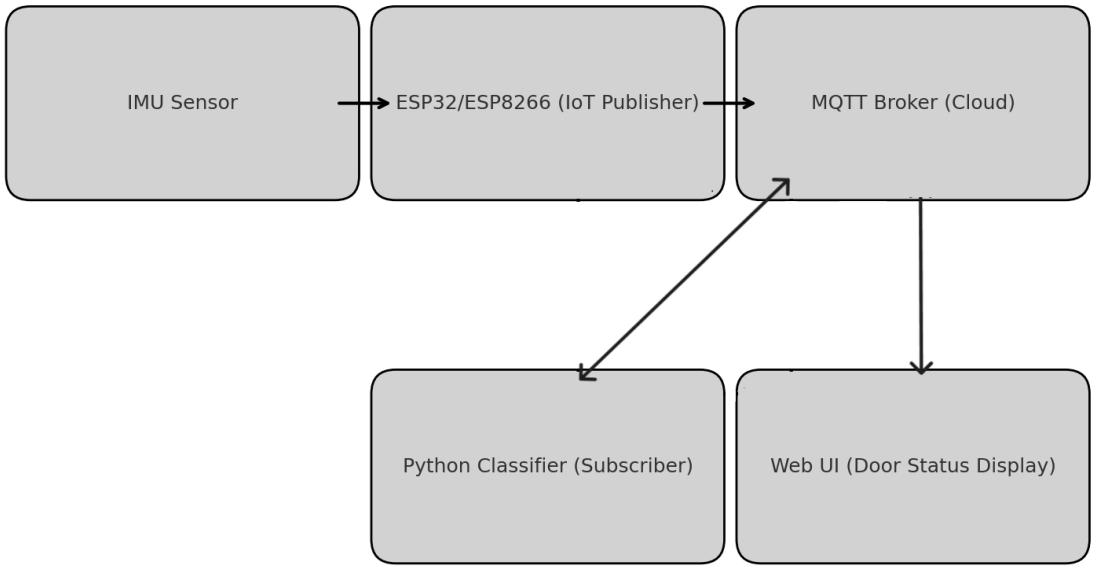
Alex Harvey - 25%
Amy Lee - 25%
Sayali Patukale -25%
Seth Schallau - 25%

	Seth	Sayali	Alex	Amy
IMU	20%	20%	30%	30%
Data Collection	30%	30%	20%	20%
Classifier Training	20%	30%	20%	30%
Classify talk	30%	20%	30%	20%
Cloud and MQTT Communication	30%	20%	20%	30%
UI Laptop	30%	30%	20%	20%
Report	20%	20%	30%	30%

1. System Architecture Overview

This project implements a smart door event detection system using an Inertial Measurement Unit (IMU), an ESP8622 IoT device, and cloud-based analytics. The system detects door open and close events by analyzing motion data—captured from the IMU's accelerometer and gyroscope components—and classifies these events using a pre-trained Support Vector Machine (SVM) model. The ESP8622 acts as the MQTT publisher, sending real-time sensor data to a cloud MQTT broker our team set up in AWS running on an EC2 instance. Another EC2 web server runs a service that is pulling from the MQTT broker and running the classification SVM model (svm_model.pkl) on the incoming sensor data.This server publishes the results of it’s classification back to another topic on the MQTT broker. A Flask-based web interface (web_server.py) can be run to display the real-time door status.

2. System Diagram



3. Feature Extraction and Selection Methods

Sensor data is initially recorded by the sensor in the serial monitor and is transformed into a csv file (self_labeling_data.csv). Feature extraction is a critical step in preparing the IMU data for training the SVM classifier. Key statistical and motion-based features are calculated to capture the dynamics of door movement. These include:

- Mean and standard deviation of acceleration and gyroscope signals along each axis.
- Maximum and minimum acceleration values to detect motion intensity.
- Angular velocity and net displacement estimation to capture door rotation and movement direction.
- Duration of the motion event for temporal context.
- Windowed temporal sequences (e.g., 3-second windows) to structure the data consistently.

These features are extracted from the IMU/ESP serial monitor. The selected features are chosen for their strong relevance to door motion patterns such as rotation, sudden stops, and directional changes. Once extracted, the dataset is labeled by event type (stable, moving (opening or closing)) and prepared for training the SVM classifier.

4. Classifier Training Method

The door event classifier is implemented using the Support Vector Machine (SVM) algorithm from the libSVM library, with training handled via the train_svm.py script. The training dataset consists of IMU sensor data collected from multiple door open and close instances, with manual labeling of data segments as "stable," "opening," "closing". Each data sample includes a feature vector composed of statistical features derived from sensor readings, formatted from cleaned CSV files. To enhance model performance, the data undergoes normalization and vector formatting before training. An SVM with a Radial Basis Function (RBF) kernel is used for its effectiveness in handling non-linear data distributions. Cross-validation is employed to evaluate model accuracy and optimize key hyperparameters such as the regularization parameter (C) and kernel coefficient (gamma). A confusion matrix is used to assess classification performance. The final trained model is saved as svm_model.pkl, which is later used in classify_talk.py to classify real-time sensor data.