



LUND
UNIVERSITY

LTH

FACULTY OF
ENGINEERING

EITP40

Final Project Presentation

HARRY, JIA, TIANCI



HUMAN ACTIVITY CLASSIFICATION

OBJECTIVE:

LABEL	CATEGORY
0	PASSIVE
1	A BIT ACTIVE
2	VERY ACTIVE

We strived to achieve as good as possible classification metrics:

- Accuracy
- Precision
- Recall
- F1

FEATURE COLLECTION - IMU DATA

Sliding Window Size: 128 samples

Feature Extraction:

Sensor	Axes	Features
Accelerometer	X, Y, Z	mean + std
Gyroscope	X, Y, Z	mean + std
Magnetometer	X, Y, Z	mean + std

Total features per window: 18

Normalization: Uses Exponential Moving Average (EMA)

- EMA smoothing factor: 0.005
- Normalization: $\tilde{x}_i = x_i - \text{EMA}_i$

ARCHITECTURE

LIGHTWEIGHT RESERVOIR NETWORK - A NO LIB APPROACH

```
10
11 struct ESN {
12     float reservoir[RESERVOIR_SIZE];
13     float W_in[RESERVOIR_SIZE][INPUT_SIZE];    // Input weights
14     float W_res[RESERVOIR_SIZE][RESERVOIR_SIZE]; // Reservoir recurrent weights
15     float W_out[OUTPUT_SIZE][RESERVOIR_SIZE];  // Output weights (trainable)
16 };
17
```

Problem:

Each window of features is not independent from the last - a sequence of “passive” is likely followed by “passive” irrespective of the current window’s feature composition. The model needs to capture this crucial observation

Solution:

Memoryless output layered trained on a stateful reservoir of values updated through non-linear activations - We achieve RNN capabilities without the need for BPTT

ARCHITECTURE

LIGHTWEIGHT RESERVOIR NETWORK - A NO LIB APPROACH

STATE UPDATE:

$$x_i(t+1) = (1 - \alpha) x_i(t) + \alpha \tanh \left(\sum_{j=1}^{N_u} W_{ij}^{in} u_j(t) + \sum_{j=1}^{N_r} W_{ij}^{res} x_j(t) \right)$$

in: random +/- 1

recurrent: random +/- 0.5

```
69  for (uint8_t i = 0; i < RESERVOIR_SIZE; i++) {
70      float sum = 0.0f;
71
72      // Input contribution
73      for (uint8_t j = 0; j < INPUT_SIZE; j++) {
74          sum += esn.W_in[i][j] * fv.features[j];
75      }
76
77      // Reservoir recurrent contribution
78      for (uint8_t j = 0; j < RESERVOIR_SIZE; j++) {
79          sum += esn.W_res[i][j] * esn.reservoir[j];
80      }
81
82      // Apply tanh activation
83      new_state[i] = (1 - LEAKY) * esn.reservoir[i] + LEAKY * tanh(sum);
84  }
```


ARCHITECTURE

LIGHTWEIGHT RESERVOIR NETWORK - A NO LIB APPROACH

TRAINING OUTPUT LAYER ON A GIVEN STATE:

```
98 // Compute output (linear)
99 float out[OUTPUT_SIZE];
100 for (uint8_t i = 0; i < OUTPUT_SIZE; i++) {
101     out[i] = 0.0f;
102     for (uint8_t j = 0; j < RESERVOIR_SIZE; j++) {
103         out[i] += esn.W_out[i][j] * esn.reservoir[j];
104     }
105 }
106
107 // Softmax
108 float max_val = out[0];
109 for (uint8_t i = 1; i < OUTPUT_SIZE; i++)
110     if (out[i] > max_val)
111         max_val = out[i];
112 float sum_exp = 0.0f;
113 for (uint8_t i = 0; i < OUTPUT_SIZE; i++) {
114     out[i] = exp(out[i] - max_val);
115     sum_exp += out[i];
116 }
117 for (uint8_t i = 0; i < OUTPUT_SIZE; i++)
118     out[i] /= sum_exp;
119
120 // Compute error (one-hot)
121 for (uint8_t i = 0; i < OUTPUT_SIZE; i++) {
122     float target = (i == y[s]) ? 1.0f : 0.0f;
123     float error = target - out[i];
124
125     // Gradient descent update for W_out
126     for (uint8_t j = 0; j < RESERVOIR_SIZE; j++) {
127         esn.W_out[i][j] += learning_rate * error * esn.reservoir[j];
128     }
129 }
```

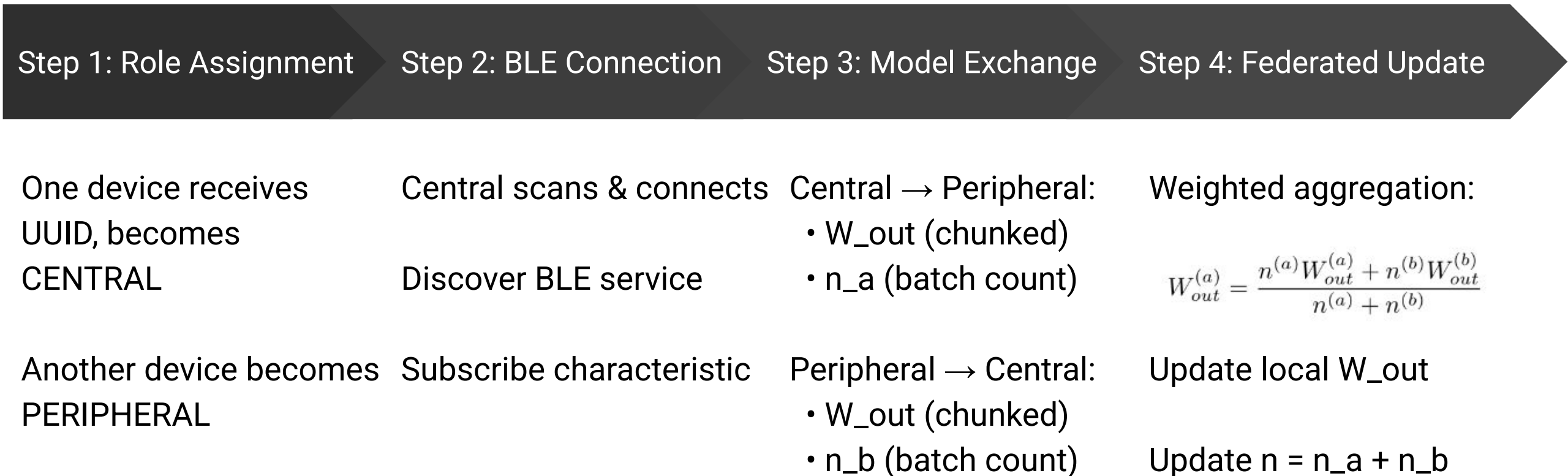
$$\hat{p} = \text{softmax}(W_{out}x)$$
$$W_{out} := W_{out} + \eta (y - \hat{p}) x^\top$$

prediction is just argmax of the first line with or without the softmax applied

ARCHITECTURE

LIGHTWEIGHT RESERVOIR NETWORK - A NO LIB APPROACH

FEDERATED LEARNING:



Evaluation and Results

Hyperparameter setups:

- RESERVOIR_SIZE = 20
- LEARNING_RATE = 0.01
- WINDOW_SIZE = 128
- BATCH_SIZE = 8
- EMA_ALPHA = 0.005

Evaluation matrices:

$$\text{Accuracy} = \frac{\sum_{i=1}^N TP_i}{\text{Total number of samples}}$$

$$\text{Recall}_{macro} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FN_i}$$

$$\text{Precision}_{macro} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision}_{macro} \times \text{Recall}_{macro}}{\text{Precision}_{macro} + \text{Recall}_{macro}}$$

Evaluation and Results - before federate

Device 1: trained on 26 batches

Accuracy = 79.17% Precision = 87.18%

Recall = 79.17% F1-Score = 82.98%

Confusion matrix:

<div>Predict Actual</div>	Labe 0	Label 1	Label 2
Label 0	6	10	0
Label 1	0	16	0
Label 2	0	0	16

Device 2: trained on 24 batches

Accuracy = 95.83% Precision = 96.30%


Recall = 95.83% F1-Score = 96.06%

Confusion matrix:

<div>Predict Actual</div>	Labe 0	Label 1	Label 2
Label 0	16	0	0
Label 1	2	14	0
Label 2	0	0	16

Evaluation and Results - after federate


Device 1: 26 + 24 = 50 batches

 Accuracy = 95.83% Precision = 96.30%
Recall = 95.83% F1-Score = 96.06%

Confusion matrix:

<div>Predict Actual</div>	Labe 0	Label 1	Label 2
Label 0	16	0	0
Label 1	2	14	0
Label 2	0	0	16

Device 2: 24 + 26 = 50 batches

 Accuracy = 100.00% Precision = 100.00%
Recall = 100.00% F1-Score = 100.00%

Confusion matrix:

<div>Predict Actual</div>	Labe 0	Label 1	Label 2
Label 0	16	0	0
Label 1	0	16	0
Label 2	0	0	16

Highlights

- ESN model selection
- EMA normalization method
- Online on-device training that ensures privacy
- IO interface for both BLE and Serial communications
- Interaction design: button and LEDs
- On-device Federated Learning without Cloud
- Decentralized: roles (central/peripheral) are assigned dynamically
-

Thank you!



LUND
UNIVERSITY

LTH

**FACULTY OF
ENGINEERING**