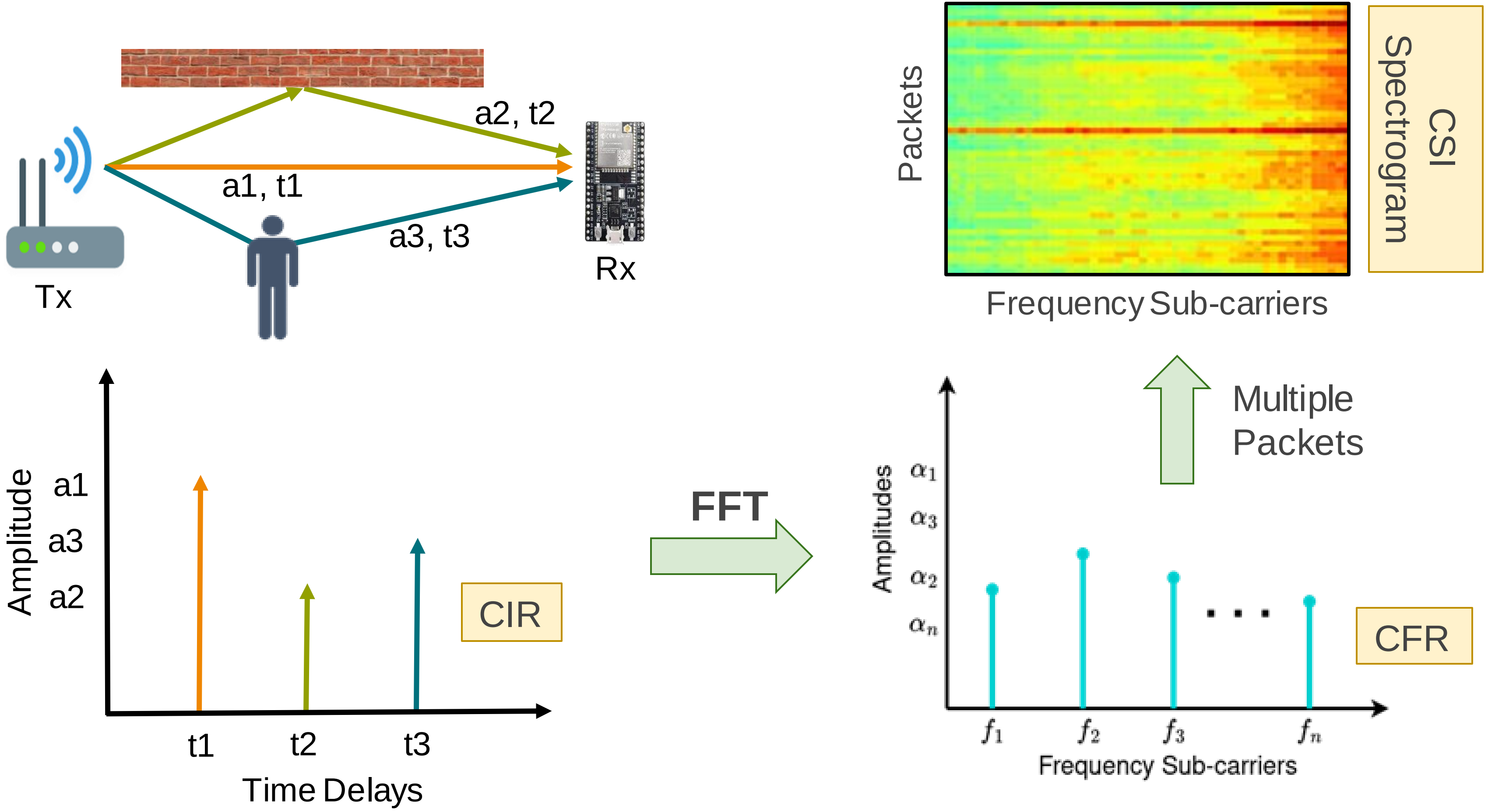
On-Device Deep Learning for IoT-based Wireless Sensing Applications

Manoj Lenka and Ayon Chakraborty

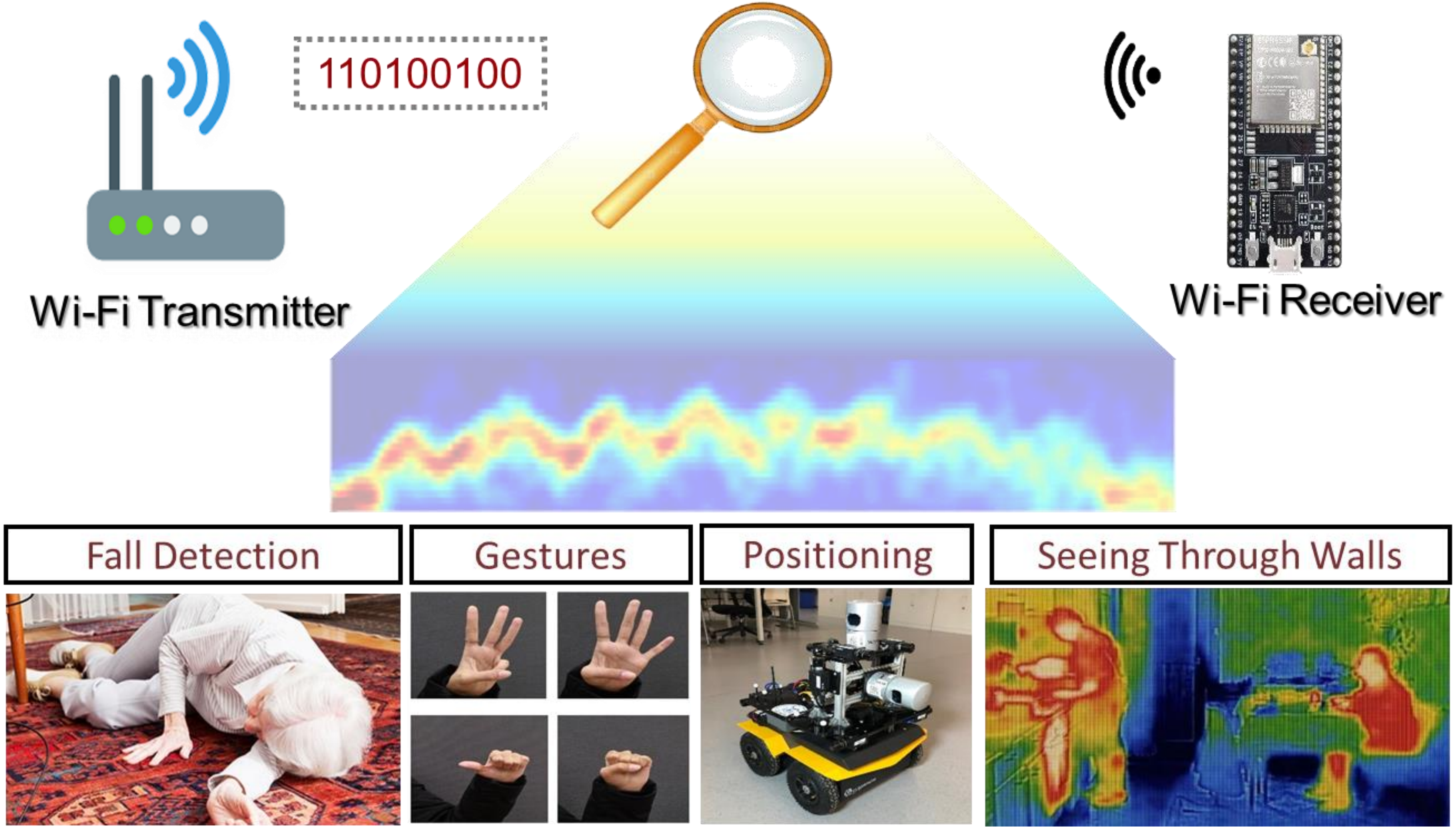
SENSE Lab
IIT Madras

Sensing and Networked Systems Engineering
@IIT Madras

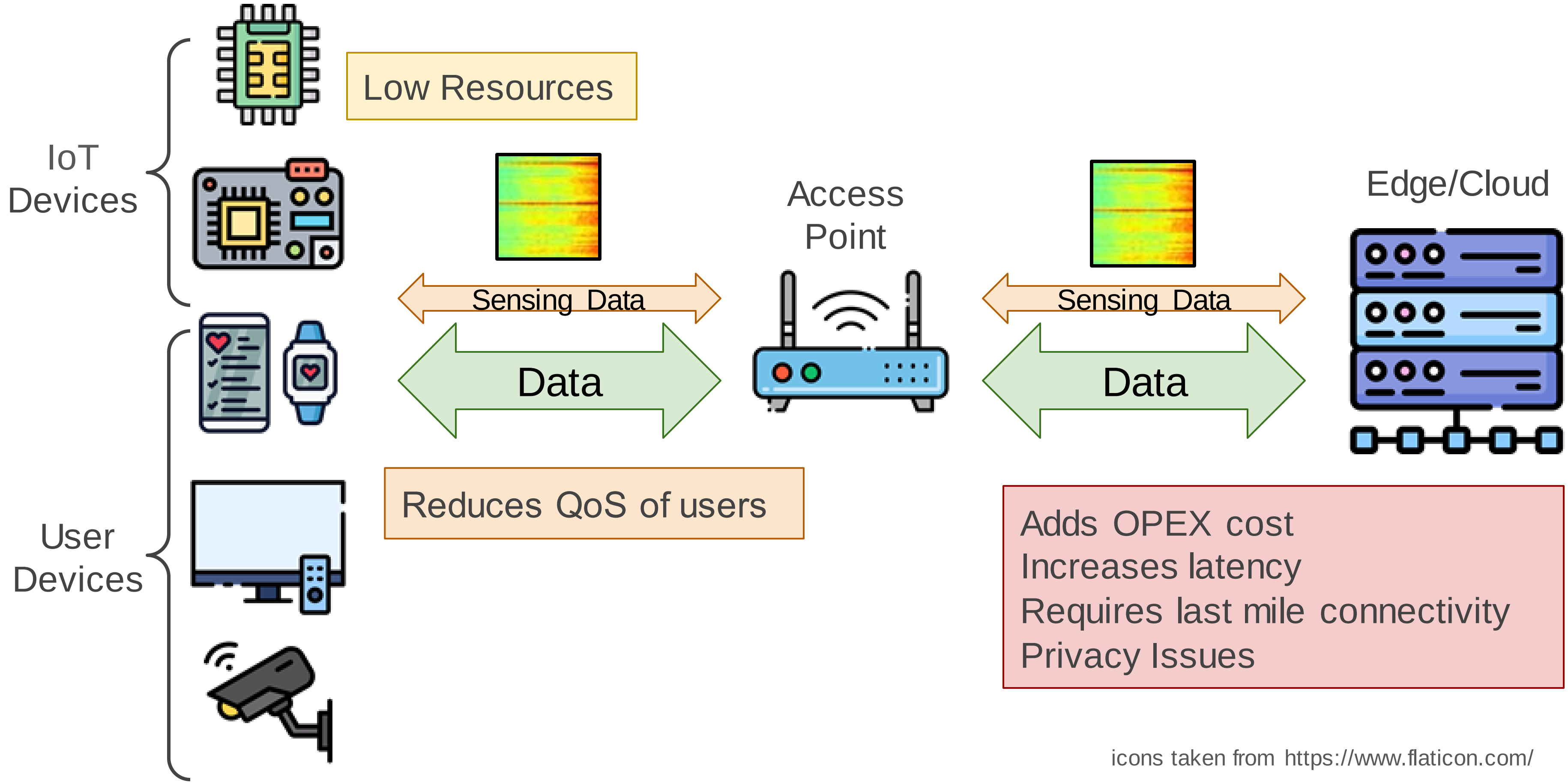
Wireless Sensing 101



CSI as a Sensing Primitive



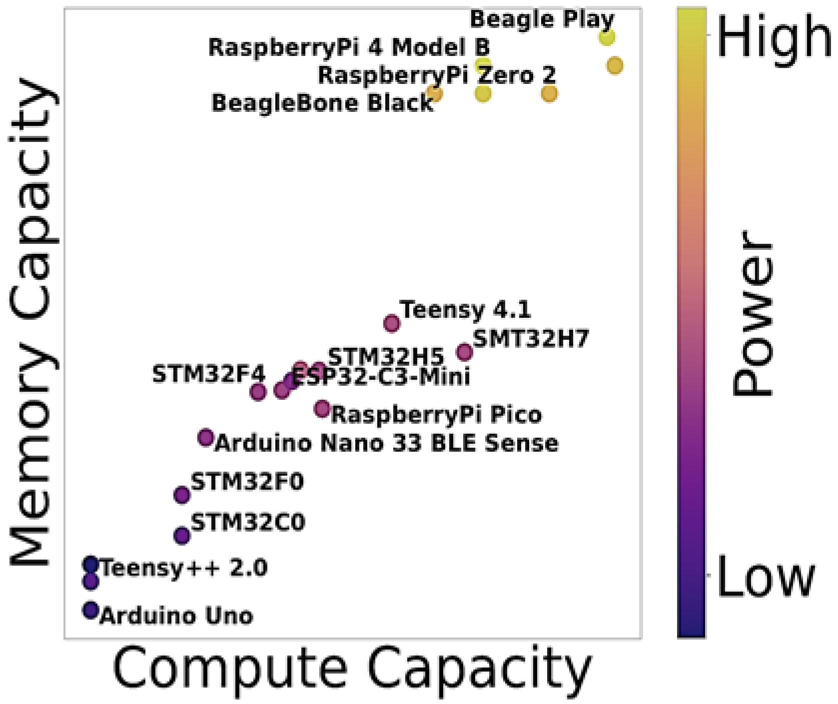
Why Not Edge Based Sensing?



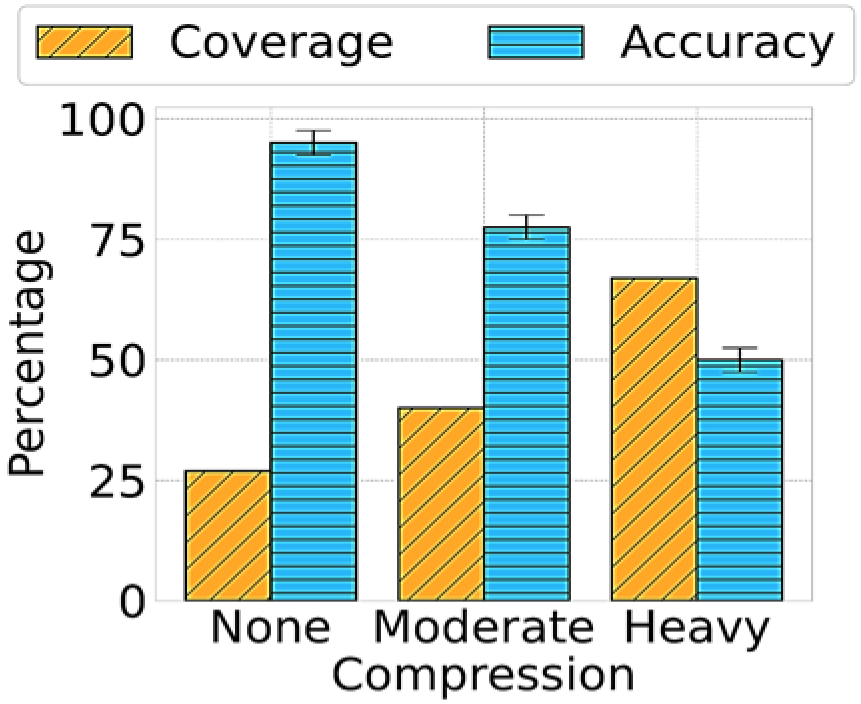
icons taken from https://www.flaticon.com/

Challenges of Inferencing on IoT Devices

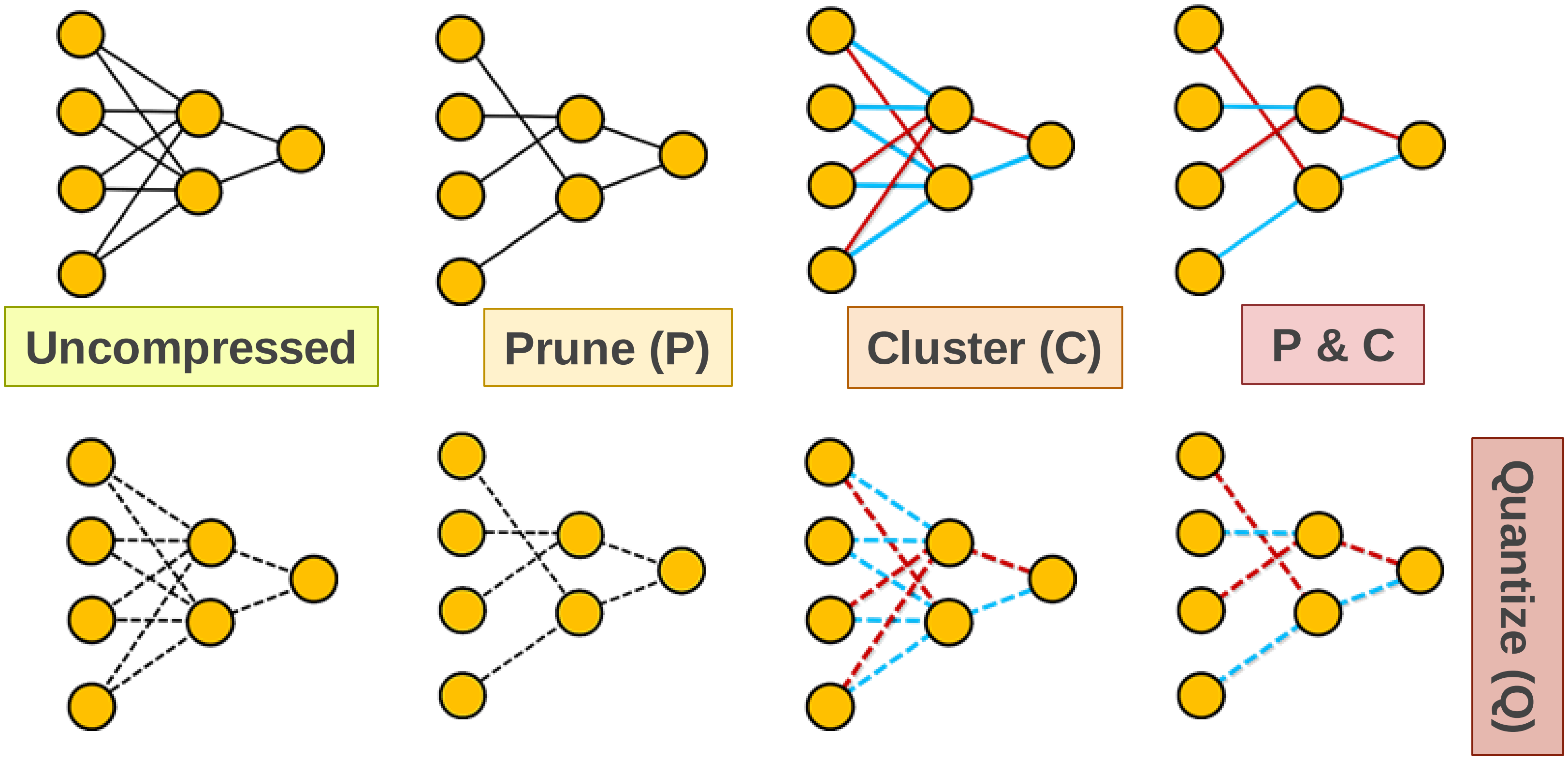
More Resources = More Energy



No one-size-fits-all solution



Compressing a Neural Network



Related Work and Research Gaps

Traditional Wi-Fi Sensing

Mainly focuses on improving performance and finding new and innovative applications. Less interest in actual system implementation

System Consideration for IoT

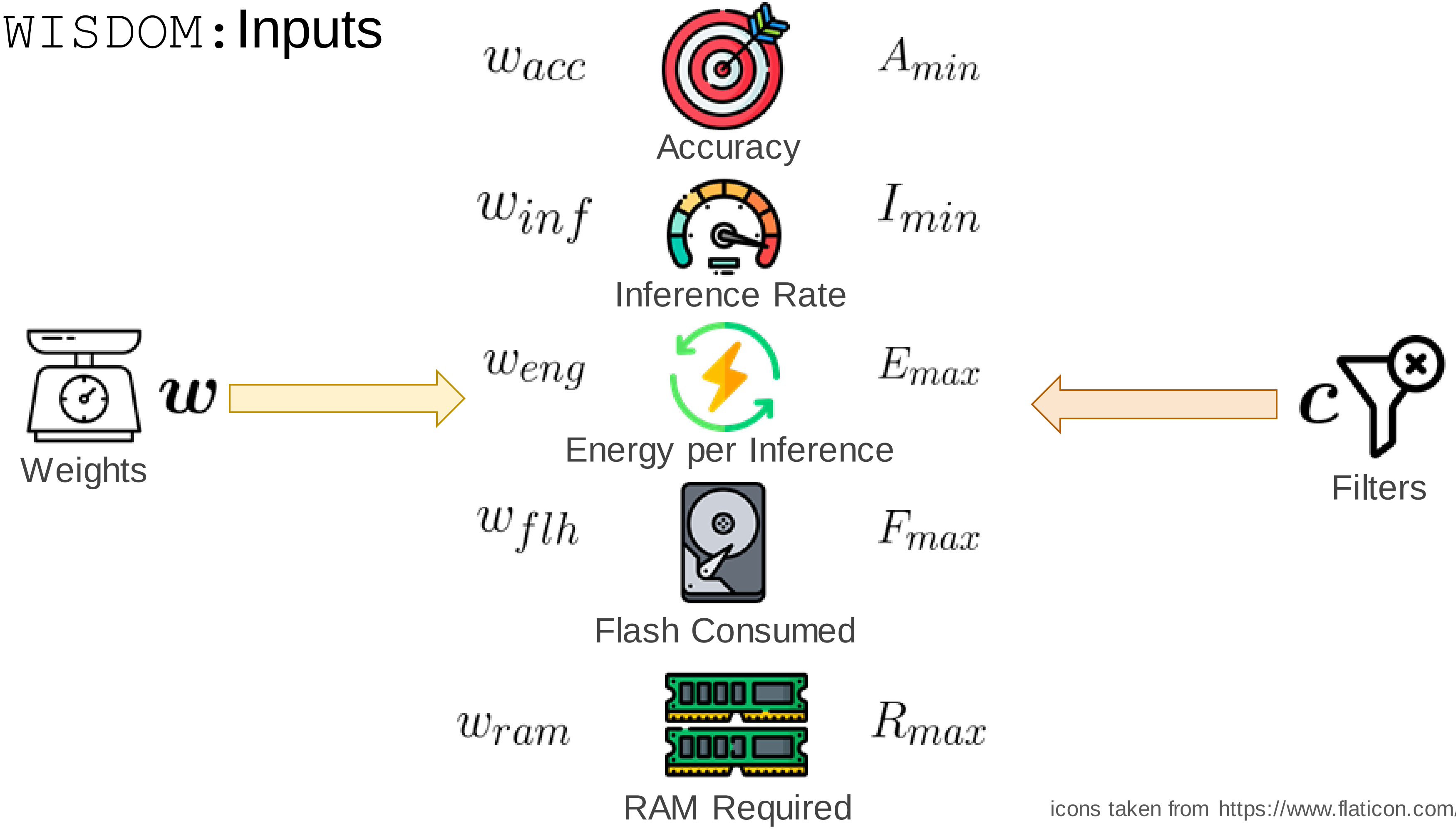
Some recent work do look into on-device Wi-Fi sensing on microcontrollers (like ESP-32) from a quantization perspective. Works like EfficientFi look into edge-based deployment.

TinyML Related Work

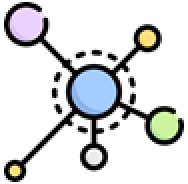
Has developed techniques like quantization, pruning, etc. Tools like TensorFlow Lite and Micro. Does not specifically focus on wireless sensing

Design a framework that provides a best-effort compressed neural network for a Wi-Fi sensing application such that the user can tune the trade off between performance and cost

WISDOM: Inputs



`WISDOM`: Outputs

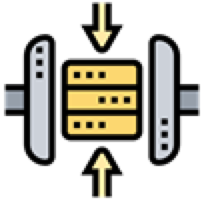


Architecture Type

$$\mathbf{T} = \{FCN, CNN, LSTM\}$$

Number of Parameters

$$\mathbf{N} = \{250, 1.5K, 3K, 6K, \ldots, 180K\}$$



Compression Techniques

$$\mathbf{O} = \{none, prune, cluster, qat, \ldots, pcptq\}$$

Neural Network Configuration

$$\mathbf{I} = \{[t, n, o] | t \in \mathbf{T}, n \in \mathbf{N}, o \in \mathbf{O}\}$$

Total of 324 models

icons taken from https://www.flaticon.com/

WISDOM: Utility Function

Utility | Performance | Cost

$$\mathcal{U}_{w,c} : \mathbf{I} \to \mathbb{R}$$

$$\mathcal{U} = \mathcal{P} - \mathcal{C}$$

Energy per Inference | RAM Required | Flash Consumed

$$\mathcal{C} = w_{eng}\mathcal{E} + w_{ram}\mathcal{R} + w_{flh}\mathcal{F} \quad where$$

$$\mathcal{E} \le E_{max} \qquad \mathcal{R} \le R_{max} \qquad \mathcal{F} \le F_{max}$$

Accuracy | Inference Rate

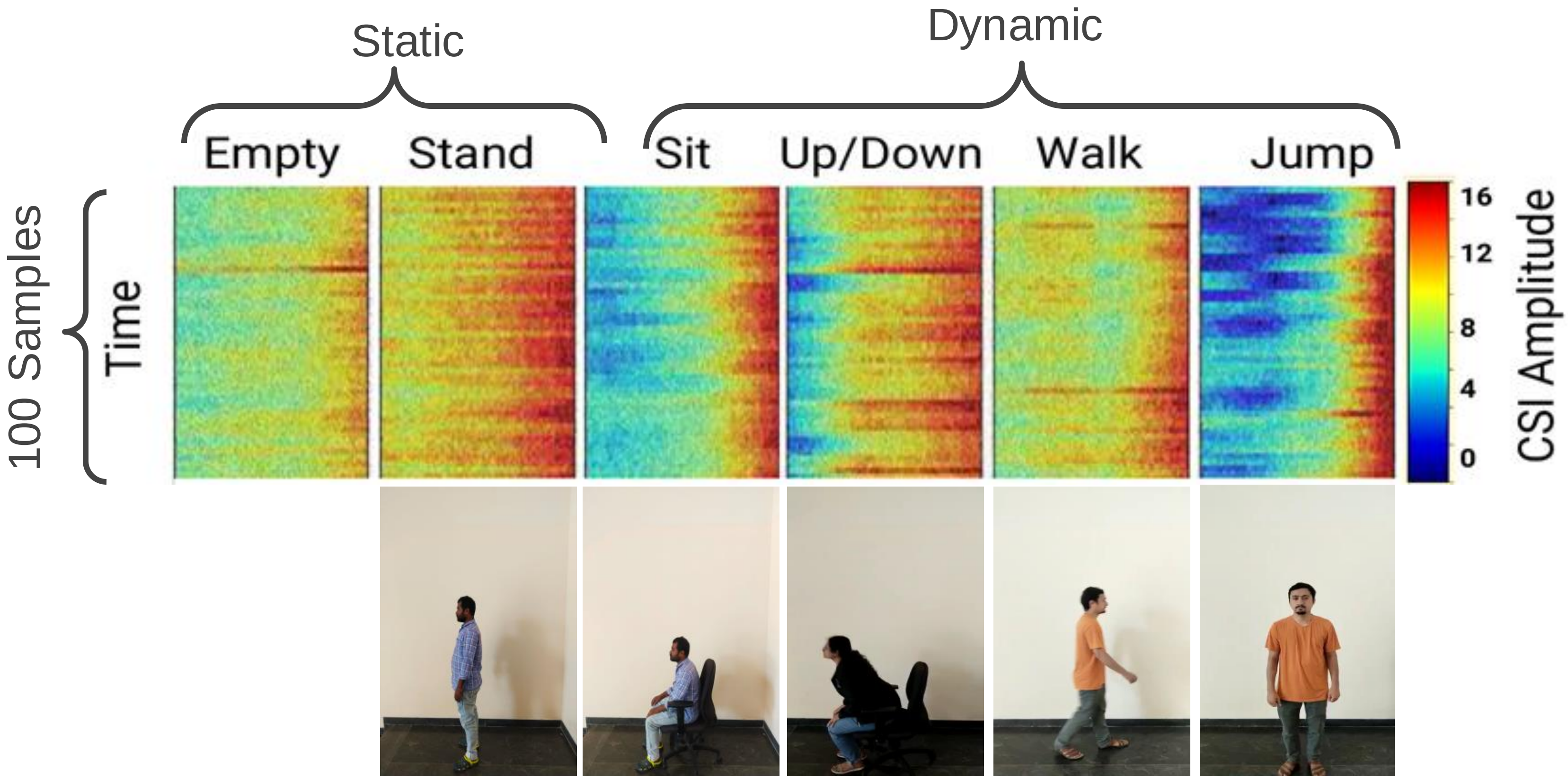
$$\mathcal{P} = w_{acc}\mathcal{A} + w_{inf}\mathcal{I} \quad where$$

$$\mathcal{A} \ge A_{min} \qquad \mathcal{I} \ge I_{min}$$

All metrics are normalized between 0 and 1 for a fair comparison

icons taken from https://www.flaticon.com/

Application Use Case: Human Activity Recognition



Dataset is open-sourced at: https://cse.iitm.ac.in/~sense/wisdom/

WISDOM: Optimization Problem

$$\mathcal{U}_{\boldsymbol{w},\boldsymbol{c}}(\texttt{Wisdom}(\boldsymbol{w},\boldsymbol{c})) \approx \mathcal{U}_{\boldsymbol{w},\boldsymbol{c}}(\boldsymbol{i}^{OPT})$$

$$\boldsymbol{i}^{OPT} = \underset{\boldsymbol{i}}{\operatorname{argmax}}\, \mathcal{U}_{\boldsymbol{w},\boldsymbol{c}}(\boldsymbol{i}) \ \forall\ \boldsymbol{i} \in \mathbf{I}$$

icons taken from https://www.flaticon.com/

WISDOM: Training

$$\mathcal{U}_{\boldsymbol{w},\boldsymbol{c}}\left(\texttt{Wisdom}\underbrace{(\boldsymbol{w},\boldsymbol{c})}_{\text{Samples}}\right) \approx \mathcal{U}_{\boldsymbol{w},\boldsymbol{c}}\underbrace{(\boldsymbol{i}^{OPT})}_{\text{Ground Truth}}$$

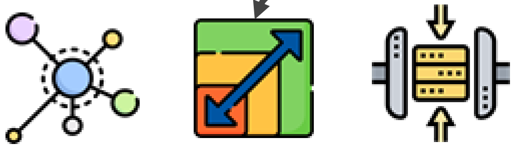
~27K

Generation Script

Decision Tree

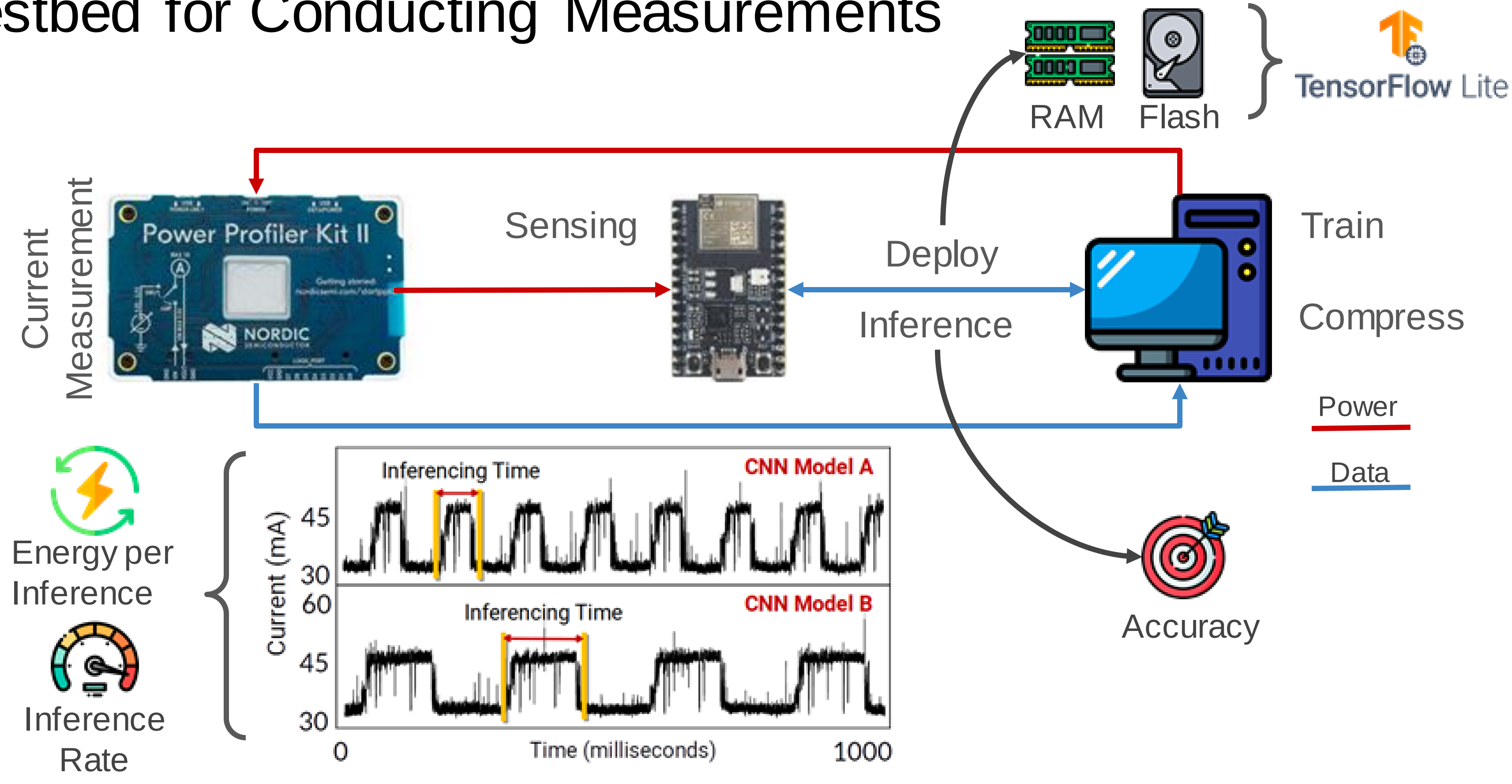
$$\boldsymbol{i}^{OPT} = \underset{\boldsymbol{i}}{\operatorname{argmax}} \, \mathcal{U}_{\boldsymbol{w},\boldsymbol{c}}(\boldsymbol{i}) \ \forall \ \boldsymbol{i} \in \mathbf{I}$$

Measure



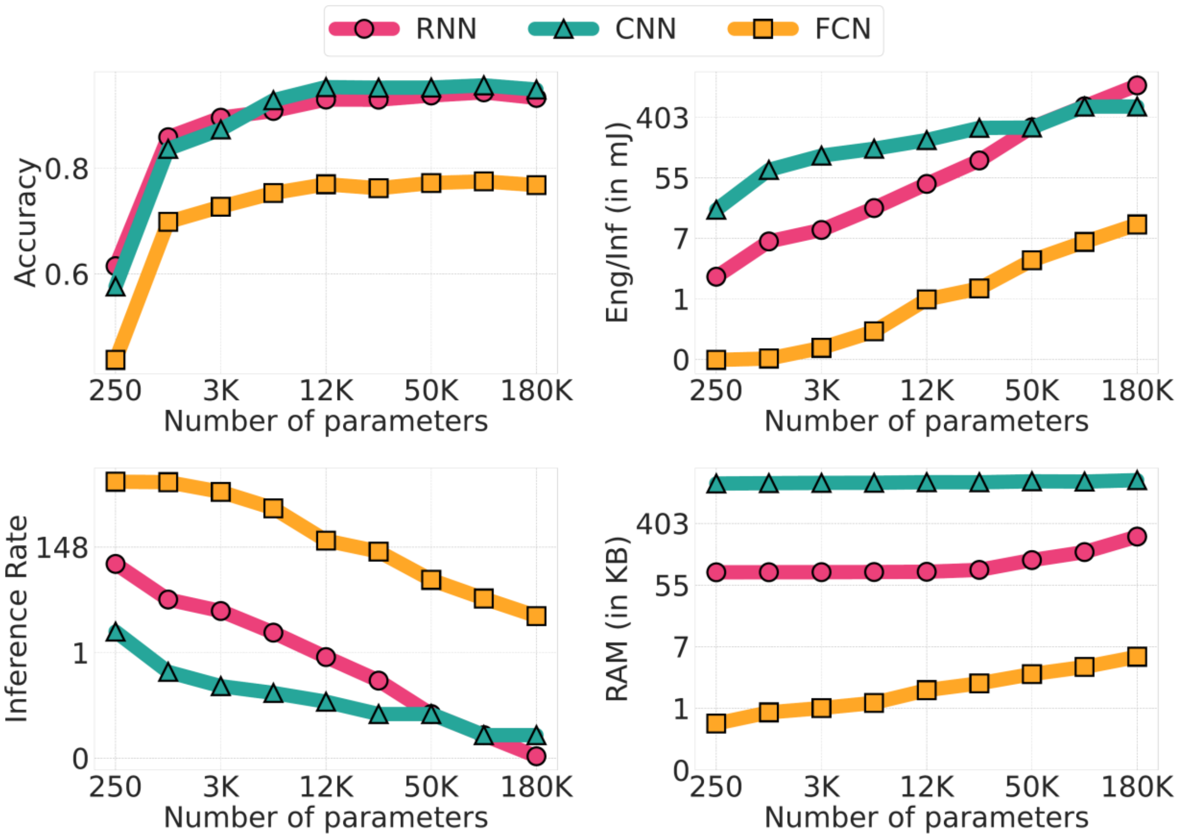
icons taken from https://www.flaticon.com/

Testbed for Conducting Measurements



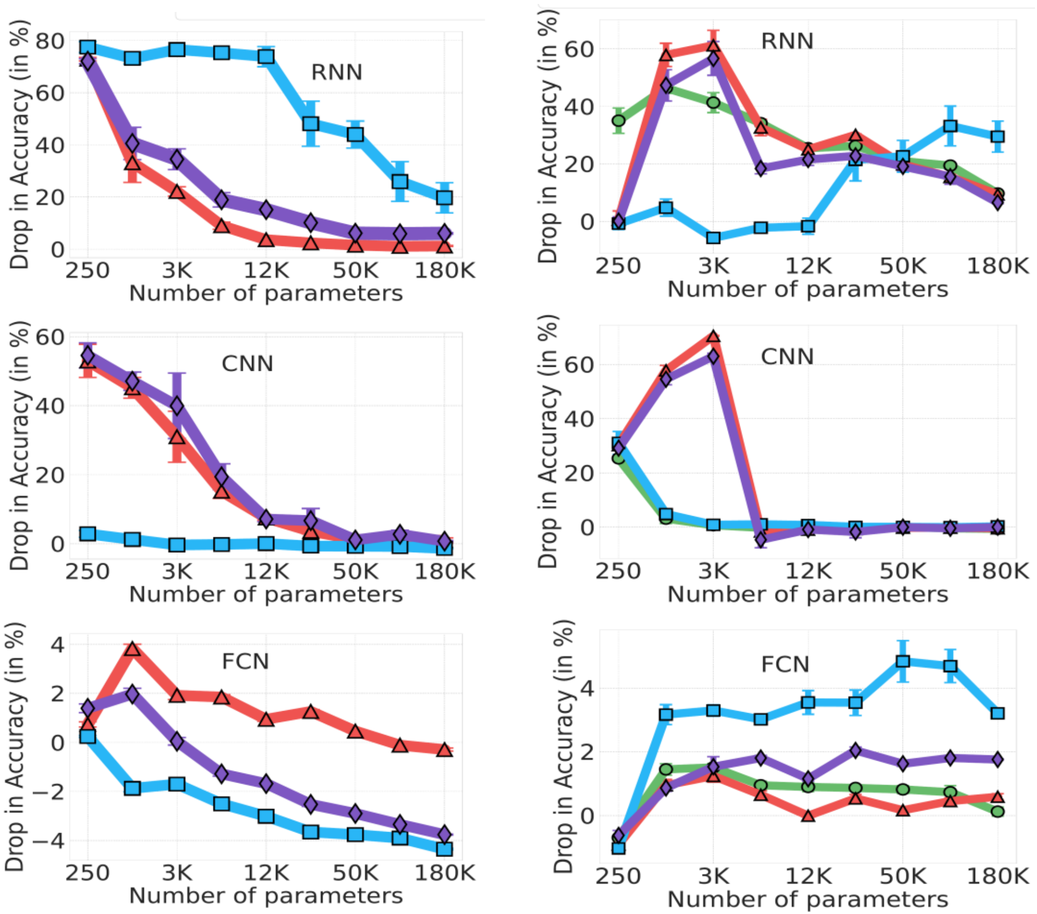
icons taken from https://www.flaticon.com/

Key Insights (1)



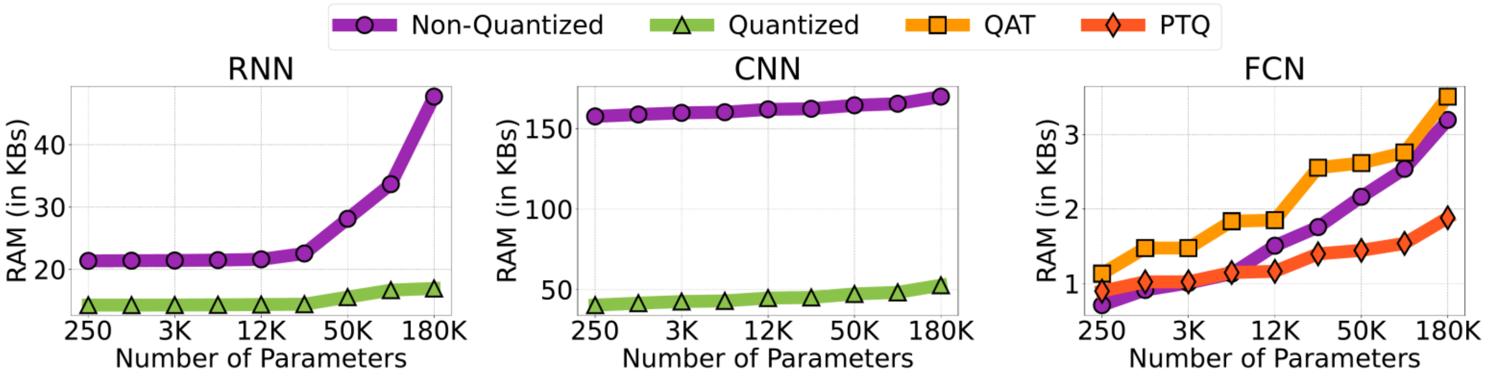
CNNs/RNNs are more accurate but also consume more resources compared to FCNs

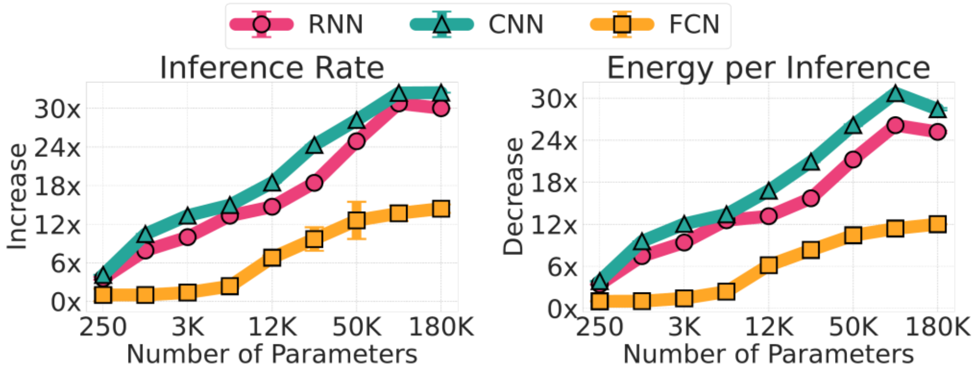
Key Insights (2)



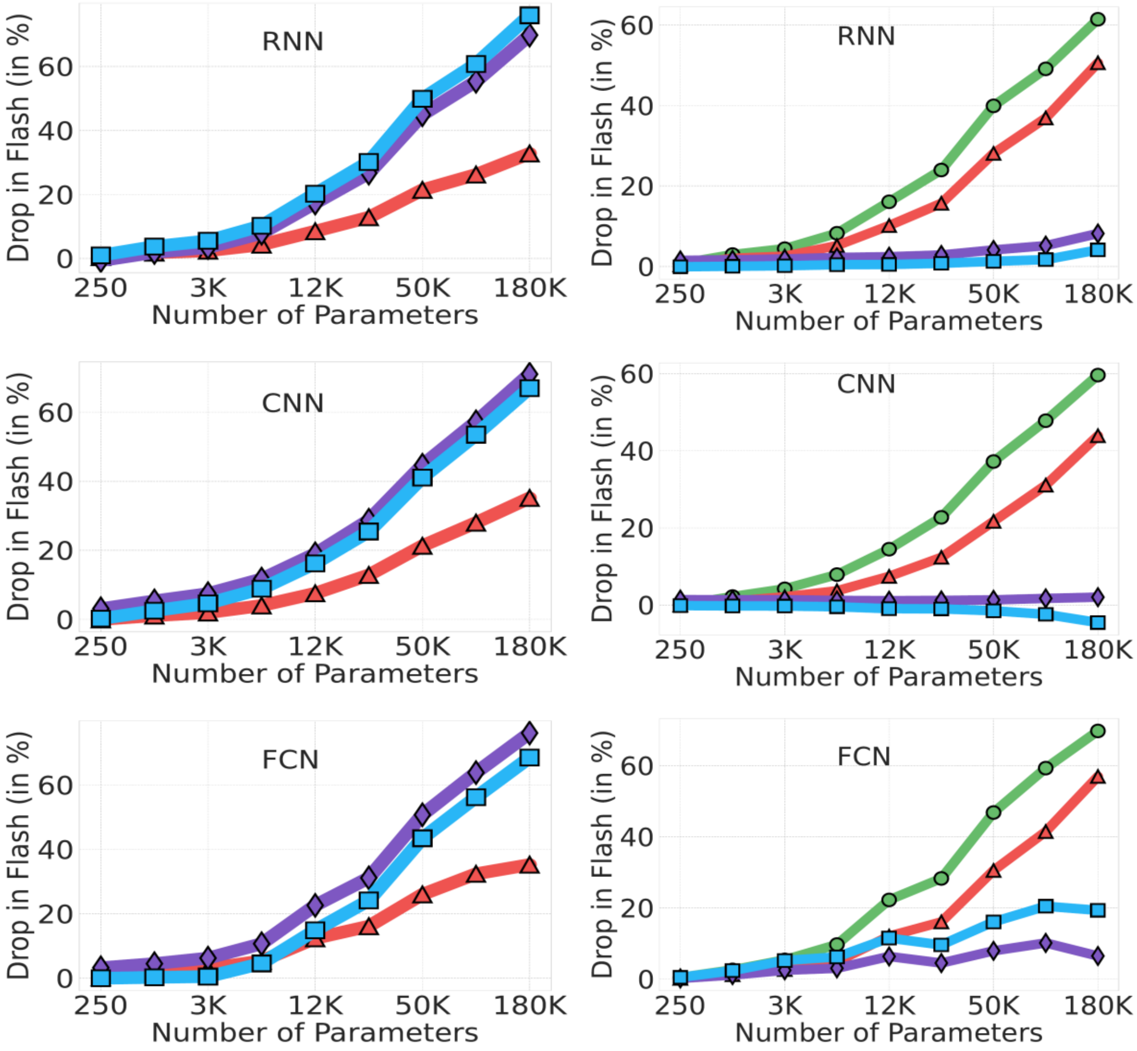
RNNs are more adversely affected by compression compared to CNNs/FCNs

Key Insights (3)

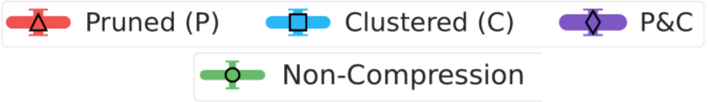




Quantization reduces the RAM and energy requirement, while increasing inferencing rate



Clustering provides significant reduction in flash, but quantization along with clustering is not reasonably effective



Key Insights (4)

Key Insights (5)

Compressing a model with higher number of parameters yields a more accurate model than an uncompressed model with lesser number of parameters

(while having a similar footprint in terms of energy and memory)

Baseline Models and Scenarios Used for Testing

There are additional 126 different test cases with different combination of weights

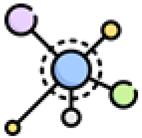
S1: More weight to accuracy
$w_{acc} = 0.9, w_{inf} = 0.1$
$w_{flh} = w_{ram} = w_{eng} = 0.3$

S2: Equal weight to accuracy and inference rate
$w_{acc} = w_{inf} = 0.5$
$w_{flh} = w_{ram} = w_{eng} = 0.3$

S3: More weight to flash usage
$w_{acc} = 0.7, w_{inf} = 0.3$
$w_{flh} = 0.8, w_{ram} = w_{eng} = 0.1$

S4: More weight to RAM and energy per infer
$w_{acc} = 0.7, w_{inf} = 0.3$
$w_{flh} = 0.1, w_{ram} = w_{eng} = 0.45$

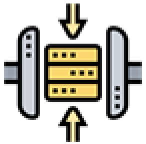
Scenarios


Type

RNN, CNN, FCN


Size

1.5K, 6K, 24K

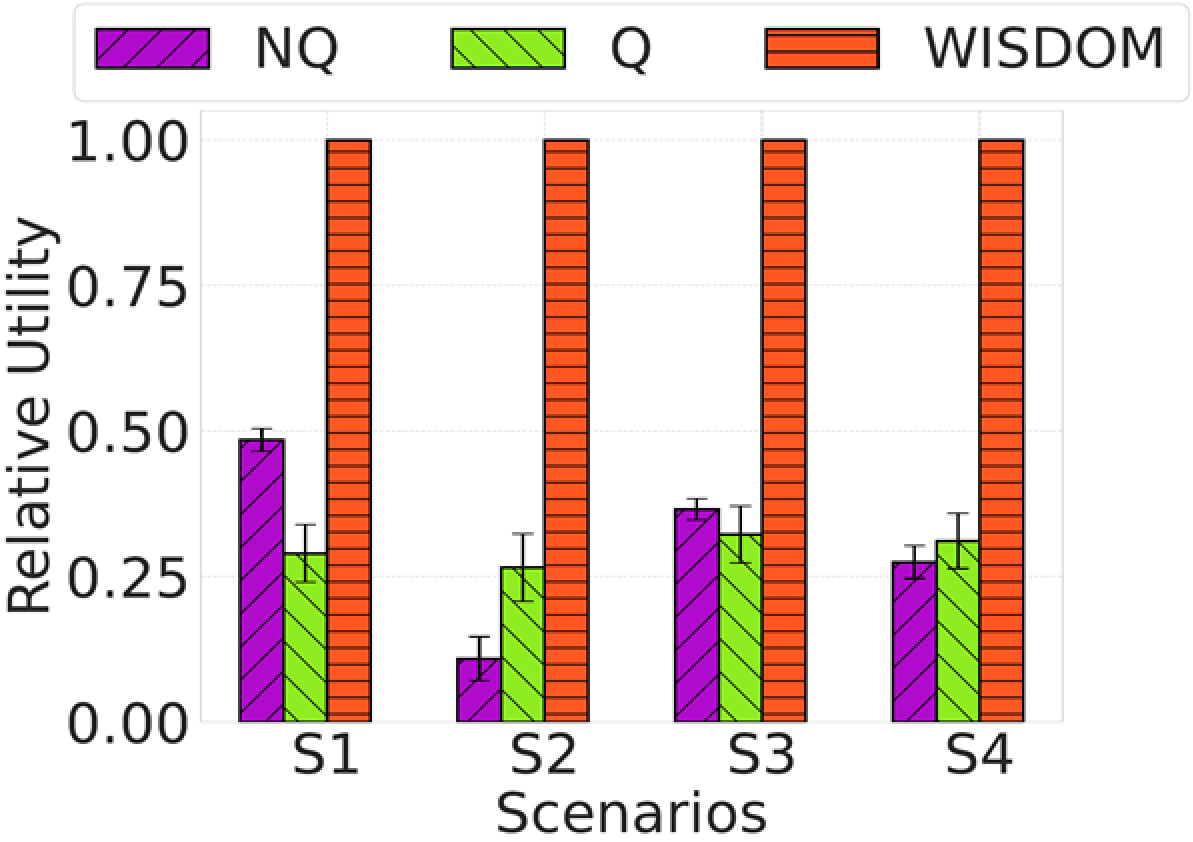

Compression

None (NQ), Quantized (Q)

18 baseline models (naive choice)

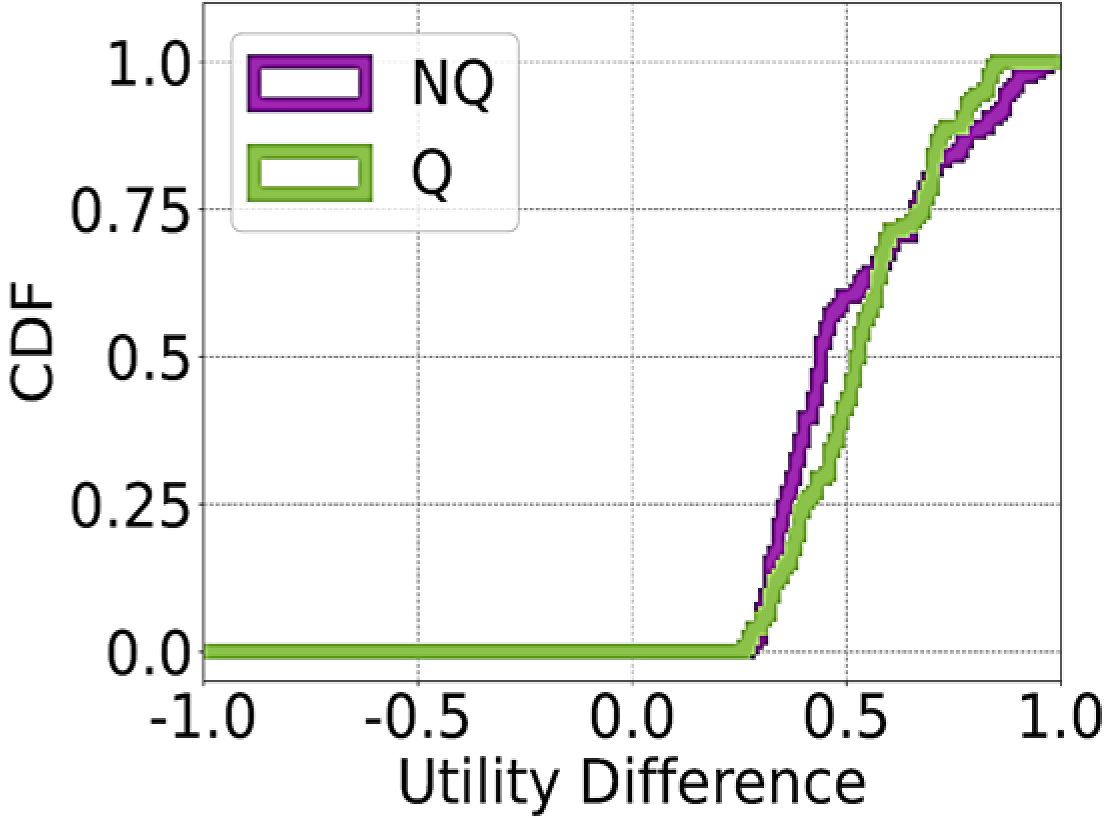
icons taken from https://www.flaticon.com/

Results: Models Chosen by `WISDOM` have Higher Utility



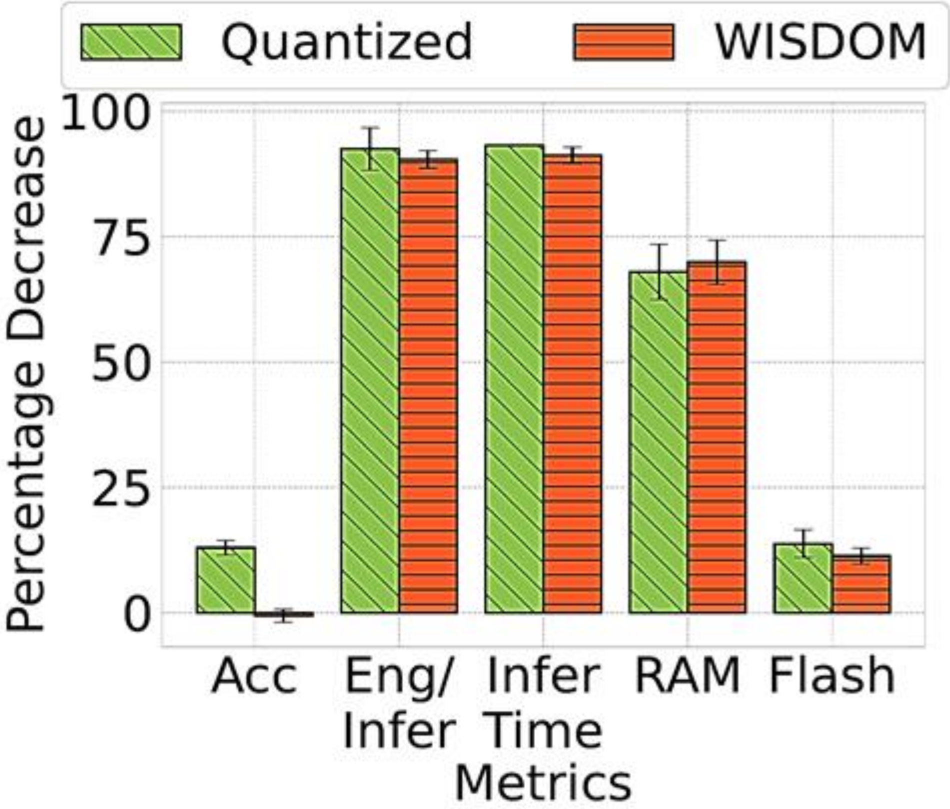
Relative utility of NQ and Q models are lower than `WISDOM` chosen models. Relative utility is w.r.t to the optimal model i.e., $\frac{\mathcal{U}(\boldsymbol{i})}{\mathcal{U}(\boldsymbol{i}^{OPT})}$

Results: Models Chosen by `WISDOM` have Higher Utility



The CDF of utility difference between Q or NQ model and WISDOM chosen model for all 126 test cases is always positive and starts increasing after 0.5

Results: Models Chosen by `WISDOM` Uses Less Resources While Maintaining High Accuracy



`WISDOM` chosen models show a percentage decrease similar to Q models for resource consumption, but still maintains higher accuracy of ~15% compared to Q models. The percentage decrease is w.r.t NQ models

`WISDOM` chosen model outperforms the best quantized model 83% of time, and the best non-compressed model 99% of time

Thank You, Questions?

Sensing and Networked Systems Engineering
@IIT Madras

Manoj Lenka, lenka98.github.io and **Ayon Chakraborty**, cse.iitm.ac.in/~ayon

Contact: cs22s008@cse.iitm.ac.in, ayon@cse.iitm.ac.in

Artifacts available at: https://cse.iitm.ac.in/~sense/wisdom/