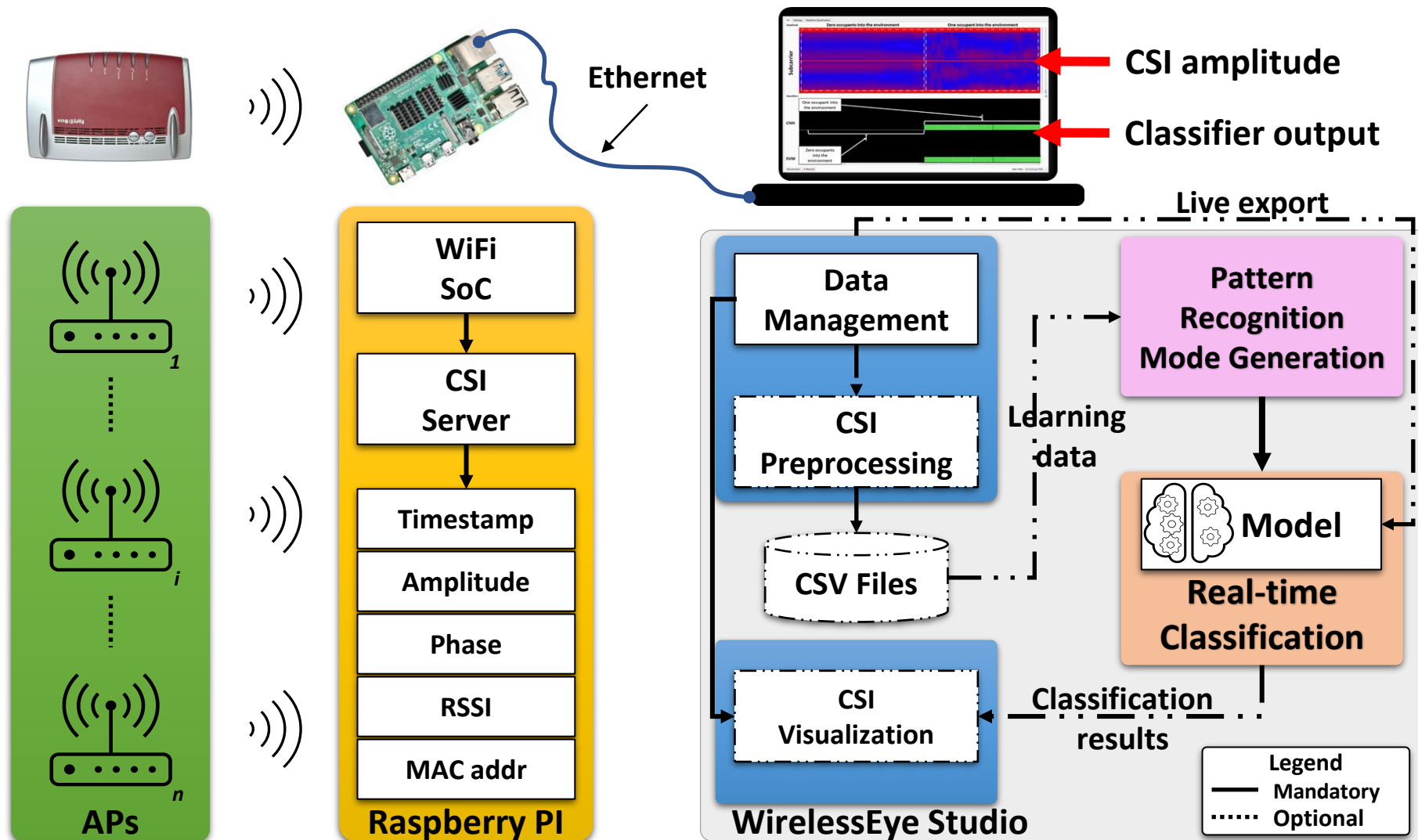
Enable **accessibility** to **WiFi sensing systems**, simplifying usage for all levels of **technical proficiency**.

Make **WirelessEye** freely accessible to the community.

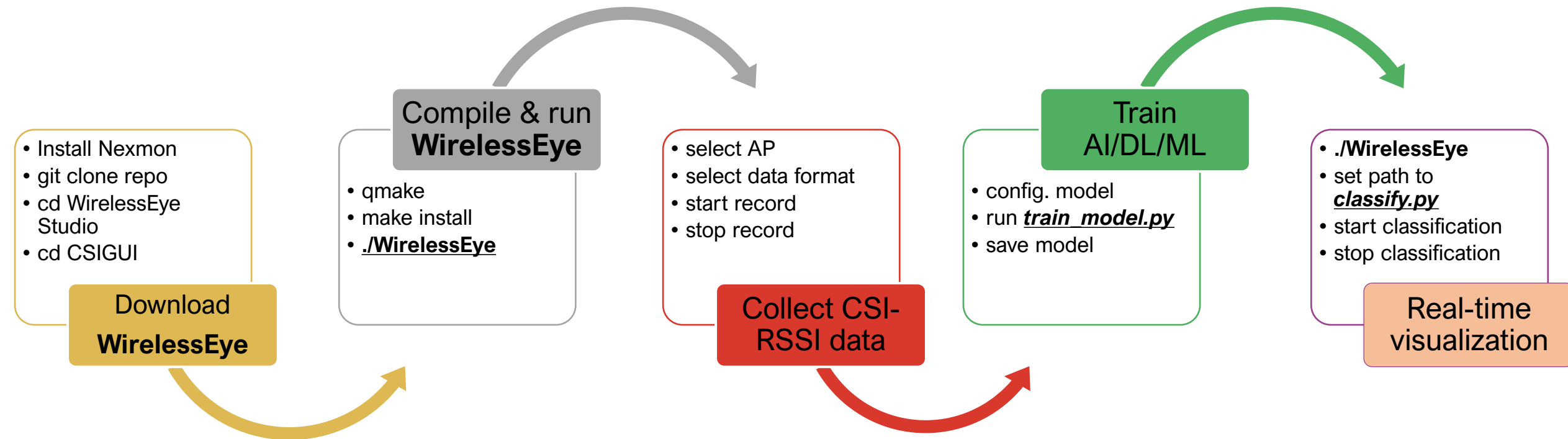
Git Page:  
[WirelessEye](#)



# Illustration of WirelessEye's workflow

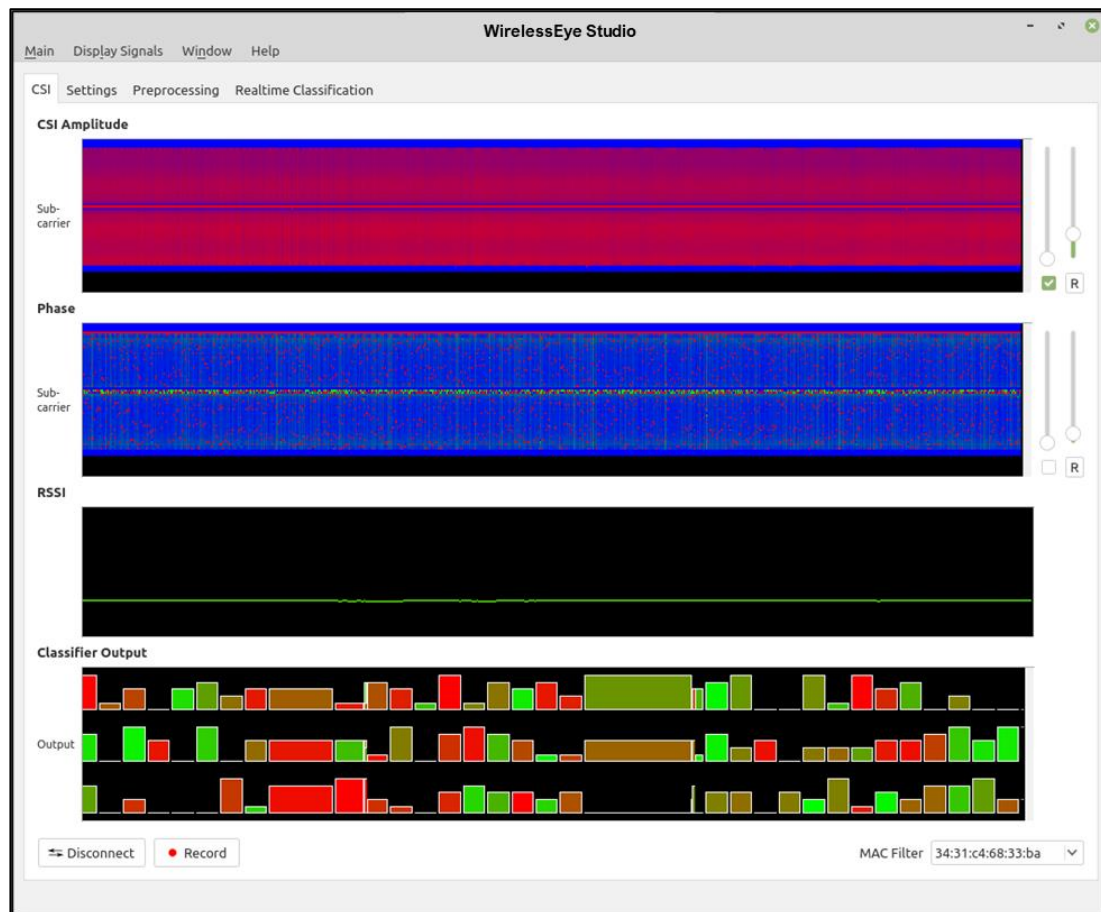


# WiFi Sensing pipeline setup using WirelessEye





# WirelessEye tabs (1)



The screenshot shows the **CSI Studio** application window, specifically the **Preprocessing** settings tab. The interface includes a menu bar (Main, Display Signals, Window, Help) and a list of preprocessing options:

- Corrects Gain Compensation using RSSI:** Priority 50,  Activated
- Exponential smoothing for the RSSI signal:** Priority 10,  Activated
- Add RSSI to CSI Amplitude:** Priority 1,  Activated
- Subcarrier Reordering:** Priority 1,  Activated
- Phase Unwrapping:** Priority 60,  Activated
  - Description: Unwraps the CSI Phase
  - Compensate Phase Difference Between Consecutive Frames:  activated
  - Form Difference Between two Consecutive Subcarriers:  activated
  - Support multiple sender MACs:  activated
  - Exclude invalid subcarriers (guard carriers and similar):  activated
  - Subcarriers to Exclude (if excluding is activated): `0,1,2,3,31,61,62,63,64`
- A simple sample filter:** Priority 1,  Activated

A **Refresh Filters** button is located at the bottom of the settings panel.

The screenshot shows the **Visualization** settings tab in the application. The interface includes a menu bar (Visualization, Settings, Preprocessing, Real-Time Classification) and a **Connection** section:

- Connection Type:**
  - UDP Broadcast on Port 5500
  - TCP Connection with Seryer
- Raspi IP or Hostname:** `raspberrypi`

# WirelessEye tabs (2)

Visualization Settings Preprocessing Real-Time Classification

Connection CSI Recording Visualization Preprocessing

CSI Settings

Actual Channel Bandwidth 80 MHz

Display Bandwidth 80 MHz

Export Bandwidth 80 MHz

MAC Filter

Apply to File Recording

Apply for Live Export to Classifier



Visualization Settings Preprocessing Real-Time Classification

Connection CSI Recording Visualization Preprocessing

Recording

Automatic Filename

Static Filename: CSICapture.cvs

Data Format (does not apply to live export)

Simple CSV (One Line per Subcarrier)

WifEye Binary

Compact CSV (One Line per Frame)

The format of the exchanged data is specified in the file *fileFormat* which can be found in the same folder as the WifEye executable

Visualization Settings Preprocessing Real-Time Classification

Connection CSI Recording Visualization Preprocessing

Generic Display Settings

Display Area Width  Adjust to Window

Static Size

2000px

Scroll Delay x1

Refresh Period 42ms

Signals to Display

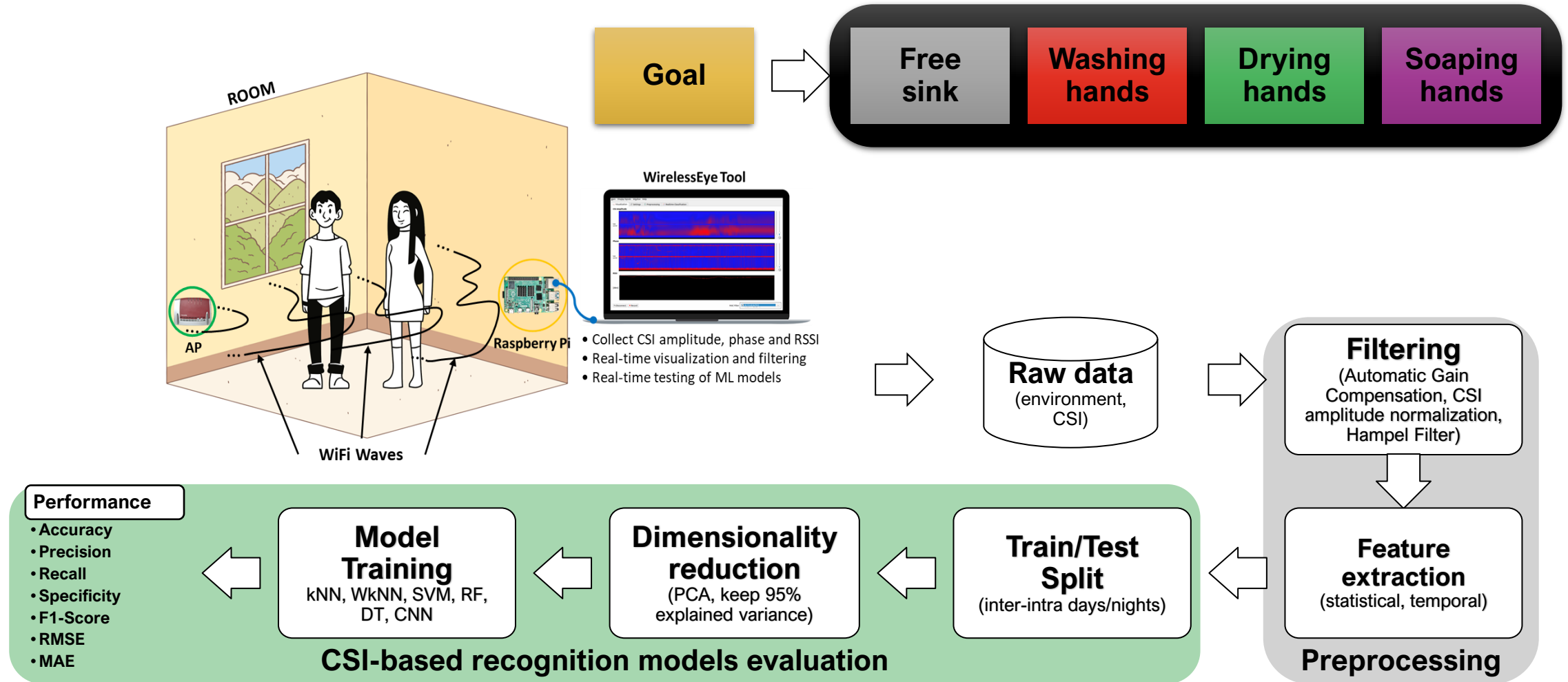
Display Amplitude

Display Phase

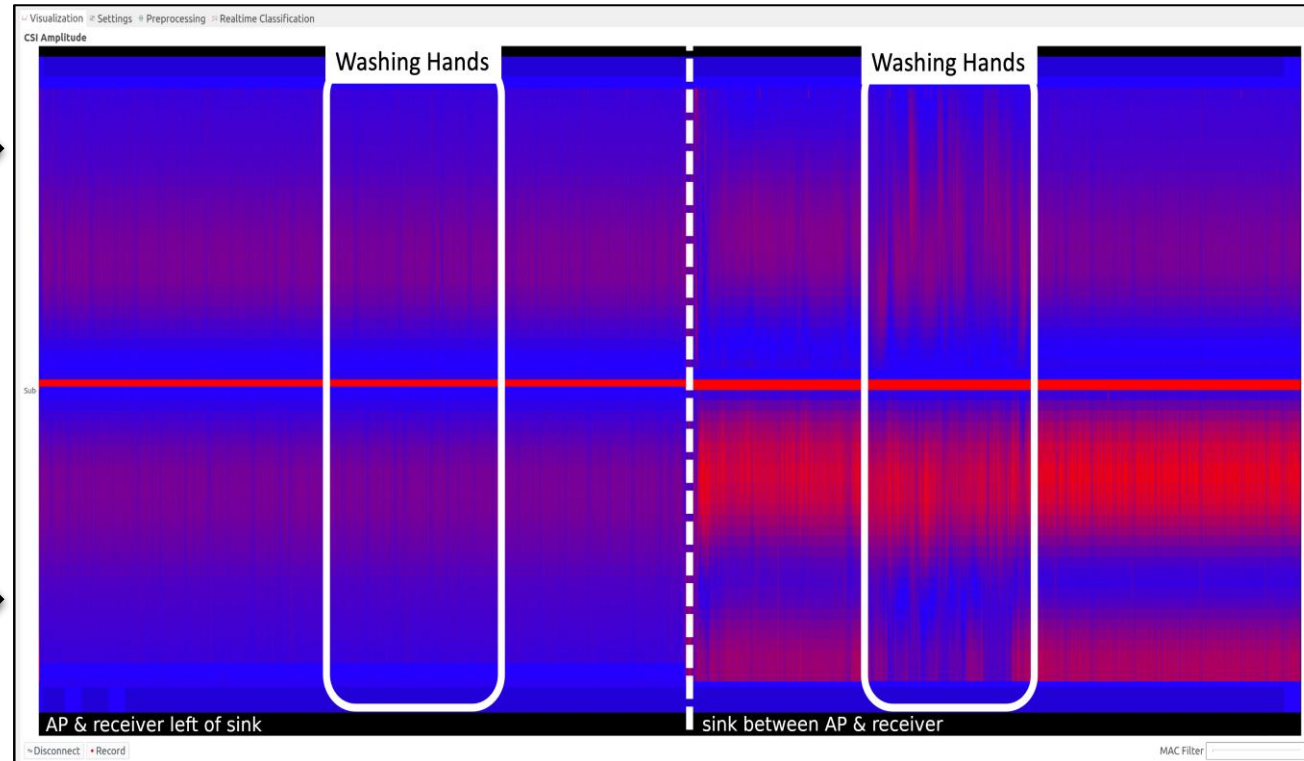
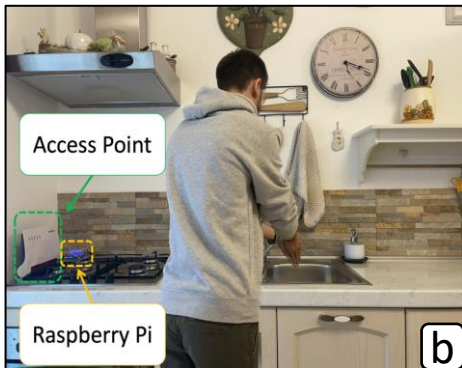
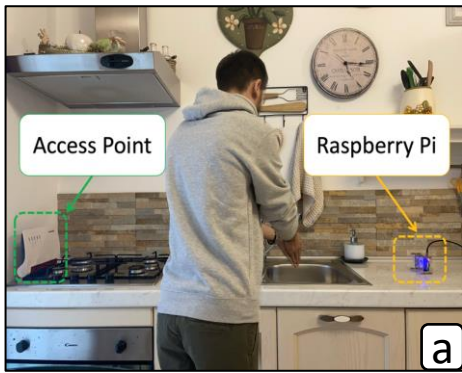
Display RSSI

Display Graphical Classifier Output Wid

# Let's recognize the Washing Hands action



# Recognizing washing hands with WirelessEye

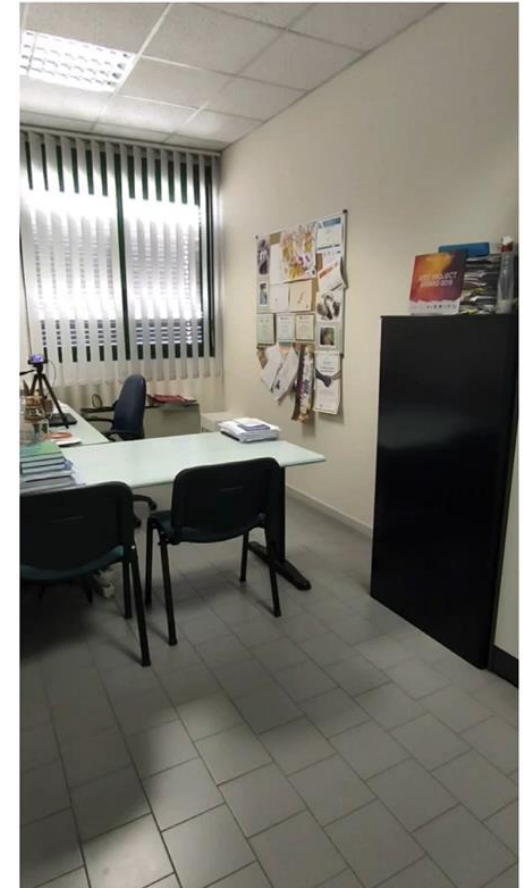
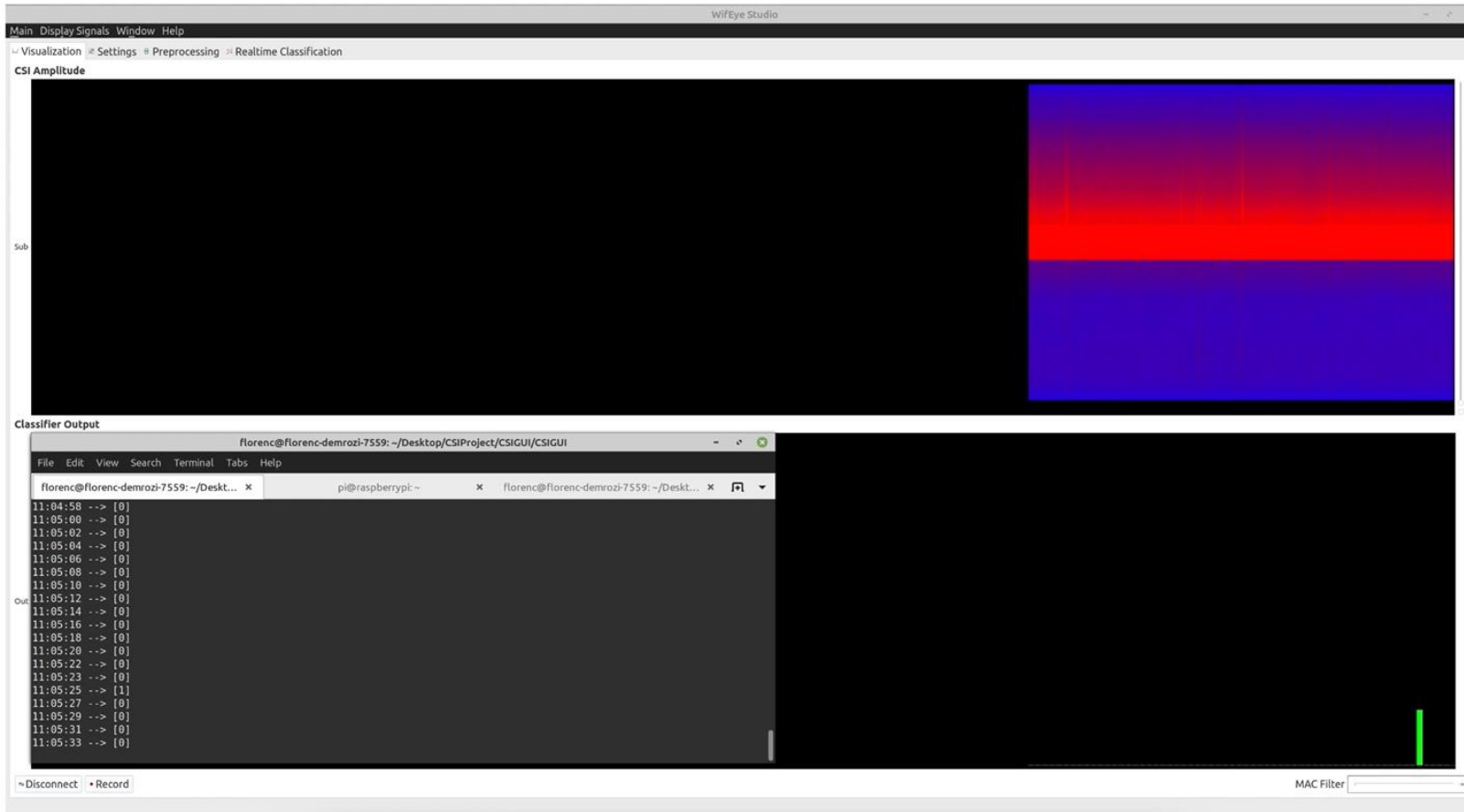


free sink	97.2% 70/72	0% 0//72	0% 0//72	0% 0//72
washing hands	1.3% 1/75	98.7% 74/75	0% 0/75	0% 0/75
drying hands	0% 0/66	0% 0/66	95.5% 63/66	4.5% 3/66
soaping	0% 0/71	0% 0/71	2.8% 2/71	97.2% 69/71
	free sink	washing hands	drying hands	soaping

free sink	78.7% 59/75	4.0% 3/75	6.7% 5/75	10.7% 8/75
washing hands	0% 0/85	97.6% 83/85	0.0% 0/85	2.4% 2/85
drying hands	50% 35/70	24.3% 17/70	2.9% 2/70	22.9% 19/70
soaping	6.7% 5/75	44.0% 33/75	2.7% 2/75	46.7% 35/75
	free sink	washing hands	drying hands	soaping

Moreover, WirelessEye have been used to collect data for seven consecutive days from multiple AP simultaneously without data loss

# WirelessEye Demo





# Conclusions and Future work

Git Page:  
[WirelessEye](#)



## Conclusions

- WirelessEye is open-source and available on Github.
- Includes all steps of the design of WiFi-sensing systems.
- High stability during the data collection and processing.
- Highly configurable.
- Real-time visualization of WiFi-sensing model results.
- Accepted also as Artifact in PerCom 2024.

## Future Work

- Implement as a WebApp.
- Sensor fusion with a Wireless Body Area Network and Video stream for the implementation of more complex Human Activity Recognition tasks.





# WirelessEye -- Seeing over WiFi Made Accessible

Git Page:

[WirelessEye](#)



Philipp H. Kindt, Cristian Turetta, Alejandro Masrur,  
Samarjit Chakraborty, Graziano Pravadelli, and [Florenc Demrozi](#)

[email: florenc.demrozi@uis.no](mailto:florenc.demrozi@uis.no)

