Smartwatch-Based Sensing Framework for Continuous Data Collection: Design and Implementation

UbiComp2023 Workshop (Mental Health: S&I)
@2023.10.8 in Cancun, Mexico

Yuuki Nishiyama & Kaoru Sezaki

Center for Spatial Information Science
The University of Tokyo, Japan

nishiyama@csis.u-Tokyo.ac.jp

as a Passive Sensing Platform

as a Passive Sensing Platform

Resources

Almost the same sensors are inside the smartwatch:

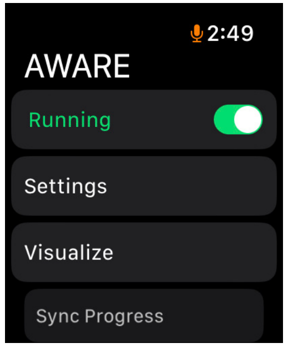
- Motion
- Heartrate
- Microphone
- Bluetooth
- Location
- Compass
- Wi-Fi

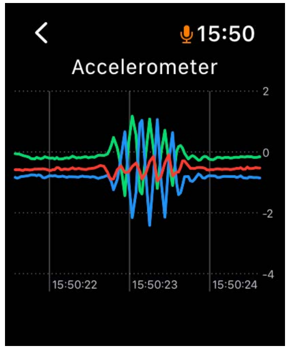
.. etc

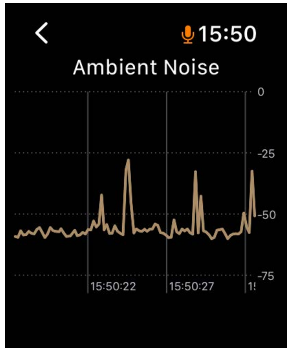
Limitation

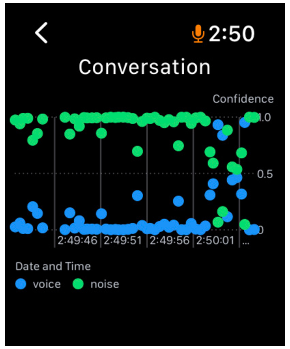
- Battery
- Communication cost between a hosted device
- CPU power
- Background sensing

We **designed and implemented a framework for smartwatch-based passive sensing**



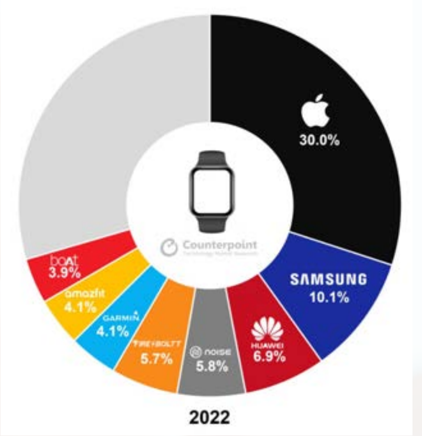






Background

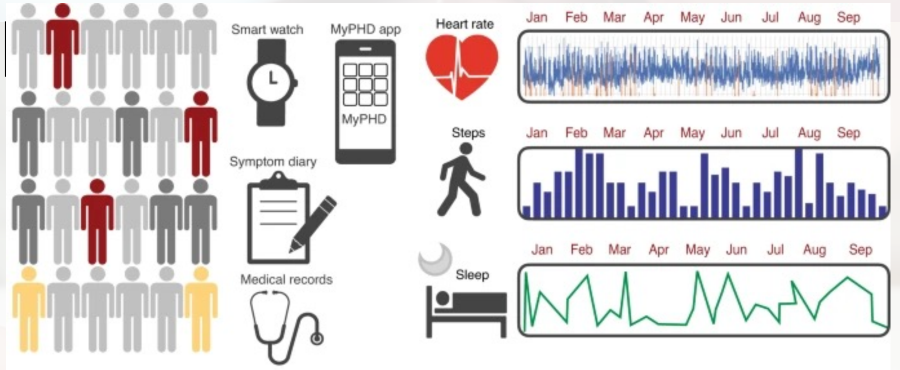
- The market for wearable devices, such as smartwatches, smart bands, and rings, has significantly expanded [1].



Market share of smartwatch

- A smartwatch has been **powerful** and has **rich sensors**: location, heart rate, motion, microphone, Wi-Fi, Bluetooth ··· etc

- Several passive mobile sensing researches used the limited sensor data from smartwatches, bands, or rings [2,3].



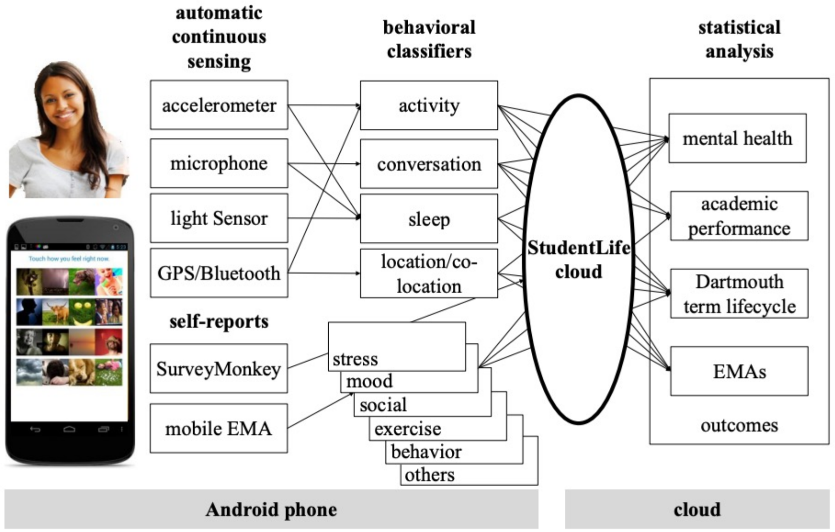
Overview of research method [2]

[1] https://www.counterpointresearch.com/insights/global-smartwatch-shipments-grow-yoy-2022/
[2] Mishra, T., Wang, M., Metwally, A.A. et al. Pre-symptomatic detection of COVID-19 from smartwatch data. Nat Biomed Eng 4, 1208–1220 (2020).
[3]Doryab, A., et al. 2019. Identifying Behavioral Phenotypes of Loneliness and Social Isolation with Passive Sensing: Statistical Analysis, Data Mining and Machine Learning of Smartphone and Fitbit Data. JMIR Mhealth Uhealth 7, 7 (Jul 2019), e13209.

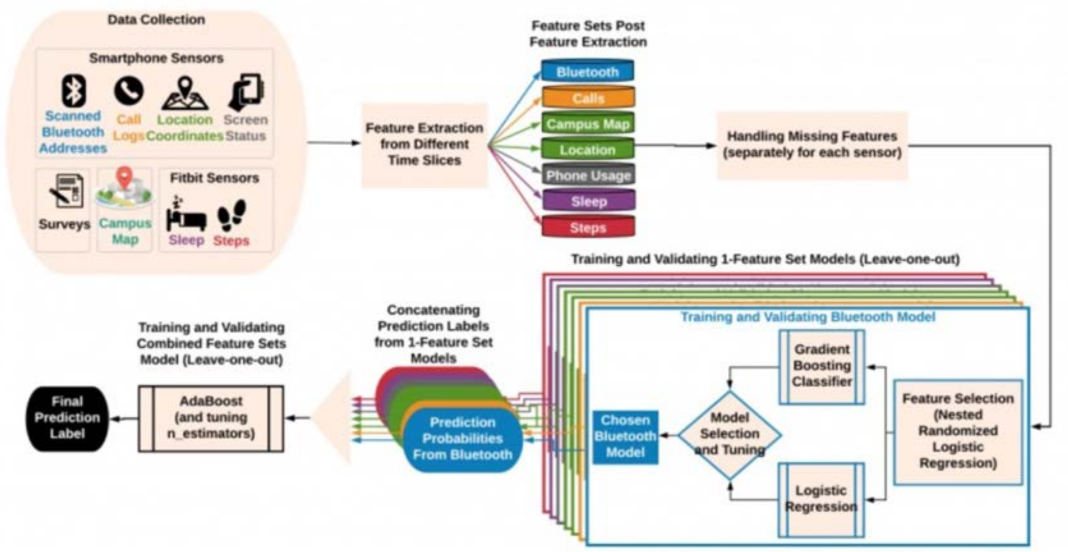
Related Works①:
Passive mobile sensing

Passively collected sensor data from mobile/wearable devices allow us to detect human conditions over their behaviors

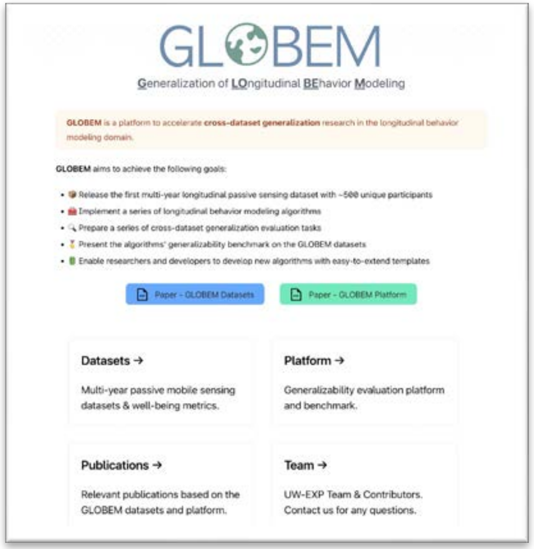
e.g., illness[2], mental health[3,4], and addiction[6]



[4]Collected sensor data on passive mobile sensing for mental health


Data Collection
Smartphone Sensors
Scanned Bluetooth Addresses
Call Logs
Location Coordinates
Screen Status
Surveys
Campus Map
Fitbit Sensors
Sleep
Steps
Feature Extraction from Different Time Slices
Feature Sets Post Feature Extraction
Bluetooth
Calls
Campus Map
Location
Phone Usage
Sleep
Steps
Handling Missing Features (separately for each sensor)
Training and Validating 1-Feature Set Models (Leave-one-out)
Training and Validating Bluetooth Model
Gradient Boosting Classifier
Feature Selection (Nested Randomized Logistic Regression)
Model Selection and Tuning
Logistic Regression
Chosen Bluetooth Model
Concatenating Prediction Labels from 1-Feature Set Models
Prediction Probabilities From Bluetooth
Training and Validating Combined Feature Sets Model (Leave-one-out)
AdaBoost (and tuning n_estimators)
Final Prediction Label

[3] Data processing pipeline for loneliness detection



GLOBEM

Generalization of LOngitudinal BEhavior Modeling

GLOBEM is a platform to accelerate **cross-dataset generalization** research in the longitudinal behavior modeling domain.

GLOBEM aims to achieve the following goals:

- Release the first multi-year longitudinal passive sensing dataset with ~500 unique participants
- Implement a series of longitudinal behavior modeling algorithms
- Prepare a series of cross-dataset generalization evaluation tasks
- Present the algorithms' generalizability benchmark on the GLOBEM datasets
- Enable researchers and developers to develop new algorithms with easy-to-extend templates

Datasets →

Multi-year passive mobile sensing datasets & well-being metrics.

Platform →

Generalizability evaluation platform and benchmark.

Publications →

Relevant publications based on the GLOBEM datasets and platform.

Team →

UW-EXP Team & Contributors. Contact us for any questions.

[5] Long-term and large-scale human behavior tracking study

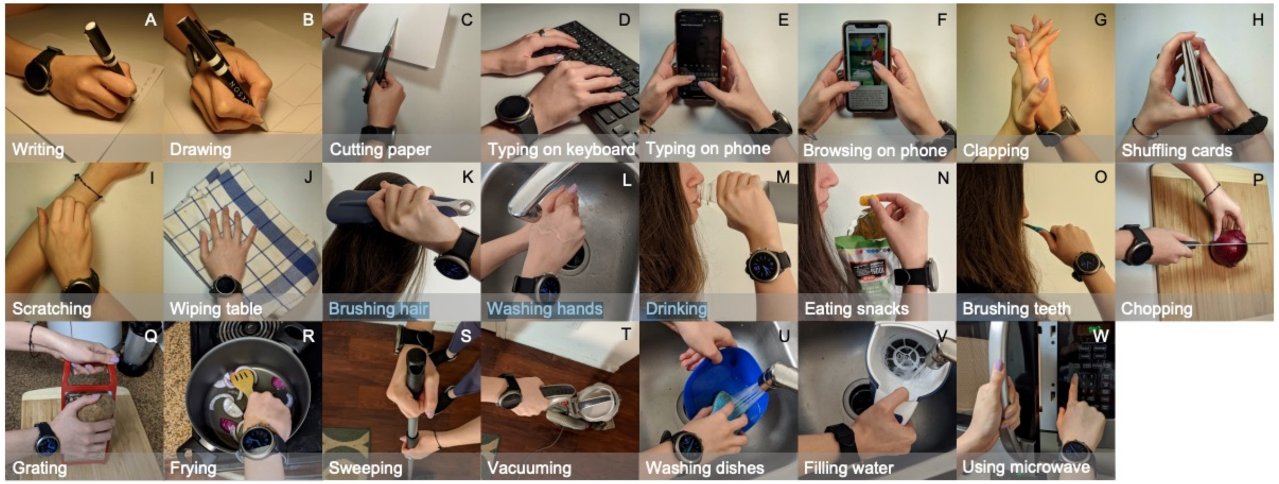
[4] Wang, R., et al.. StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students using Smartphones" In Proceedings of the ACM Conference on Ubiquitous Computing. 2014

[5] Xu X., et al., GLOBEM Dataset: Multi-Year Datasets for Longitudinal Human Behavior Modeling Generalization. In Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, page 18, 2022.

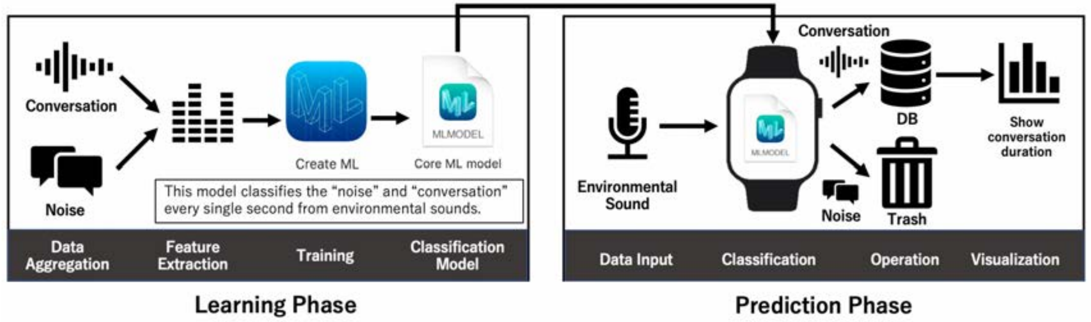
[6] Bae, S., et al., Mobile phone sensor-based detection of subjective cannabis intoxication in young adults: A feasibility study in real-world settings, In: Drug and Alcohol Dependence, pp. 108972, 2021.

Related Works②:
Context recognition using smartwatch

- **Activates of Daily Living (ADL)** detection by using **audio** and **motion sensors** on a smartwatch[7].
- Recording **daily conversation duration** by using a microphone on a smartwatch[8]



Target 23 types of ADL

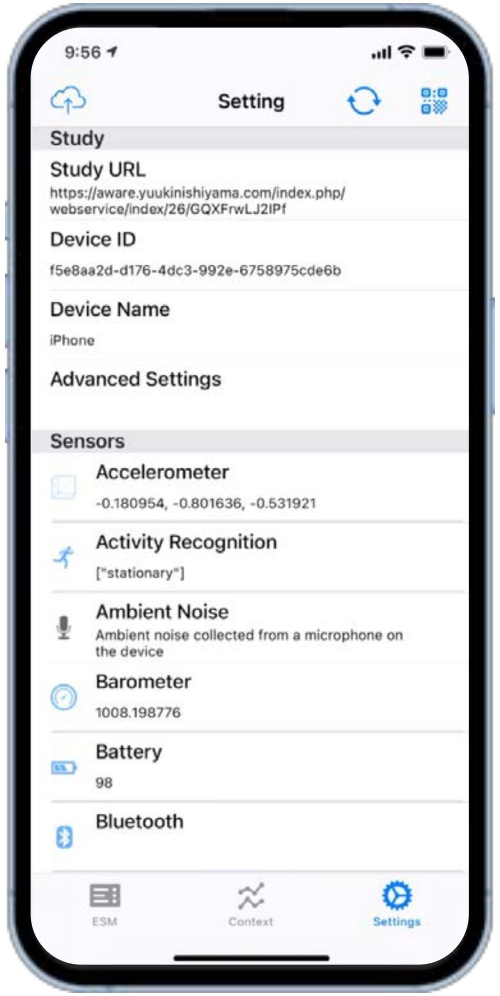


Pipeline for conversation detection

[7] Bhattacharya, S., et al., 2022. Leveraging Sound and Wrist Motion to Detect Activities of Daily Living with Commodity Smartwatches. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 6, 2, Article 42 (July 2022), 28 pages.
[8] Komatsu, Y. et al., "Toward Measuring Conversation Duration Using a Wristwatch-type Wearable Device," 2022 IEEE International Conference on Smart Computing (SMARTCOMP), Helsinki, Finland, 2022, pp. 150-152.

Related Works③:
Mobile sensing frameworks

- Several **passive mobile sensing frameworks** are available: **AWARE**[9,10]*, **CARP****, **Sensus*****, and more.
  - These frameworks allow us to stably and quickly collect sensor data on smartphones with a minimum workload.
- However, these frameworks **do not support collecting raw-sensor data** and **processing on the watch in the background.**



Study

Study URL
https://aware.yuukinishiyama.com/index.php/webservice/index/26/GQXFrwLJ2IPf

Device ID
f5e8aa2d-d176-4dc3-992e-6758975cde6b

Device Name
iPhone

Advanced Settings

Sensors

Accelerometer
-0.180954, -0.801636, -0.531921

Activity Recognition
["stationary"]

Ambient Noise
Ambient noise collected from a microphone on the device

Barometer
1008.198776

Battery
98

Bluetooth

[9] Ferreira, D., Kostakos, V., Dey, A.K. **“AWARE: mobile context instrumentation framework**.” Frontiers in ICT (Vol 2, Issue 6), 2015.
[10] Nishiyama, Y, Ferreira, D., Eigen, Y., Sasaki, W., Okoshi, T., Nakazawa, T., Dey, A.L., Sezaki, K., **“iOS Crowd-Sensing Won't Hurt a Bit!: AWARE Framework and Sustainable Study Guideline for iOS Platfor**m,” Distributed, Ambient and Pervasive Interactions, pp. 223–243, Springer International Publishing, Cham, 2020.
* https://www.awareframework.com
** https://carp.cachet.dk/
*** https://predictive-technology-laboratory.github.io/sensus/

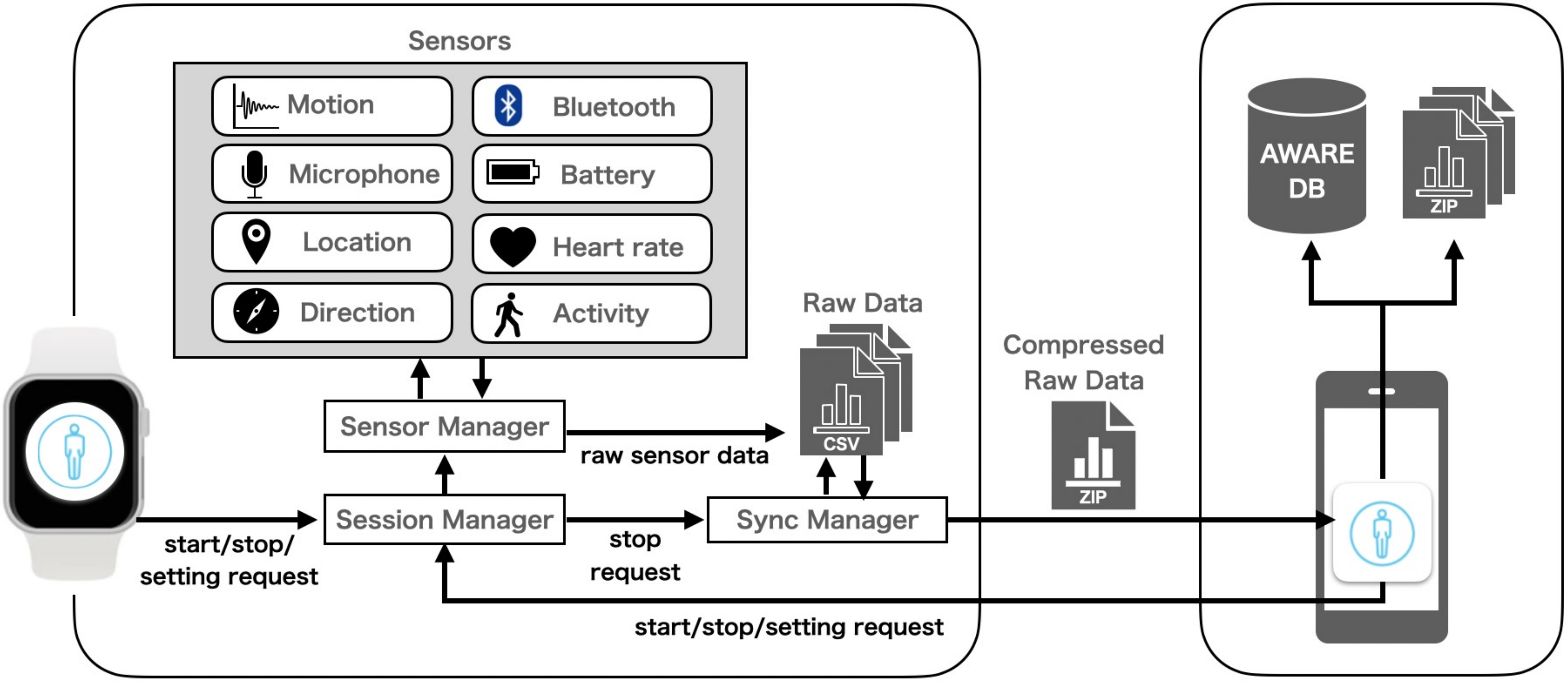
Motivation & Challenges

- **Motivation:**
  A framework collecting and processing raw sensor data continually creates lots of opportunities for the passive mobile sensing research field

- **Challenges:**
  - Designing and developing a **passive mobile sensing framework on smartwatches**, considering the limitations of smartwatch
  - Investigating and optimizing **battery, storage, communication, and background processing usage** on smartwatch

Design and Implementation:

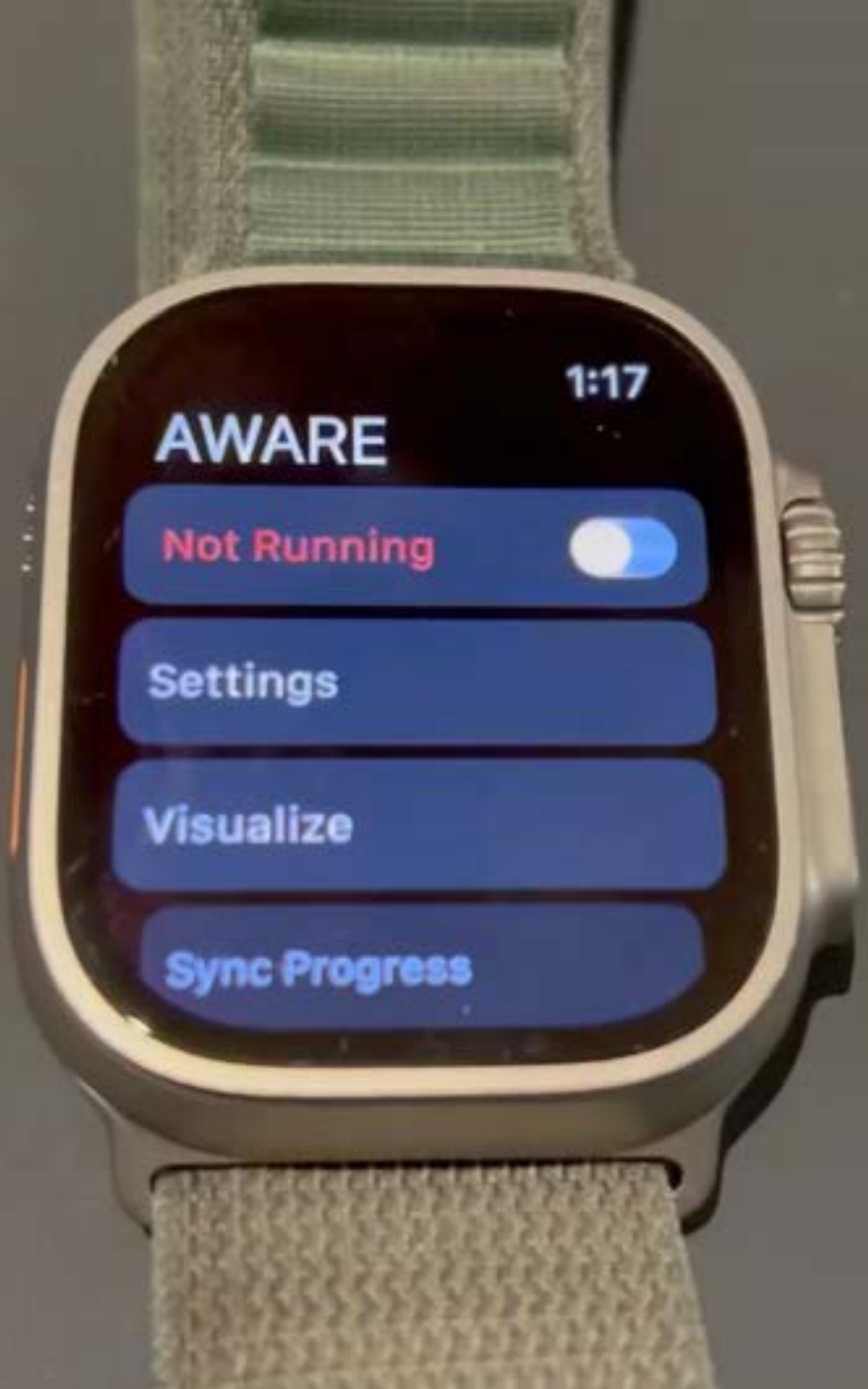
Passive mobile sensing framework on *watchOS*



Smart Watch (Apple Watch)

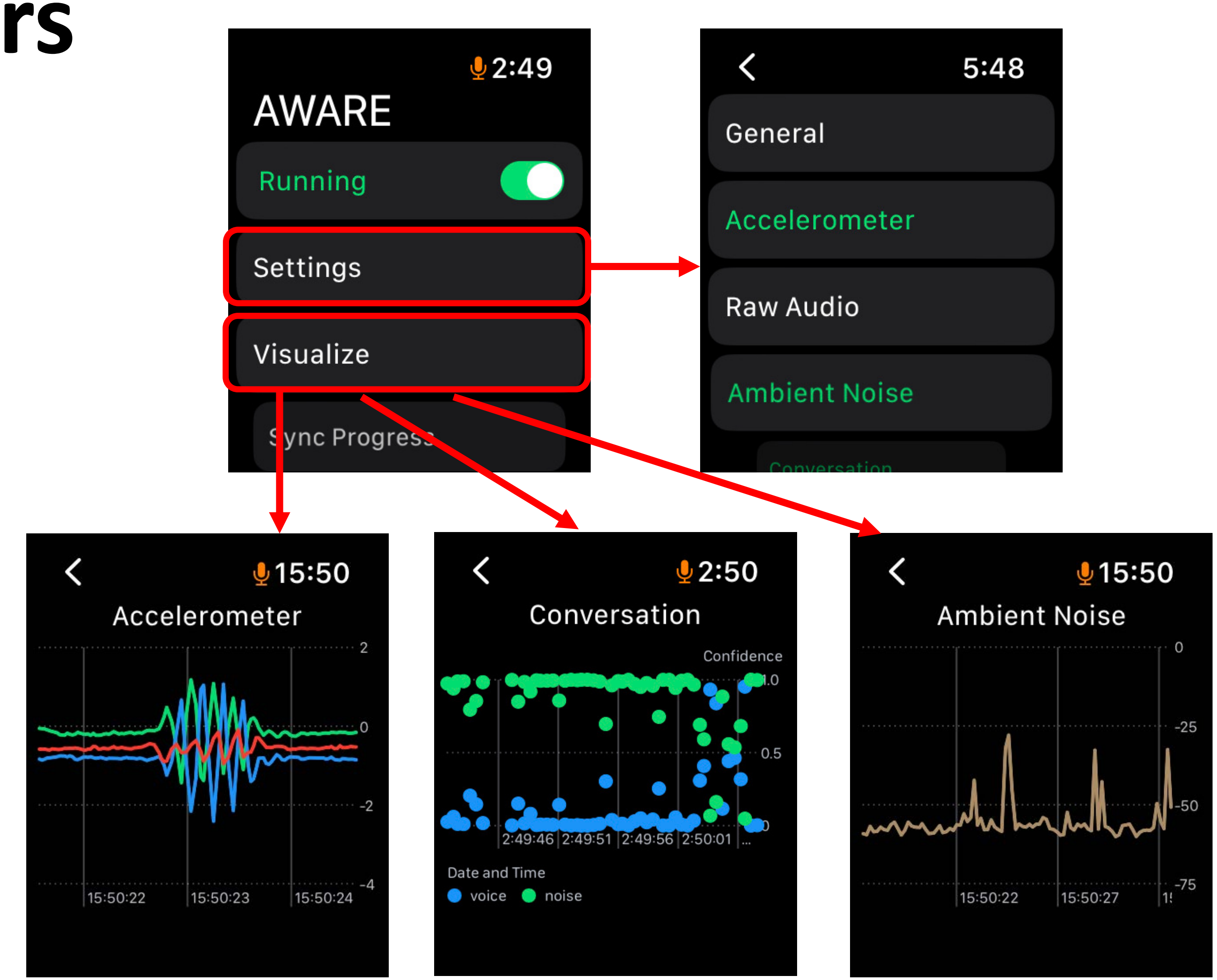
Smartphone (iPhone)

Demo: Sample Application



Supported Sensors

- Accelerometer
- Ambient noise
- Battery
- Barometer
- Bluetooth
- Conversation[8]
- Direction
- Gyroscope
- Heart rate
- Raw audio
- Location



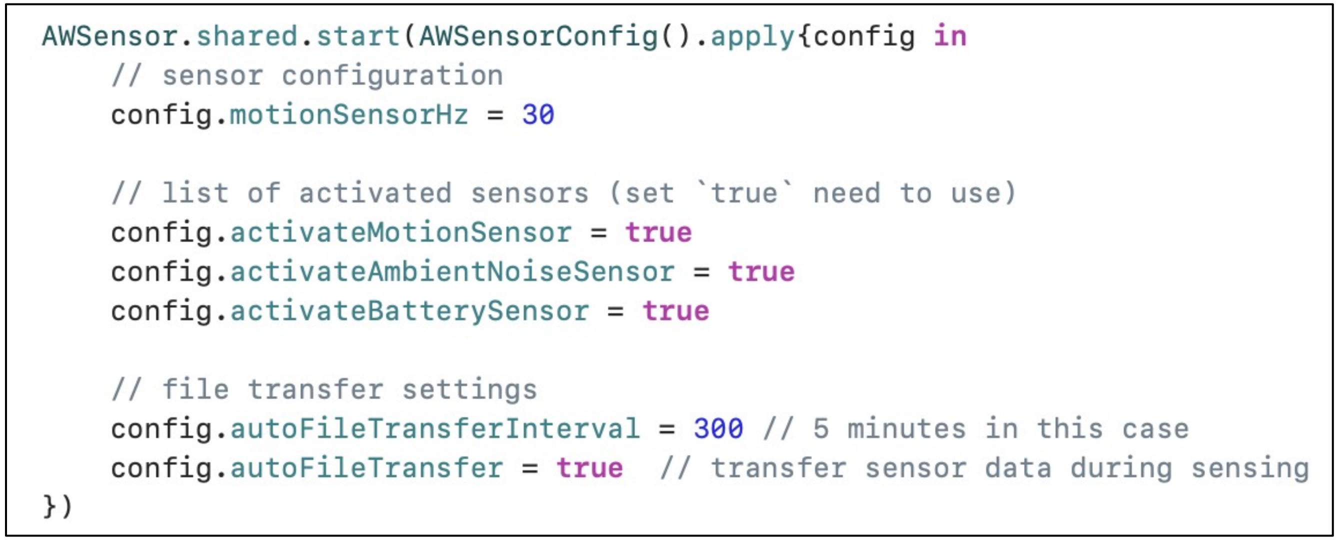
Screenshots of a sample app

[8] Komatsu, Y. et al., "Toward Measuring Conversation Duration Using a Wristwatch-type Wearable Device," 2022 IEEE International Conference on Smart Computing (SMARTCOMP), Helsinki, Finland, 2022, pp. 150-152.

AWARE-watchOS as a library

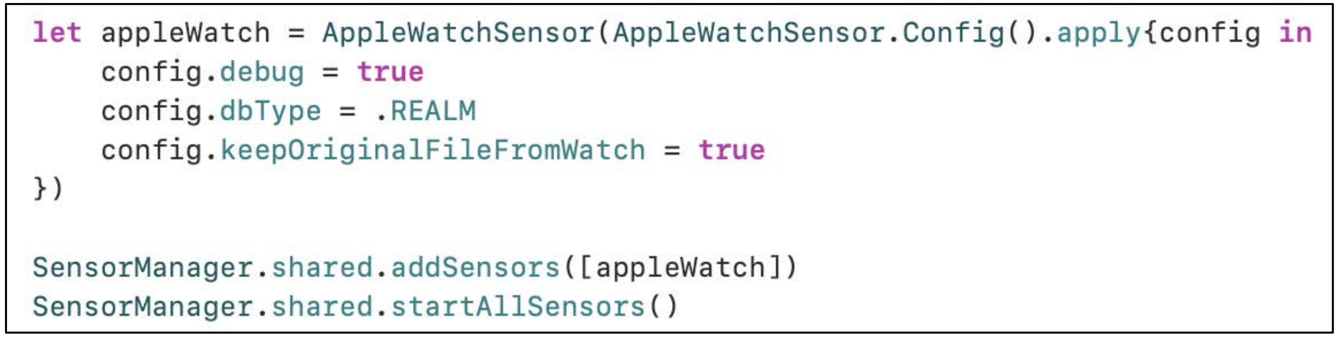
Install this library via **CocoaPods** and **insert the following code** into iOS and watchOS projects.

watchOS側



```
AWSensor.shared.start(AWSensorConfig().apply{config in
    // sensor configuration
    config.motionSensorHz = 30

    // list of activated sensors (set `true` need to use)
    config.activateMotionSensor = true
    config.activateAmbientNoiseSensor = true
    config.activateBatterySensor = true

    // file transfer settings
    config.autoFileTransferInterval = 300 // 5 minutes in this case
    config.autoFileTransfer = true  // transfer sensor data during sensing
})
```

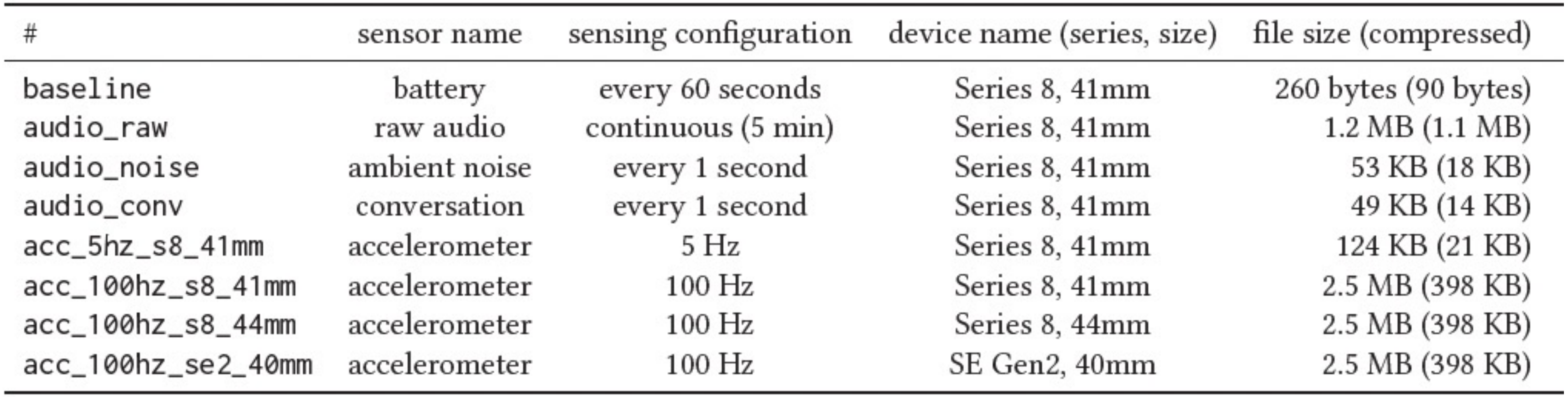
iOS側



```
let appleWatch = AppleWatchSensor(AppleWatchSensor.Config().apply{config in
    config.debug = true
    config.dbType = .REALM
    config.keepOriginalFileFromWatch = true
})

SensorManager.shared.addSensors([appleWatch])
SensorManager.shared.startAllSensors()
```

Performance Evaluation

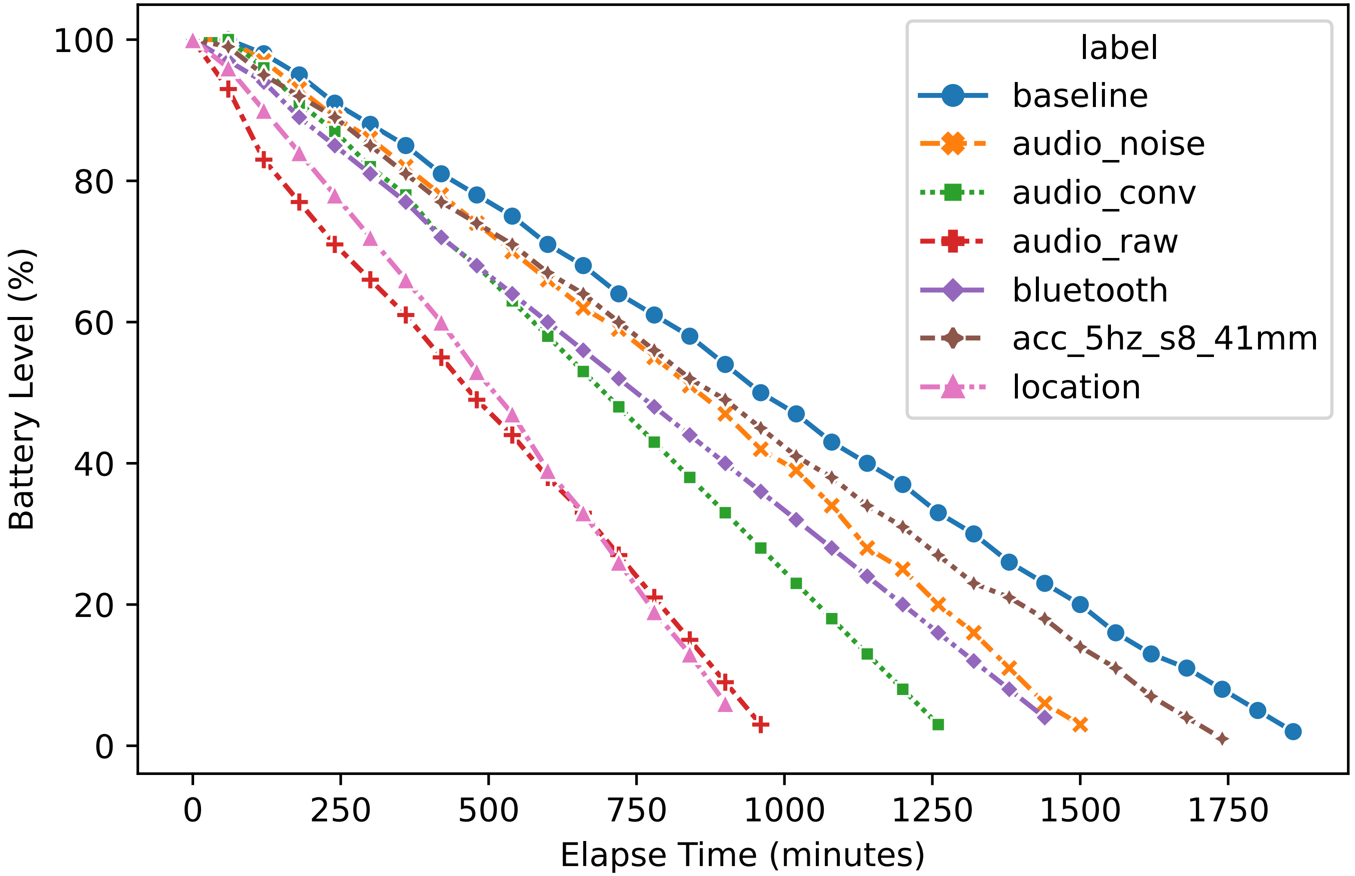
- Controlled battery consumption study
  - Case①: **Sensors**
  - Case②: **Settings**
  - Case③: **Hardware**
  - Case④: **Free-living**

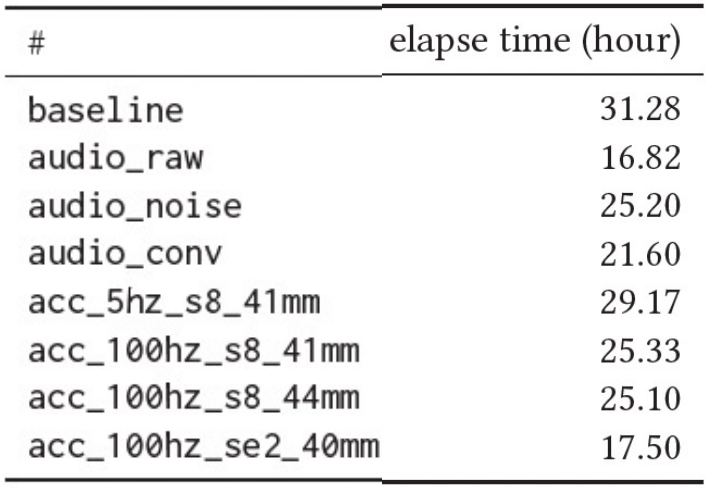


#	sensor name	sensing configuration	device name (series, size)	file size (compressed)
baseline	battery	every 60 seconds	Series 8, 41mm	260 bytes (90 bytes)
audio_raw	raw audio	continuous (5 min)	Series 8, 41mm	1.2 MB (1.1 MB)
audio_noise	ambient noise	every 1 second	Series 8, 41mm	53 KB (18 KB)
audio_conv	conversation	every 1 second	Series 8, 41mm	49 KB (14 KB)
acc_5hz_s8_41mm	accelerometer	5 Hz	Series 8, 41mm	124 KB (21 KB)
acc_100hz_s8_41mm	accelerometer	100 Hz	Series 8, 41mm	2.5 MB (398 KB)
acc_100hz_s8_44mm	accelerometer	100 Hz	Series 8, 44mm	2.5 MB (398 KB)
acc_100hz_se2_40mm	accelerometer	100 Hz	SE Gen2, 40mm	2.5 MB (398 KB)

The list of sensors and their settings

Battery consumption:
①Difference with Sensor Types

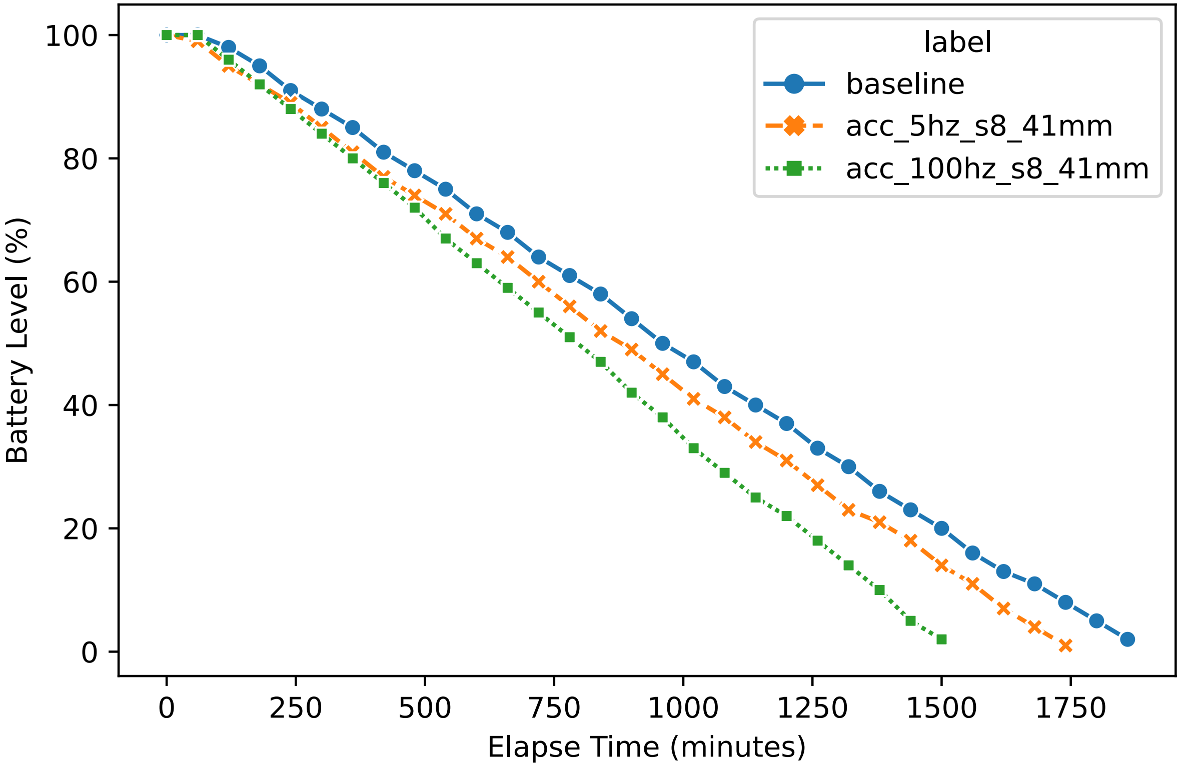




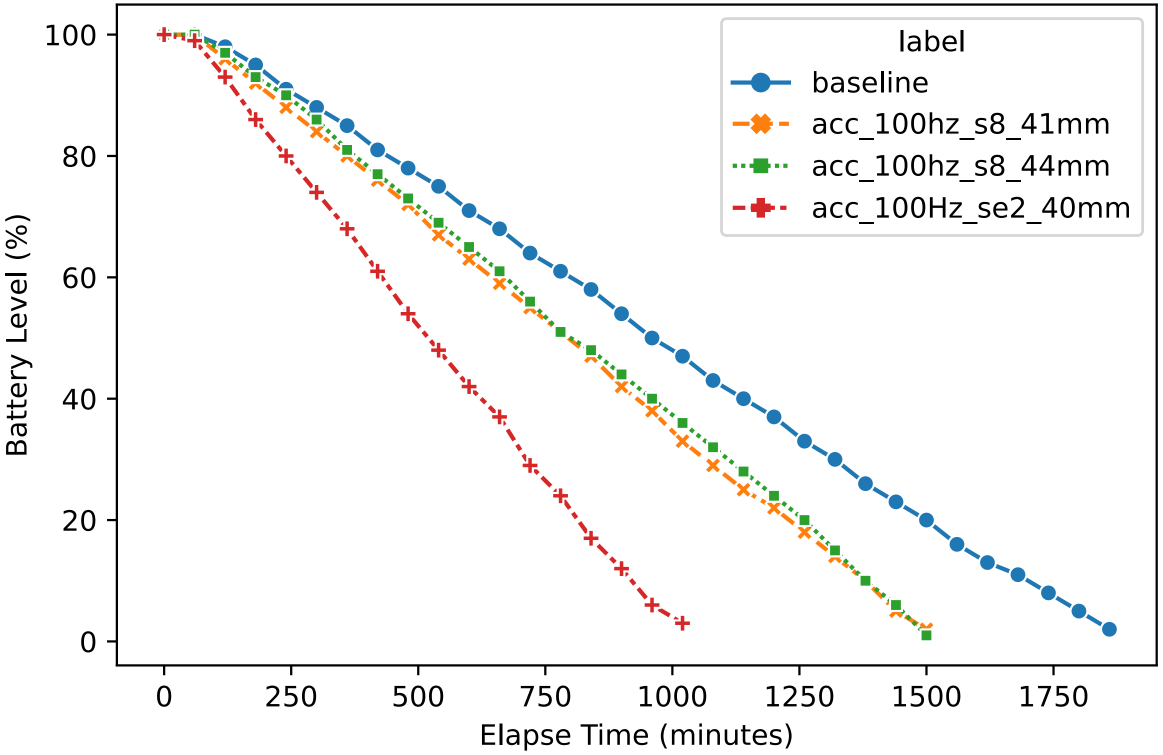
#	elapse time (hour)
baseline	31.28
audio_raw	16.82
audio_noise	25.20
audio_conv	21.60
acc_5hz_s8_41mm	29.17
acc_100hz_s8_41mm	25.33
acc_100hz_s8_44mm	25.10
acc_100hz_se2_40mm	17.50

baseline > acc_5hz > audio_noise > bluetooth > audio_conv > audio_raw > location

Battery consumption:
②③Effects of Sensing Frequency and Hardware

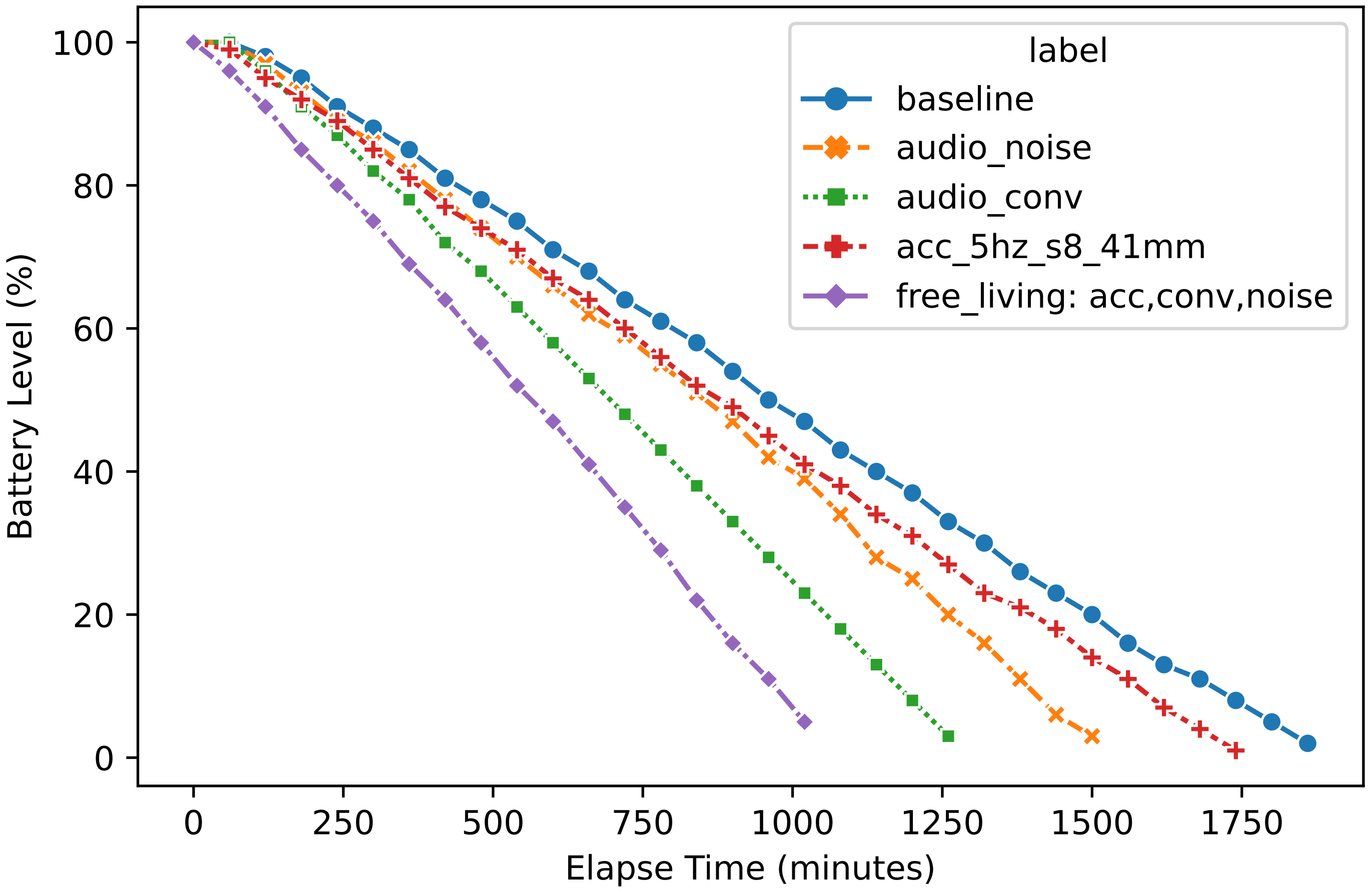


baseline > acc_5hz > acc_100Hz



baseline > s8_41mm & s8_44mm > se2_40mm

Battery consumption:
④Free-living and Multiple Sensors Condition

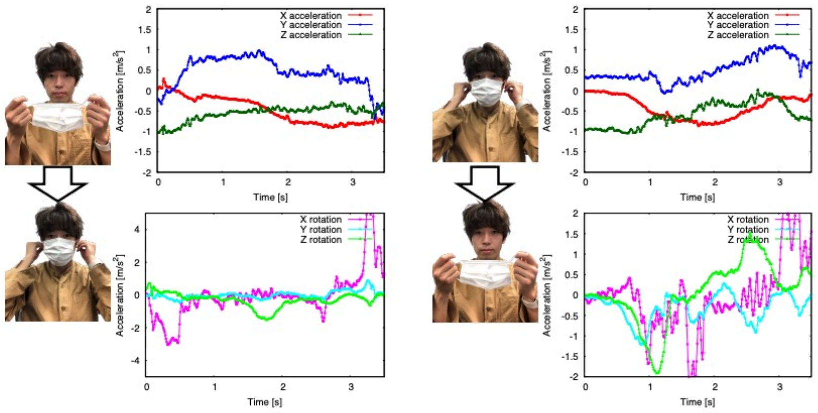


free_living (16h) ≒ (baseline – audio_noise) + (baseline – audio_con) + (baseline – acc_5hz)

Discussion & Limitation

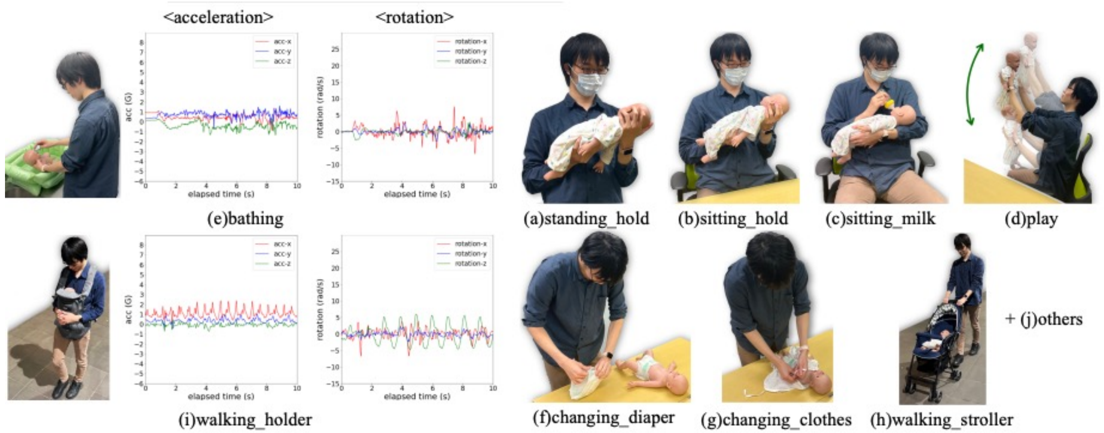
- **Battery consumption and its optimization**
  - Baseline: **31.28h**, Shortest: **16.82h**, Free-living(motion&noise&conversation): **16h**
  - The results of battery consumption are closely related to our hypothesis.
    - **Sensors with a higher sampling frequency** = consume more power
    - **Larger file transfers** = consume more power
  - Duty cycle
- **Application**
  - This framework realizes **seamless integration of passive sensing functionality** on passive sensing studies
  - Also, the framework allows us to record more **granular daily activities** without a smartphone
  - Intervention based on the collected sensor data
- **Limitations**
  - The framework might need more battery in the wild condition than this result.

Applications based on the Proposed Framework

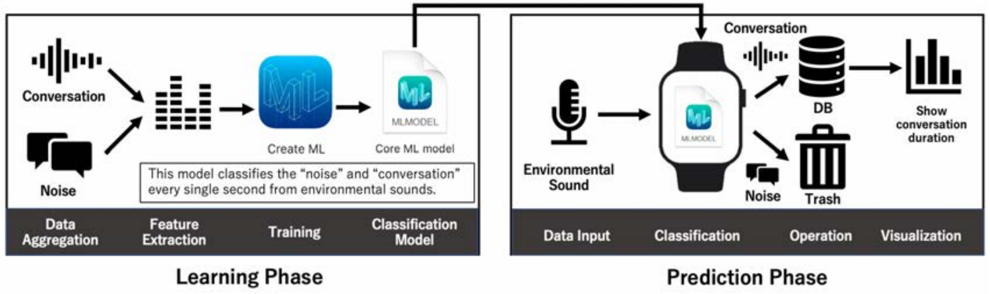


a) Wearing the mask b) Take off the mask

[11] **Face-mask usage detection** by motion sensors



[12] **Child-care activities detection** by motion sensors



Conversation

Noise

Create ML

Core ML model

MLMODEL

This model classifies the "noise" and "conversation" every single second from environmental sounds.

Data Aggregation | Feature Extraction | Training | Classification Model

Learning Phase

Environmental Sound

Conversation

DB

Show conversation duration

Noise

Trash

Data Input | Classification | Operation | Visualization

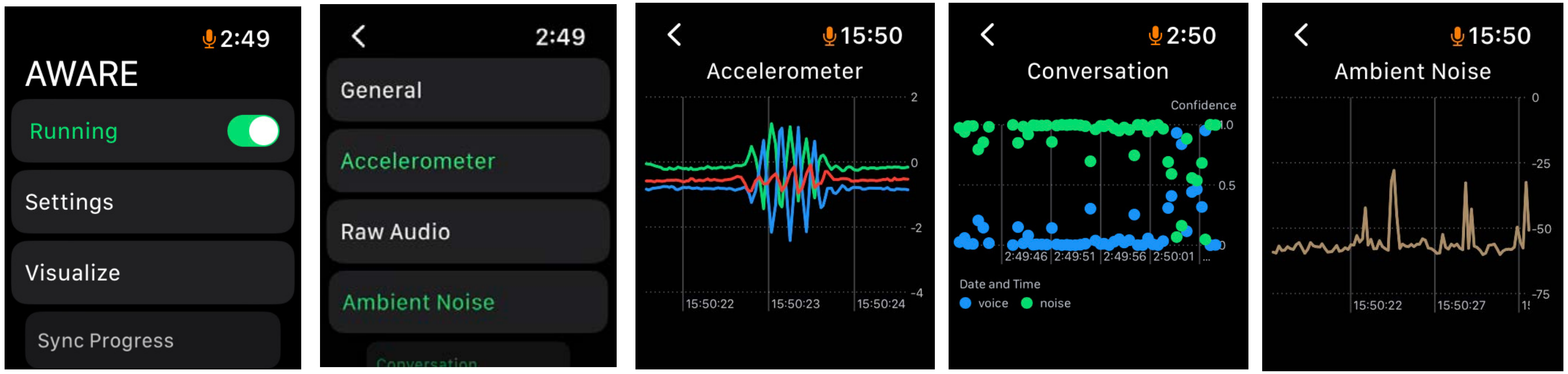
Prediction Phase

[13] **Conversation detection** by microphone

[14] **Disinfection behavior detection** by motion and audio

[11] S. Ono, et al., "Detecting Face-Mask Wearing Status Using Motion Sensors in Commercially Available Smartwatches," 2022 IEEE International Conference on E-health Networking, Application & Services (HealthCom), Genoa, Italy, 2022, pp. 107-112.
[12] Y. Kasahara, et al., "Detecting Childcare Activities Using an Off-the-shelf Smartwatch," 2022 IEEE International Conference on Smart Computing (SMARTCOMP), Helsinki, Finland, 2022, pp. 159-161.
[13] Y. Komatsu et al., "Toward Measuring Conversation Duration Using a Wristwatch-type Wearable Device," 2022 IEEE International Conference on Smart Computing (SMARTCOMP), Helsinki, Finland, 2022, pp. 150-152.
[14] H. Zhuang, et al., "Detecting Hand Hygienic Behaviors In-the-Wild Using a Microphone and Motion Sensor on a Smartwatch" In: Streitz, Norbert A.; Konomi, Shiníchi (Ed.): Distributed, Ambient and Pervasive Interactions, pp. 470–483, Springer Nature Switzerland, Cham, 2023.

Conclusion



Background: **The market for smartwatches has significantly expanded** and has also been **utilized in various research projects**.

Issue: **Smartwatches have resource limitations**, necessitating specific design considerations for **long-term passive sensing**.

Approach: We **designed and implemented a framework for smartwatch-based passive sensing** that supports **eight different sensors,** and **background sensing, automatic file transfer functions.**

Result: Our battery evaluations in **four scenarios** show the range of data collection time was between **16-31 hours**, depending on the settings. We are currently preparing to **open-source** the proposed framework.

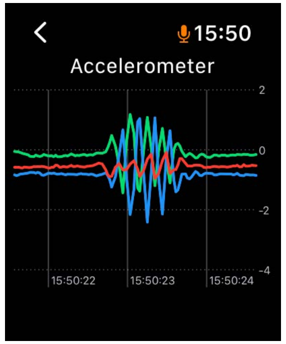
Thank you.

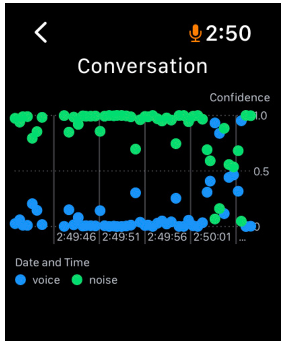
Please let me know if you are interested in using this framework!
I’m happy to share the source code and collaborate with you.

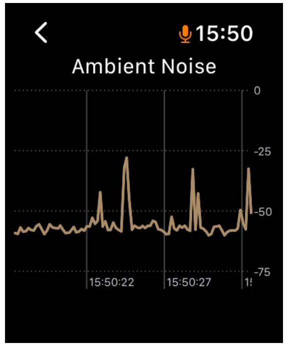
Name: **Yuuki NISHIYAMA**

Affiliation: **CSIS, The University of Tokyo**

Contact: nishiyama@csis.u-tokyo.ac.jp
Home page: https://www.yuukinishiyama.com

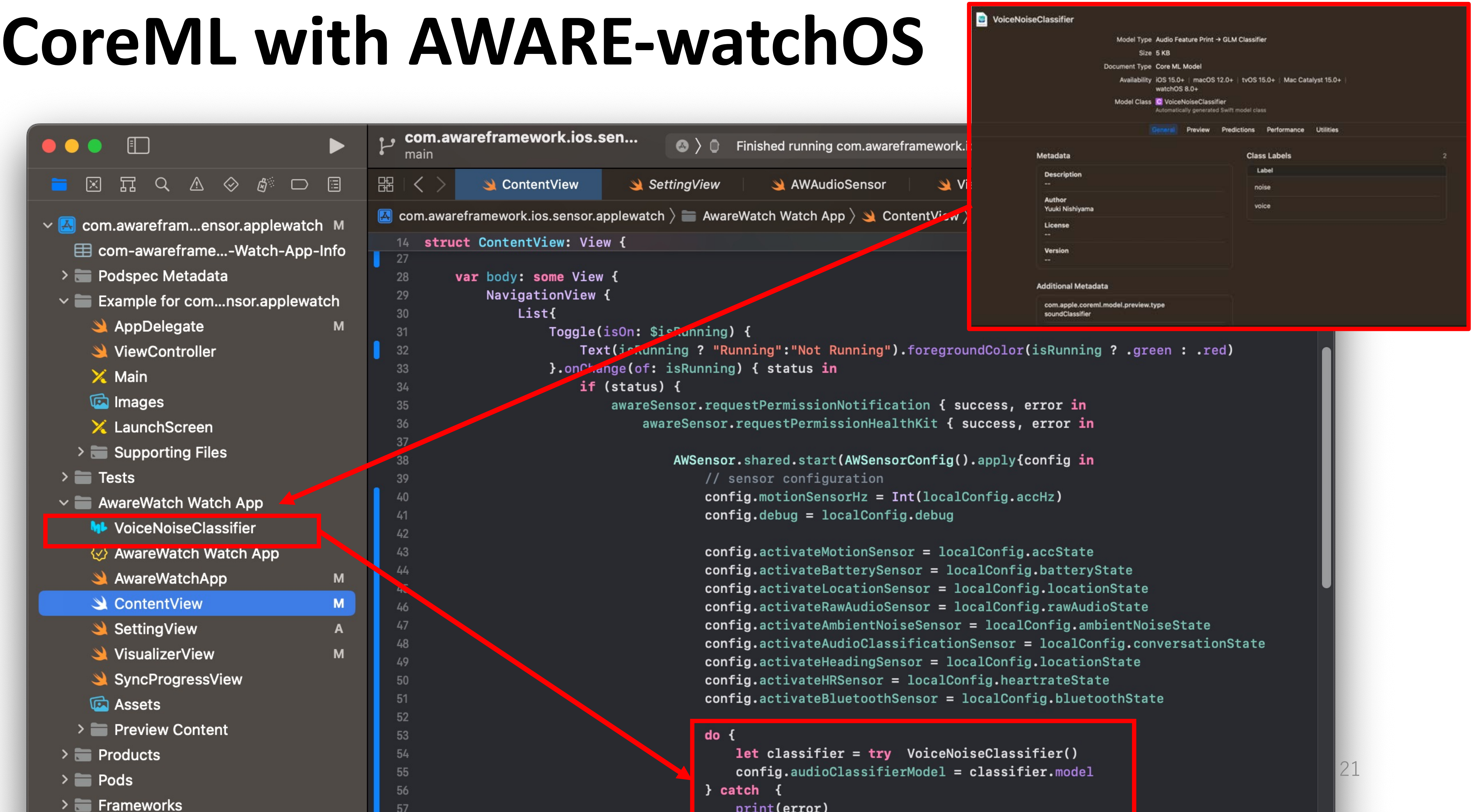






Create a classification model by **CreateML** or **PyTorch**

CoreML with AWARE-watchOS



VoiceNoiseClassifier

Model Type Audio Feature Print → GLM Classifier
Size 5 KB
Document Type Core ML Model
Availability iOS 15.0+ | macOS 12.0+ | tvOS 15.0+ | Mac Catalyst 15.0+ | watchOS 8.0+
Model Class VoiceNoiseClassifier
Automatically generated Swift model class

Metadata

Description
--

Author
Yuuki Nishiyama

License
--

Version
--

Additional Metadata

com.apple.coreml.model.preview.type
soundClassifier

Class Labels 2

Label
noise
voice

com.awareframework.ios.sen...
main

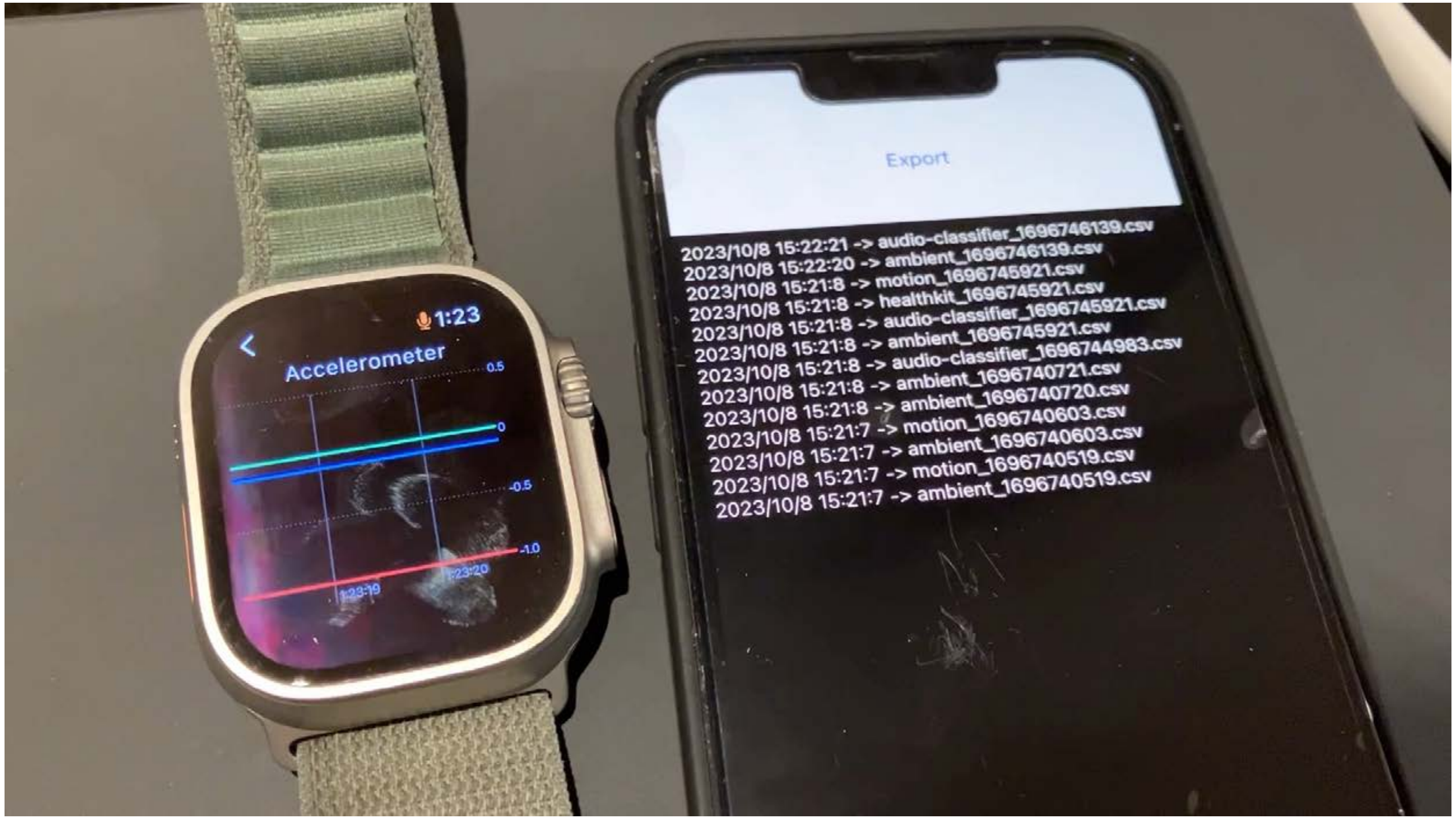
Finished running com.awareframework.i

com.awareframework.ios.sensor.applewatch 〉 AwareWatch Watch App 〉 ContentView 〉

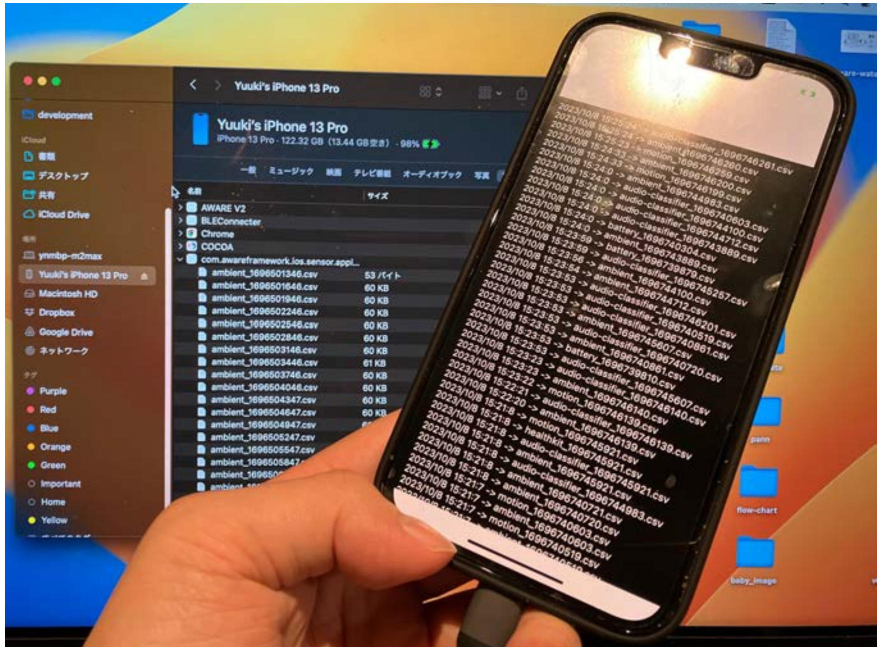
- com.awarefram...ensor.applewatch M
  - com-awareframe...-Watch-App-Info
  - Podspec Metadata
  - Example for com...nsor.applewatch
    - AppDelegate M
    - ViewController
    - Main
    - Images
    - LaunchScreen
    - Supporting Files
  - Tests
  - AwareWatch Watch App
    - VoiceNoiseClassifier
    - AwareWatch Watch App
    - AwareWatchApp M
    - ContentView M
    - SettingView A
    - VisualizerView M
    - SyncProgressView
    - Assets
    - Preview Content
  - Products
  - Pods
  - Frameworks

```swift
struct ContentView: View {

    var body: some View {
        NavigationView {
            List{
                Toggle(isOn: $isRunning) {
                    Text(isRunning ? "Running":"Not Running").foregroundColor(isRunning ? .green : .red)
                }.onChange(of: isRunning) { status in
                    if (status) {
                        awareSensor.requestPermissionNotification { success, error in
                            awareSensor.requestPermissionHealthKit { success, error in

                                AWSensor.shared.start(AWSensorConfig().apply{config in
                                    // sensor configuration
                                    config.motionSensorHz = Int(localConfig.accHz)
                                    config.debug = localConfig.debug

                                    config.activateMotionSensor = localConfig.accState
                                    config.activateBatterySensor = localConfig.batteryState
                                    config.activateLocationSensor = localConfig.locationState
                                    config.activateRawAudioSensor = localConfig.rawAudioState
                                    config.activateAmbientNoiseSensor = localConfig.ambientNoiseState
                                    config.activateAudioClassificationSensor = localConfig.conversationState
                                    config.activateHeadingSensor = localConfig.locationState
                                    config.activateHRSensor = localConfig.heartrateState
                                    config.activateBluetoothSensor = localConfig.bluetoothState

                                    do {
                                        let classifier = try  VoiceNoiseClassifier()
                                        config.audioClassifierModel = classifier.model
                                    } catch  {
                                        print(error)
```

How to extract the collected data?



```
2023/10/8 15:22:21 -> audio-classifier_1696746139.csv
2023/10/8 15:22:20 -> ambient_1696746139.csv
2023/10/8 15:21:8 -> motion_1696745921.csv
2023/10/8 15:21:8 -> healthkit_1696745921.csv
2023/10/8 15:21:8 -> audio-classifier_1696745921.csv
2023/10/8 15:21:8 -> ambient_1696745921.csv
2023/10/8 15:21:8 -> audio-classifier_1696744983.csv
2023/10/8 15:21:8 -> ambient_1696740721.csv
2023/10/8 15:21:8 -> ambient_1696740720.csv
2023/10/8 15:21:7 -> motion_1696740603.csv
2023/10/8 15:21:7 -> ambient_1696740603.csv
2023/10/8 15:21:7 -> motion_1696740519.csv
2023/10/8 15:21:7 -> ambient_1696740519.csv
```



Siri? No problem