Magic Sensors

Team Name: sddec21-09
Team Members: Abdelaziz Hassan, Jared Hermon, Chinar Kaul, Ryan Lanciloti, Alyssa Marshall, Ben Pierre
Team Email: sddec21-09@iastate.edu
Team Website: https://sddec21-09.sd.ece.iastate.edu/
Advisor/Client: Dr. Daji Qiao, daji@iastate.edu

Problem Statement

"Currently in the market, door sensors (used in home security systems) are battery powered or need to be wired to a control unit.

The goal of this project is to design and implement innovative 'magic sensors' which are wireless and batteryless.

These 'magic sensors' will 'report' the status of the door (whether it is open or closed) to the control unit without a power source [on the door or frame] (such as battery)." (Professor Qiao, Project Abstract)

Requirements

Functional

- Detect if a door is open or closed with 85% accuracy with false-positive reporting around 5% to 10% and false-negative reporting less than 5%
- Detect a door's state up to 10 feet away from the base unit
- No powered no wired sensor on the door
- Alert customer via an application on their phone within 5 seconds if a door opens or closes
- User capable of arming and disarming system

Non Functional

- $300 budget for total system
- The door module must be less than $30 to allow more reflectors to be added to the system in the future
- UI must conform to current ARIA standards
- The door module must weigh less than one pound and be less than 4" by 4" when installed

Technical/Other Constraints

Constraints:

- The system will only be used inside
- Doors will be within 10 feet of the base station
- The user owns a device capable of connecting to our web app

Technical Constraints:

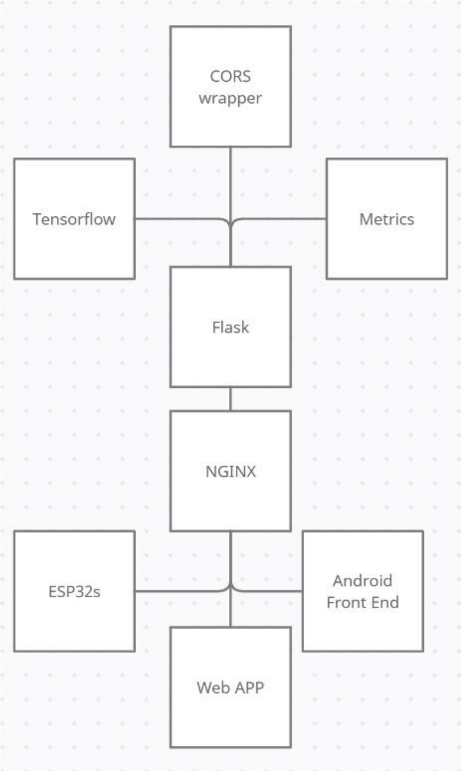
- Door module must not use a battery
- Door module must not use wires for power
- Door module must not permanently modify the door

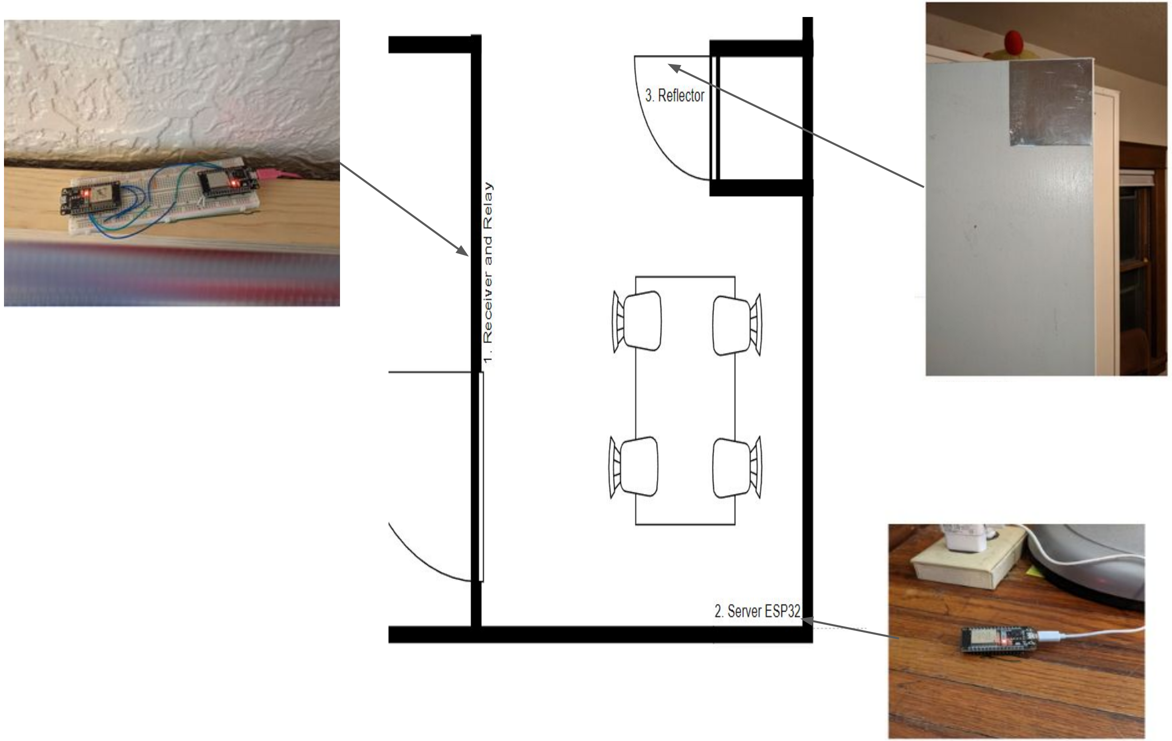
Our Solution

Our solution was to use Channel State Information (CSI) data. This came from research and guidance from our client and advisor.

We established a link between two ESP32 microcontrollers in the same room as the door, with a reflector placed upon the door. When the door is moved we used the ESP32s to monitor changes to the CSI data, which we passed to a machine learning algorithm to generate a prediction on door state.

System Diagram





Resource/Cost Estimate

- ESP 32 - 3x - 3*$5 = $15
- Reflector - 3in x 3in aluminum = $3
- Server Rental = Highly Variable = Estimated at $5/client/month

Total = $20 + $5/Month

Design - Hardware

- Access Point
  - This component acts as a wifi access point, broadcasting an SSID others can connect to. Acts as a mirror, replying to anything it receives with a duplicate.
    - This operates on channel 8, which is blocked on our home internet.
- Connector
  - This ESP32 connects to the wifi network being broadcast by the Access Point ESP32.
  - It then floods it with UDP packets whose payloads are simply the char ‘1’.
  - On response from the AP it grabs the CSI header from that response, calculates CSI values and writes over UART.
    - Calculations are adjustable depending on the scenario
    - MAC filtered
- Relay
  - This ESP32 is connected to the Connector ESP32 via an 115200 bps uart connection.
  - It is also connected to the internet via our home wifi network.
  - It reads the UART information from the Connector ESP32 and posts it to an end point on our server.
  - We see approximately 3 data points per second with this configuration.

Design - Reflector

- Choose best reflective material from options: copper, aluminum, steel
- Benefit of Aluminum: cheaper, doesn't rust, and good reflector of RF waves
- Movement of metal on door will cause an observable change in both amplitude and phase
- Aluminum provides the minimum alteration to the appearance of the door and fits our weight and size requirements- tape width is 2” wide and Aluminum foil adhesive thickness: 3.6 mil
- Flat design with strong yet non-invasive adhesive

Design - Server

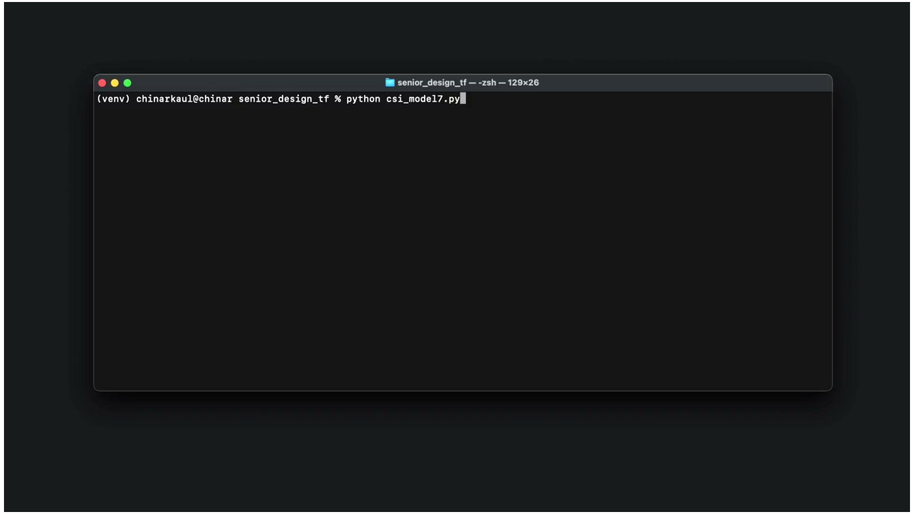
- We have a VM spun up on ISU's servers courtesy of ETG.
    - As this is behind a firewall, we also have an NGINX reverse proxy setup that allows us to access it from the outside networks.
- The server on ISU's network is running two services
    - Apache web server hosting module documentation (port 20001)
    - Flask service acting as an API for the front end (port 20002)
- The backend flask server handles all of the following logic:
    - Parse and throw away bad packets from the ESP32s
    - Train the machine learning model
    - Run inference on incoming CSI data
    - General API functionality

Design - Machine Learning

- Multi-class neural network to predict door state based on incoming CSI data.
- Incoming CSI data received, model called by endpoint, prediction generated.
- Model was developed using Keras and Scikit-learn in Python.
  - Loss function: categorical_crossentropy to handle multiple angle labels.
  - Optimizer: Adam, very common adaptive optimizer algorithm.

Layer	Nodes	Activation Function	Justification
1	53	sigmoid	Can predict probability [0, 1]
2	200	relu	Non-linear activation function.
3	200	relu	Non-linear activation function.
4	7	softmax	Common output function for multiclass models.

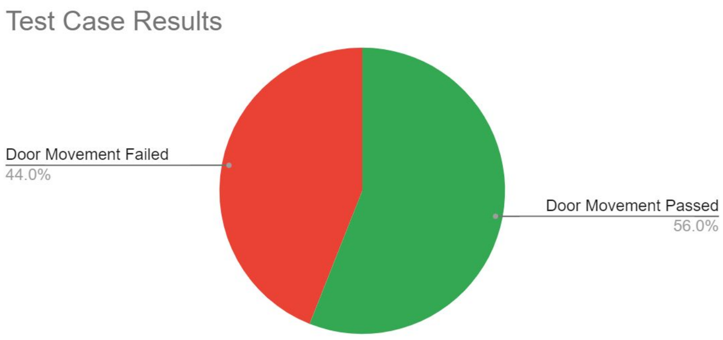
Demo


senior_design_tf — -zsh — 129×26
(venv) chinarkaul@chinar senior_design_tf % python csi_model7.py

Testing

- Developed acceptance testing document that outlined, setup, verification, and testing procedures that allowed us to quantify the results in terms of pass/fail accuracy.
- Acceptance testing was done with local machine learning algorithm output

			in terms of buckets			
#	expected result	actual result	Pass or fail	Timing	Pass or Fail	comı
3.1	0-15	0-15	Pass	5.5	Fail	
3.2	0-15	0-15	Pass	6	Fail	
3.3	0-15	0-15	Pass	11.2	Fail	
3.4	0-15	0-15	Pass	16.6	Fail	
3.5	0-15	15-30	Fail	6.7	Fail	
3.6	15-30	0-15	Fail	15.6	Fail	
3.7	15-30	15-30	Pass	10.5	Fail	
3.8	15-30	0-15	Fail	11.7	Fail	
3.9	15-30	15-30	Pass	11.5	Fail	
3.10	30-45	30-45	Pass	18.6	Fail	
3.11	30-45	15-30	Fail	4.8	Pass	
3.12	30-45	30-45	Pass	10.5	Fail	
3.13	30-45	45-60	Fail	17.7	Fail	
3.14	45-60	45-60	Pass	4.7	Pass	
3.15	45-60	15-30	Fail	11.3	Fail	
3.16	45-60	60-75	Fail	14.2	Fail	



Acknowledgements and Thank you!

- Dr. Daji Qiao
- Dr. Andrew Bolstad
- Dr. Mani Mina
- Wei Shen Theh
- Steven M. Hernandez
- Jonathan Muller

Thanks for Listening!

Questions?