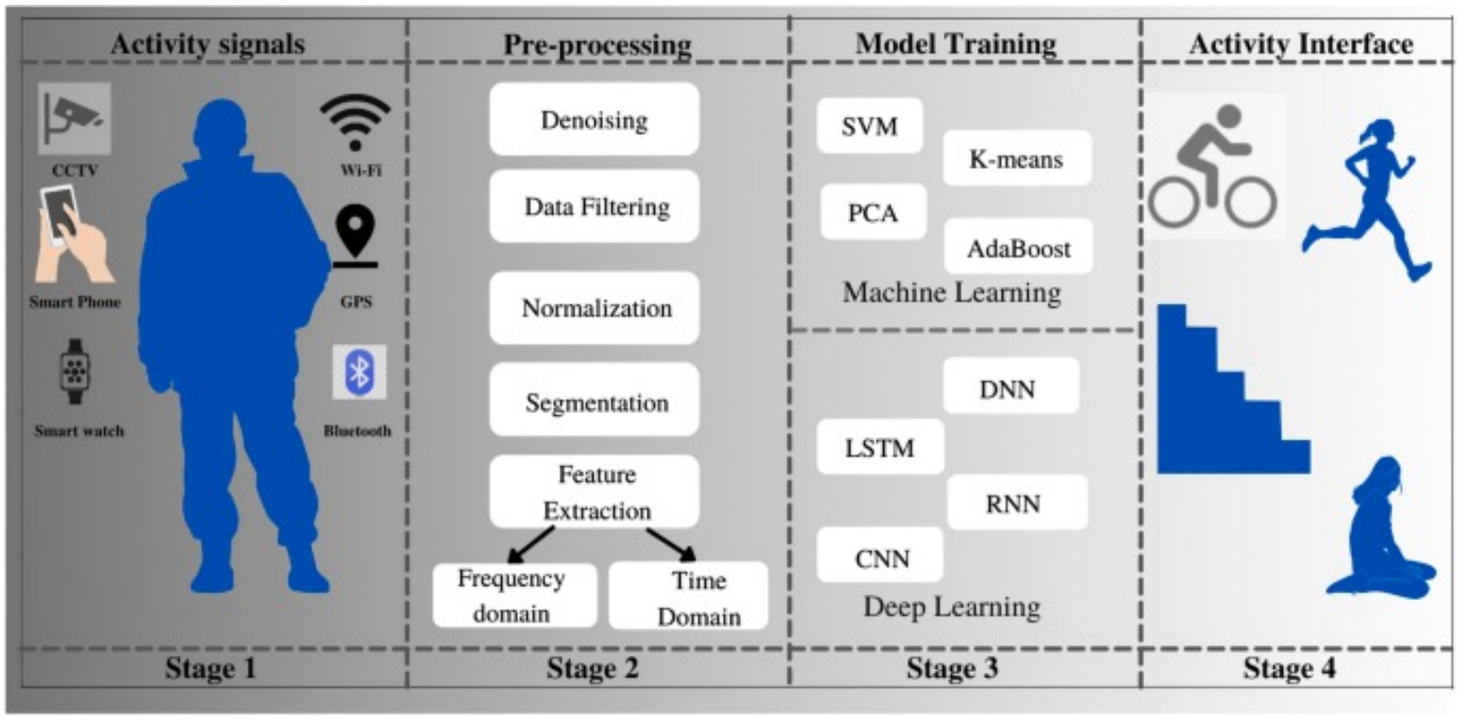
The 1st CUK Annual Colloquium on Artificial Intelligence; CUK AI Colloquium 2022

Human Activity Recognition System using Semi-supervised Learning and Incremental Learning

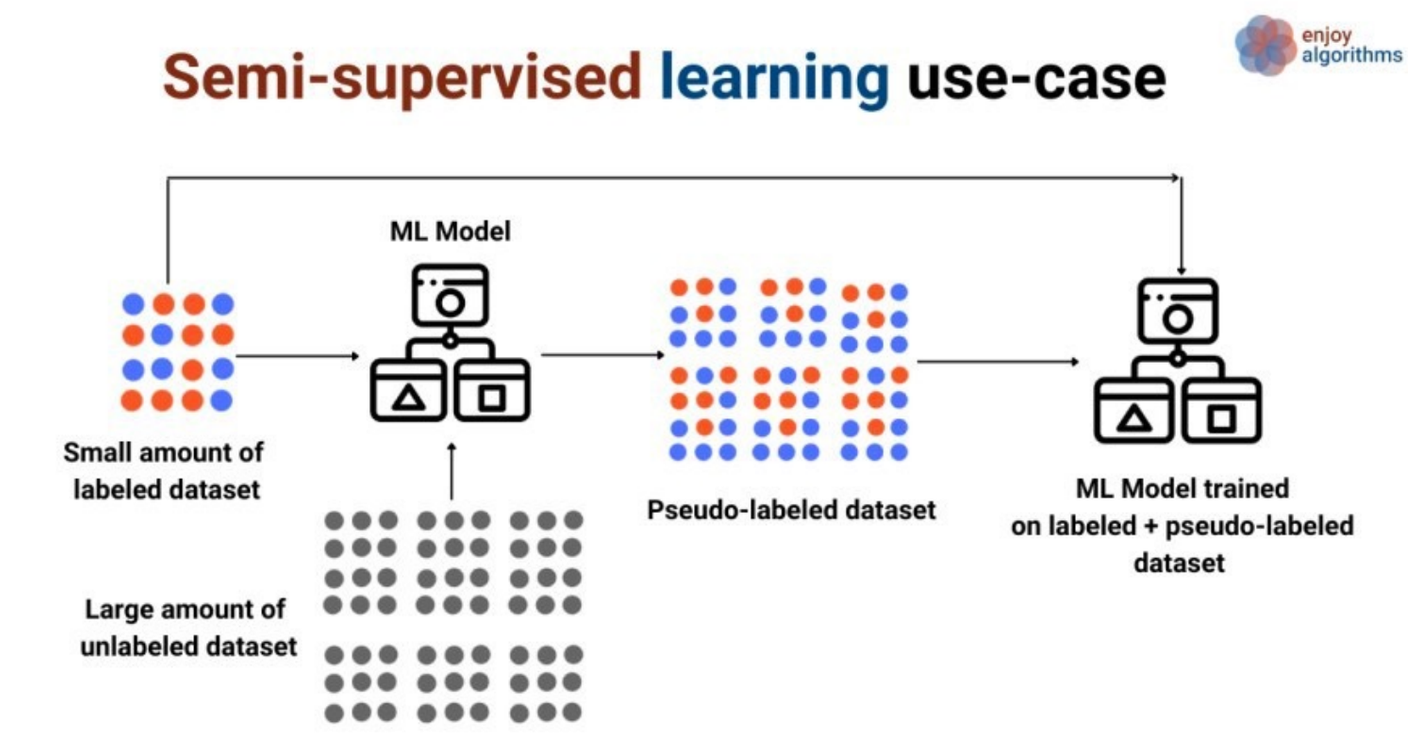
Seol Jeon, Seong-Ho Ahn, and Dong-Hwa Jeong

가톨릭대학교 인공지능학과

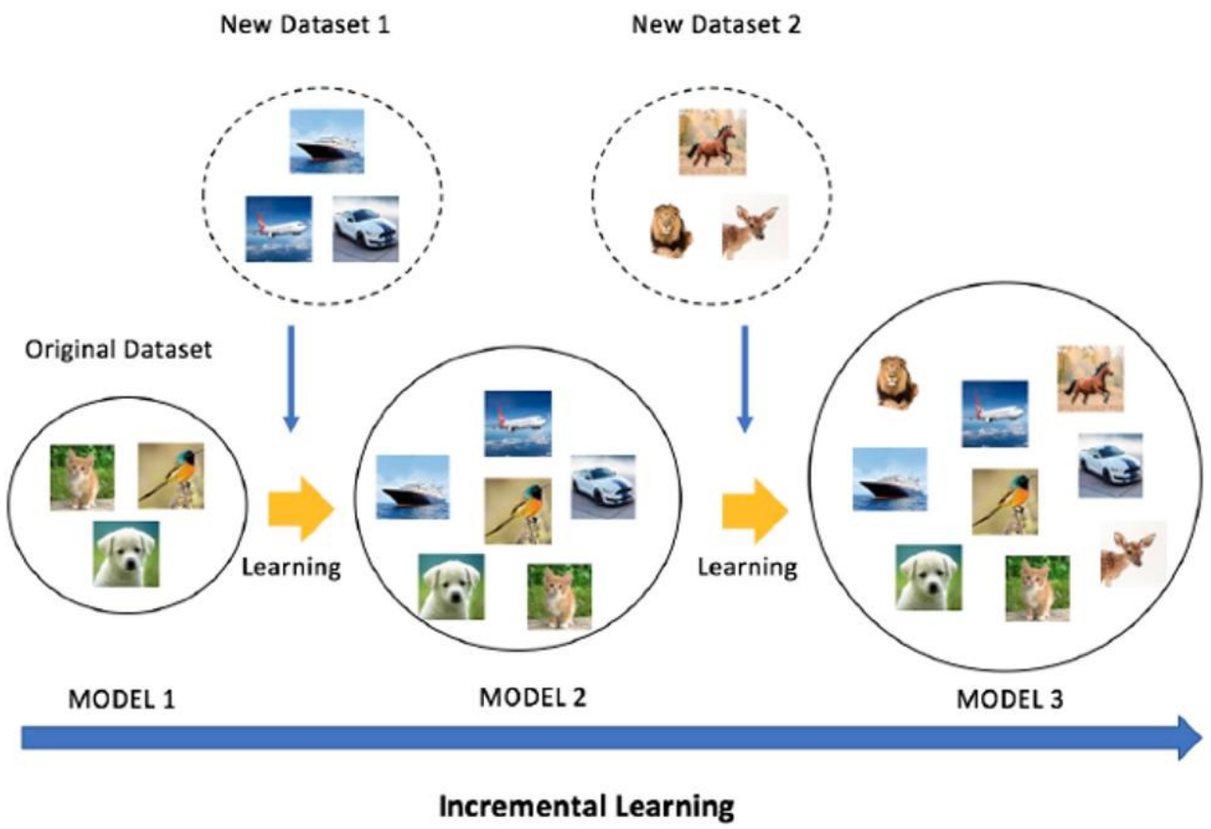
▶️ Human Activity Recognition



▶️ Semi-supervised Learning



▶️ Incremental Learning



▶️ Extremely Fast Decision Tree

If $G_t(X_a) - G_t(X_\emptyset) > \varepsilon, G_t(X_a) - G_t(X_{current}) > \varepsilon$

Replace l by an internal node that splits on Xa

Hoeffding Bound:

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}} \qquad (1)$$

Hoeffding AnyTime Tree makes a simple change to the current de facto standard for incremental tree learning. The current stateof- the-art Hoeffding Tree aims to only split at a node when it has identified the best possible split and then to never revisit that decision.

In contrast HATT aims to split as soon as a useful split is identified, and then to replace that split as soon as a better alternative is identified.

- Personal information issues may occur while passing through the server

- The time and storage space required increase when retraining

- Sensor data is difficult to label

- Personal information issues may occur while passing through the server - (1)
- The time and storage space required increase when retraining - (2)
- Sensor data is difficult to label - (3)

- Personal information problem using Incremental Learning - (1)
- Time and storage space problems using Incremental Learning - (2)
- Labeling problem using Semi-supervise Learning - (3)

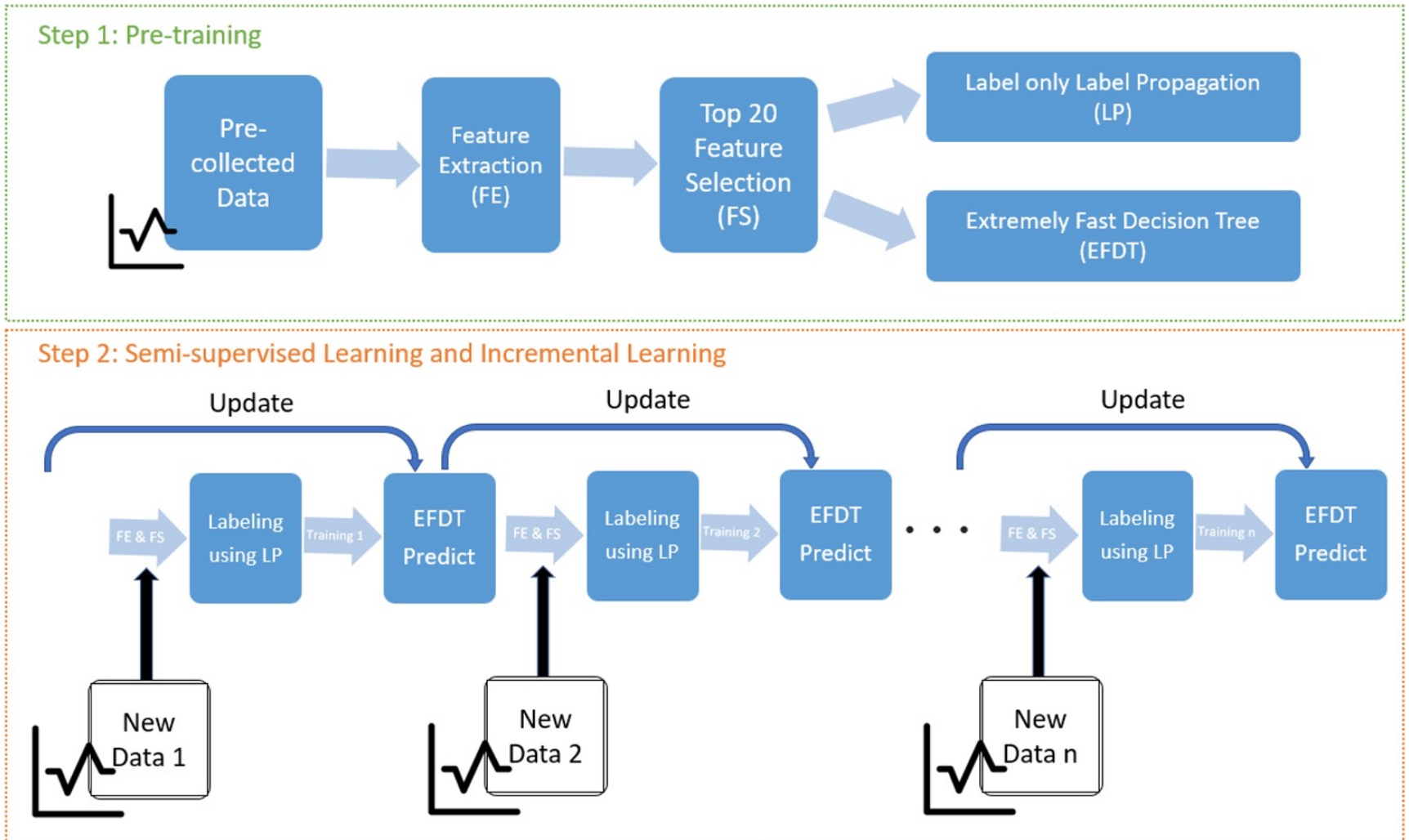
▶️ DATA SET : UCI-Human Activity Recognition Using Smartphones Data Set

-activity
1 WALKING
2 WALKING_UPSTAIRS
3 WALKING_DOWNSTAIRS
4 SITTING
5 STANDING
6 LAYING

-Constant Speed: 50 Hz using an accelerometer and a gyroscope
-Using a Sliding window : 2.56 seconds
- Feature Selection : using the top 20 features with high importance as feature importance using random forest
-To establish an environment similar to the actual use environment, the experiment was conducted by pre-training first, continuously adding new data, and performing SSL by LP on the added data

▶️ Experiment Environment
: Google colab and utilized the machine learning library scikit-learn

▶️ Framework Visualization



▶️Result

Table 1. Comparison of model test performance

Model	Accuracy	Precision	Recall	F1-score
DT	75.67%	75.89%	75.46%	75.43%
EFDT	84.73%	84.65%	84.31%	84.34%
LP+DT	80.32%	80.65%	79.63%	79.78%
LP+EFDT	83.17%	83.28%	82.46%	82.56%

Table 2. Comparison of model training times

Model	Mean	Std
EFDT	0.0146s	0.0033s
LP+DT	0.0997s	0.0227s
LP+EFDT	0.0333s	0.0152s

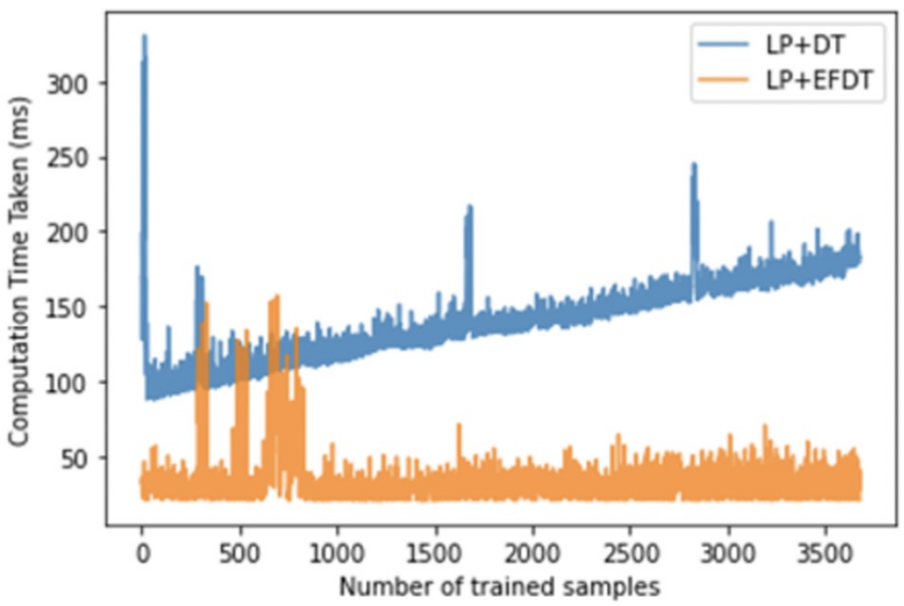
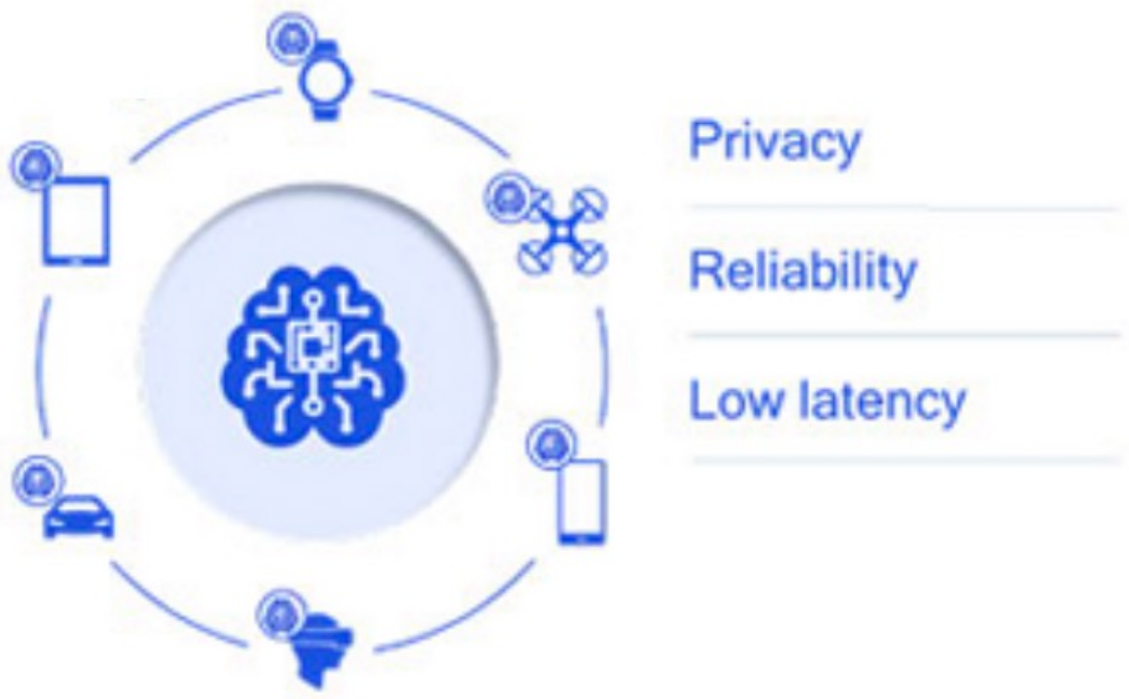


Fig. 2. Computation time of our proposed model per trained samples

The method of this study showed that it is a more useful method in actual use environments!

▶️Conclusion



THANK YOU