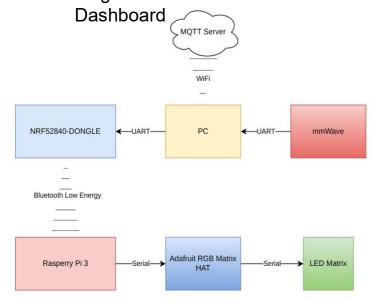
GESTURE RECOGNITION WITH MMW RADAR TRACKING

Dominic R, Hayden D, Ryan L and Sean C: Kratos-Red

SYSTEM OVERVIEW

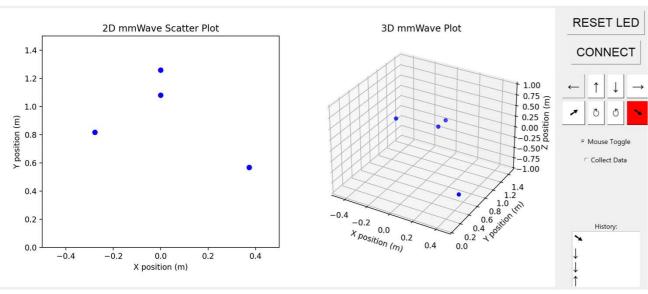
PC Node

 mmWave Sensor connected to PC (GUI). Data passed to ML model to classify gestures. Gestures correspond to HID (mouse) actions and LED Matrix transitions. Closest Target data transmit to Web



Raspberry Pi Node

 NRF Dongle connected to PC connects to Raspberry Pi Node using BLE. RGB LED Matrix adjusted through Pi controlled Adafruit HAT



KEY PERFORMANCE INDICATORS

Less than 2 second latency sending information to the MQTT dashboard

High-speed 1 second latency

Greater than 60% accuracy in recognising left, right, up and down gestures

 Achieved through filtering out static points with zero velocity

Less than 20% variation in recognition of the same gestures between different people.

Parameter Optimisation

Less than 20cm error in hand position

Achieved through thresholding

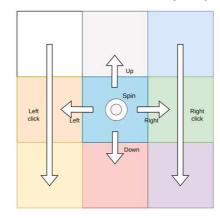
Can filter out background noise (people/objects) – Only one person can do gesture recognition

Achieved by focusing data

RESULTS

Gestures

Up - moves mouse & ref square up
Down - moves mouse & ref square down
Left - moves mouse & ref square left
Right - moves mouse & ref square right
Spin - rotates mouse & ref square 90°
Left Click - on mouse & flip square up
Right Click - on mouse & flip square down



ML Model

- Gesture recognition algorithms
- 240 Input Size
- LSTM 2 hidden layers (size 32)

Adversity

CONCLUSIONS

- CNN, RNN, Transformer ML approaches tests and failed
- mmWave config had to be modified for UART parsing & noise
- Multithreading in python (GIL) causing latency and timing issues
- WINDOWS vs LINUX

Future Improvements

- More complex threshing and model architecture
- More training data
- Hardware upgrades to include full data collection
- Improved and expanded gesture
- Generic accessibility for hardware and software

BIG DATA - MMW - ML - MQTT - GPU ACCEL. -

SQL - CLOUD DB - WIFI - AGILE DEV

