Unlocking value in big data in IoT

Dr. Zubair Fadlullah
Department of Computer Science
Western University

Layout

Preliminaries
- IoT
- AI : data-driven models
- Cyber-Physical Systems and Networks
- Edge Computing
- Big Data

From unstructured data to Edge AI
- Big data
- Data classification
- Data Cascades in high-stakes AI
- Collecting data in a volatile world
- Domain expertise and data understanding
- Data Analysis

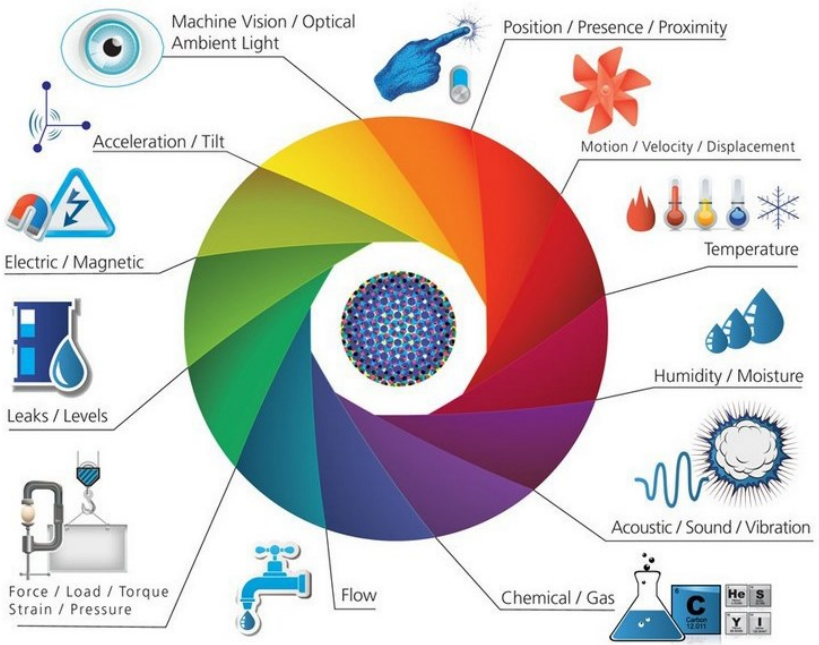
Smart IoT sensor-based bio-magnetic activity detection
- Problem statement
- Data collection
- AI-based noise processing
- Federated learning

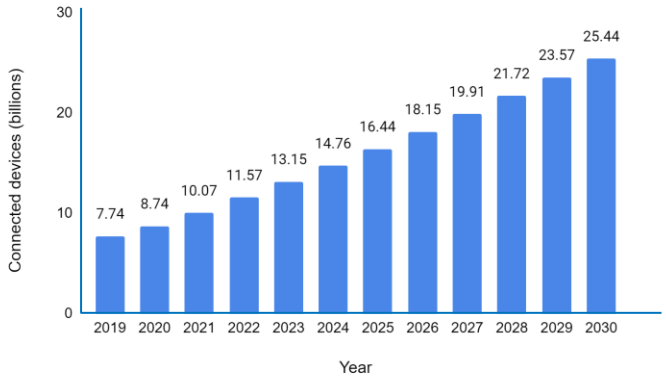
Wearable-based Human Activity Recognition
- Problem statement
- Data collection
- Data structuring
- Data processing
- Data analysis
- Federated Learning

WiFi-based Human Authentication
- Problem statement
- WiFi sensing
- Public dataset
- Data processing
- Central and local data analysis
- Federated Learning with knowledge distillation

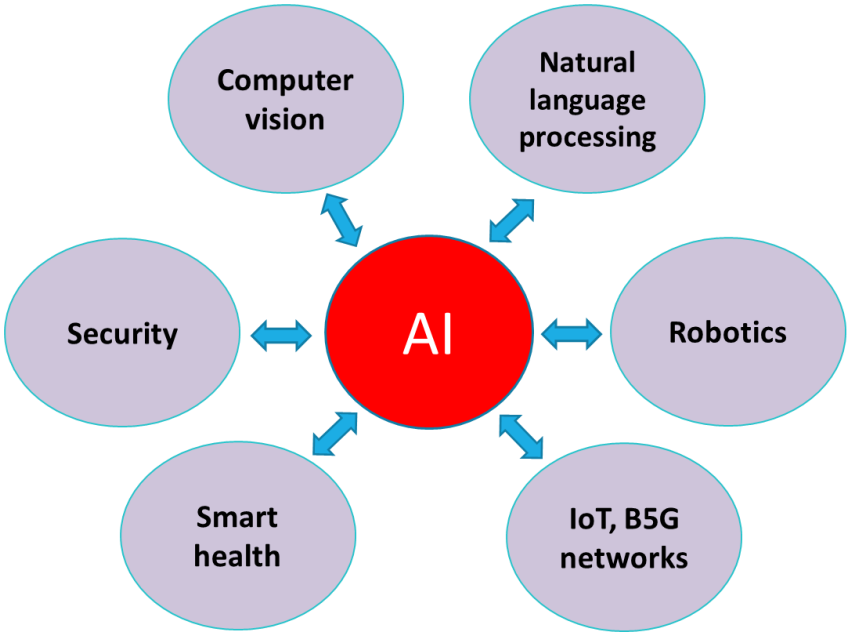
Preliminary Terms

Internet of Things (IoT)





Artificial Intelligence (AI)

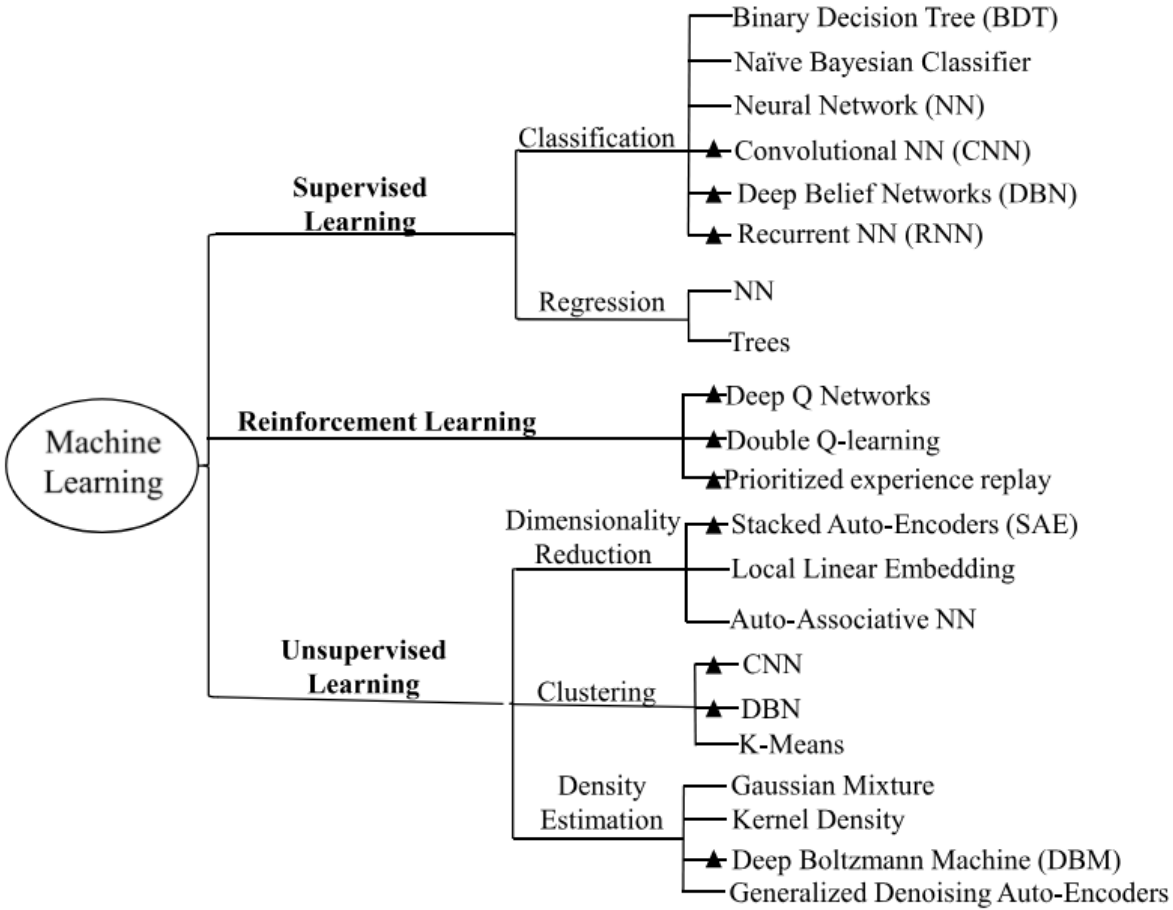


Availability of large datasets

Radical computational power

Leap in deep neural networks

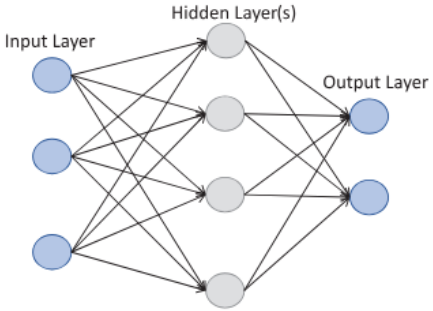
Machine Learning Taxonomy



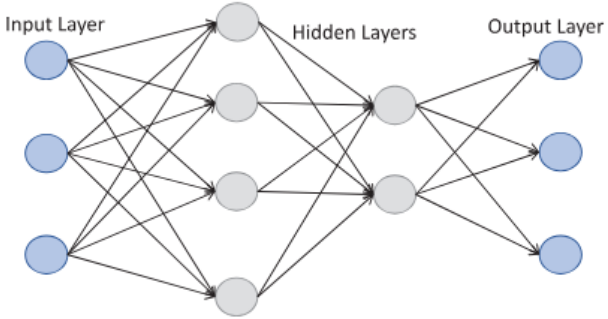
Applications

Image classification; Character recognition; Facial recognition; Surveillance systems (intrusion detection, etc); Diagnostics
Advertising and business intelligence (Google ads, etc); Weather forecasting; Market forecasting; Political campaigns
Real-time decisions; Game Artificial Intelligence (AI); Learning tasks; Skill acquisition; Personal assistants (Google Now, Microsoft Cortana, Apple Siri, etc); Autonomous ("self-driving") cars
Big data visualization; Feature elicitation; Structure discovery; Meaningful compression
Recommendation engines (Amazon web service, Netflix, etc); Customer segmentation; Targeted marketing; Filtering algorithms; Newsfeeds
Economics (risk prediction, etc)

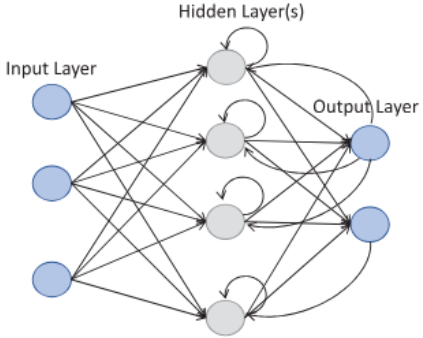
Deep Learning architectures



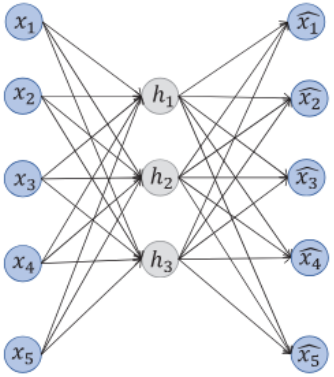
(a) Shallow ANN architecture.



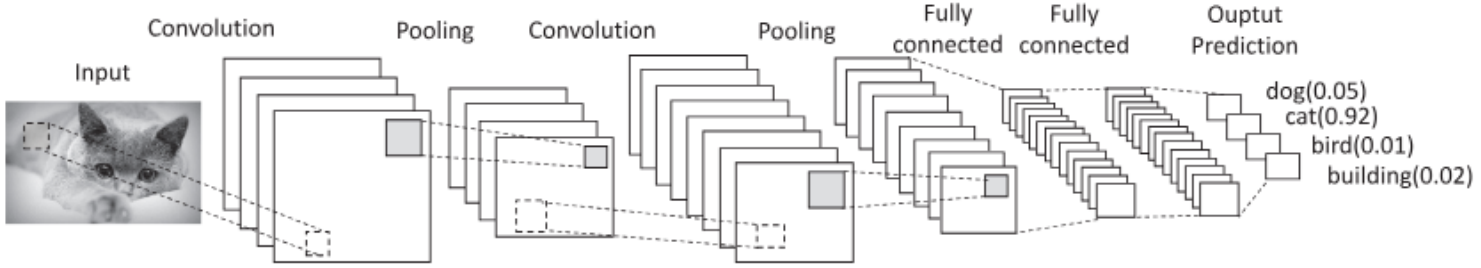
(b) Deep ANN architecture.



(d) RNN architecture.

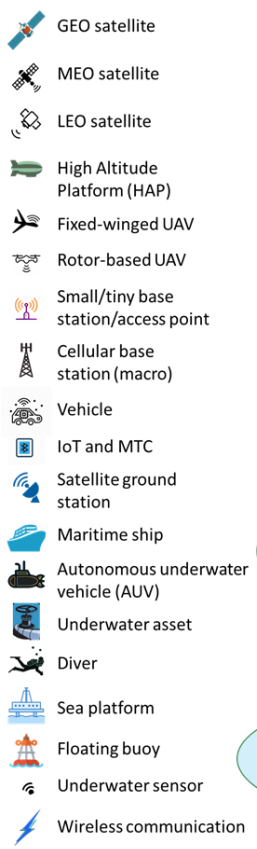


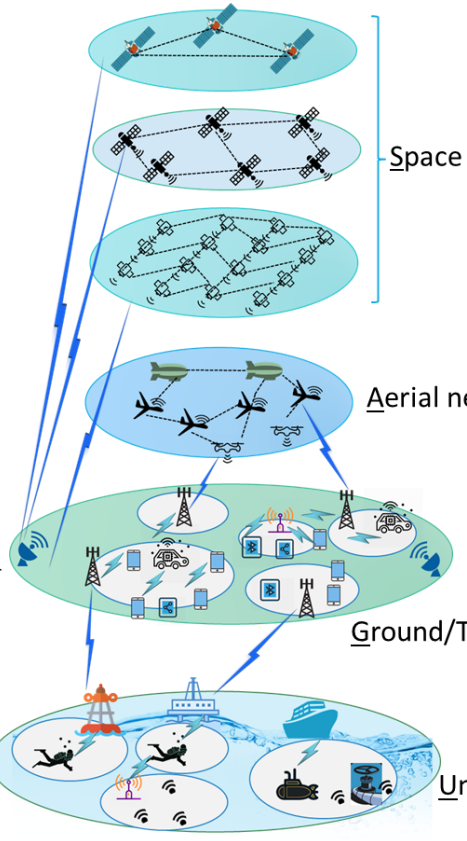
(e) Auto-Encoder.



(c) CNN architecture.

Emerging B5G and 6G Networks

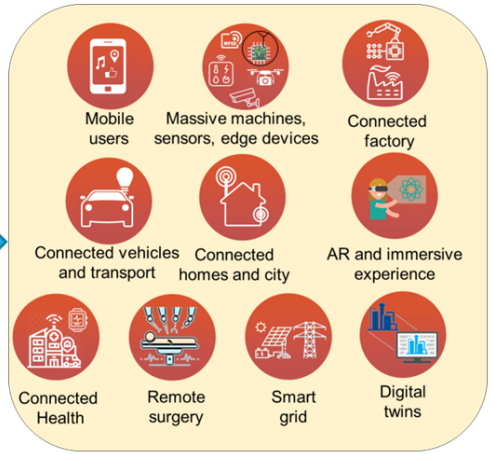




SAGUIN Integrated Networks

Challenge

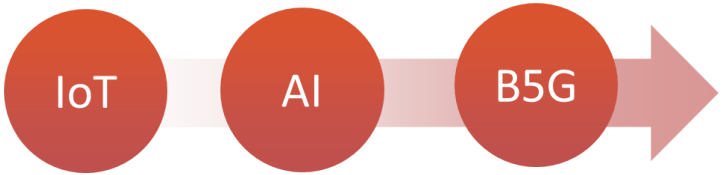
Different Radio Access Technologies (RATs) with different inherent properties



Merit

Provide ubiquitous connectivity supporting emerging applications

Ubiquitous connectivity in smart communities



IoT and communication networks generate big data!

Big data

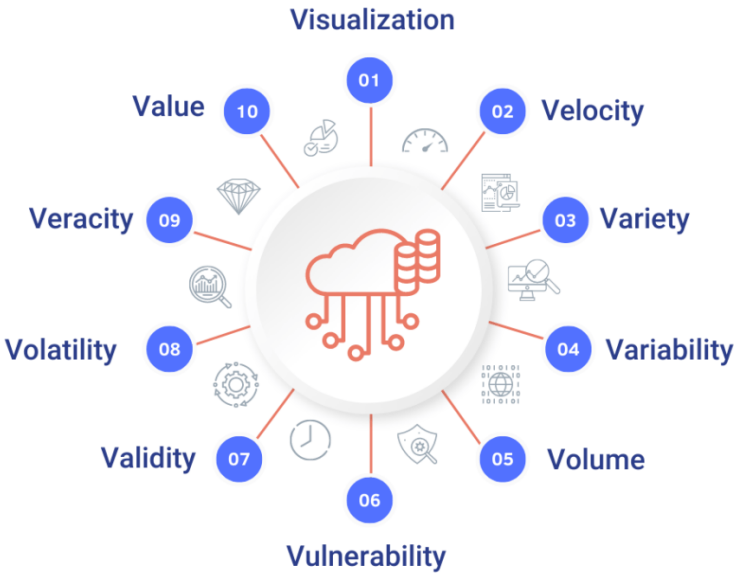
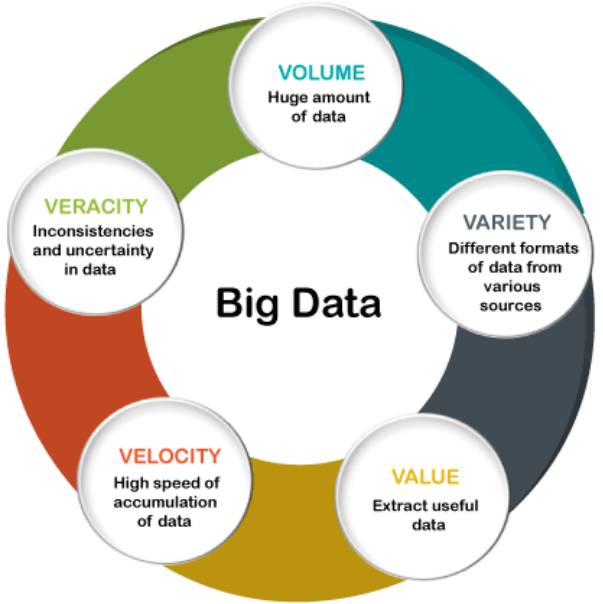


Figure source: https://datasciencedojo.com/blog/10-vs-of-big-data/

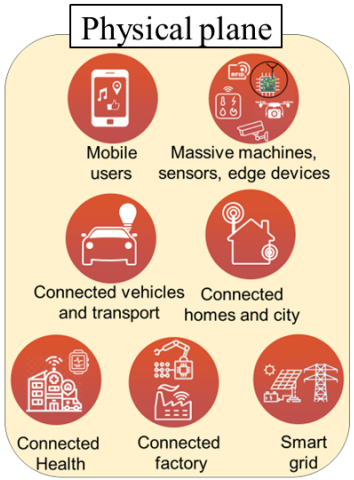


https://www.javatpoint.com/big-data-characteristics

Cyber-Physical System scope and impact of Big Data

CPS = Connected Embedded Systems

IoT + Communication network + Computing for control processing



Network effects

HW, microprocessor, OS effects

OS and software system effects

Cyber plane

Cloud

AI, optimization, analytics

Inaccurate modeling of real-world processes, virtual prototyping, digital twins, etc

Presence of big, unstructured data in each layer is inevitable → how to handle?

CPS: Cyber Physical System IoT: Internet of Things HW: hardware OS: Operating System AI: Artificial Intelligence

The 3C's

My research program on CPS consists of three research planes: the 3C's

Network edge

Physical plane

Cyber plane

Cloud

Quality signal acquisition

Communication Requirements

Operation Requirements

Security Requirements

Privacy Requirements

C1. Smart IoT sensor conceptualization

C2. Optimizing CPS network

C3. Privacy-preserving Edge Computing

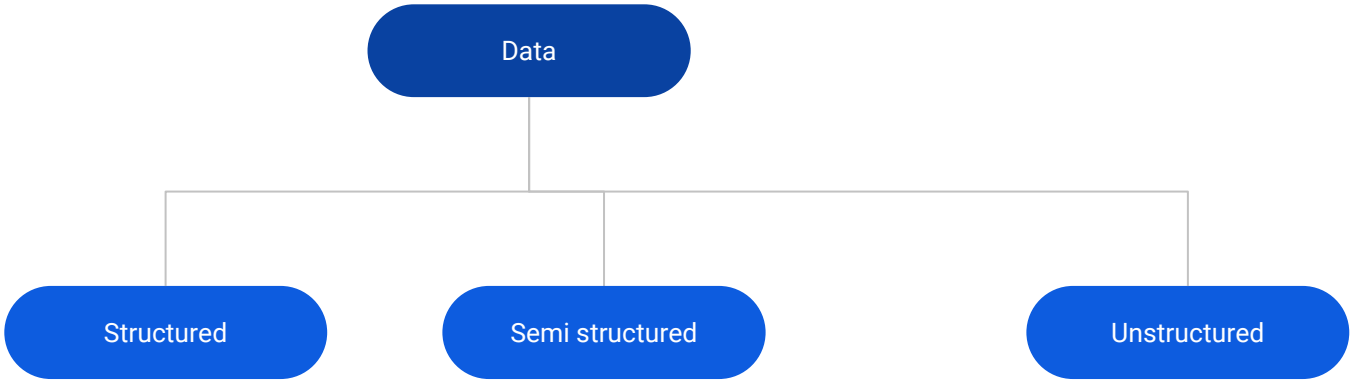
From unstructured data to edge AI

Big data

- Big data refers to the creation, collection, and utilization of data to create value in businesses.
- It is not the amount of data that matters, but what businesses do (how to create value) with the data that counts.
- Businesses leverage big data to propel growth

Data Classification

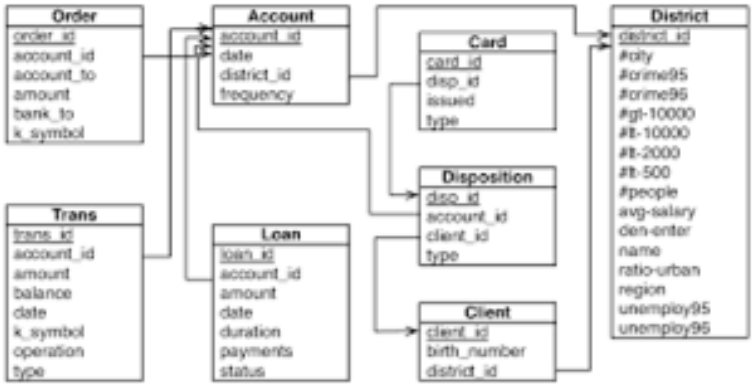
- In the domain of big data, it is important to study the classifications of data to be able to understand and utilize the tools that make managing and analyzing these classes of data possible.
- Data can be broadly classified into three categories.



Structured data

Definition

- Structured data is usually contained in rows and columns and its elements can be mapped into fixed predefined fields.
- Structured data data is the easiest to search through and organize.
- Structured data follow a schema (a data model).



Structured data

Frameworks

- Structured data are often managed using Structured Query Language (SQL).

Examples

- Structured data can be created by humans or machines. Examples of structured data include financial data like accounting transactions, address details, demographic information, customer reviews, machine logs.

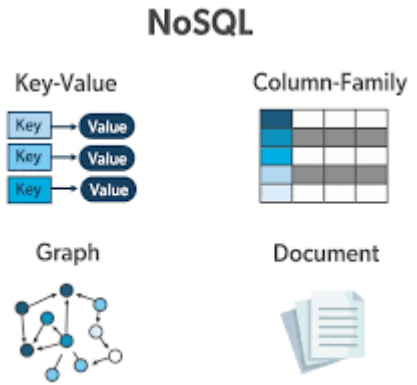
Unstructured data

Definition

- Data that is not contained in a row-column database and does not have an associated data model.
- The lack of structure makes unstructured data difficult to organize and analyze.
- Companies often discard unstructured data. However, Artificial Intelligence (AI) and machine learning (ML) made it easier to manage unstructured data.

Frameworks

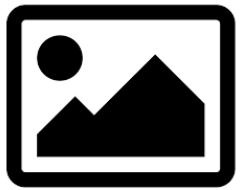
- AI and ML training and inference frameworks are often utilized to make sense of unstructured data.
- Instead of relational databases and spreadsheets, unstructured data are often stored in data lakes, NoSQL databases, and data warehouses.



Unstructured data

Examples

- Examples of unstructured data include photos, videos, audio files, text files, social media content, presentations, PDFs, and scraped website content.



Semi-structured data

Definition

- Data that has some defining or consistent characteristics but does not conform to a structure as rigid as is expected with a relational database.
- Semi-structured data has some organizational properties such as semantic tags or metadata to make it possible to search and manage, but there is still fluidity in the data.

Frameworks

- Depending on the task, a combination of the tools for structured and unstructured data can be used.

Examples

- An example of Semi-structured data is emails. While the email body is unstructured, it contains structured data such as the name and email addresses of the sender and the recipient(s), the time stamp of the email, etc.

Job interview - Data class analogy

To understand the difference between different data classifications, we will use job interview as an analogy.

Structured Interview <-> Structured data

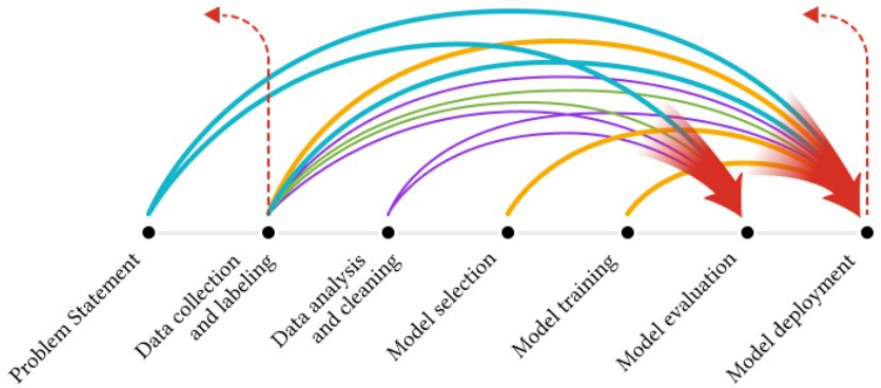
- In a structured interview, the interviewer follows a strict script defined by the HR department for all candidates.
- Likewise, the types and shape of structured data follows a predefined schema.

Unstructured Interview <-> Unstructured data

- In an unstructured interview, the interviewer does not follow a script. Instead, it is up to the interviewer, which questions to ask and to which depth based on the scenario.
- Likewise, the format of unstructured data does not follow a data model and can change with different scenarios.

Data cascades in high-stakes AI

- Data quality is central to AI development.
- Issues in earlier stages of AI development associated with data collection and data handling have great negative impact on the later phases of the AI pipeline.



Data issues impacts model performance

Collecting data in a volatile world

In big data, data and query representation, and indexing models methods are crucial for efficient data management.

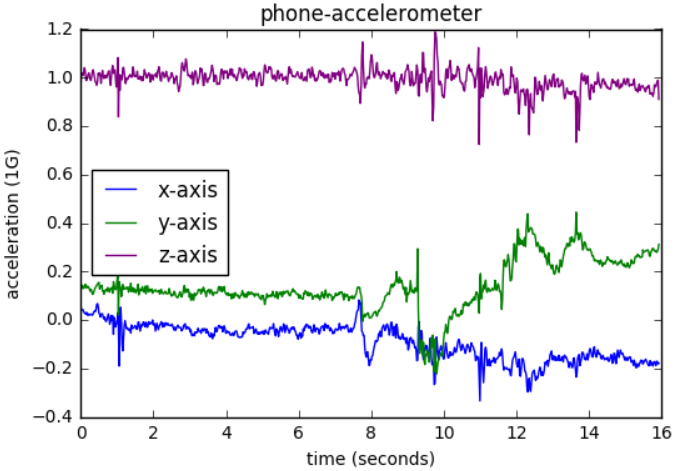
But what do we mean by these terms?

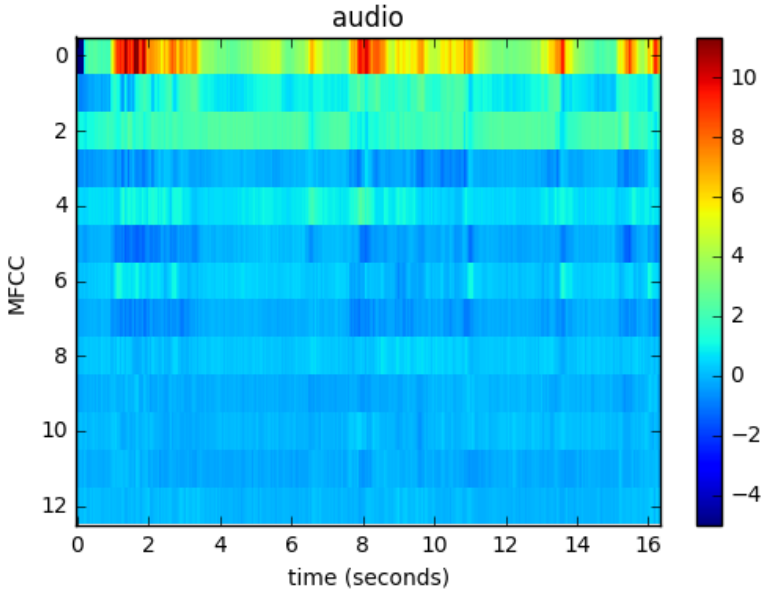
In the context of IoT,

- **Data representation** refers to data structure and encoding for storage, retrieval, and analysis.
- **Query representation** refers to the users' queries formulation to retrieve specific information from data.
- **Indexing is** the process of organizing data to allow for efficient retrieval of data based on certain criteria.

Due to the dynamic nature of data, data representation structures and indexing methods need to be updated as the data format changes.

Example of sensor data

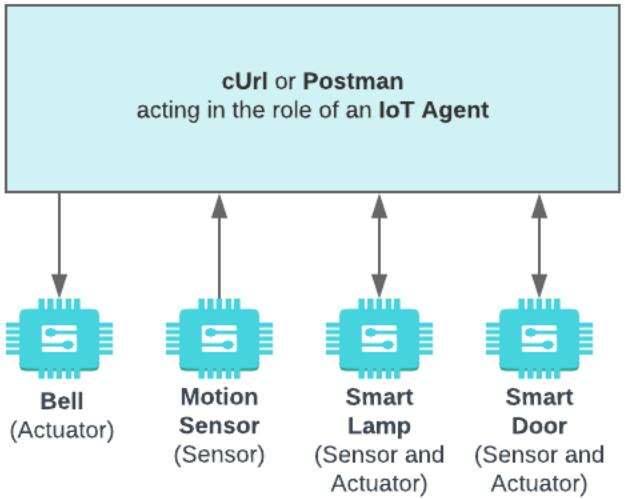




Raw-measurements recorded from various sensors during the 20-second window.

http://extrasensory.ucsd.edu/

Example of sensor data



Commands are going Southbound on Port 3001

Measurements are going Northbound to the IoT Agent listening on Port 7896

Devices within Store **urn:ngsi-ld:Store:001**

- **door001** s|LOCKED
- **bell001** s|OFF
- **motion001** c|4
- **lamp001** s|ON|l|1000

Devices within Store **urn:ngsi-ld:Store:003**

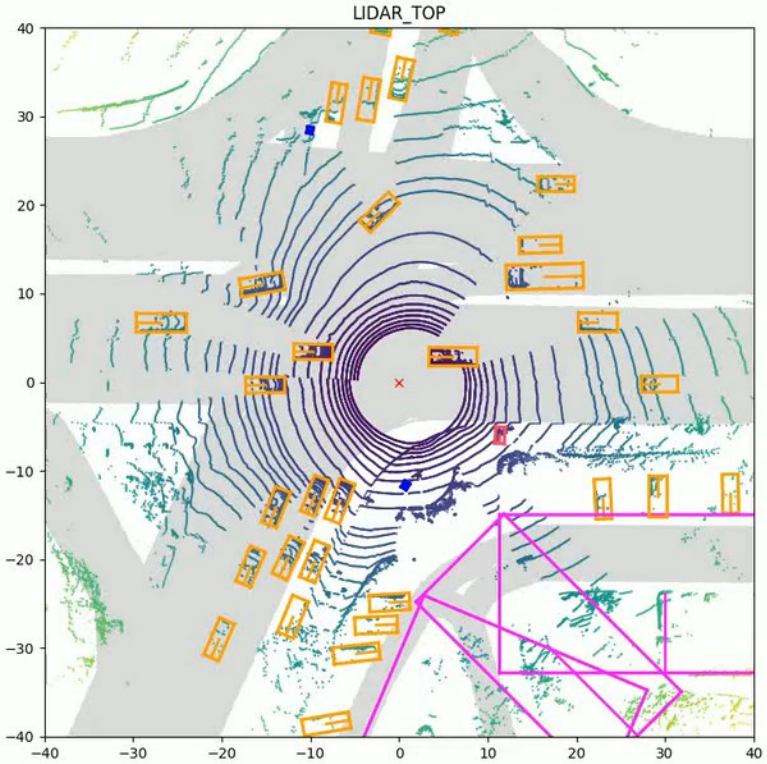
- **door003** s|LOCKED
- **bell003** s|OFF
- **motion003** c|0
- **lamp003** s|OFF|l|0

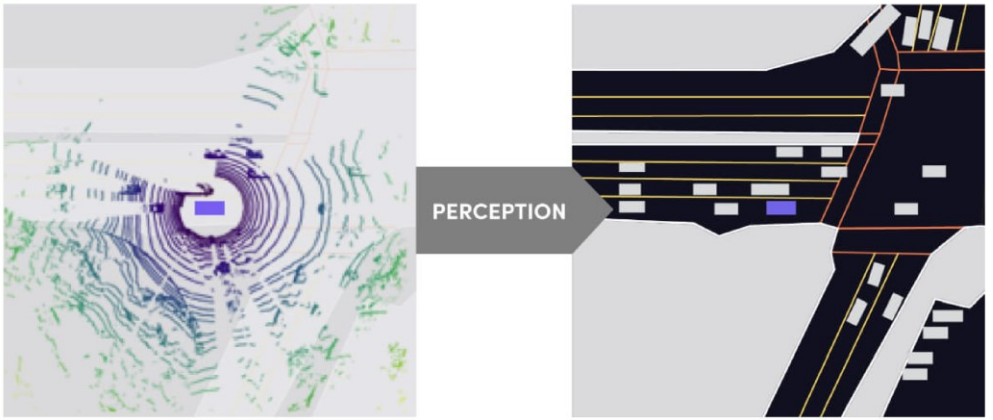
Northbound Traffic

- 2:09:09 PM - /iot/d?i=lamp001&k=1234&d=s|ON|l|1400
- 2:09:12 PM - /iot/d?i=lamp001&k=1234&d=s|ON|l|1300
- 2:09:15 PM - /iot/d?i=lamp001&k=1234&d=s|ON|l|1200
- 2:09:18 PM - /iot/d?i=lamp001&k=1234&d=s|ON|l|1100
- 2:09:21 PM - /iot/d?i=lamp001&k=1234&d=s|ON|l|1000

https://github.com/FIWARE/tutorials.NGSI-v2/blob/master/docs/iot-sensors.md

Example of sensor data



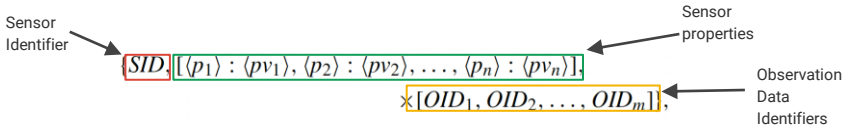


https://woven.toyota/en/perception-dataset/

Representation models

Sensor model

A sensor model allows observation data to be registered from a sensor. Mingliu et al [4] proposed a sensor model with a three-tuple format, including the identifier, sensor-related data, and sensor data.



Observation model

The observation data is represented as a five-tuple model $\{OID, Location, Value, Timestamp, Duration\}$

Heterogeneity-enabled sensor model

Air Quality Egg

Image from https://airqualityegg.com/egg

Sensor model

```
{
  "SID": 111,
  "Property": { "url": "https://airqualityegg.wickeddevice.com/portal/", "Measurement": "Temperature",
                "Accuracy": "0.2°C", "Frequency": "5s", "Mobility": "static", "Location": "Home" },
  "Observations": [111_1, 111_2, 111_3, ...]
}
```

```
{
  "SID": 112,
  "Property": {"url": "https://airqualityegg.wickeddevice.com/portal/", "Measurement": "Humidity",
               "Accuracy": "1.8²%", "Frequency": "5s", "Mobility": "static", "Location": "Home" },
  " Observations ": [112_1, 112_2, 112_3, ... ]
}
```

```
{
  "SID": 113,
  "Property": {"url": "https://airqualityegg.wickeddevice.com/portal/", "Measurement": "CO2",
               "Frequency": "5s", "Mobility": "static", "Location": "Home" },
  " Observations ": [ 113_1, 113_2, 113_3,...]
}
```

Observation model

OID	Location	Value	Timestamp	Duration
111_1	$76.51°W, 42.43°N$	$14.7°C$	2018-01-21 14:58:38	182s
111_2	$76.51°W, 42.43°N$	$14.6°C$	2018-01-21 15:01:40	183s
112_1	$76.51°W, 42.43°N$	35%	2018-01-21 14:58:38	3813s
112_2	$76.51°W, 42.43°N$	36%	2018-01-21 16:02:11	1953s
113_1	$76.51°W, 42.43°N$	$701ppm$[1]	2018-01-21 14:58:38	180s
113_2	$76.51°W, 42.43°N$	$686ppm$	2018-01-21 15:01:38	122s
...	...	...	...	...

[1] ppm = parts per million.

Query model

To remotely inspect and control things, we need a query model that can retrieve data for further analysis.

According to different search conditions, queries can be divided into two categories:

1. Search for property value or observation data based on specific identifiers.
    a. Ex. asking for health condition or heart rate of a specific patient name?
2. Search for property value or observation data, given a series of constraints in sensor property values or observation data.
    a. Ex. what is the current weather in New York?

Query model

Mingliu et al. [4] proposes to convert all queries into a combination of four basic query models.
These four basic query models are:

1. Searching for a property value based on the sensor identifier and the designated property name.

$$[SID]\&\&[qp_1, qp_2, \ldots, qp_n] \Rightarrow [pv_1, pv_2, \ldots, pv_n]$$

2. Searching for the observation data based on the sensor identifier or the observation identifier.

$$[SID]|||[OID_1, OID_2, \ldots, OID_n] \Rightarrow [o_1, o_2, \ldots, o_n]$$

3. Search for sensors that satisfy the properties' constraints in a range query.

$$[qp_1 : (qpv_{1l}, qpv_{1h}), qp_2 : (qpv_{2l}, qpv_{2h}), \ldots, qp_n : (qpv_{nl}, qpv_{nh})] \Rightarrow [SID_1, SID_2, \ldots, SID_m]$$

4. Search for observation identifiers that match the queried interval in location, time, or data value.

$$[qol_1, qol_2, \ldots, qol_m]|||[qot_1, qot_2, \ldots, qot_n] ||[qov_1, qov_2, \ldots, qov_p] \Rightarrow [OID_1, OID_2, \ldots, OID_q]$$

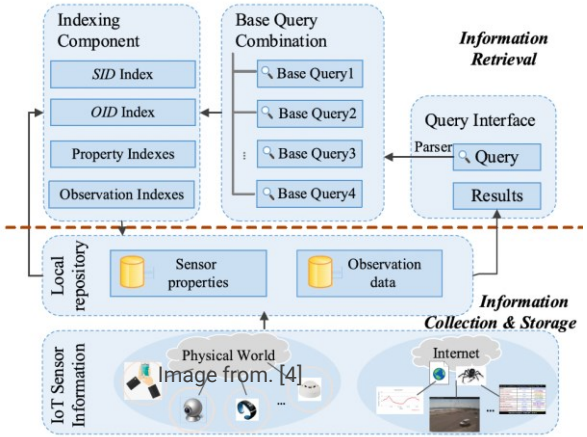
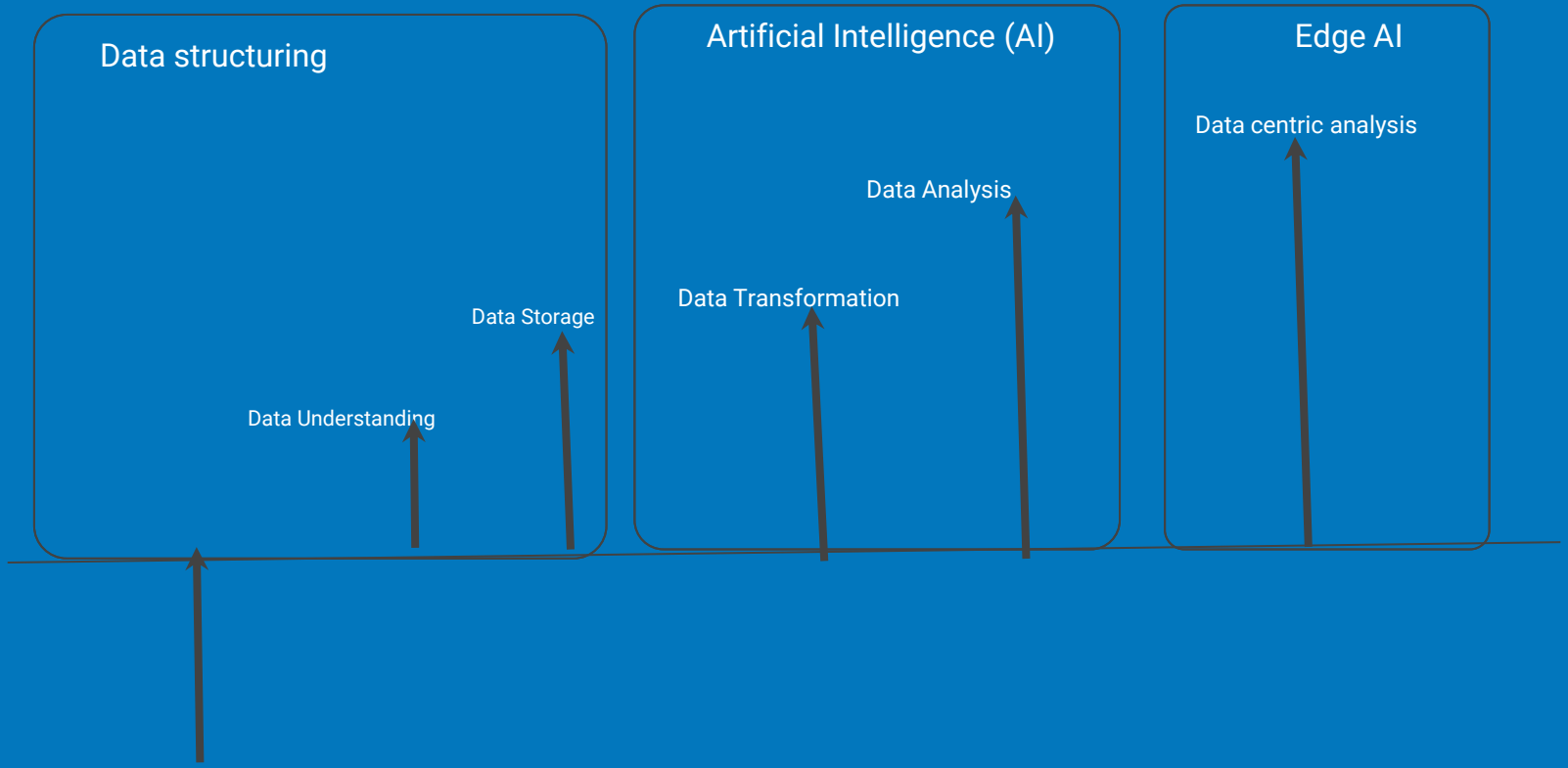
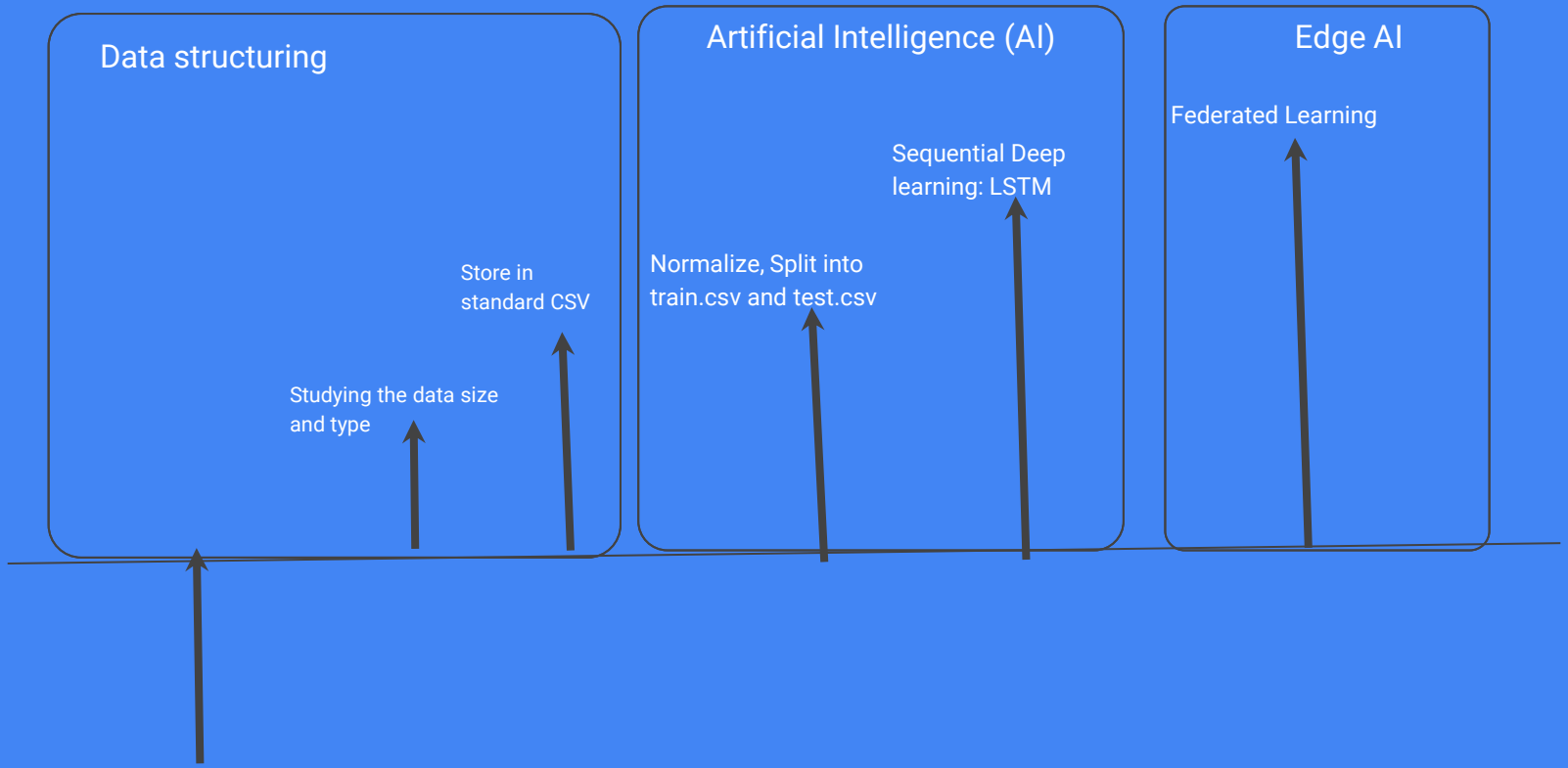


Image from. [4]

Blueprint

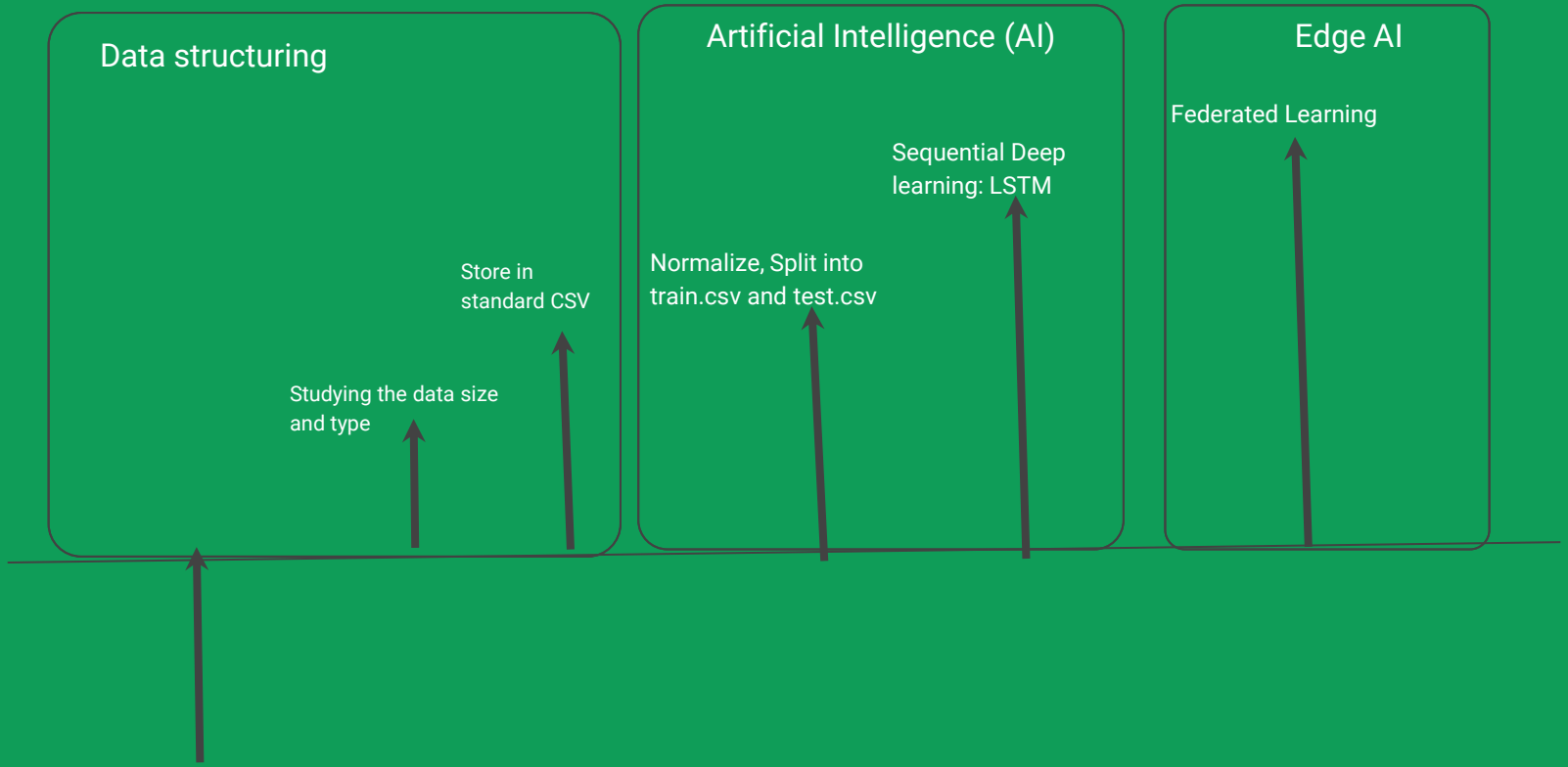

Data structuring
Data Understanding
Data Storage
Artificial Intelligence (AI)
Data Transformation
Data Analysis
Edge AI
Data centric analysis
Unstructured data

Ex1: Smart IoT sensor for biomagnetic signal monitoring

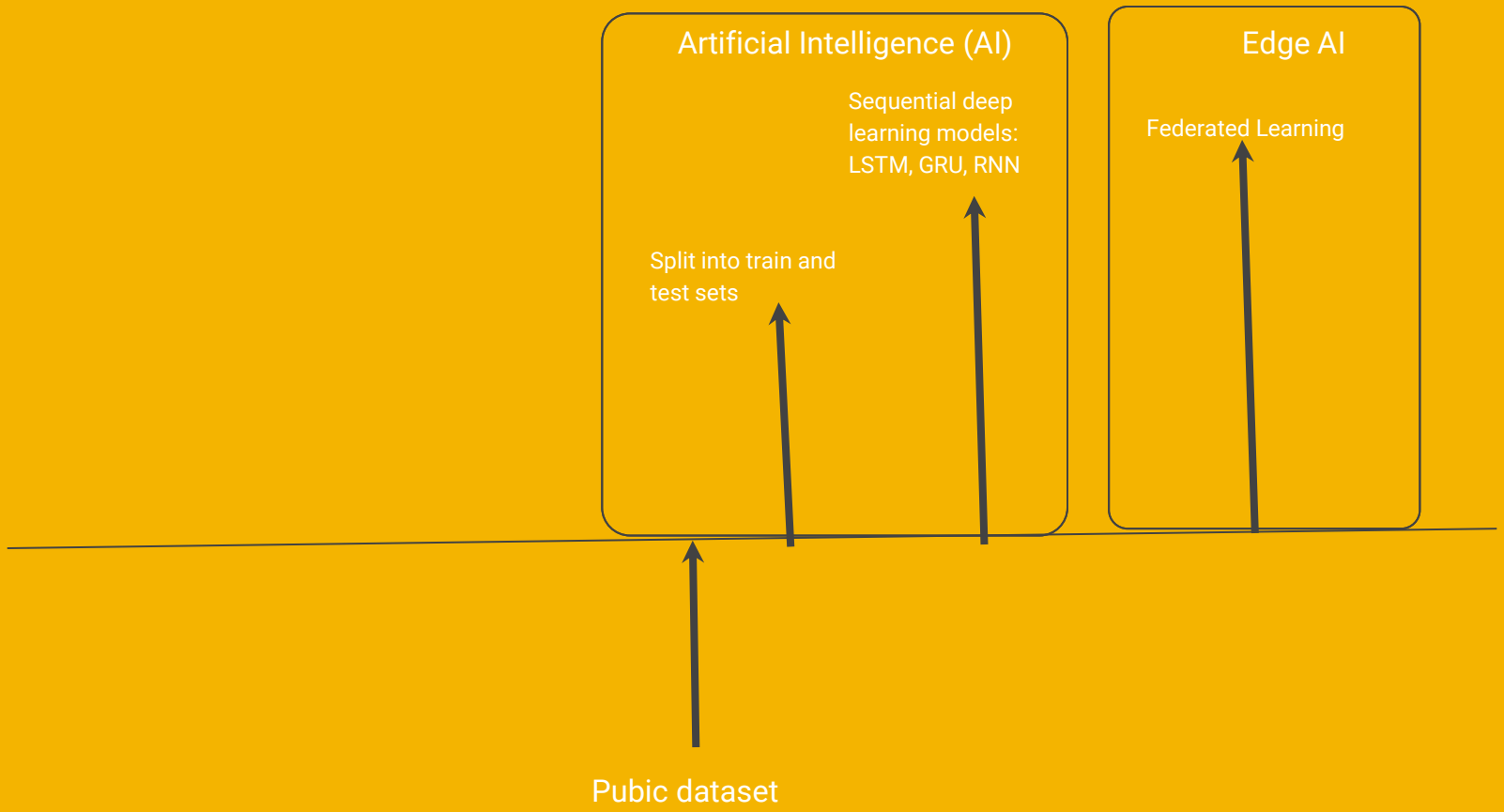


Magnetic Sensor-based portable MCG monitor

Ex2: Human Activity Recognition on edge supported by UAV Ad Hoc network



Ex3: Wifi-based Human Authentication

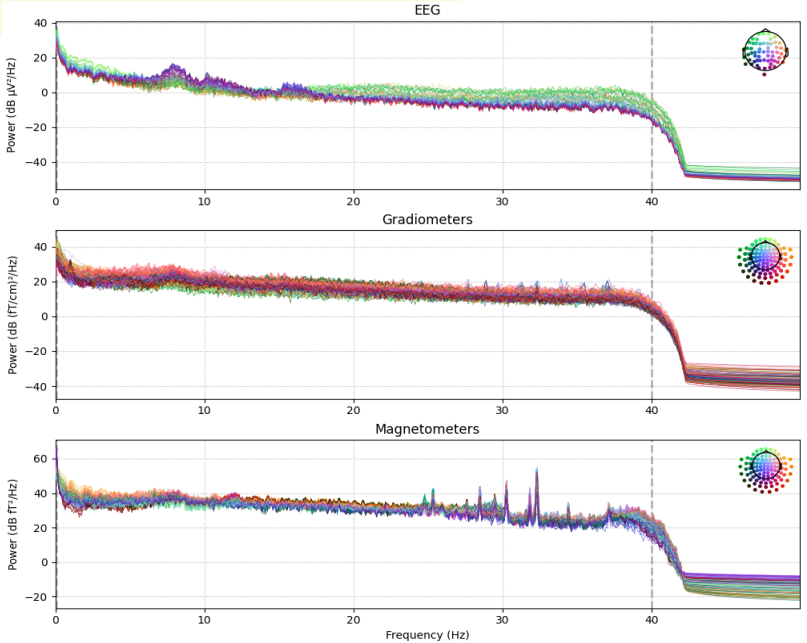


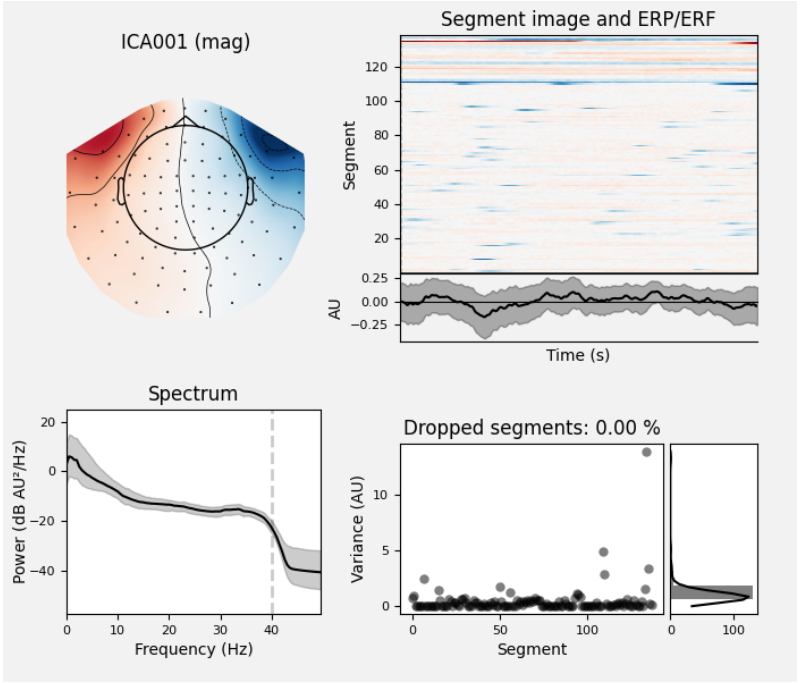
Ex 1. Smart IoT Sensor for Bio-magnetic Activity Monitoring

Biomedical signal is unstructured!

```
<Raw | sample_audvis_filt-0-40_raw.fif, 376 x 41700 (277.7 s), ~3.3 MB, data not loaded>
<Info | 14 non-empty values
 bads: 2 items (MEG 2443, EEG 053)
 ch_names: MEG 0113, MEG 0112, MEG 0111, MEG 0122, MEG 0123, MEG 0121, MEG ...
 chs: 204 Gradiometers, 102 Magnetometers, 9 Stimulus, 60 EEG, 1 EOG
 custom_ref_applied: False
 dev_head_t: MEG device -> head transform
 dig: 146 items (3 Cardinal, 4 HPI, 61 EEG, 78 Extra)
 highpass: 0.1 Hz
 hpi_meas: 1 item (list)
 hpi_results: 1 item (list)
 lowpass: 40.0 Hz
 meas_date: 2002-12-03 19:01:10 UTC
 meas_id: 4 items (dict)
 nchan: 376
 projs: PCA-v1: off, PCA-v2: off, PCA-v3: off, Average EEG reference: off
 sfreq: 150.2 Hz
>
```

https://mne.tools/dev/auto_tutorials/intro/10_overview.html#sphx-glr-auto-tutorials-intro-10-overview-py



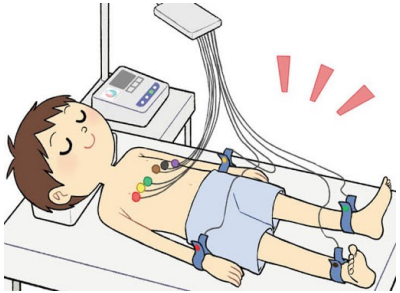


Use case

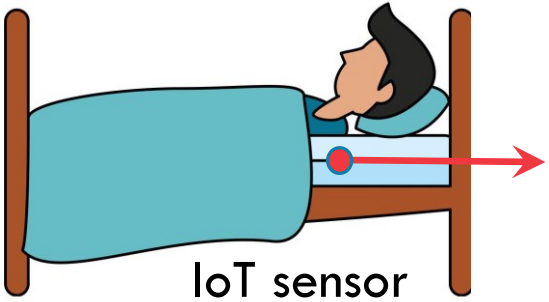
- EEG
  - Brain signal monitoring

- ECG
  - Cardiac signal monitoring

Electro vs Magnetocardiography



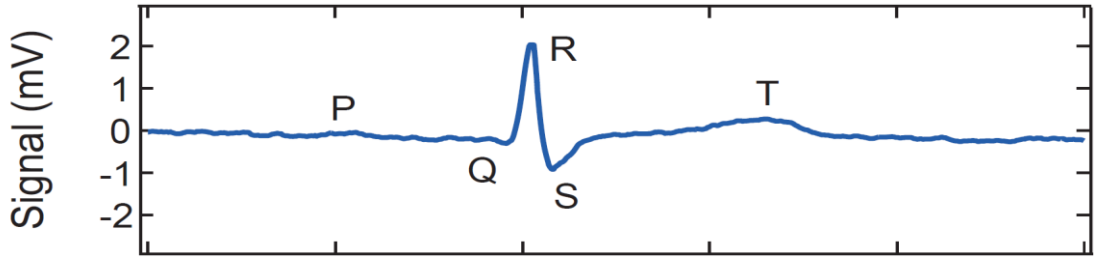
ECG

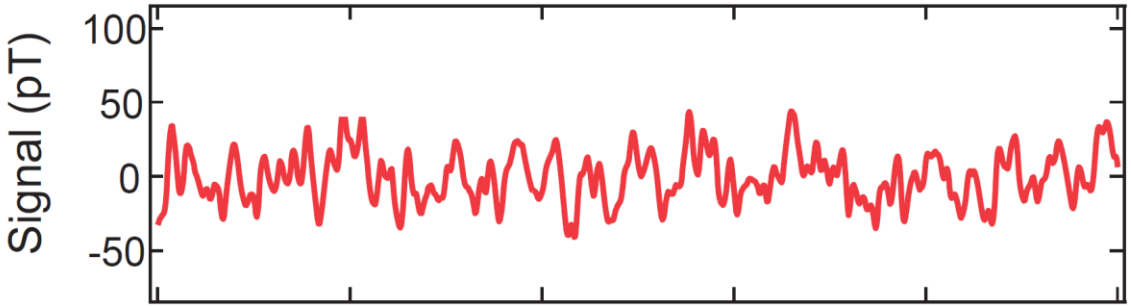


IoT sensor

Cardiac magnetic field (10^{-12}T)

by tiny TMR sensors





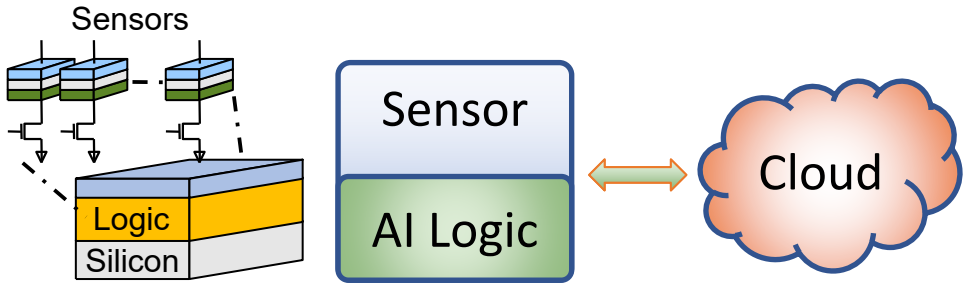
Diagnosis of heart diseases

(arrythmia, cardiac infraction, etc)

Smart IoT sensor concept

Conventional IoT flow: QoS * issues
delay, communication overhead, **energy**

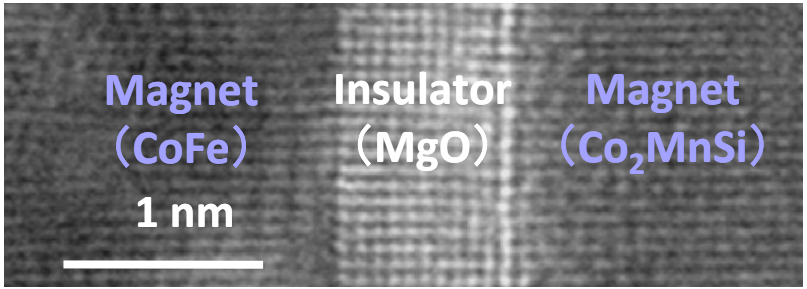
New direction:
Logic-in-sensor Improved QoS



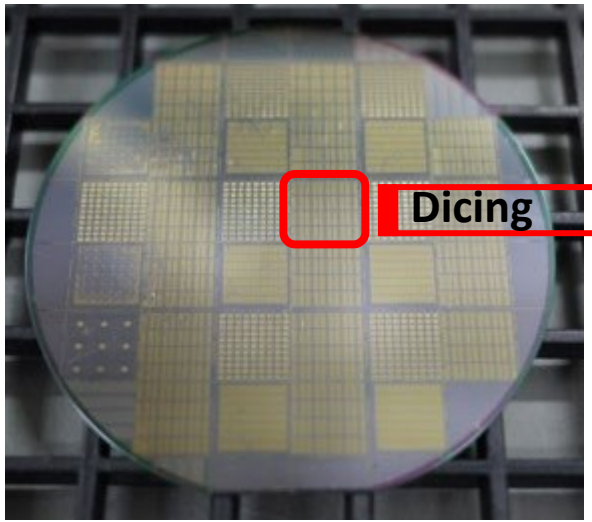
Magnetic IoT Sensing Example

- Tunnel Magneto-Resistance (TMR) sensors
  - used as HDD reading heads

- Magnetic tunnel junction (MTJ)

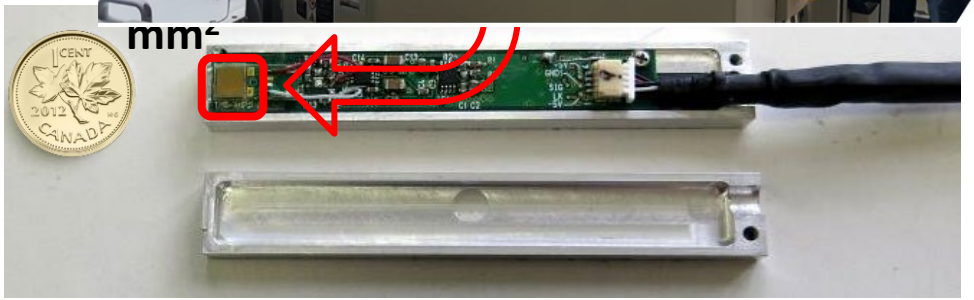


Tunnel Magneto-Resistance Sensor



array (10
5 x 5 mm

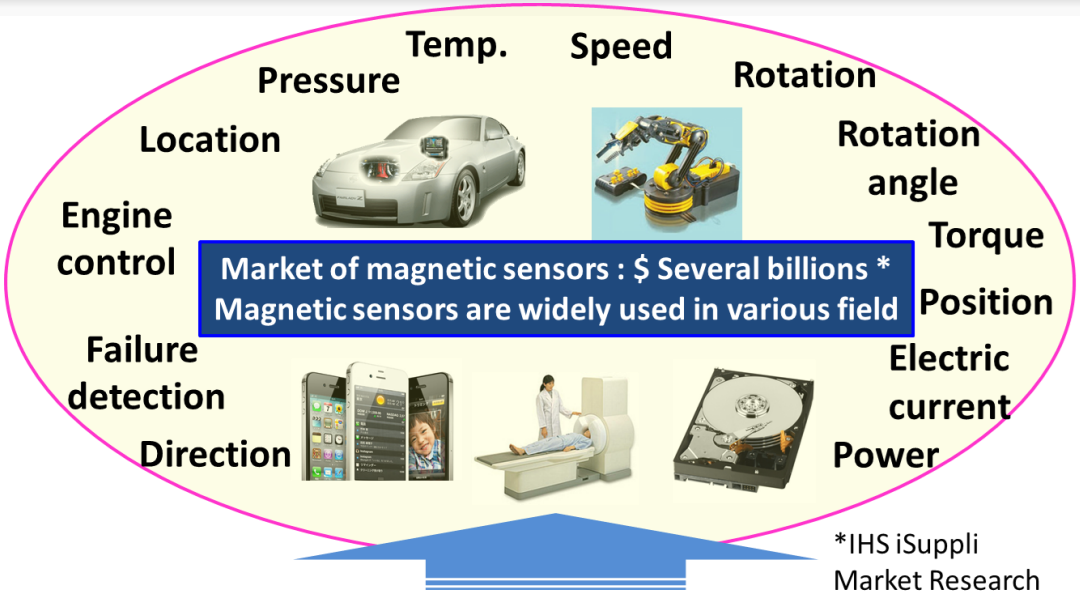
mm²



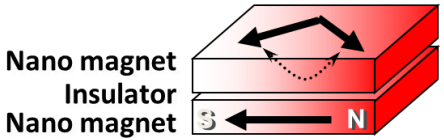
Courtesy of M. Oogane
Press release, 21 July 2015

Applications of Magnetic IoT Sensors

- Magnetic field measurement can be applied to other IoT sensing applications



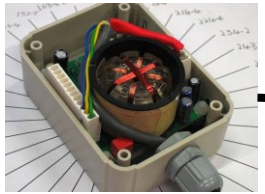
Magnetic tunnel junctions



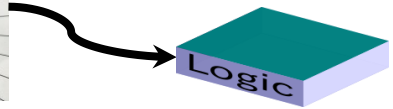
Advantages

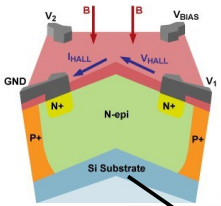
- ✓ High sensitivity
- ✓ Small size
- ✓ Low power consumption
- ✓ Low price

Integrating Magnetic Sensor and Logic



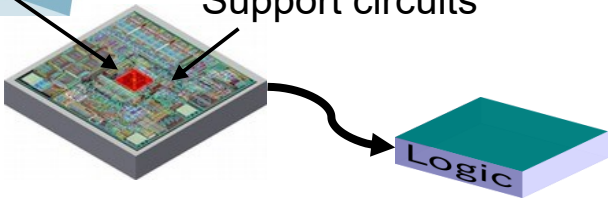
Fluxgate: bulky





Hall sensors:
Use silicon area

Support circuits



Source: www.allegromicro.com

- Communication is very expensive
  - Bluetooth Low-Energy: 10~500 mW, 2 Mbit/s
    vs.
  - 32-bit processor: ~1 mW, 30 MHz
  - TMR sensor-based MCU 50 µW, 200 MHz
    (Collaborator - Tohoku University)

A Big Challenge: High 1/f noise

- Cardio-rhythm and 1/f noise share the same spectral frequencies

PSD of low-frequency noise

sensor susceptibility

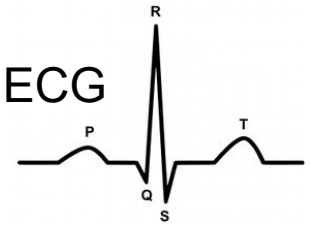
$$S_v \propto \frac{\chi}{M_s V} \frac{1}{f^{\beta}}$$

sensor saturation magnetization

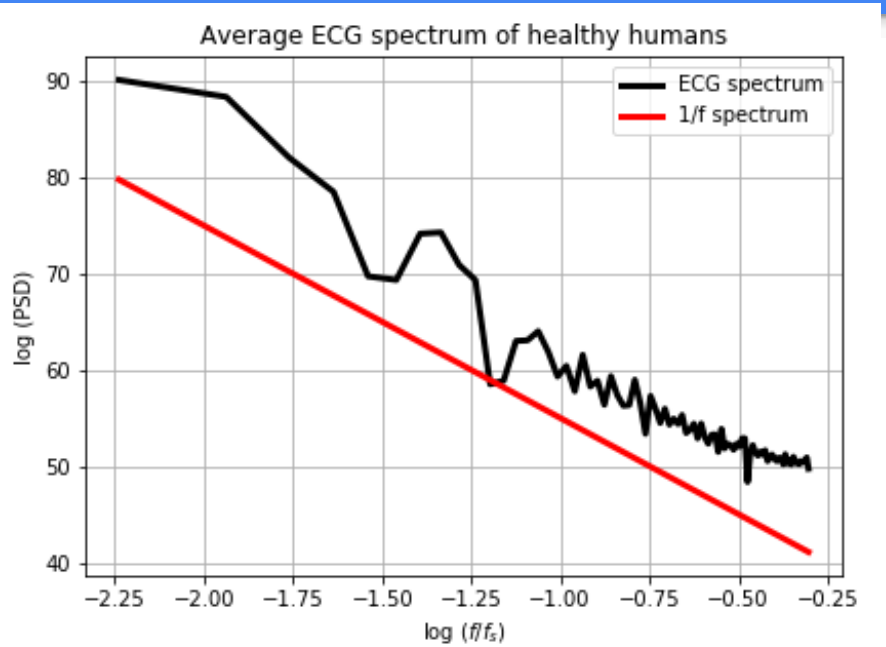
sensor volume

spectral frequency

exponent of noise spectrum



Pure 1/f Noise

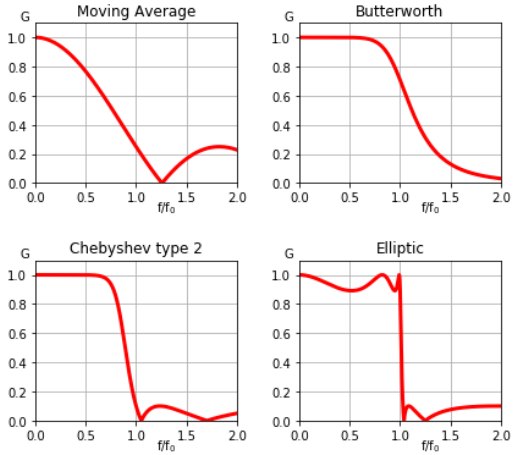


❑ Denoising MCG obtained by TMR sensors is needed at the IoT node

Traditional Filters

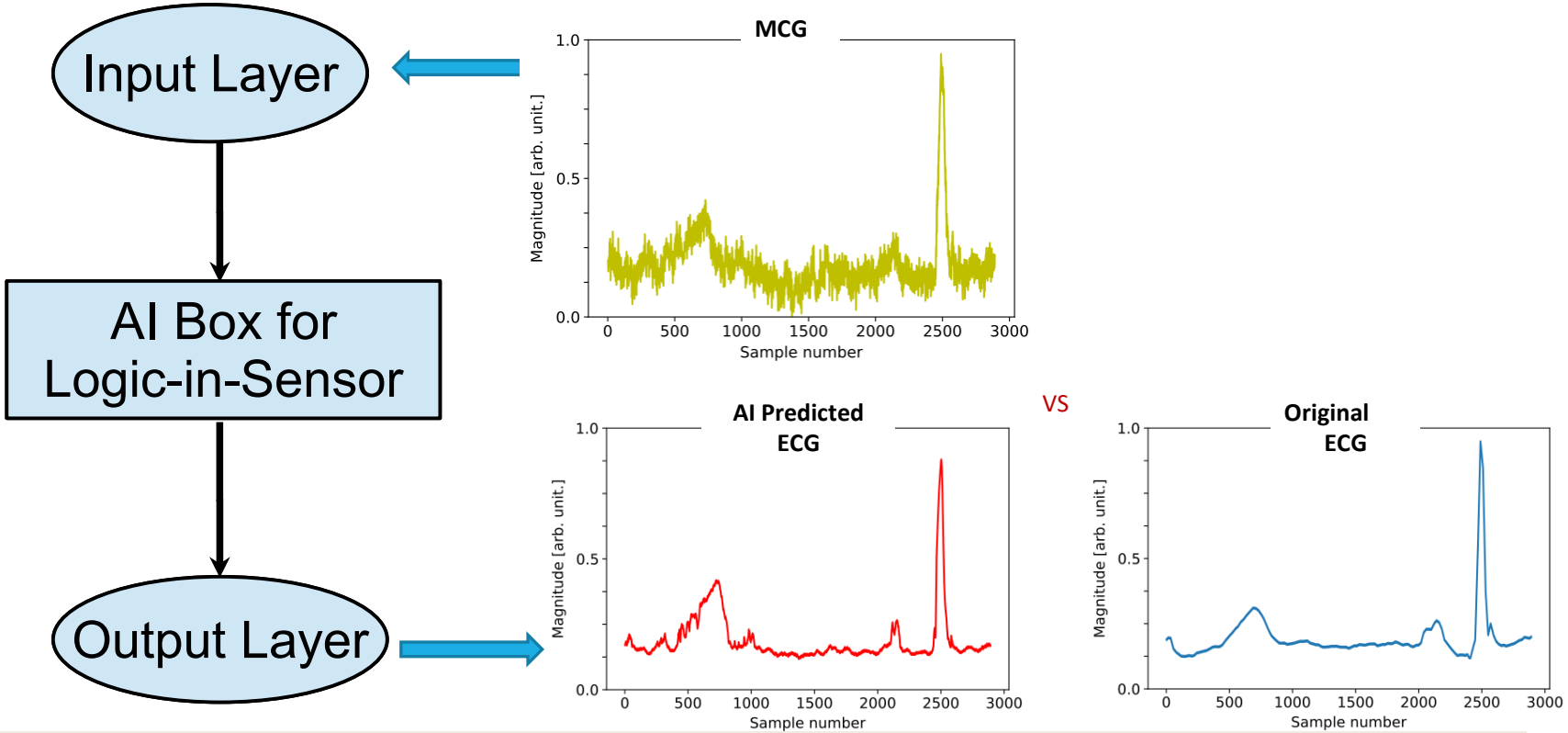
- All traditional linear filters cannot distinguish between signal and noise
  - This is more problematic for 1/f noise

Transfer function of linear low-pass filter



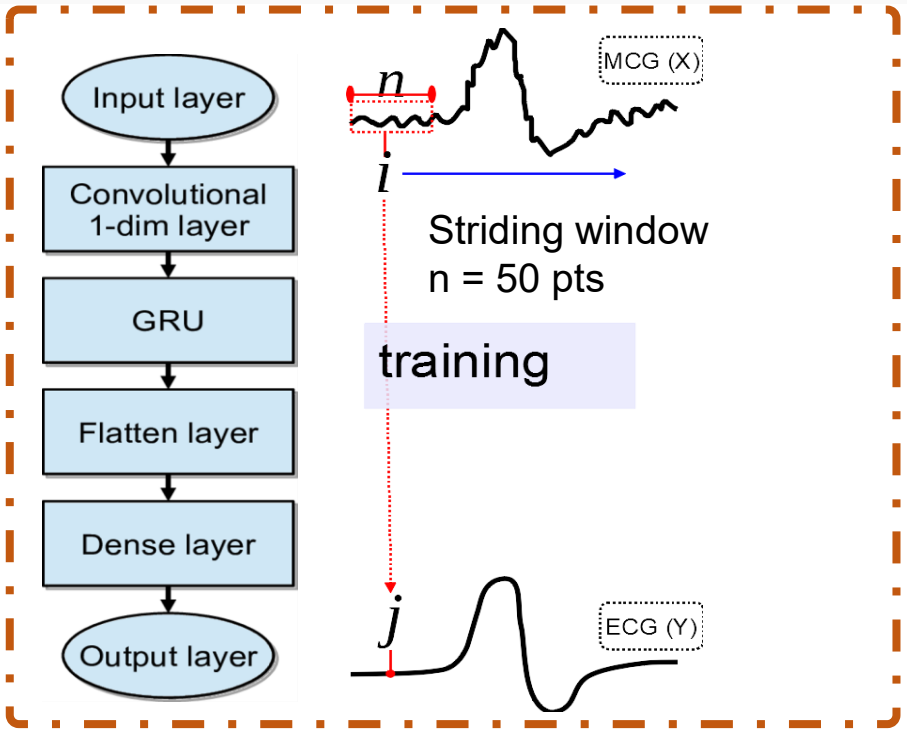
❑Maxwell filter can remove noise outside the volume covered by sensor array but not the noise of the sensor array

Proposed Deep Learning Method



A. Mohsen, M. Al-Mahdawi, M. M. Fouda, M. Oogane, Y. Ando, and **Z. M. Fadlullah**, "AI Aided Noise Processing of Spintronic Based IoT Sensor for Magnetocardiography Application," Proc. IEEE International Conference on Communications (ICC), Dublin, Ireland, Jun. 2020.

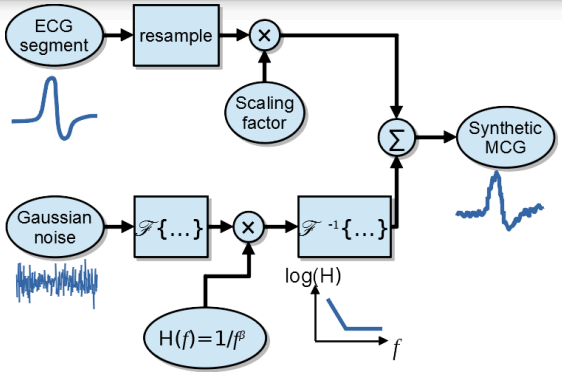
Proposed Deep Learning Method

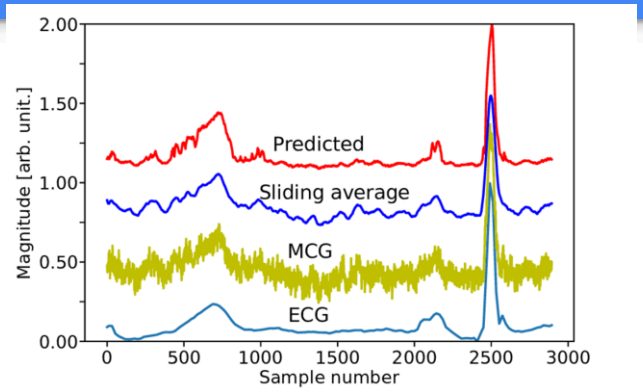


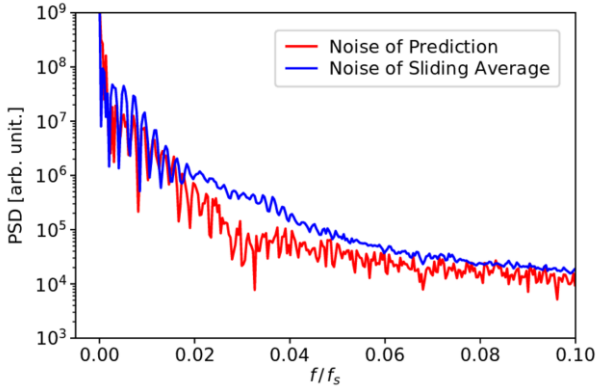
- Combination of CNN and GRU is very effective for separating chaotic attractors
  - Like eyes + brain

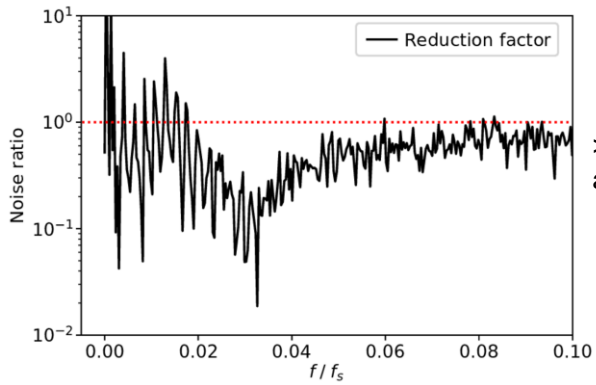
A. Mohsen, M. Al-Mahdawi, M. M. Fouda, M. Oogane, Y. Ando, and **Z. M. Fadlullah**, "AI Aided Noise Processing of Spintronic Based IoT Sensor for Magnetocardiography Application," Proc. IEEE International Conference on Communications (ICC), Dublin, Ireland, Jun. 2020.

Performance Evaluation









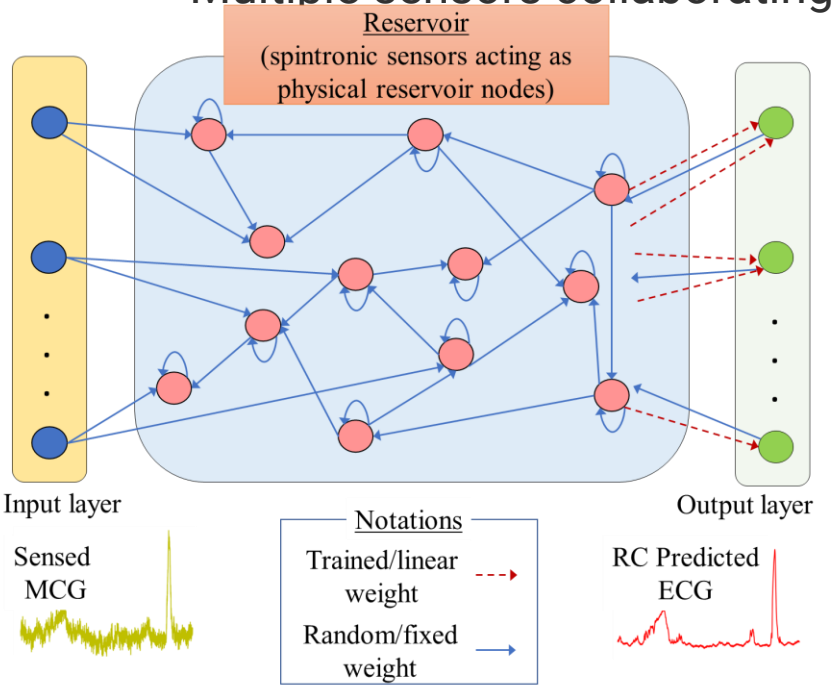
>10 times better than avg.

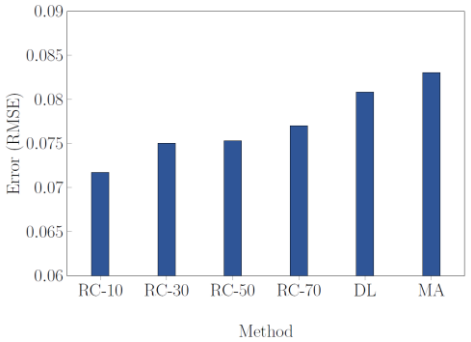
A. Mohsen, M. Al-Mahdawi, M. M. Fouda, M. Oogane, Y. Ando, and **Z. M. Fadlullah**, "AI Aided Noise Processing of Spintronic Based IoT Sensor for Magnetocardiography Application," Proc. IEEE International Conference on Communications (ICC), Dublin, Ireland, Jun. 2020.

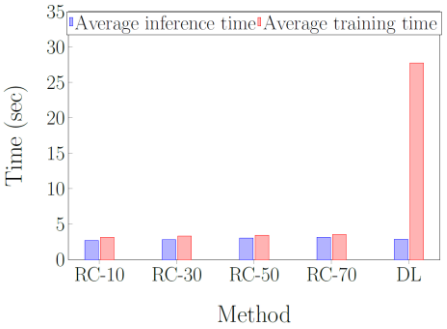
Sadman Sakib, Mostafa M. Fouda, Muftah Al-Mahdawi, Attayeb Mohsen, Mikihiko Oogane, Yasuo Ando, and **Zubair Md Fadlullah**, "Deep Learning Models for Magnetic Cardiography Edge Sensors Implementing Noise Processing and Diagnostics," in IEEE Access, vol. 10, pp. 2656-2668, 2022, doi: 10.1109/ACCESS.2021.3138976

Enhanced Solution

- Multiple sensors collaborating together as a reservoir computer

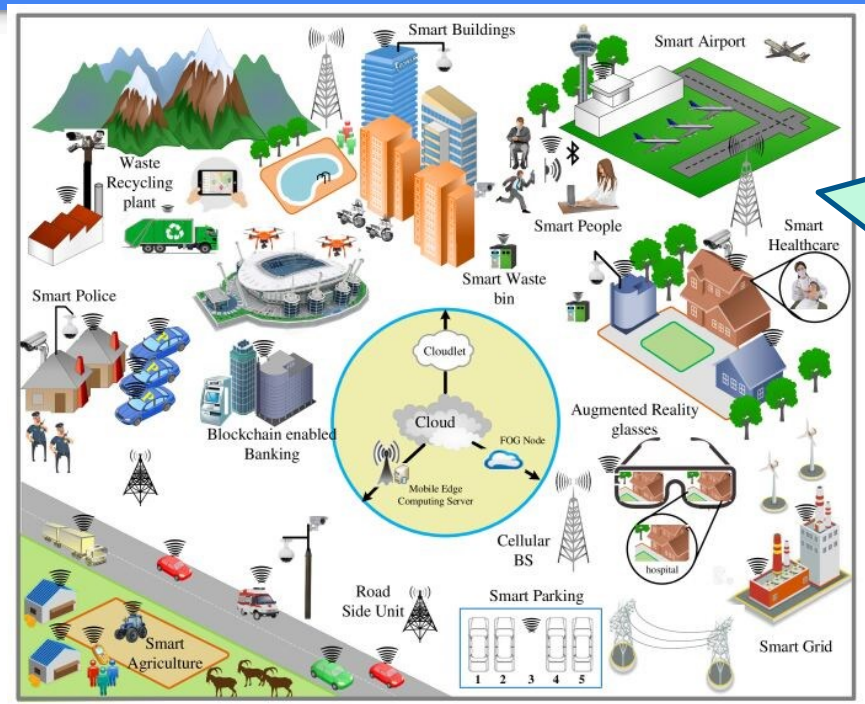


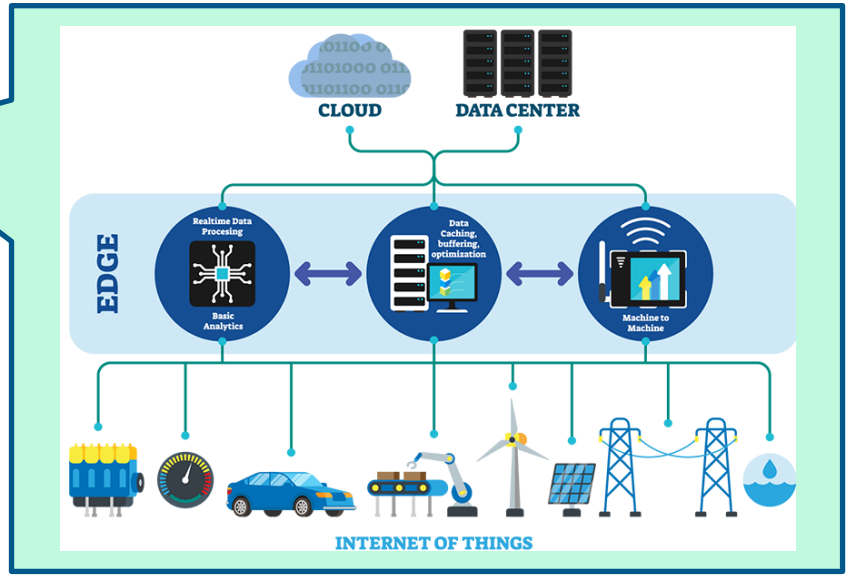




Sadman Sakib, Mostafa M. Fouda, Muftah Al-Mahdawi, Attayeb Mohsen, Mikihiko Ooagne, Yasuo Ando, and **Zubair Md Fadlullah**, "Noise-Removal from Spectrally-Similar Signals Using Reservoir Computing for MCG Monitoring," ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1-6, doi:10.1109/ICC42927.2021.9500993

Edge Computing

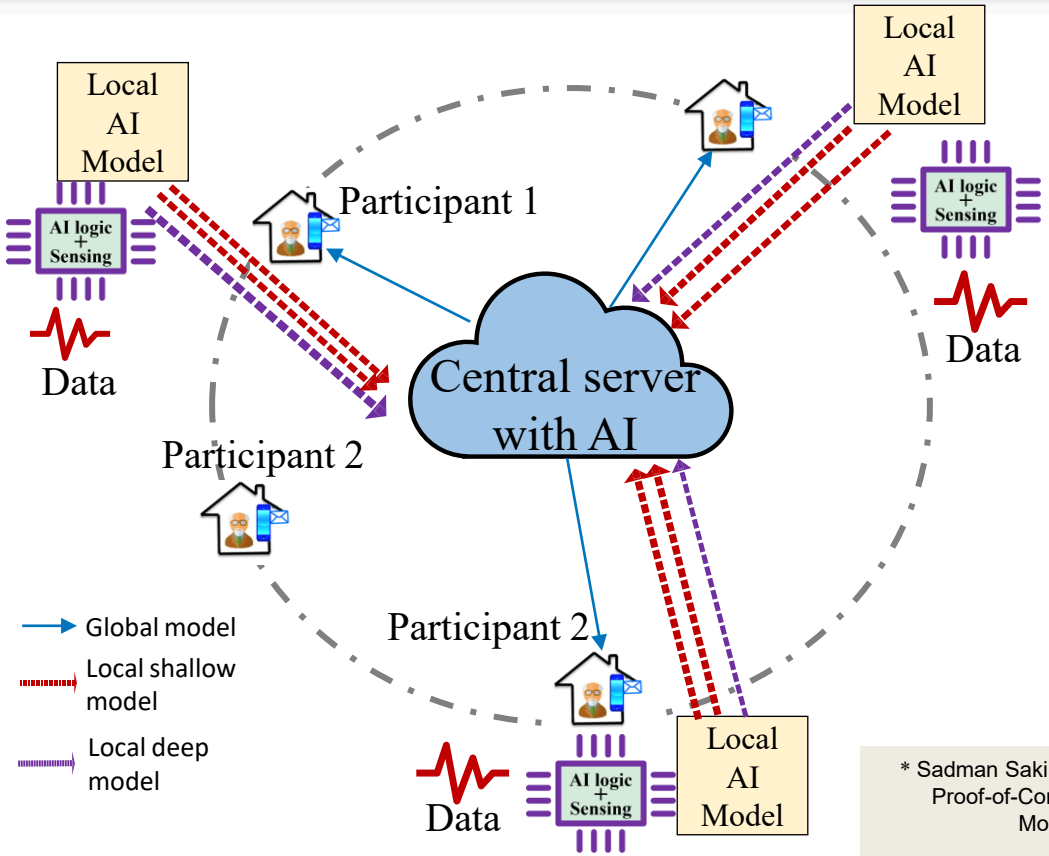




Edge Computing with mobile users and IoT nodes becomes more privacy-sensitive

L. U. Khan et al., "Edge-Computing-Enabled Smart Cities: A Comprehensive Survey," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10200-10232, Oct. 2020.

Cardiac Irregularity Detection in Edge IoT



*Centralized AI model does the job but requires participant data

Can edge devices (participants/users) locally train model?

** Find a **distributed collaborative learning approach**

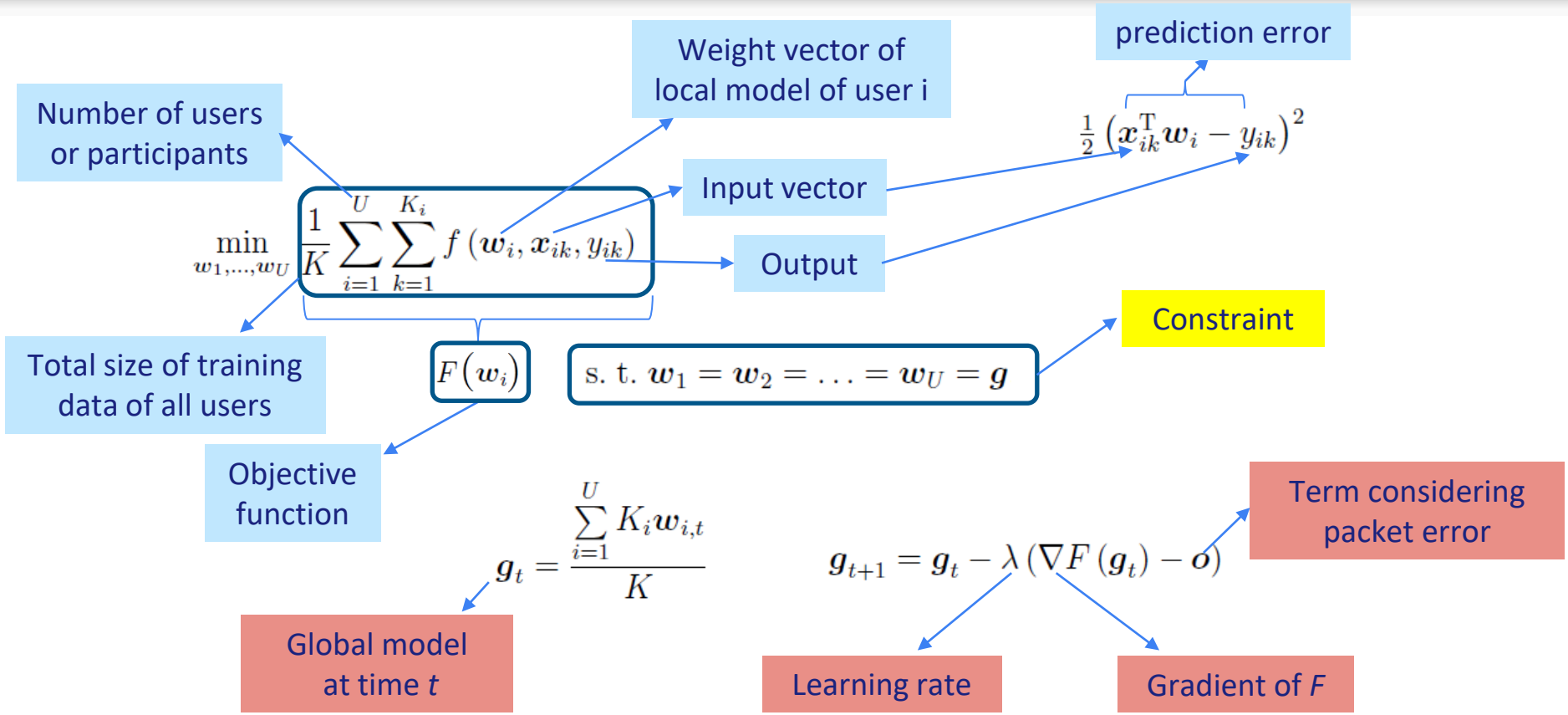
Federated Learning (FL)

Design considerations?

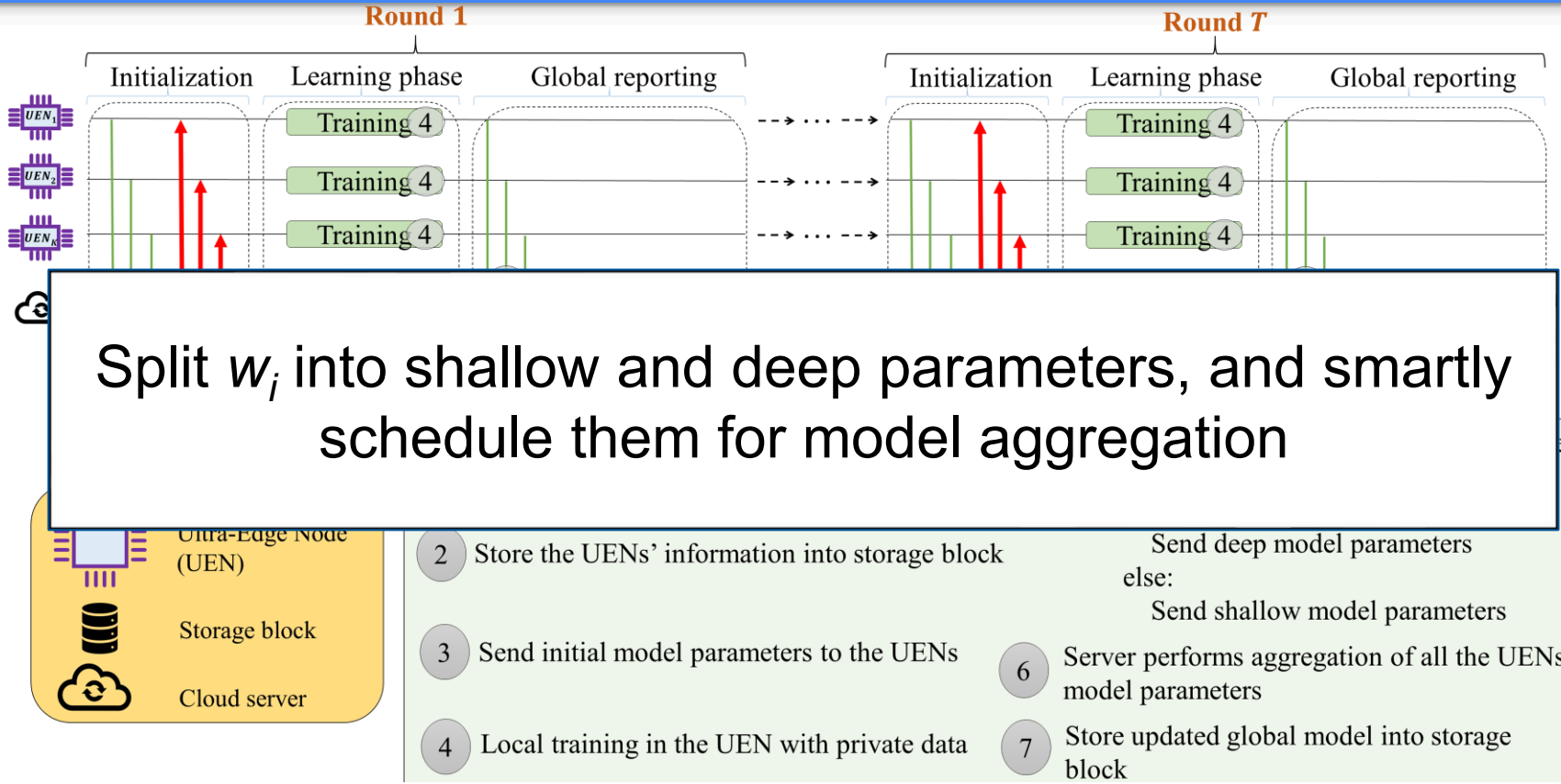
* Sadman Sakib, Mostafa M. Fouda**, Zubair Md Fadlullah**, Nidal Nasser and Waleed Alasmary, "A Proof-of-Concept of Ultra-Edge Smart IoT Sensor: A Continuous and Lightweight Arrhythmia Monitoring Approach," in IEEE Access, vol. 9, pp. 26093-26106, 2021, doi: 10.1109/ACCESS.2021.3056509.

** Sadman Sakib, Mostafa M. Fouda, **Zubair Md Fadlullah**, Khalid Abualsaud, Elias Yaacoub and Mohsen Guizani, "Asynchronous Federated Learning-based ECG Analysis for Arrhythmia Detection," 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), 2021, pp. 277-282

System Model



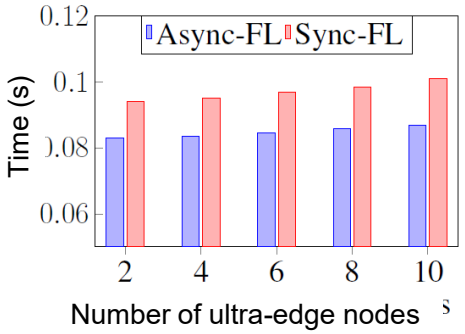
Proposed Asynchronous Weight Updating FL

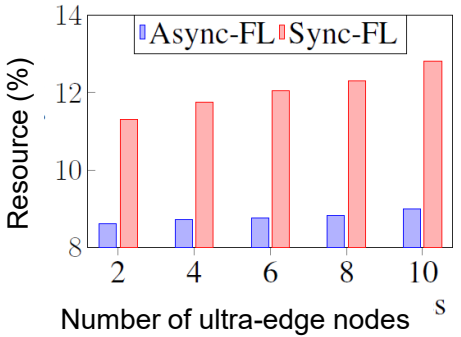


Sadman Sakib, Mostafa M. Fouda, **Zubair Md Fadlullah**, Khalid Abualsaud, Elias Yaacoub and Mohsen Guizani, "Asynchronous Federated Learning-based ECG Analysis for Arrhythmia Detection," 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), 2021, pp. 277-282

Performance Evaluation

Method	Dataset #	Metrics	Number of Ultra-edge nodes				
			2	4	6	8	10
Conventional sync-FL	DS1	Accuracy	0.888	0.889	0.889	0.878	0.877
		Precision	0.878	0.879	0.874	0.868	0.871
		F1-score	0.880	0.881	0.881	0.871	0.871
	DS2	Accuracy	0.892	0.889	0.886	0.895	0.863
		Precision	0.884	0.867	0.859	0.881	0.838
		F1-score	0.867	0.861	0.867	0.865	0.849
	DS3	Accuracy	0.767	0.777	0.753	0.732	0.748
		Precision	0.941	0.952	0.948	0.950	0.950
		F1-score	0.613	0.783	0.834	0.820	0.791
Proposed async-FL	DS1	Accuracy	0.868	0.879	**0.891**	**0.879**	**0.878**
		Precision	0.872	0.878	0.883	0.875	0.877
		F1-score	0.869	0.875	0.887	0.875	0.875
	DS2	Accuracy	0.870	0.886	**0.894**	**0.895**	**0.898**
		Precision	0.841	0.859	0.889	0.881	0.891
		F1-score	0.852	0.867	0.867	0.868	0.873
	DS3	Accuracy	0.740	0.737	0.732	**0.769**	**0.762**
		Precision	0.950	0.948	0.944	0.953	0.947
		F1-score	0.811	0.824	0.735	0.844	0.695





Sadman Sakib, Mostafa M. Fouda, **Zubair Md Fadlullah**, Khalid Abualsaud, Elias Yaacoub and Mohsen Guizani, "Asynchronous Federated Learning-based ECG Analysis for Arrhythmia Detection," 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), 2021, pp. 277-282

Ex 2. Wearable-based Human Activity Recognition

Wearable-based Human Activity Recognition

- Wearable-based Human Activity Recognition classifies activities using sensory data from wearable devices.
- Traditional machine learning methods offer accurate predictions but requires data preprocessing steps such as data normalization, feature selection, and feature engineering.
- Deep learning, however, extracts patterns from raw data more efficiently, improving accuracy at the expense of increased computational resources.

Tracking activities

Health monitoring

Sleep tracking

Problem statement

- Multiple participant users residing in a rural community that lacks internet connection (e.g. Northern Ontario) each is given a wearable device (e.g. Miband 4 smart watch) and a data collection device.

- Data collected from a single user is not enough to be used for training a Human Activity Recognition (HAR) system. Therefore data from multiple users need to be integrated to train a global HAR model.

- How can users **collaboratively yet privately** train a human activity recognition system using the sensory data extracted from the sensors of the wearable device..

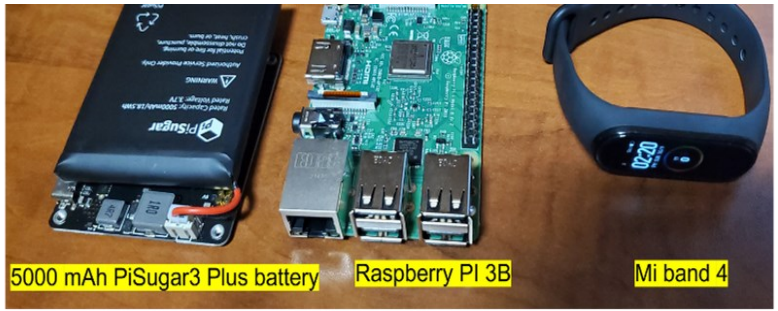
Problem statement

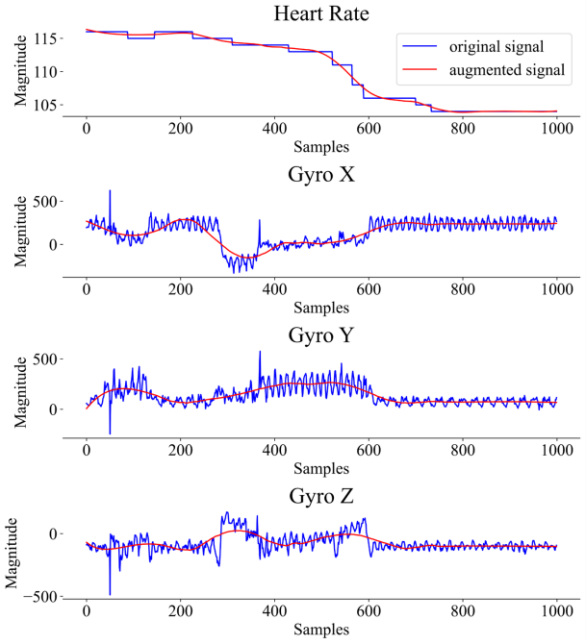
- To combine users' data and utilize it for deep learning training of a HAR model, we can follow one of the following two approaches:
    - **Central deep learning**
        - Sending users' data to a central server to train a central global model.
    - **Federated learning**
        - Training multiple local models and send the local models to a central server to be aggregated into a global model

Data collection

We use a commercial smart watch (Xiaomi Mi band 4) and a custom-built data collection device (Raspberry pi) to collect and annotate a Human Activity Recognition (HAR) dataset.

The data has four time-series features:
1- Heart rate
2- Gyro x (gyroscope measurement in the x-axis)
3- Gyro y (gyroscope measurement in the y-axis)
4- Gyro z (gyroscope measurement in the z-axis)



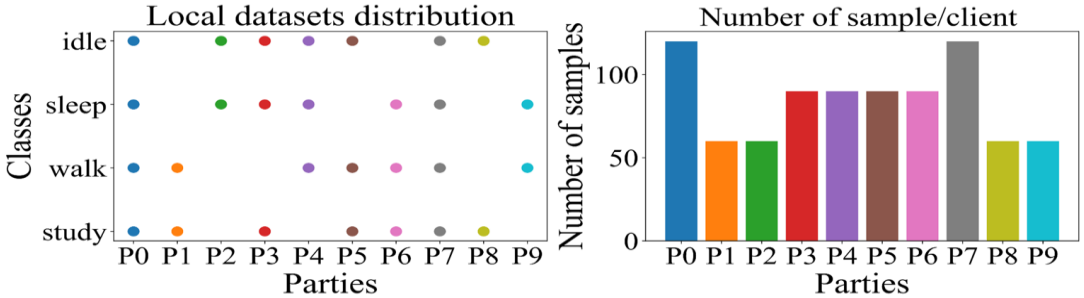


Wearable-based dataset

- Participants were able to collect an annotated HAR dataset from the wearable device.
- The dataset was collected from 10 subjects and its ground truth has four labels including:

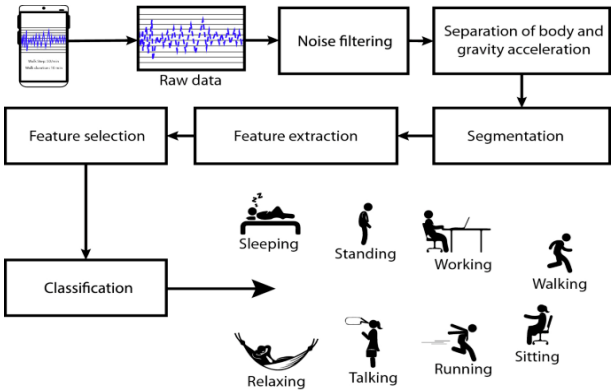
  idle, sleep, walk, and study.

- Some subjects lack data belonging to certain classes, therefore subjects need to collaborate with each other to train a powerful HAR model that can classify all defined human activities collected across subjects.

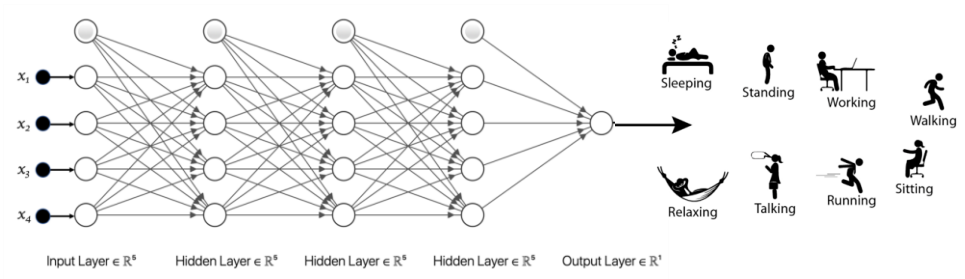


Data Analysis

We decide to use Deep learning to train a HAR model instead of machine learning, because deep learning requires less data preprocessing and trains higher performance models.



Machine learning based HAR

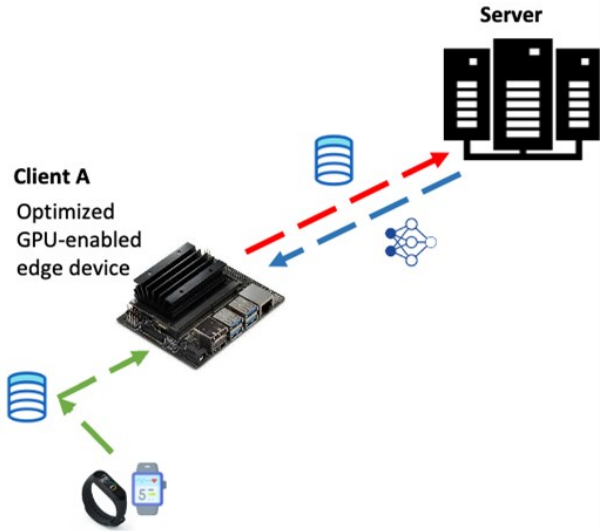


Deep learning based HAR

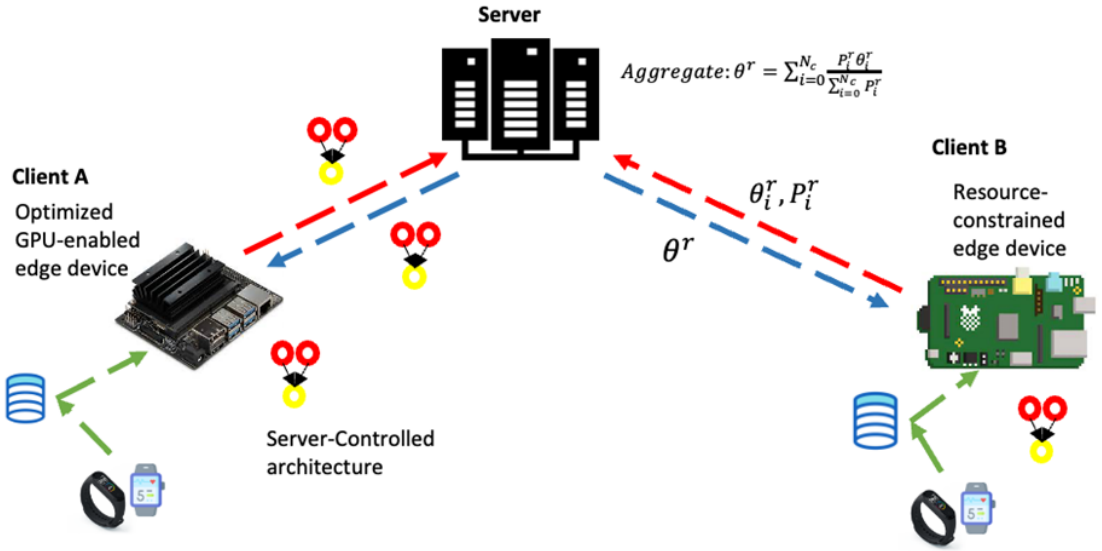
Data training paradigm

Choosing the training paradigm depends on our privacy-performance tradeoff.

In central deep learning, data is transferred to a central server for training. However, moving data from edge devices (where they are born) to a central server poses privacy risks, high communication cost, and is computationally inefficient.



Federated learning



Model sharing Federated Learning

PROS

✓ Federated Learning trains the distributed local models θ_i^r without sharing users' data.

CONS

✗ Shared parameters can still pose privacy risks.

✗ Communication bandwidth is high.

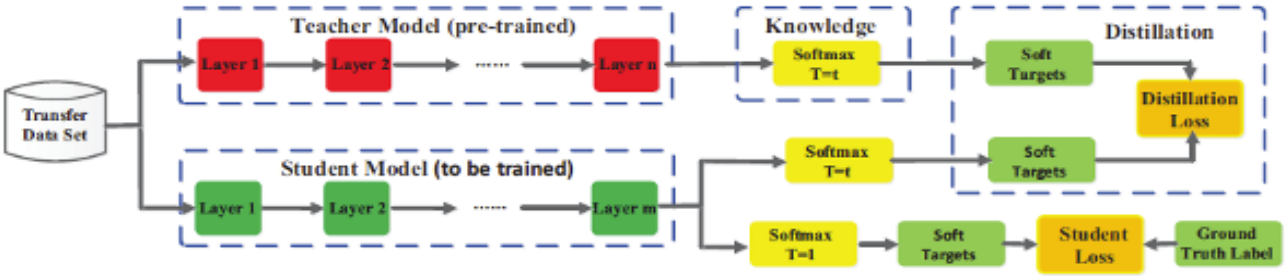
✗ GPU-enabled edge devices unable to use optimized architecture due to different HW/SW stack.

Deep learning model training in standard federated setting

Knowledge Distillation

Knowledge Distillation is a recent deep learning technique to train a student model using a trained teacher model.

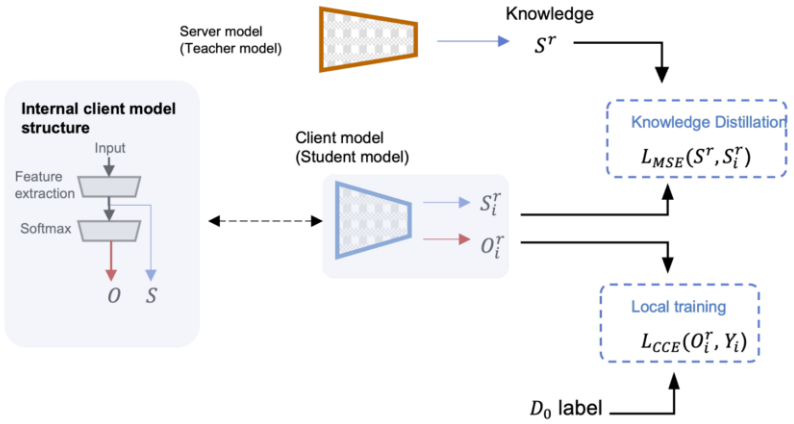
Knowledge Distillation-based federated learning is **communication-efficient** and allows **independently-designed local models.**



Knowledge Distillation in central setting

Knowledge Distillation-based Federated Learning

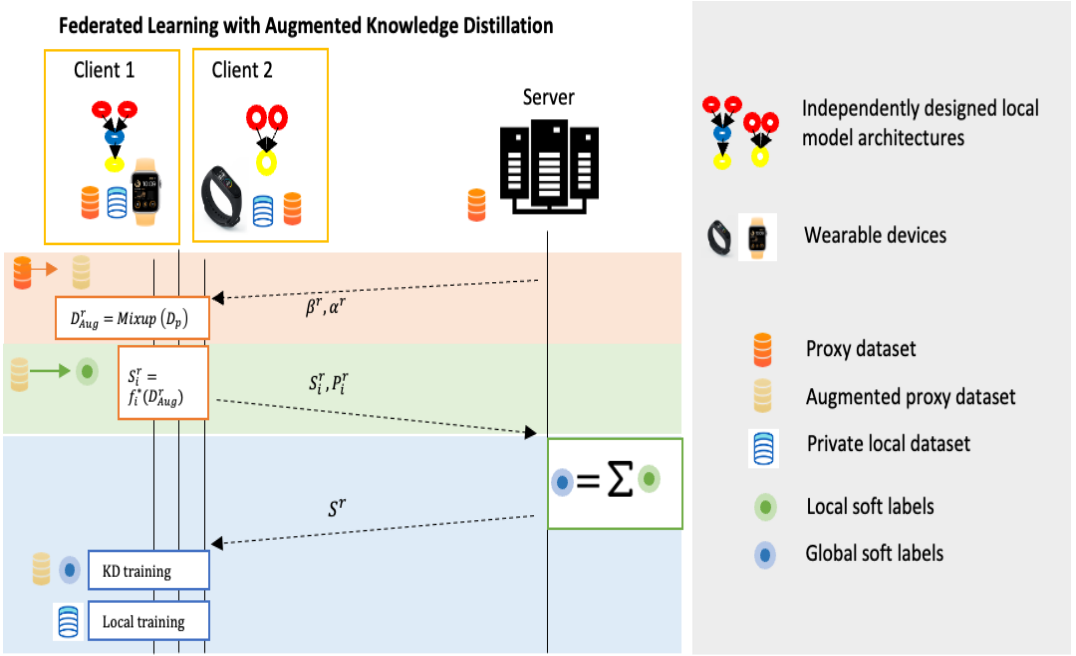
Knowledge Distillation based FL relies on a shared **public data** to distill the knowledge (patterns) of the **private data.**



Knowledge Distillation in Federated learning

Federated Learning via Knowledge Distillation

We propose Federated Learning via Augmented Knowledge Distillation (FedAKD).



Algorithm 1 FedAKD Algorithm

Input: Public dataset: D_p, Test dataset: D_t, Local dataset of client i: D_i, Independently designed local model of client i: f_i, Number of communication rounds: R, Number of epochs for local training: E_l, Number of epochs for KD training: E_{KD}, Loss function for local training: L_l, Loss function for KD training: L_{KD}, Total number of participating clients: N_c, Fraction of clients participating at any given round: K

Output: Collaboratively trained local model f_i

Initialize f_i

for round $r = 1$ to R **do**

  $N_k = K \cdot N_c$

  Select N_k clients randomly

  $\rho^r, \alpha^r \leftarrow$ server randomly generated

  Broadcast ρ^r, α^r

  for client $i = 1$ to N_k **do**

    $D^r_d \leftarrow$ permute(D_p, ρ^r)

    $D^r_{Aug} \leftarrow$ mixup(D_p, D^r_d, α^r)

    $S^r_i \leftarrow$ calculate soft labels on D^r_{Aug}

    $P^r_i \leftarrow$ calculate accuracy on D_t

    Send S^r_i, P^r_i to Server

  end for

  $S^r \leftarrow$ aggregate S^r_i weighted by P^r_i

  Broadcast S^r

  for client $i = 1$ to N_k **do**

    for epoch $e = 1$ to E_{KD} **do**

      Compute gradients $g_{KD} = \nabla L_{KD}(f_i, D^r_{Aug}, S^r)$

      Update f_i: $f_i \leftarrow f_i - \eta \cdot g_{KD}$

    end for

    for epoch $e = 1$ to E_l **do**

      compute the gradients $g_{CCE} = \nabla L_{CCE}(f_i, D_i, Y_i)$

      Update f_i: $f_i \leftarrow f_i - \eta \cdot g_{CCE}$

    end for

  end for

end for

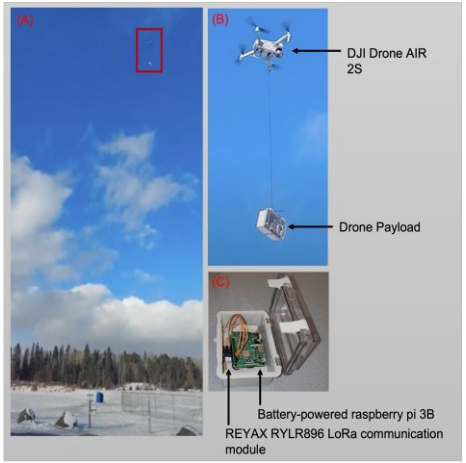
Federated learning via Augmented Knowledge Distillation

Federated Learning in the wild

We design a proof of concept for an UAV-based ad hoc network to traverse the households of participants who reside in a rural community that lacks internet connection infrastructure.

In the case of central deep learning:
The UAV is tasked with collecting users' HAR data, transferring the data to a central server, training a central HAR model, and transferring the centrally trained model back to all users.

In the case of federated learning
The UAV is tasked with collecting locally trained HAR models from all users, transferring local weights to a central server which aggregates the local weights into a global model, and transferring the aggregated global weights back to all users.



Field implementation of a drone-aided lora network with Reliable Data Transfer (RDT)

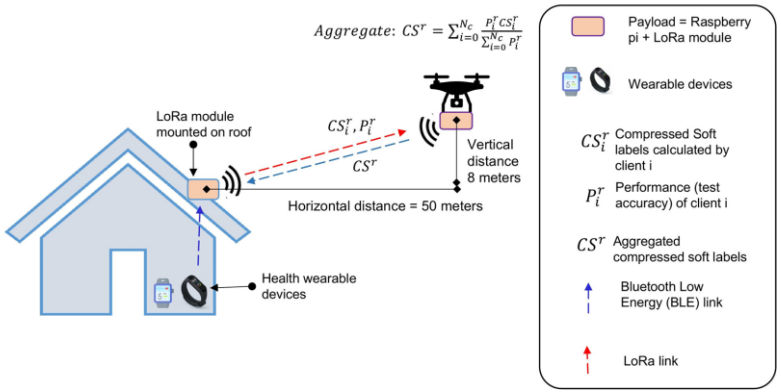
Federated Learning in the wild

Due to the limited battery capacity of the UAV, the flight time is limited to around 35 minutes, UAV operations should be optimized. We do this as follows:

1. To maximize the distance the UAV can cover, we use **Self-Organizing-Maps (SOM)** for path planning.
2. To minimize the time the UAV spends uploading/downloading data/weights, we use Knowledge Distillation-based Federated Learning (KD-based FL) instead of model-based FL due to significantly less data size, and employ compression techniques to further reduce data size.

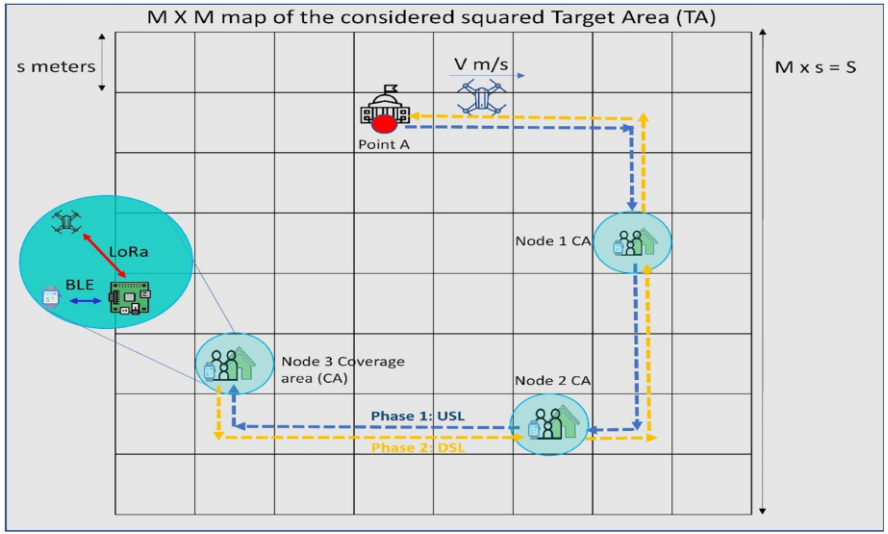
Compressed Knowledge Distillation Federated Learning (CFedAKD)

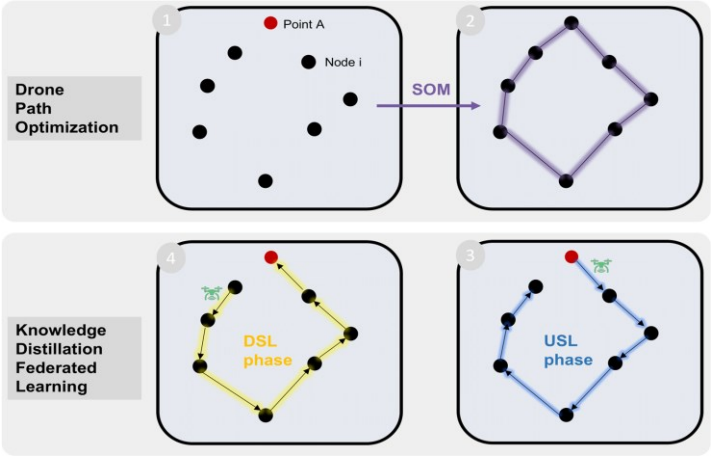
For communication efficiency, we proposed Compressed Federated Learning viai Augmented Knowledge Distillation (CFedAKD) which uses compression to reduce the size of the transmitted soft labels.



Clients share local Compressed Soft labels (CS) over LoRa Comm.

Path planning using Self-Organizing Maps

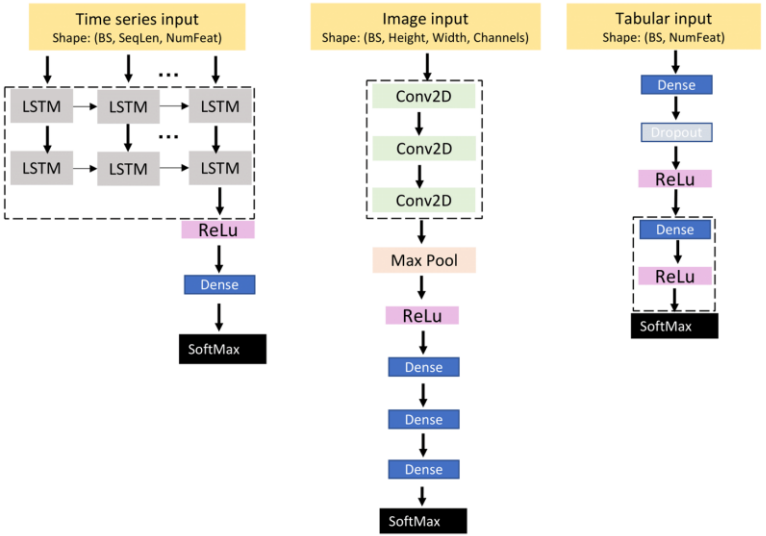




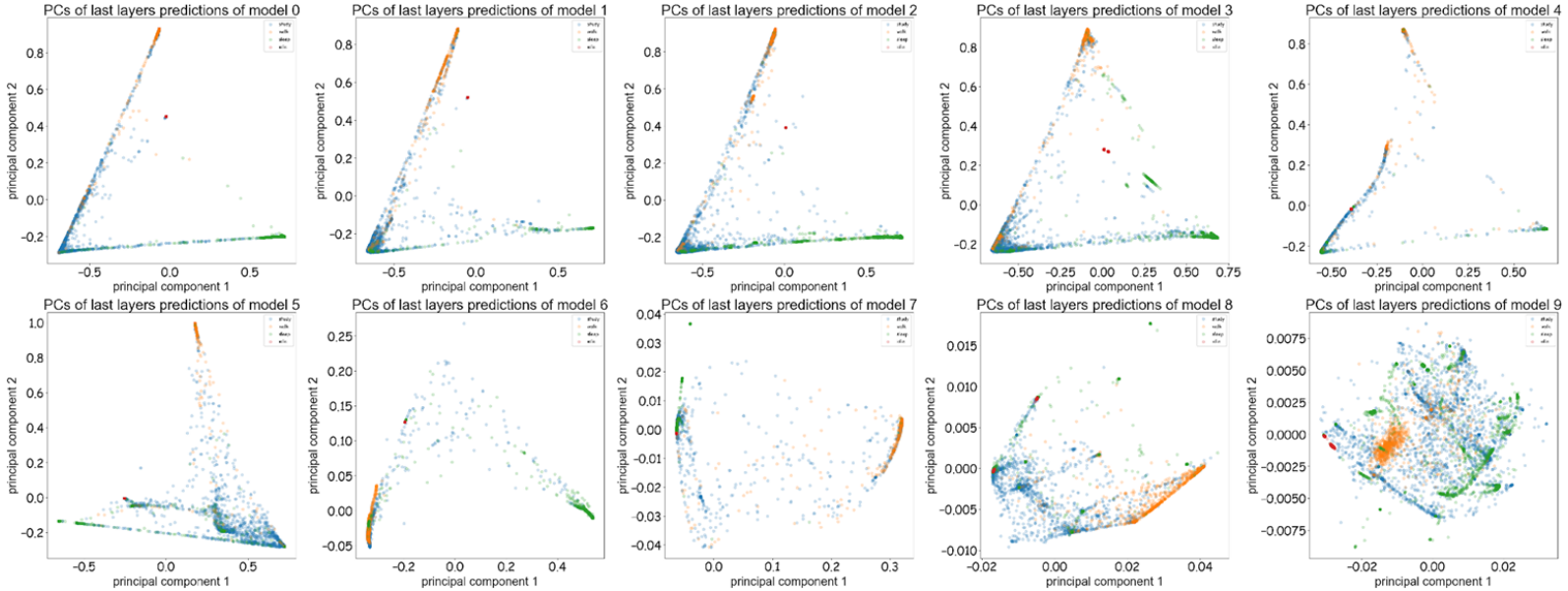
Deep learning model architectures

We use and compare different sequential model architectures including:

1. LSTM
2. CNN
3. MLP



Exploring PCA



We use Principal Component Analysis (PCA) to perform dimensionality reduction. We can then asses the separability of features by color-mapping each sample to its label.

Evaluation

Performance

- Compare the accuracies of baseline FL algorithms on wearable-based HAR datasets.

Communication cost

- Communication overhead between model-based federated learning and Knowledge Distillation-based federated learning.

Path planning efficiency

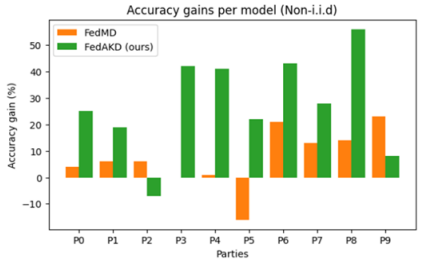
- Comparing the path generated by SOM with randomly generated paths.

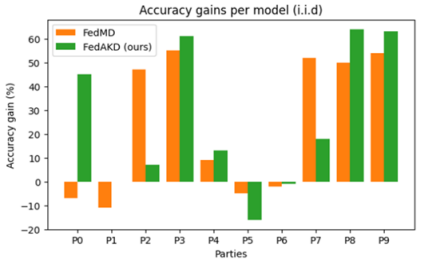
Knowledge Distillation

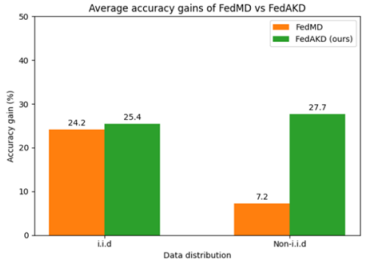
- The impact of Knowledge Distillation distance loss function on performance: Mean Squared Error vs Kullback Leibler divergence loss.

Performance

Comparing the accuracy achieved by two Knowledge Distillation-based FL algorithms in Independently and Identically Distributed (IID) vs Non-IID scenarios.







Communication cost

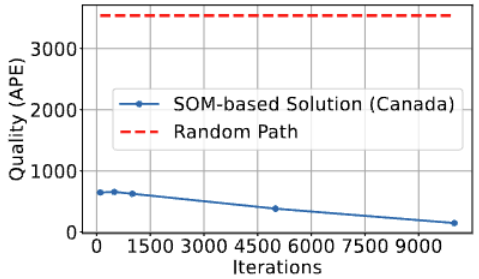
The Communication overhead of Knowledge Distillation-based FL methods is significantly less than that of model-based FL methods. This is because FedMD and FedAKD utilize soft labels to transfer knowledge across clients while FedAvg uses model weights which yields better performance but has higher communication overhead.

Communication overhead of FL methods

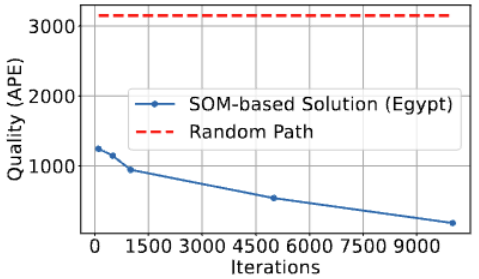
Federated learning algorithm	Communicated entities	Size
FedMD	$\|Z\|$	10 KB
FedAKD	$\|S\|$	10 KB
FedAvg	$\|\theta\|$	250 KB

Path planning

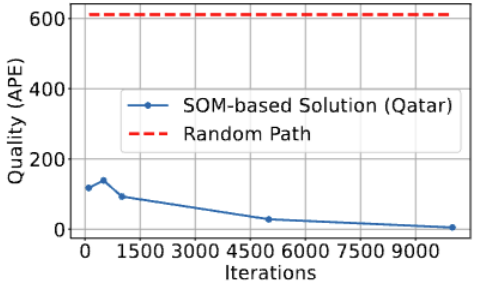
The absolute Percentage Error (APE) of our SOM-based solution compared with random path.



(a) APE for random path vs SOM path for Canada city map.



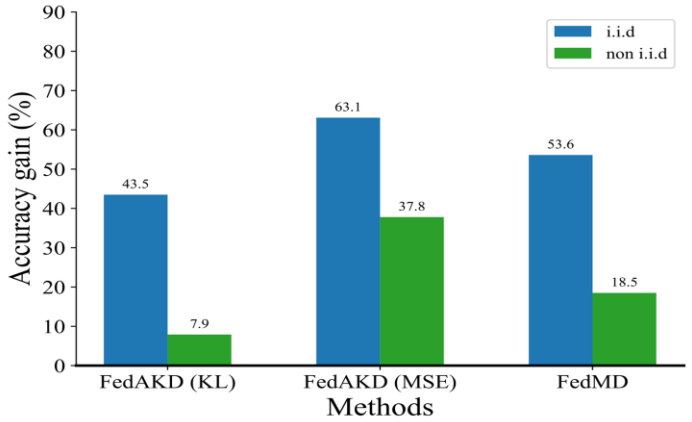
(b) APE for random path vs SOM path for Egypt city map.



(c) APE for random path vs SOM path for Qatar city map.

Comparing Knowledge Distillation losses

Using the Mean Squared Error (MSE) loss achieved better results than Kullback-Leibler (KL) divergence loss. Consistent with results reported by [3].



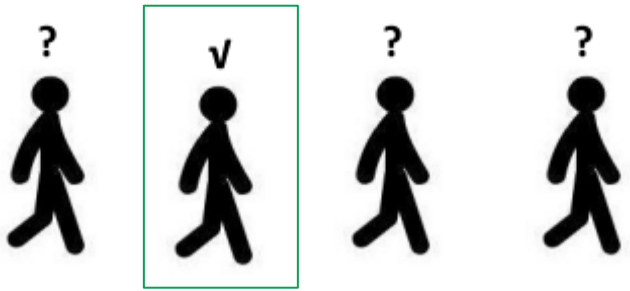
KD-based FL accuracy comparison using different loss functions MSE vs KL

Ex 3. WiFi-based Human identification

Problem statement

Using Channel State Information (CSI) for sensing purposes.

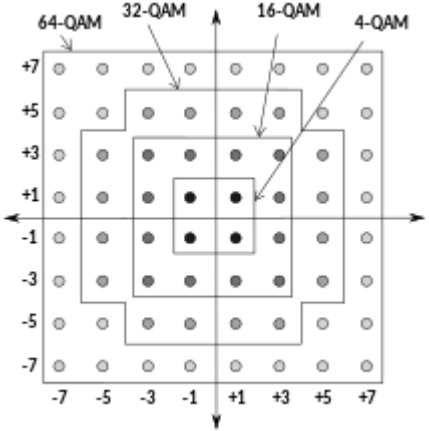
Given samples of CSI and corresponding labels of the name of a person standing between the transmitter and the receiver, predict the labels of new unseen CSI samples.



Wifi sensing

Orthogonal Frequency Domain Multiplexing (OFDM)

- OFDM is a modulation scheme used in Wifi 802.11n protocol to encode data streams into multiple subcarrier frequencies.
- Each OFDM symbol encodes binary data as a complex number. For instance, 16-QAM can represent 4 bits of data with a single complex number.

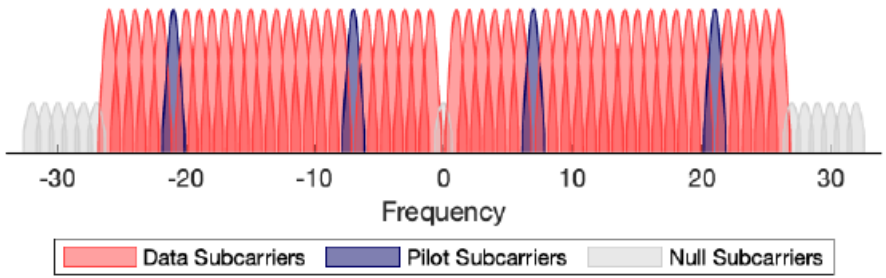


Constellation diagram

WiFi sensing

Frequency selective fading occurs due to the interference caused by obstacles in the environment. This interference does not affect all subcarriers in the same way.

Some subcarriers act as pilot subcarriers where the transmitted signal is known to both the transmitter and the receiver.
Through these pilot subcarriers, OFDM corrects for variations in the received signal through subcarrier equalization.



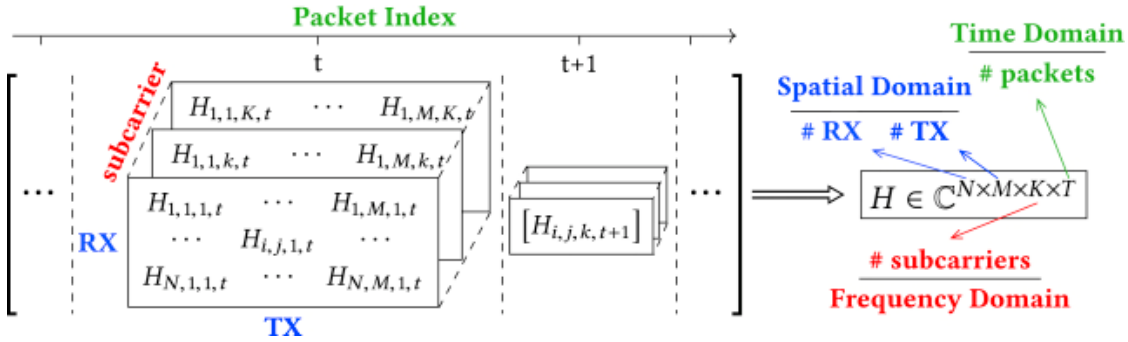
Data and pilot subcarriers in one wifi channel [1].

WiFi sensing

Channel State Information (CSI)

- CSI is the metric used in OFDM systems for describing amplitude and phase variations across subcarrier frequencies
- Channel estimation is the process used to detect variations across the subcarriers in OFDM systems through the transmission of a set of known shared pilot symbols.
- The CSI matrix (H) can be estimated as $y = Hx + \eta$

  , where x is a pilot symbol, η is additive white noise, y is the pilot symbol received at the receiver, and H is the CSI matrix.
- Each element in the CSI matrix h_i is a complex number representing the Channel Frequency Response (CFR) of a particular channel.

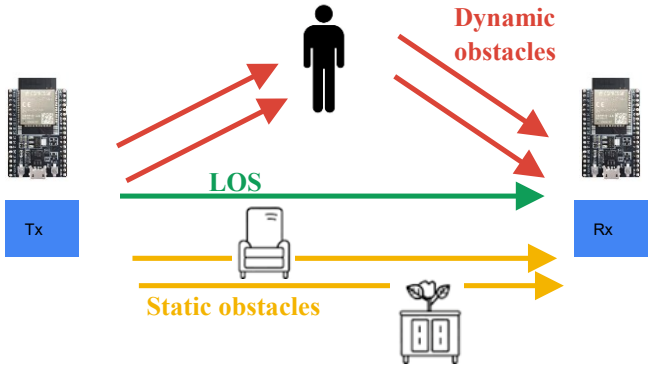


Structure of CSI [1].

WiFi sensing

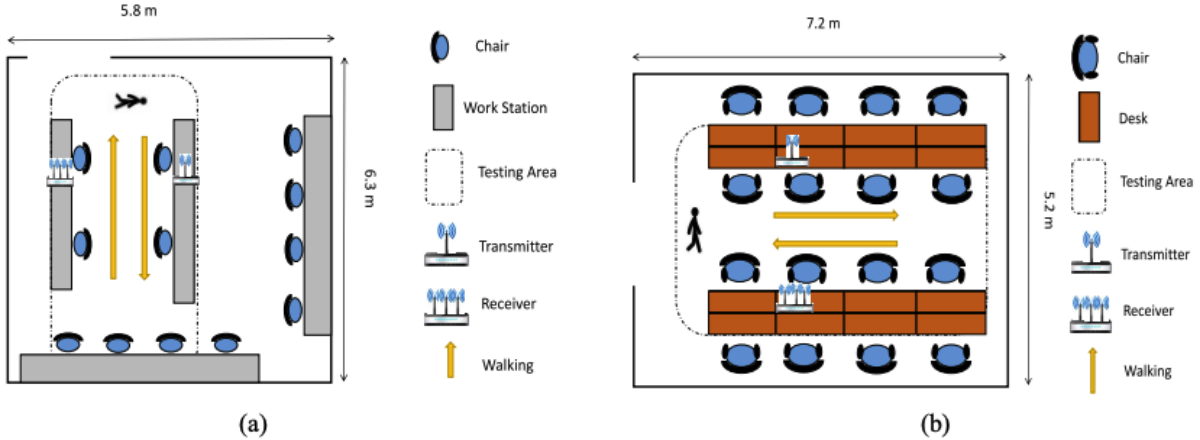
CSI is the main data used in wifi sensing

- CSI is extracted as a complex number $h_i = R(h_i) + jI(h_i) = A_i e^{j\phi_i}$
- Due to the multi-path nature of signal propagation, h_i can be represented as a summation of attenuations from N routes: $h_i = \sum_{m=1}^{N} A_m e^{\frac{-s\pi f_i d_m}{c} j\phi_m}$
- In the context of wifi sensing, we are interested in human activity recognition, so we further divide the routes that the signal takes into dynamic and static routes: $h_i = \sum_{n \in \Omega_d}^{N} A_n e^{\frac{-s\pi f_i d_n}{c} j\phi_n} + \sum_{m \in \Omega_s}^{N} A_m e^{\frac{-s\pi f_i d_m}{c} j\phi_m}$



Multi-path propagation of wifi signal through the environment.

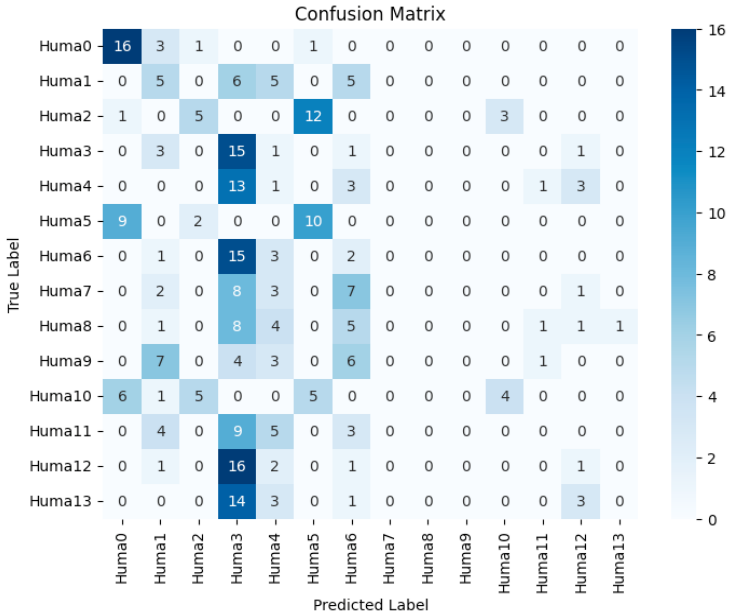
Public dataset

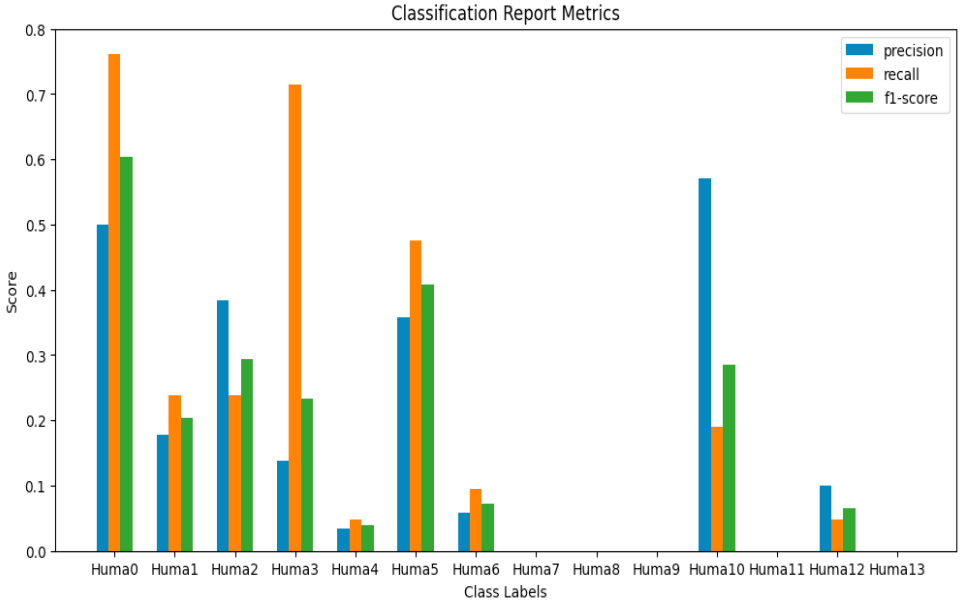


Layout of the experiment layout where CSI samples where collected with labels of person identify.

Data analysis

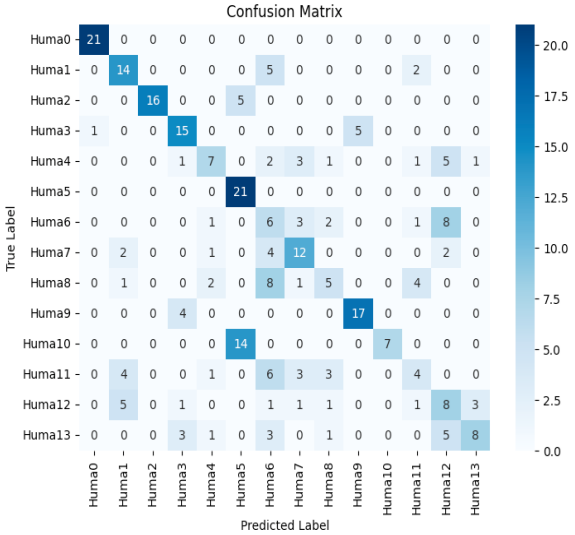
Usually local data of each users is not enough to train a powerful and representative deep learning model.

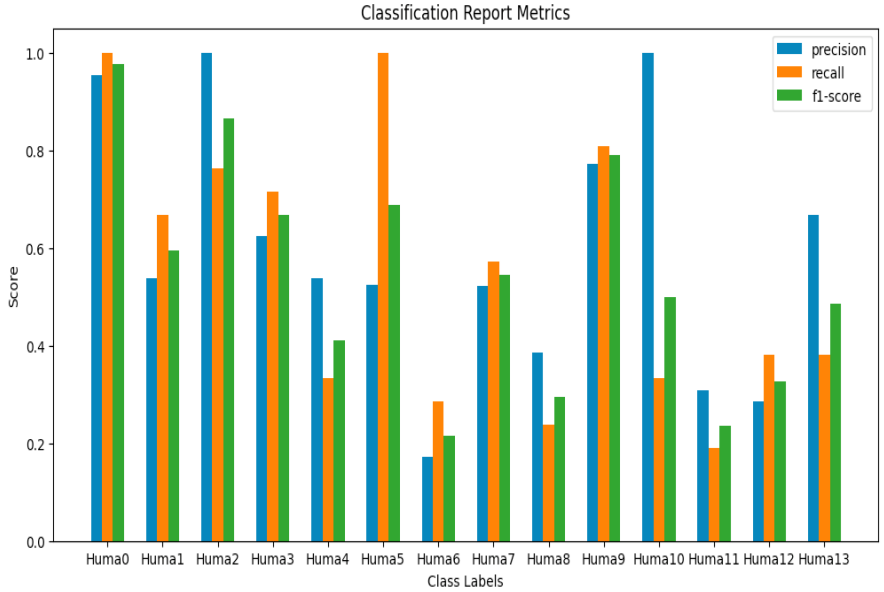




Central deep learning

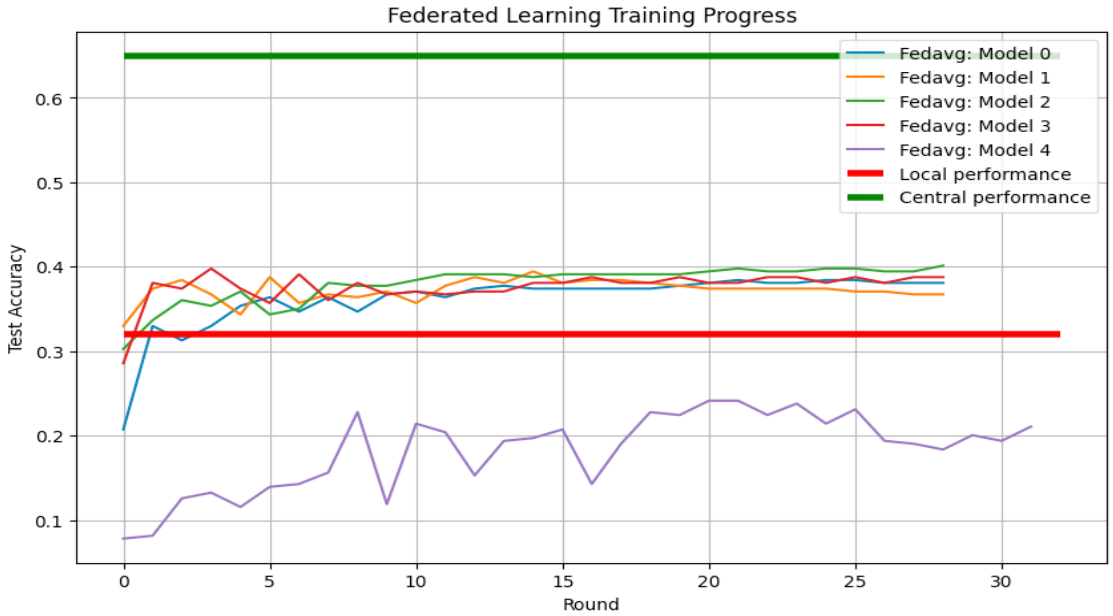
Sending local data from multiple users to the server for training a central model yields best performance, but poses privacy risks.





Federated learning

Federated Learning trains a global model collaboratively without sharing data. The performance is better than local training but is worse than central training. The main advantage of FL is preserving users privacy.



External References

1. Hernandez, S. M. (2023). WiFi Sensing at the Edge Towards Scalable On-Device Wireless Sensing Systems.
2. Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L. M. (2021, May). “Everyone wants to do the model work, not the data work”: Data Cascades in High-Stakes AI. In proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1-15).
3. Kim, T., Oh, J., Kim, N., Cho, S., & Yun, S. Y. (2021). Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. arXiv preprint arXiv:2105.08919.
4. Liu, M., Li, D., Chen, Q., Zhou, J., Meng, K., & Zhang, S. (2018). Sensor information retrieval from Internet of Things: Representation and indexing. IEEE Access, 6, 36509-36521.

Thank you