Data-driven Hierarchical Human Activity Recognition Using Wearables

Aiguo Wang

School of Electronic Information Engineering

Foshan University

AT-IOT@Chuzhou University

July 11th, 2023

Human Activity Recognition (HAR)

- **Motivation**
- Human activity recognition challenge
- Hierarchical human activity recognition
  - (Prior) Knowledge-driven hierarchical activity recognition
  - Data-driven hierarchical activity recognition
- Conclusion

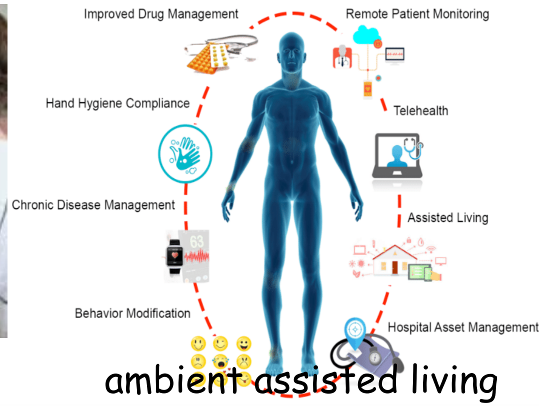
games

sports training

fall detection

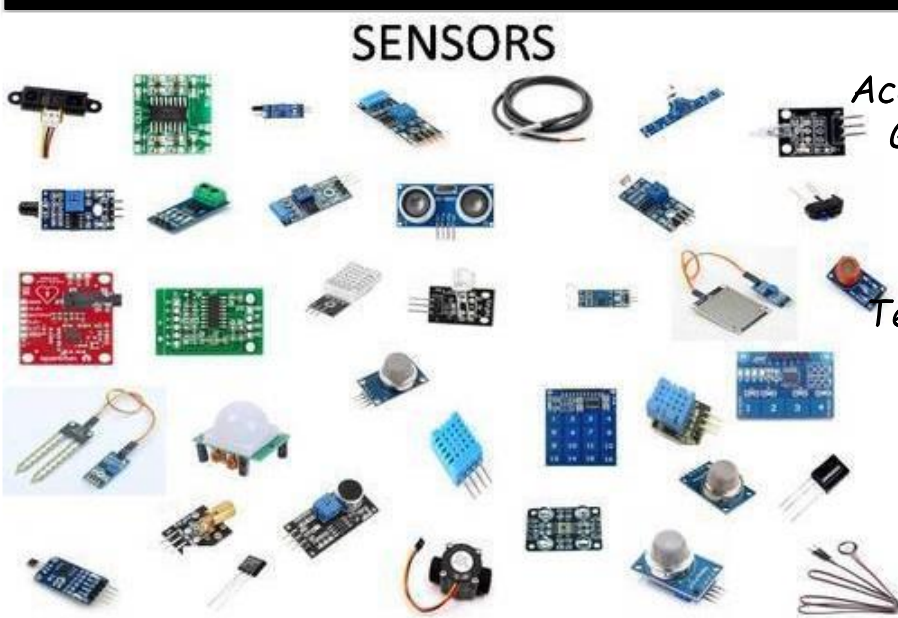
High-level applications, e.g., *smart home, elderly healthcare, ambient assisted living, human computer interaction*

rehabilitation training

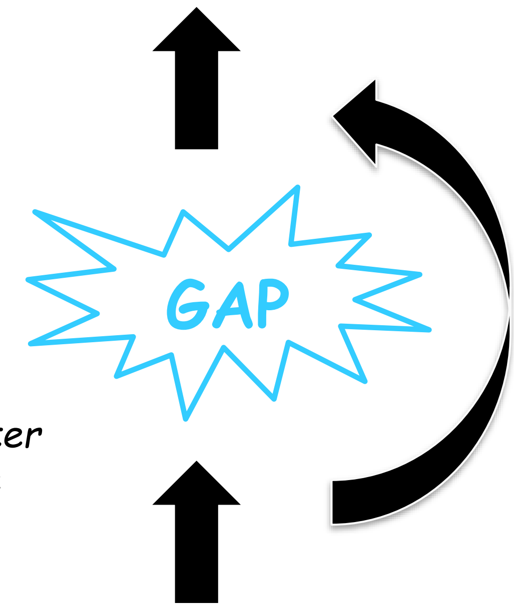


ambient assisted living

…



Camera
Accelerometer
Gyroscope
RFID
Infrared
Pressure
Temperature
Humidity
…



Human Activity Recognition

Low-level sensor data, e.g., *wearable sensors, vision sensors, environment sensors*

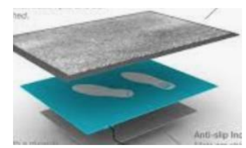
https://www.researchgate.net/publication/303096601_Progress_in_ambient_assisted_systems_for_independent_living_by_the_elderly/figures?lo=100

Categorization According to the Used Sensing Units

- **Vision-based methods**

  - ✓ utilize a camera or video to capture human movement, such as Kinect
  - ✓ easily influenced by ambient occlusion, background noise, and illumination variations
  - ✓ privacy issues, fixed place

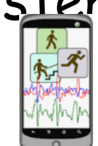
- **Ambient sensor-based methods**



  - ✓ place or embed sensors in the household objects
  - ✓ infer the on-going activities based on the interaction between an individual and the surroundings
  - ✓ fixed place, non-trivial to setup and maintain the system

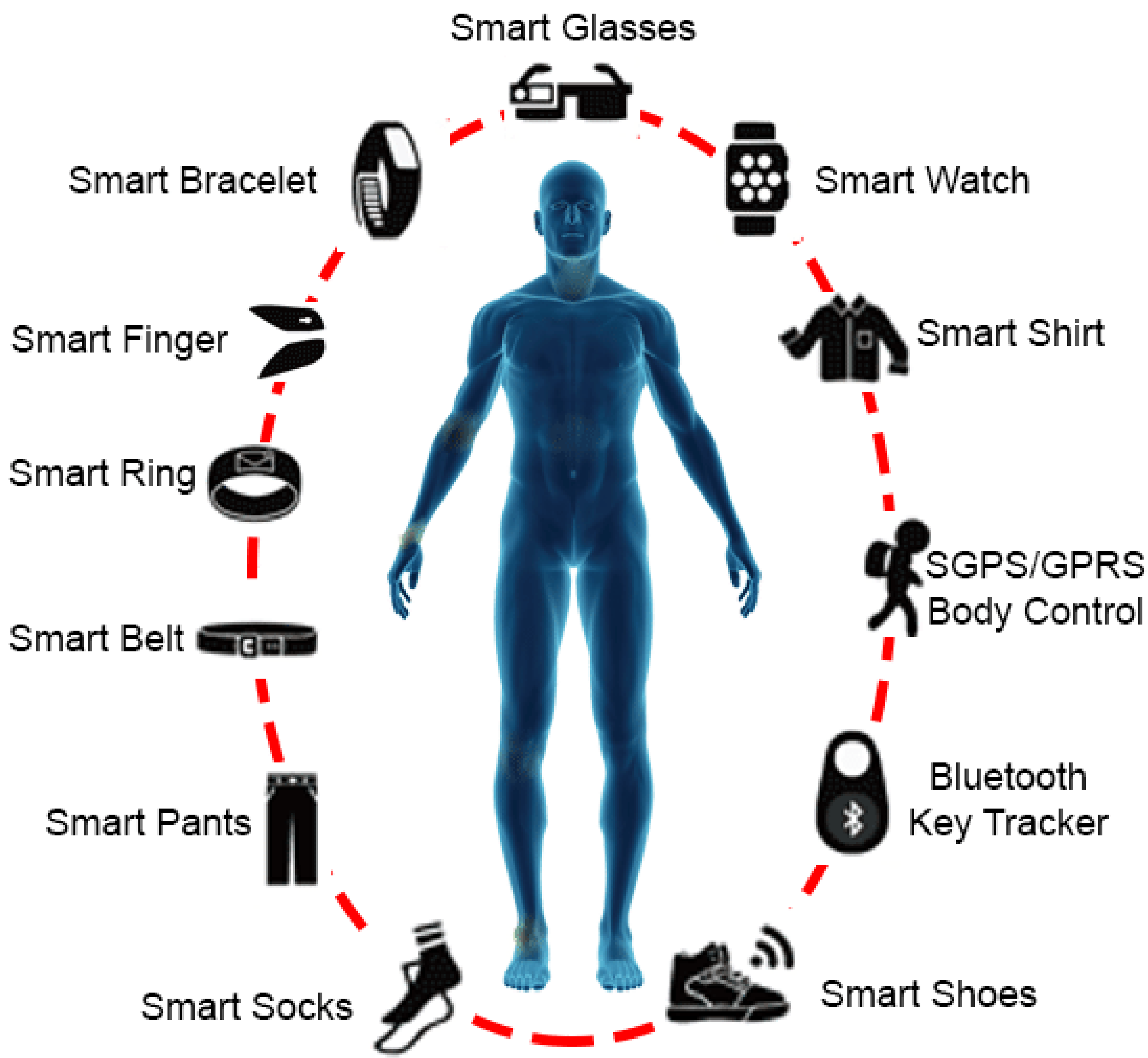
- **Wearable sensor-based methods**





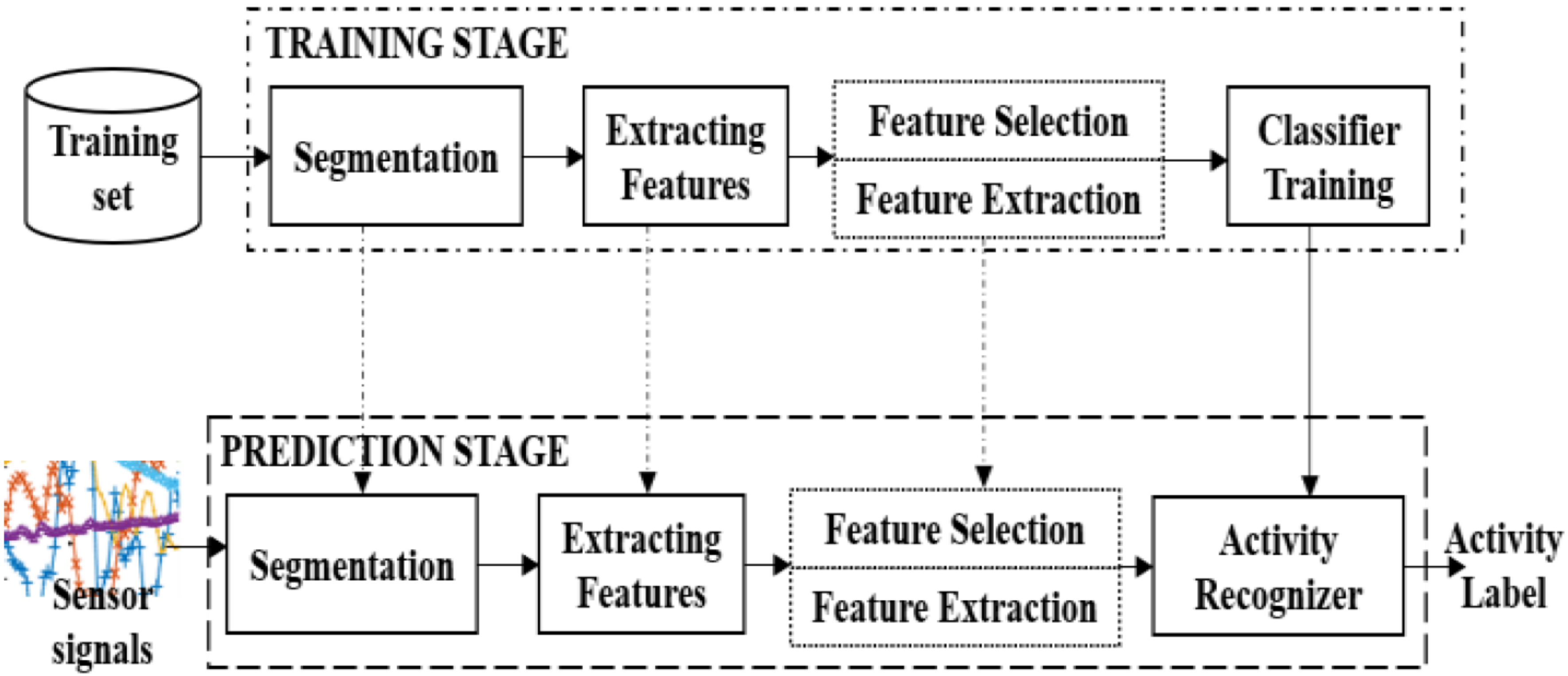
  - ✓ recognize human activities according to the wearable sensor data collected by someone performing an activity
  - ✓ suitable for both indoor and outdoor scenarios
  - ✓ less invasive to users

Wearable Sensors for Activity Recognition



https://www.researchgate.net/publication/322261039_Enabling_Technologies_for_the_Internet_of_Health_Things/figures?lo=1

Activity Recognition Chain (ARC)



- consist of the (offline) training stage and (online) prediction stage
- *key components: segmentation, extracting features, feature reduction, choice of classifiers*

Segmentation

- Divide time-series sensor data into segments
- Segmentation methods
  - ✓ explicit segmentation
  - ✓ sliding window
    - time-based vs. event-based (how many sensor events in a window)
    - fixed size vs. dynamic (adaptive) size
    - overlap vs. non-overlap between two segments
  - ✓ change-point-based
- Time-based sliding window technique is widely used and works well in practical use

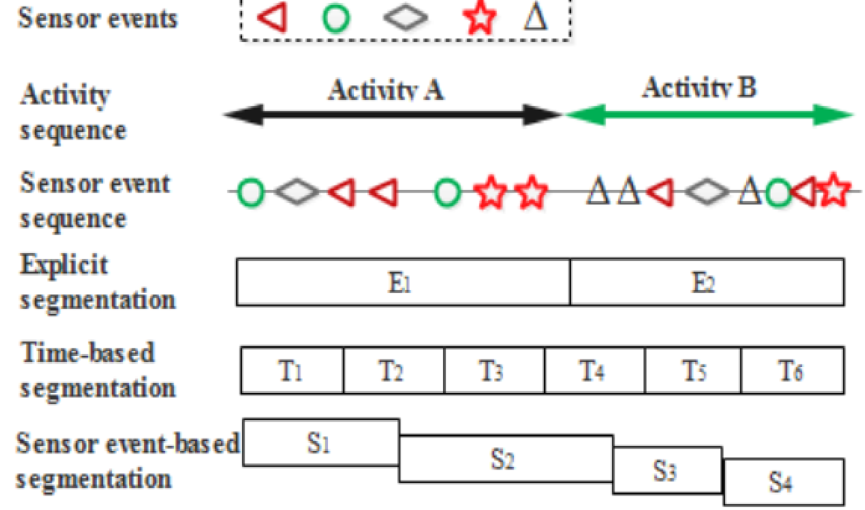
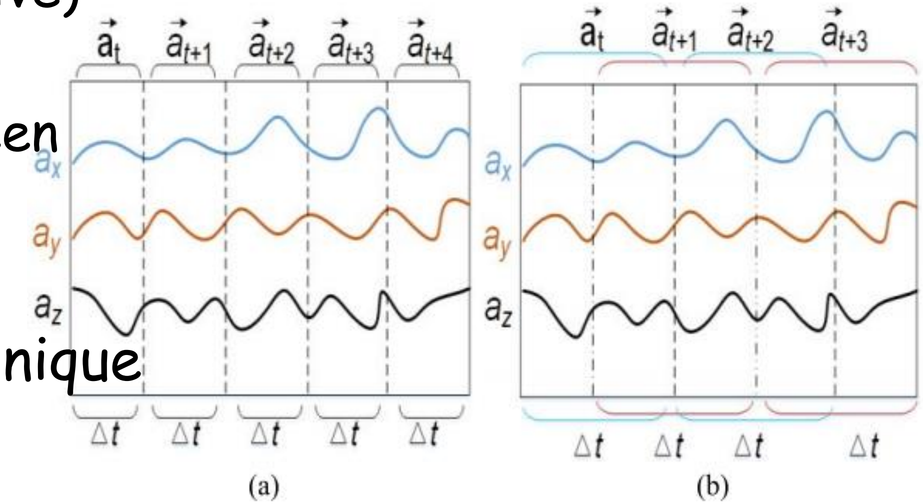
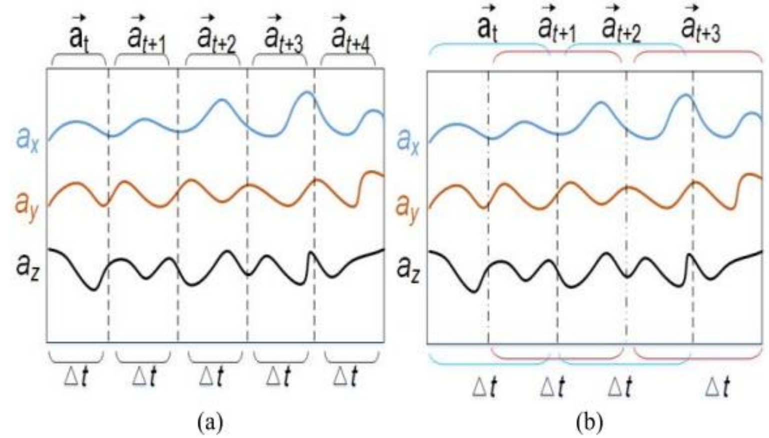


Illustration of three specific sliding window techniques. Each symbol of the sensor events denotes a specific sensor. The activity sequence consists of two activities, activity A and activity B



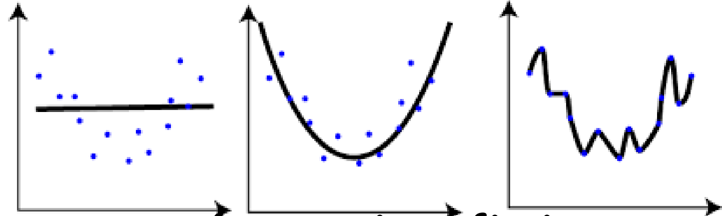
Two types of sliding window techniques. (a) Non-overlapping. (b) Overlapping[7]

Extracting Features



- **Time domain**
  - ✓ mean, std, maximum, minimum
  - ✓ autoregression coefficients
  - ✓ signal magnitude area, energy
  - ✓ correlation coefficient between two signals
  - ✓ …
- **Frequency domain (using FFT, VMD)**
  - ✓ skewness, kurtosis, the frequency component with largest magnitude
  - ✓ …
- **Time-frequency domain**
  - ✓ wavelet transformation
- **Structural features**
  - ✓ try to find interrelation or correlation between the signals
  - ✓ this means that the signal can fit a previously defined mathematical function to the current state of the variables

Dimensionality Reduction



Data points fitting

☐ Feature extraction

- ✓ project the high-dimensional data into a reduced space
- ✓ unsupervised methods (e.g., principle component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE)), supervised methods (e.g., linear discriminant analysis (LDA))

A1	A2	A3	A4	C

⇨

B1	B2	C

B1 = f(A1, A2, A3, A4)

B2 = g(A1, A2, A3, A4)

☐ Feature selection

- ✓ seek to find the minimally sized subset of features without significantly degrading the classification accuracy and changing the class distribution

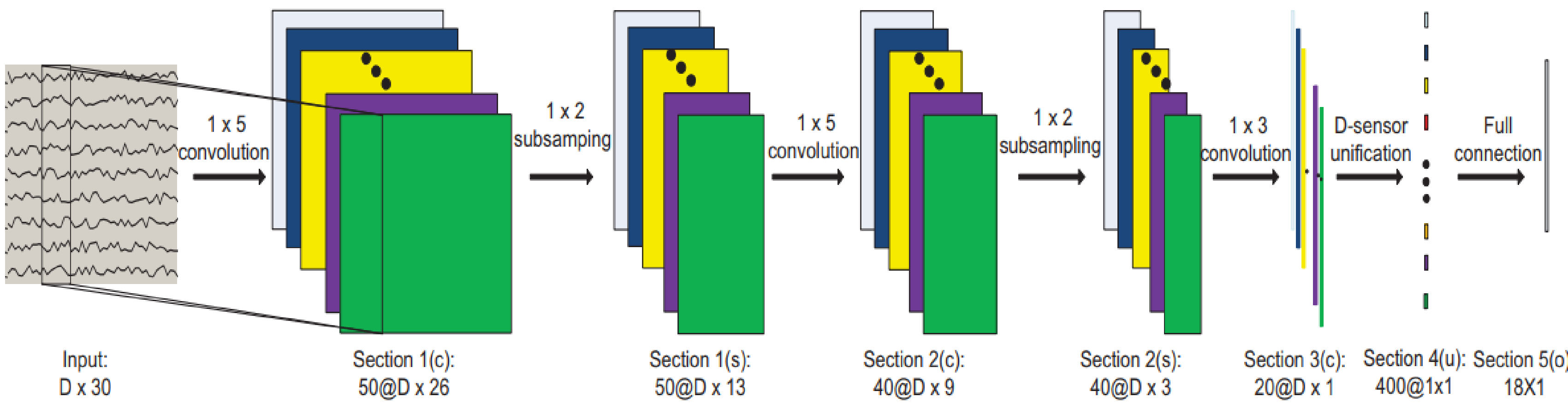
A1	A2	A3	A4	C

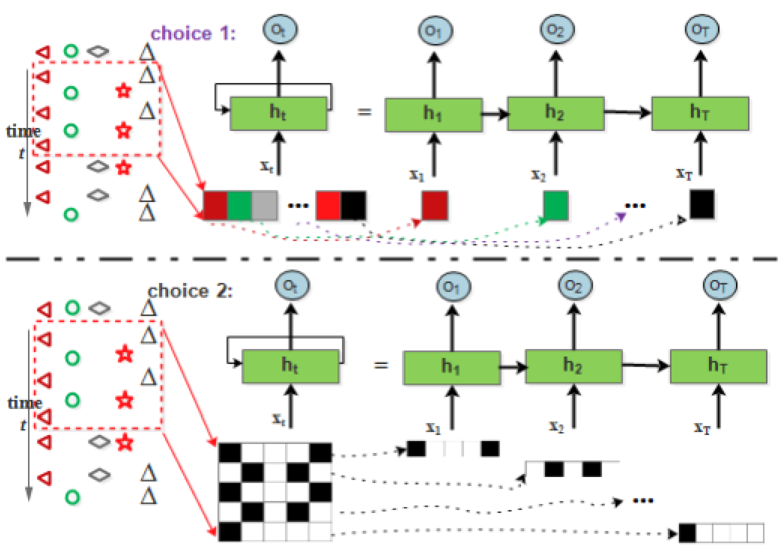
⇨

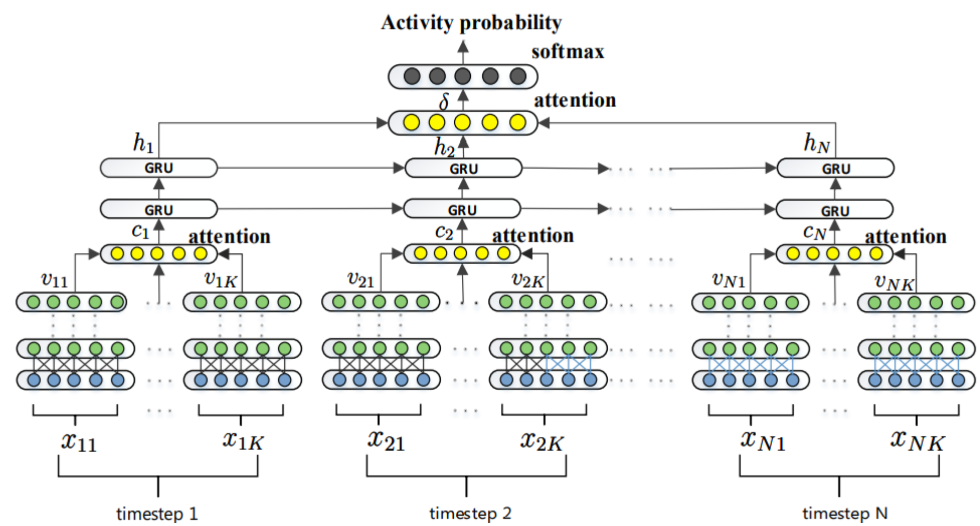
A2	A4	C

Deep Learning

- have the end-to-end learning capability to automatically learn high-level features from raw signals
- joint optimization of features and classifiers







Wang, Aiguo, et al. "Activities of Daily Living Recognition with Binary Environment Sensors Using Deep Learning: A Comparative Study." *IEEE Sensors Journal* (2020).

Human Activity Recognition (HAR)

- Motivation
- **Human activity recognition challenge**
- Hierarchical human activity recognition
  - (Prior) Knowledge-driven hierarchical activity recognition
  - Data-driven hierarchical activity recognition
- Conclusion

CHALLENGE

Modelling and Evaluation

- ✓behavior computable?
- ✓no standard evaluation metrics or systems as the context of human behavior varies
- ✓what performance should be considered (e.g., accuracy, time-efficiency, energy efficiency, robustness)
- ✓…

Data

- ✓data fragmentation
- ✓data heterogeneity
- ✓data representativeness
- ✓data sparsity
- ✓imbalanced data distribution
- ✓spatial-temporal correlation
- ✓…

Human Behavior Itself

- ✓capriciousness
- ✓evolution
- ✓null class
- ✓multiple granularity (action, activity, behavior, plan, goal, intention, etc.)
- ✓…

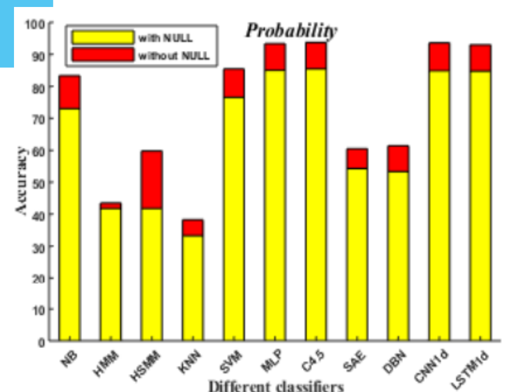


Fig. Influence of null class

Yu, Zhiwen, et al. "Ten scientific problems in human behavior understanding." *CCF Transactions on Pervasive Computing and Interaction* 1.1 (2019): 3-9.

Human Behavior Itself (cont'd)

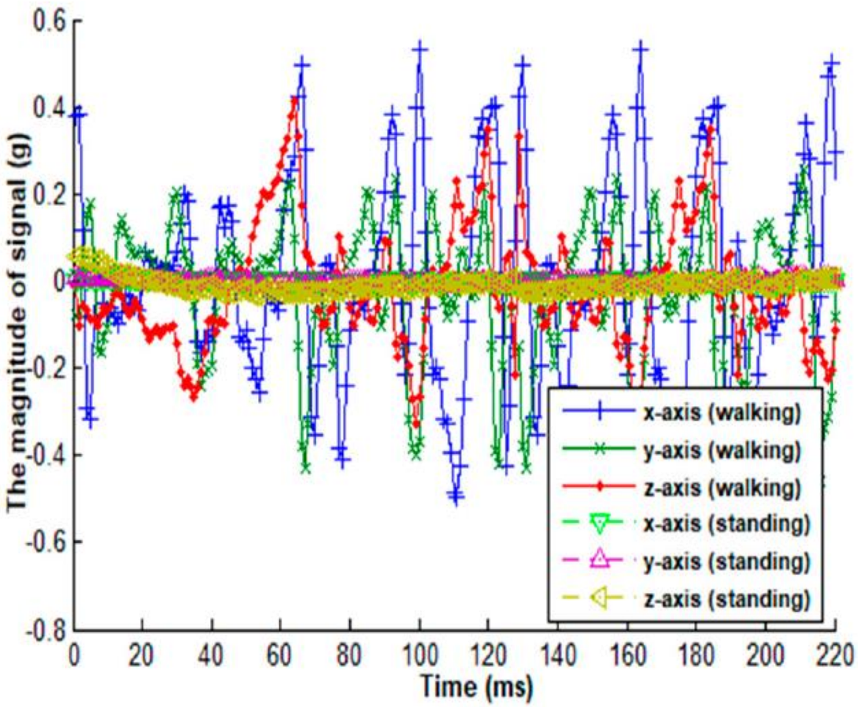
- *Inter-subject and intra-subject variations*
  - ✓ *subject dependent vs. subject independent*
- *Interleaved activities*
  - ✓ *cooking- telephone - cooking*
- *Concurrent activities*
  - ✓ *talking & watching TV*
- …

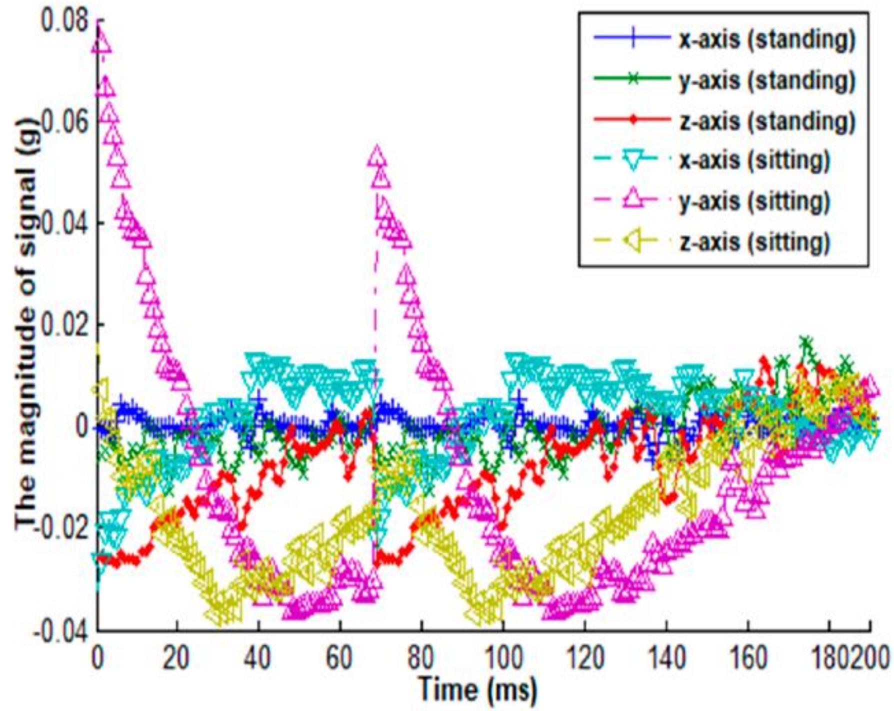
Allen's interval algebra

Allen Statements		Pictoral Example	Chronological Sequence
Relations	Inverse Relations		
X *before* Y	Y *after* X	X Y	$X_{start} < X_{end} < Y_{start} < Y_{end}$
X *equals* Y	Y *equals* X	X Y	$X_{start} = Y_{start} < X_{end} = Y_{end}$
X *meets* Y	Y *met by* X	X Y	$X_{start} < X_{end} = Y_{start} < Y_{end}$
X *overlaps* Y	Y *overlapped by* X	X Y	$X_{start} < Y_{start} < X_{end} < Y_{end}$
X *contains* Y	Y *during* X	X Y	$X_{start} < Y_{start} < Y_{end} < X_{end}$
X *starts* Y	Y *started by* X	X Y	$X_{start} = Y_{start} < X_{end} < Y_{end}$
X *finishes* Y	Y *finished by* X	X Y	$Y_{start} < X_{start} < X_{end} = Y_{end}$

Human Behavior Itself (cont'd)

- **_Confusion between similar activities_**
  - ✓ *predefined activities that trigger similar sensor signals, even they have different semantics*





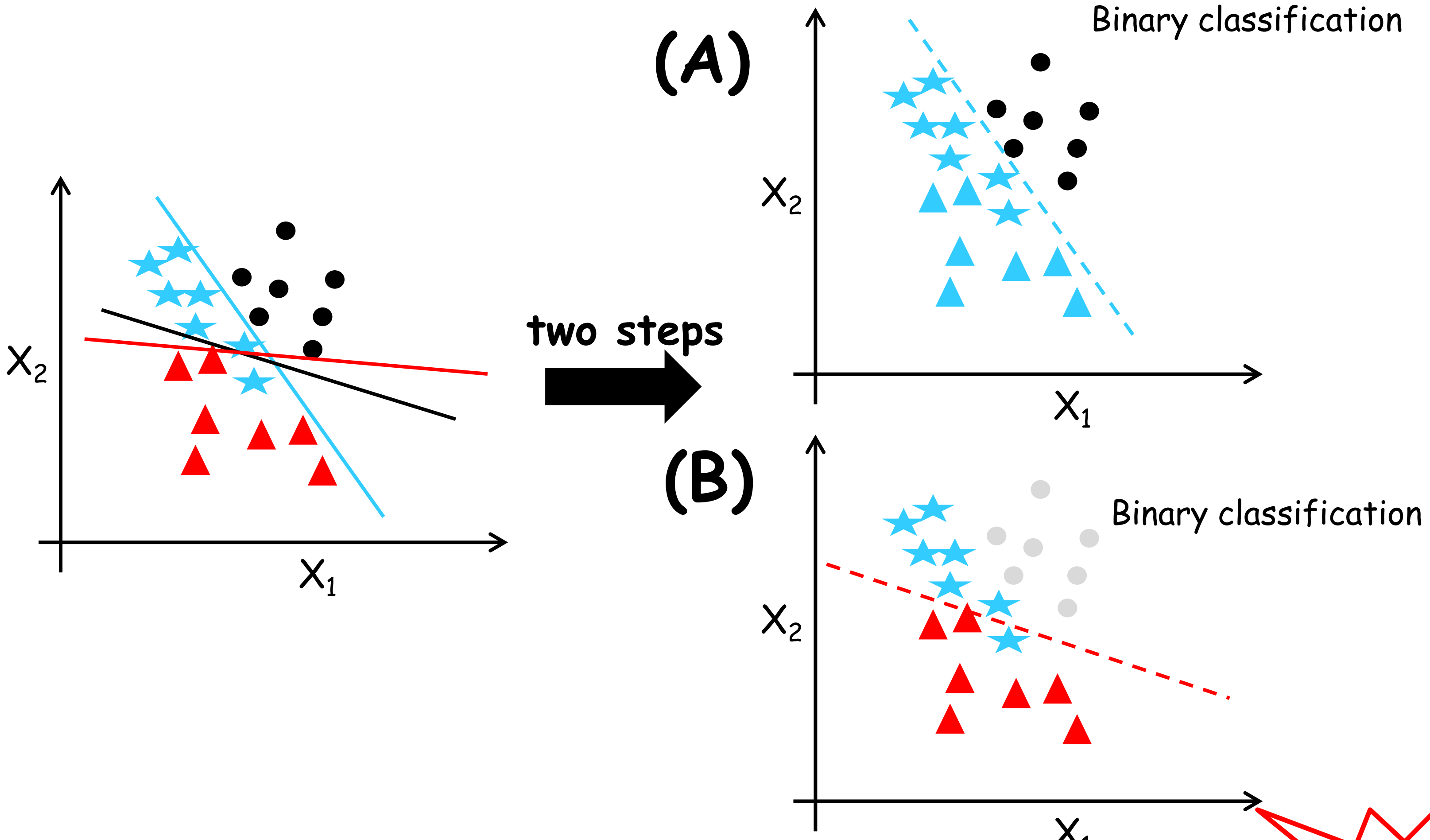
Comparison of the magnitude of a tri-accelerometer among three different activities. The accelerometer has sensor readings from three axes, i.e., x-axis, y-axis, and z-axis. (a) Comparison of walking and standing; (b) Comparison of standing and sitting.

Human Activity Recognition (HAR)

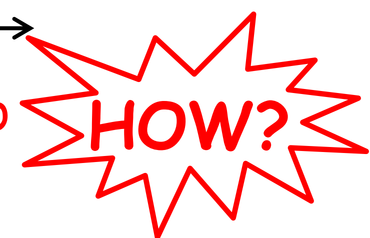
- Motivation
- Human activity recognition challenge
- **Hierarchical human activity recognition**
  - (Prior) Knowledge-driven hierarchical activity recognition
  - Data-driven hierarchical activity recognition
- Conclusion

Hierarchical Human Activity Recognition

- Motivation: for multiple-class classification problem, how to get the decision boundary?

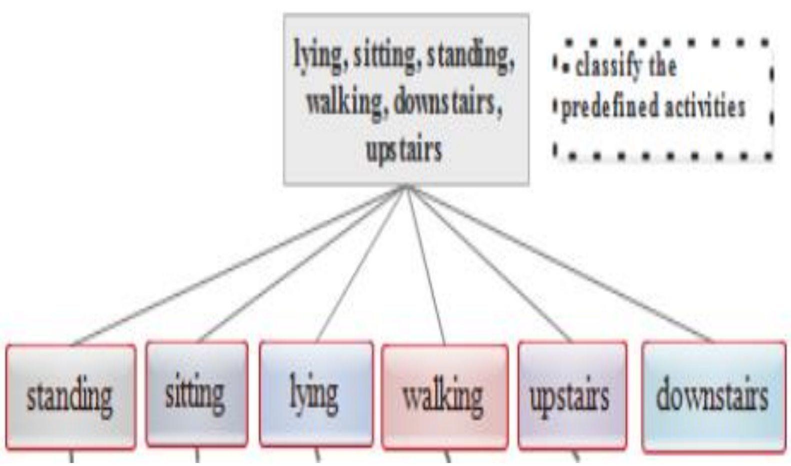


- Idea: divide the predefined activities of interest into multiple sub-groups and further recognize activities

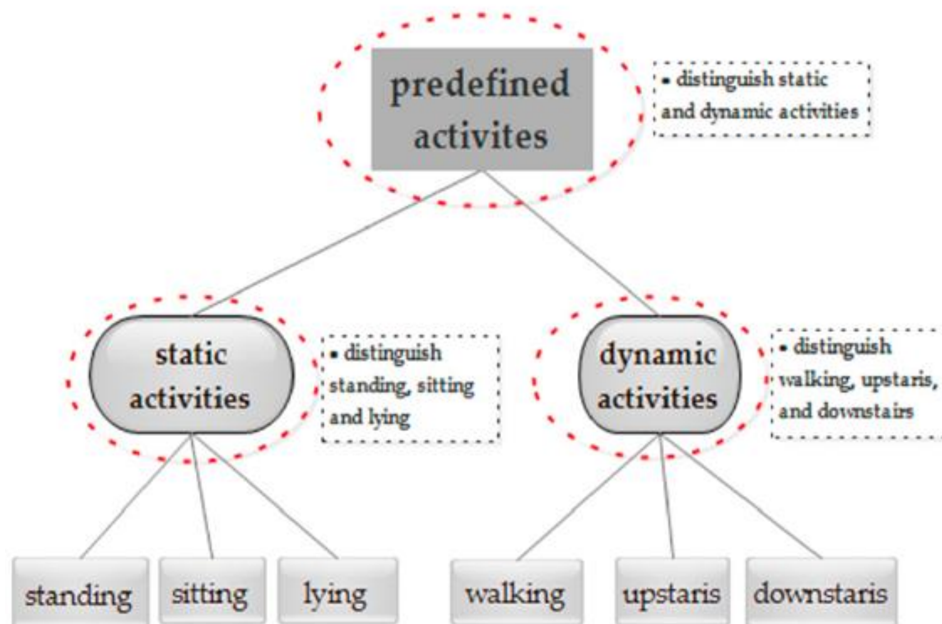


(Prior) Knowledge-driven Approach

- In some (simple) cases, we can group the activities of interest into sub-groups according to the movement state, time-spatial information, or other knowledge
- e.g., group *standing*, *sitting*, *lying* into static activity, and group *walking*, *go-upstairs*, and *go-downstairs* into dynamic activity
- Organize the procedure into a tree-structure



Flat structure



Tree-based (Hierarchical) structure

Wang, Aiguo, et al. "Towards human activity recognition: a hierarchical feature selection framework." *Sensors* 18.11 (2018): 3629.

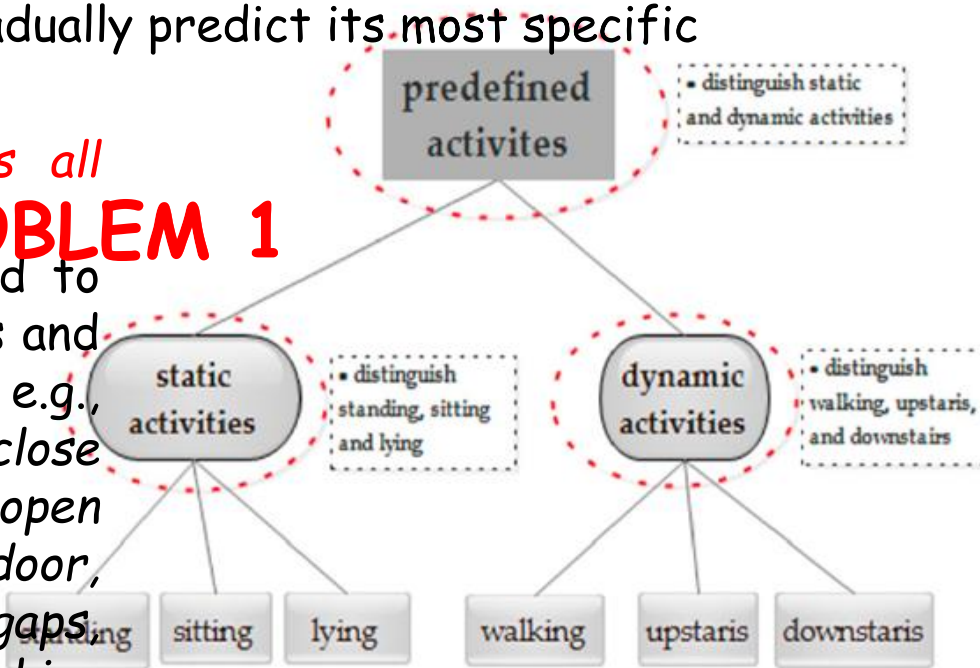
Knowledge-driven Tree-based Model

- Training stage
  - ✓build a classifier for each non-leaf node
  - ✓for each non-leaf node, its training set comes from its child nodes
- Prediction stage
  - ✓a top-down fashion is used to gradually predict its most specific activity label
- *One optimal feature subset fits all nodes?*

PROBLEM 1

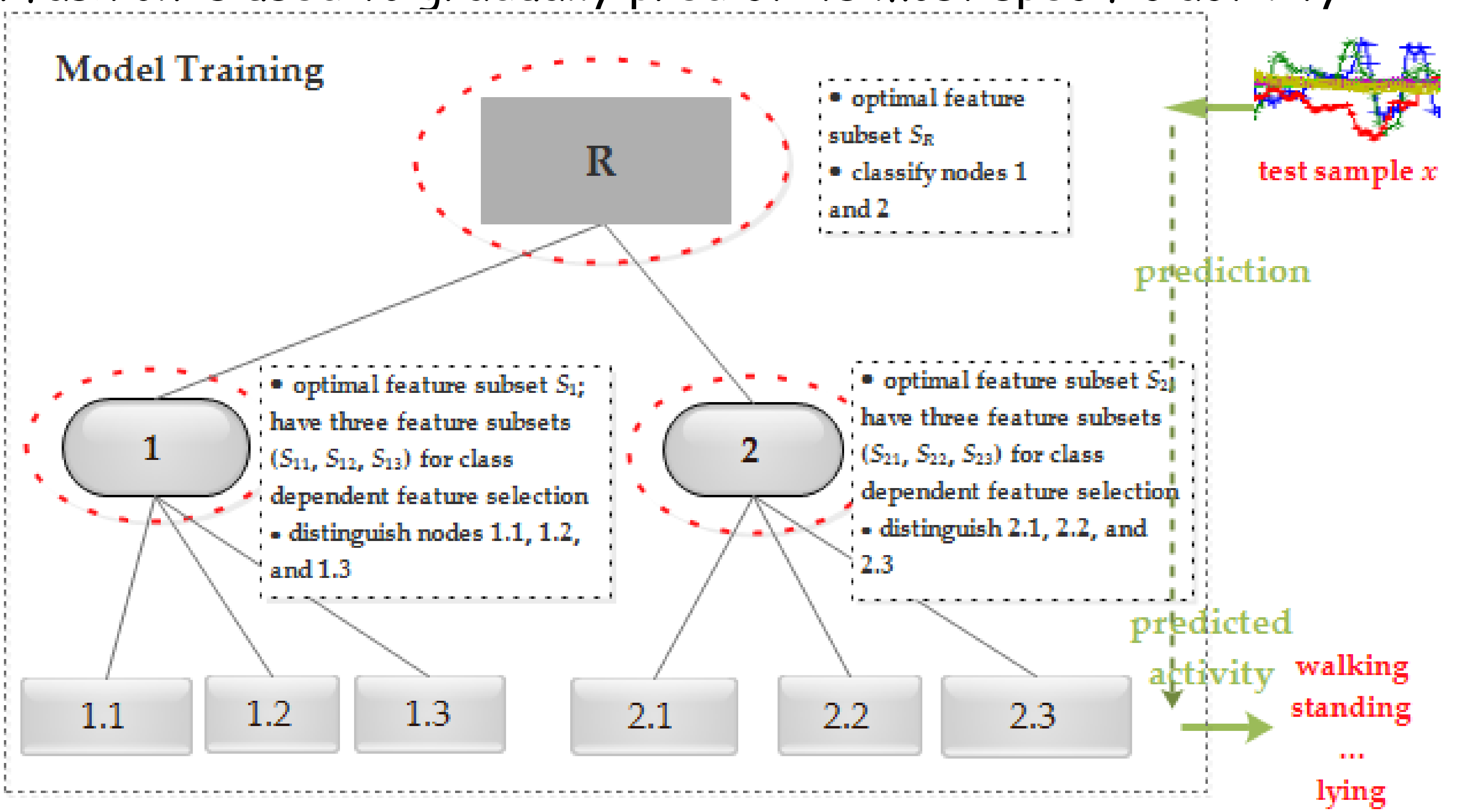
- For complex cases where we need to handle a large number of activities and expert knowledge is not available, e.g., *write on notepad, open hood, close hood, check gaps on the front, open left front door, close left front door, close both left door, check trunk gaps, open and close trunk, and checking steering wheel*
- Not easy to obtain the hierarchical structure

PROBLEM 2



Tree-based Model with Feature Selection

- Training stage
  - ✓optimize the feature space for each non-leaf node (using (existing) various feature selection algorithms)
  - ✓build a classifier for each non-leaf node
  - ✓for each non-leaf node, its training set comes from its child nodes
- Prediction stage
  - ✓a top-down fashion is used to gradually predict its most specific activity label



DONE !

Experiments & Results

- Naïve Bayes classifier, Fast Correlation-based Filter (FCBF)
- Feature selection
  - ✓ class independent: select one common feature subset for all classes
  - ✓ class dependent: select a feature subset for each class
- UCI-HAR dataset

A comparison on accuracy between hierarchical and non-hierarchical methods

Sensor	Non-hierarchical model			Hierarchical model		
	no feature selection	class independent	class dependent	no feature selection	class independent	class dependent
gyroscope	50.80	62.95	61.62	50.87	62.44	**64.91**
accelerometer	81.07	85.61	80.49	80.90	85.61	**88.29**
gyro&acc	76.99	88.16	83.61	76.86	88.39	**90.36**

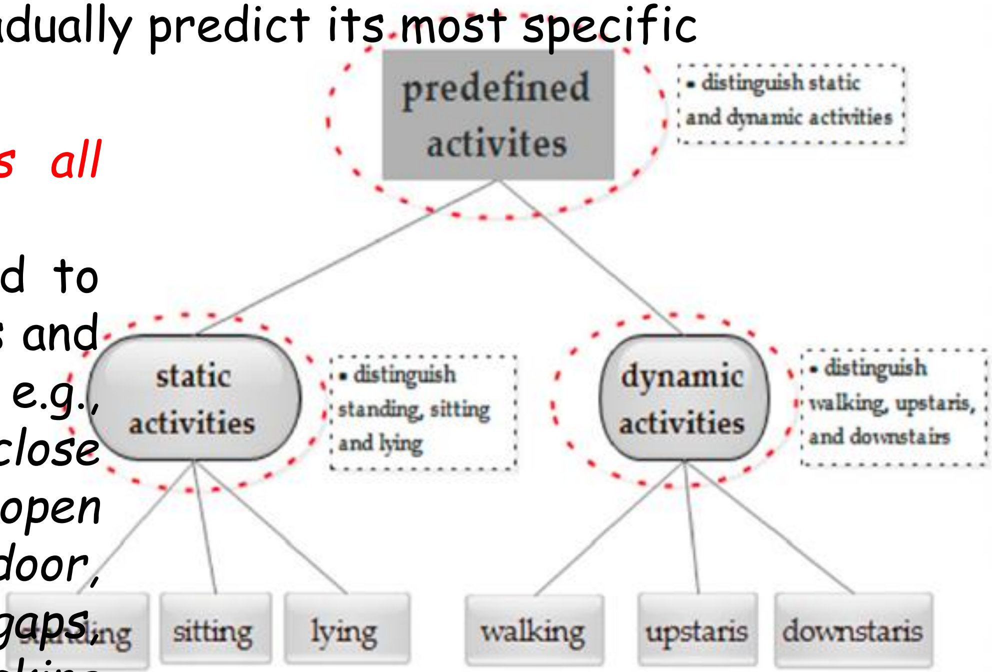
A comparison on F1 between hierarchical and non-hierarchical methods

Sensor	Non-hierarchical model			Hierarchical model		
	no feature selection	class independent	class dependent	no feature selection	class independent	class dependent
gyroscope	53.42	64.49	64.79	53.65	63.56	**66.34**
accelerometer	81.49	85.61	81.83	81.34	86.04	**88.28**
gyro&acc	78.03	88.09	84.66	77.90	88.72	**90.34**

Knowledge-driven Tree-based Model

- Training stage
  - ✓build a classifier for each non-leaf node
  - ✓for each non-leaf node, its training set comes from its child nodes
- Prediction stage
  - ✓a top-down fashion is used to gradually predict its most specific activity label
- *One optimal feature subset fits all nodes?*
- For complex cases where we need to handle a large number of activities and expert knowledge is not available, e.g., *write on notepad, open hood, close hood, check gaps on the front, open left front door, close left front door, close both left door, check trunk gaps, open and close trunk, and checking steering wheel*
- Not easy to obtain the hierarchical structure

PROBLEM 2



Skoda Mini Checkpoint

SkodaMiCP contains the sensor signals of ten manipulative gestures performed by the assembly-line worker in a car maintenance environment

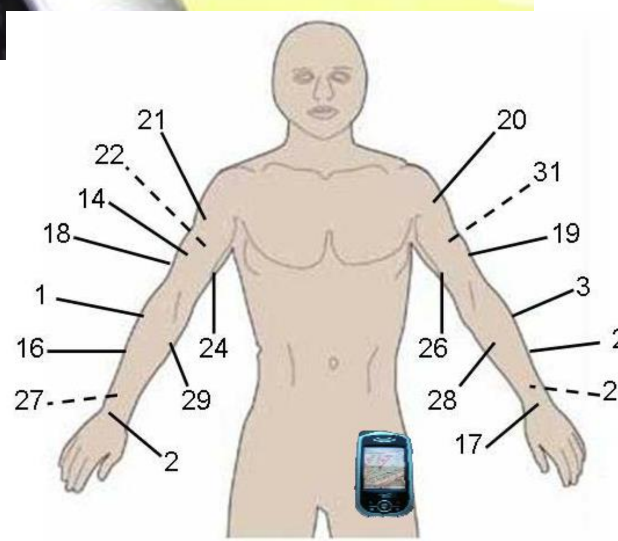


Table 1. List of activity classes to recognize from body-worn sensors.

 The user holds a notepad with his left hand and writes down a short sentence with his right hand.	 The user opens the hood with his left hand and blocks it with a stick kept with his right hand.	 The user removes the stick with his right hand while keeping the hood with his left hand then closes the hood with his left hand.	 The user checks the gaps on the front door by sliding his left and right hand over the gaps. The two hands move simultaneously.	 The user grabs the car left front door with his left hand while it is closed and opens it completely.
 The user grabs the car left front door with his left hand while it is open and closes it completely.	 The user grabs the car left front and back doors with his left and right hands than open and close completely and at the same time the two doors	 The user checks gaps on the trunk by sliding his left and right hand over the gaps. The two hands move simultaneously.	 The user opens the trunk using both hands and then moves it up and down on the top of his head three times before closing it.	 The user grabs the steering wheel with both hands and turns it clockwise and counterclockwise three times.

Human Activity Recognition (HAR)

- Motivation
- Human activity recognition challenge
- **Hierarchical human activity recognition**
  - (Prior) Knowledge-driven hierarchical activity recognition
  - **Data-driven hierarchical activity recognition**
    - ✓ **confusion matrix based hierarchical activity recognition**
    - ✓ clustering-guided hierarchical activity recognition
- Conclusion

Confusion matrix

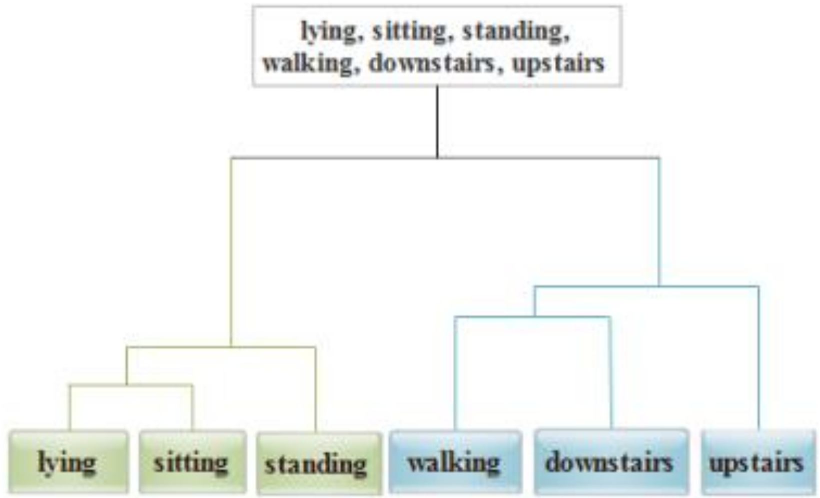
- Allow visualization of the performance of an algorithm, typically a supervised learning one
- Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa)
- Indicate the confusion among activities

	1	2	3	4	5	6
1	416	38	42	0	0	0
2	9	451	11	0	0	0
3	80	83	257	0	0	0
4	0	7	0	368	111	5
5	0	15	0	54	455	8
6	0	3	0	212	0	322

Confusion matrix for activities (1, 2, 3, 4, 5, 6)

Data-driven Tree-based Model

- Step 1: apply a clustering algorithm to the confusion matrix, and get a dendrogram that determines the clusters of activities



(a) The dendrogram of the six activities.

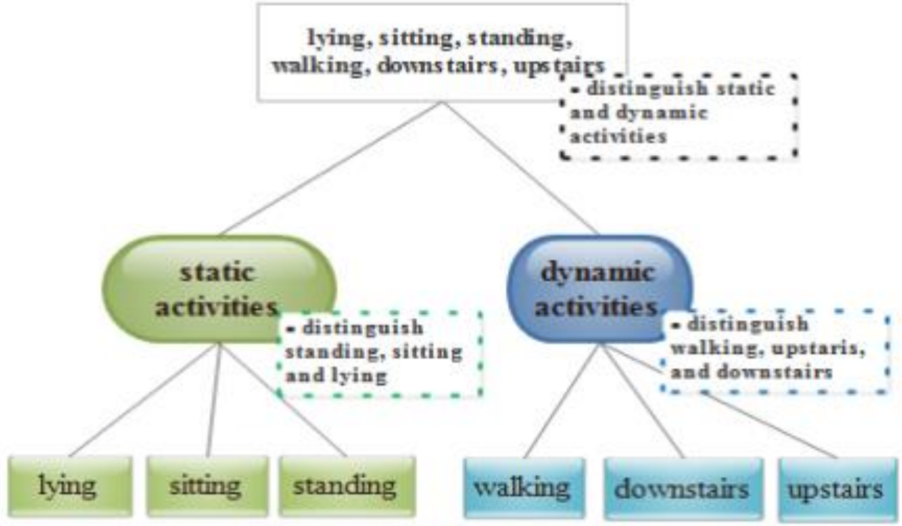
- Step 2: clip the dendrogram to get a tree structure

(b) The tree-based model

DONE !

Wang, Aiguo, et al. "HierHAR: Sensor-based Data-driven Hierarchical Human Activity Recognition." *IEEE Sensors Journal* (2020).

- What if we make wrong predictions at the first level?
- The misclassification of the top-level classifier jeopardizes the performance of the second-level classifiers



(b) The tree-based model

- In Table I, 1.7% standing instances are classified as upstairs. If an instance of standing is classified as dynamic activity by the top-level classifier, the second-level classifier can only classify it as walking, upstairs, or downstairs.

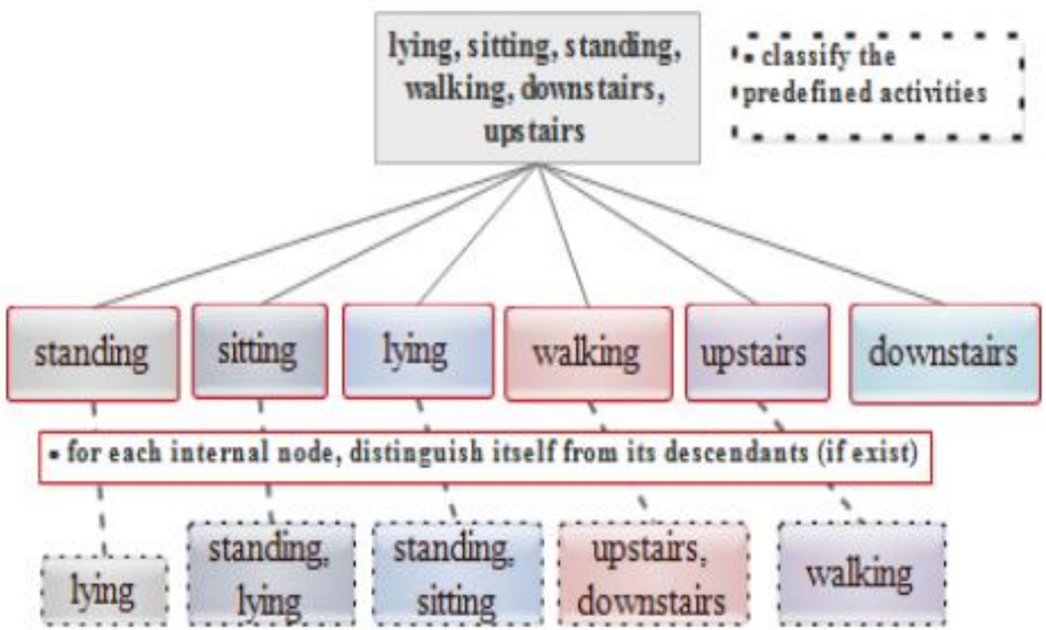
TABLE I
CONFUSION MATRIX ON UCI-HAR WITH NAÏVE BAYES.

	Walking	Upstairs	Downstairs	Sitting	Standing	Lying
Walking	0.727	0.165	0.109	0	0	0
Upstairs	0.021	0.901	0.077	0	0	0
Downstairs	0.038	0.173	0.789	0	0	0
Sitting	0	0.012	0	0.750	0.223	0.014
Standing	0.001	0.017	0	0.256	0.722	0.005
Lying	0	0.016	0	0.422	0	0.563

- Accumulated errors induced by the prediction process of the tree-based model

PROBLEM 3

- Problem: restrict the connections of activities to a hierarchy of disjoint groups
- Idea: enable connections between any two activities under certain conditions
  - Step 1: obtain the confusion matrix among the activities
  - Step 2: for each activity *A*, find the set of activities *S*(*A*) that are more easily misclassified as activity *A*
    - ✓ define a confusion threshold to obtain the confusing activities of *A*

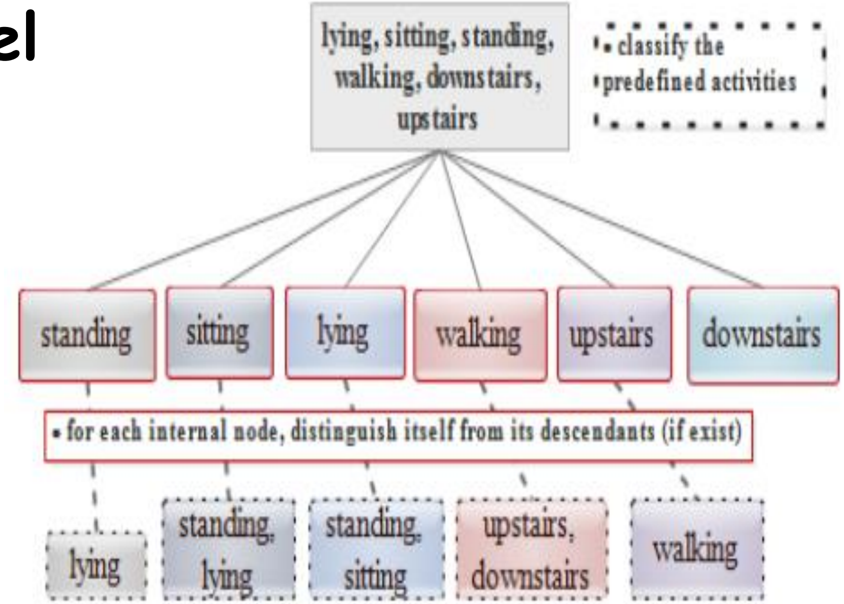


- Graph-based model **DONE !**

Wang, Aiguo, et al. "HierHAR: Sensor-based Data-driven Hierarchical Human Activity Recognition." *IEEE Sensors Journal* (2020).

Data-driven Graph-based Model

- **Training stage**
  - ✓ first trains a top-level classifier to distinguish all the predefined activities
  - ✓ for each activity *A* that has non-empty *S*(*A*), we train a second-level classifier to distinguish between *A* and *S*(*A*)
- **Prediction stage**
  - ✓ first classify it using the top-level classifier
  - ✓ if the set *S*(*A*) of the top-level prediction *A* is not empty, use the second-level classifier associated with *A* and *S*(*A*) to get the final prediction; otherwise, report the top-level result



Algorithm 2: Graph-based Activity Recognition Model

Input: a labeled train set D, activity labels L,
  a confusion threshold θ, a test sample x
Output: the activity label A of x

// the training of graph-based activity recognition model
1. calculate the confusion matrix CM on D; // return confusion matrix
2. **for** each activity A of L **do**
   2.1) $S(A)$ = { }; // initialize the set of confusing activities of A
3. **for** each activity A of L **do**
   3.1) **for** each activity B of L **do**
     if A != B and $CM(A, B)$ >= θ **do**
       $S(A)$.add(B); // B is the confusing activity of A and add it to $S(A)$
   3.2) **if** not_empty($S(A)$) **do**
       train a classifier *cls_A* to distinguish between A and $S(A)$;
4. train a classifier *cls_all* on D to distinguish all activities;

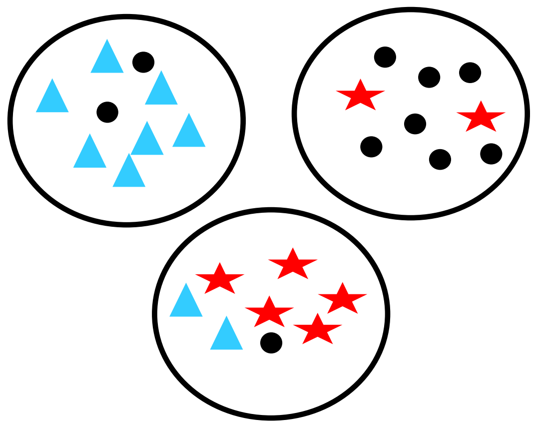
// activity recognition using the graph-based model
5. A = *cls_all* (x); // return the activity label of x using the first-level classifier
6. **if** not_empty($S(A)$) **do**
   A = *cls_A*(x); // return the label of x using the second-level classifier
7. **return** A

Human Activity Recognition (HAR)

- Motivation
- Human activity recognition challenge
- **Hierarchical human activity recognition**
  - (Prior) Knowledge-driven hierarchical activity recognition
  - **Data-driven hierarchical activity recognition**
    - ✓ confusion matrix based hierarchical activity recognition
    - ✓ **clustering-guided hierarchical activity recognition**
- Conclusion

Data Points Mixture

- Apply a clustering algorithm on the data points, the results provide the confusion information (from the viewpoint of data distribution)



- Measure the confusion among activities

Definition 1. (*Cluster Confusion Index*). Given a cluster C consisting of a subset of samples from D, the class of C is set as the label L_i ($1 \le i \le |L|$) that has the maximum number of data points in C. The number of samples with label L_j ($1 \le j \le |L|$, $i \ne j$) is defined as the cluster confusion index between L_j and L_i and is referred to as $conf_c(L_j \rightarrow L_i)$.

$$L_i = \max_{L_k \in L}\{\sum_{x \in C} \mathrm{I}(y_x = L_k)\} \quad (1)$$

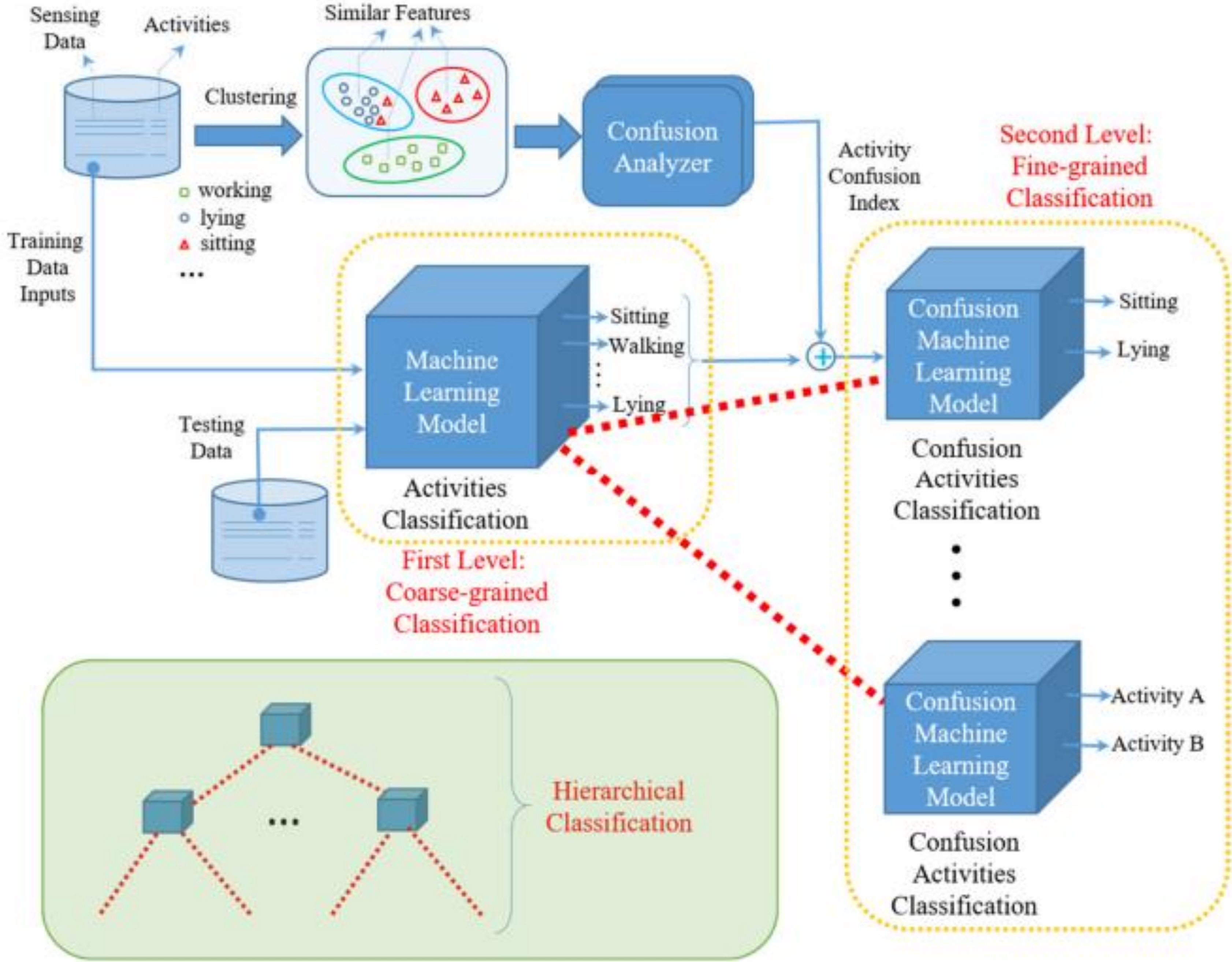
Definition 2. (*Activity Confusion Index*). Given the k clusters that are obtained by manual assignment or returned by a clustering algorithm, the activity confusion index $conf(L_j \rightarrow L_i)$ between L_j and L_i is defined as the sum of cluster confusion index of the k clusters, as given in (2).

$$conf(L_j \rightarrow L_i) = \sum_{c=1}^{k} conf_c(L_j \rightarrow L_i) \quad (2)$$

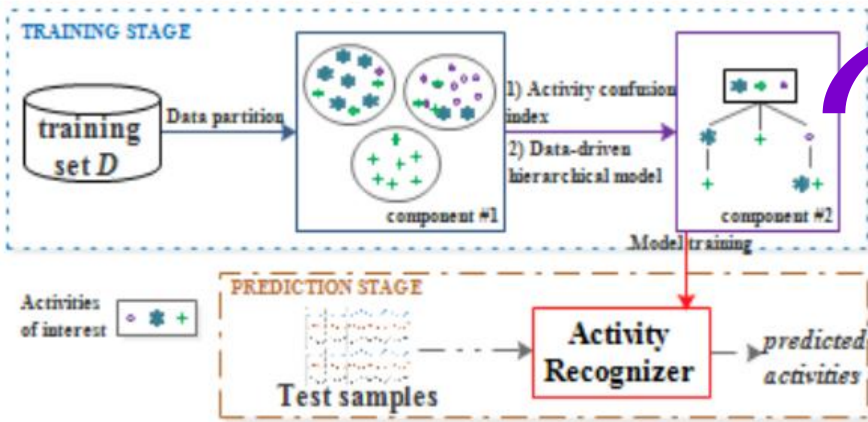
✓ use a confusion threshold θ to decide whether L_j is a potential confusing activity of L_i

$$\eta(L_j, L_i) = CM_{ji} \Big/ \sum_{j=1}^{|L|} CM_{ji} \ge \theta$$

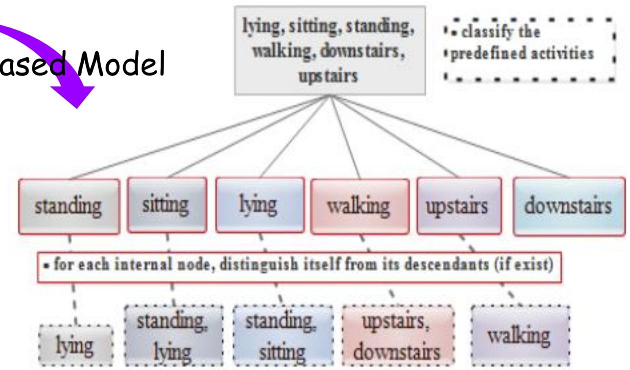
Clustering-guided Graph-based Model



Clustering-guided Graph-based Model



Graph-based Model



- lines 1-2 show the steps of quantifying the confusion among activities
- lines 3-5 denote the classification model training that mainly describes how to build a hierarchical activity recognizer under the guidance of the activity relationships
- lines 6-8 show the procedure of how to obtain the predicted label of a test sample, which involves two-level classifications

Algorithm 1: Clustering Guided Hierarchical Human Activity Recognition Framework

Input: a labeled train set D, activity labels L, confusion threshold θ, a test sample tx
Output: the predicted activity label L_A of tx

// TRAINING STAGE
1. partition D into clusters CLU; // **Component #1**
2. obtain the activity confusion matrix CM of CLU using (1) and (2);
3. train a classifier cls_all to classify all activities; // **Component #2**
4. **for** each activity L_A of L **do**
   4.1) $S(L_A) = \{\ \}$; // initialize the set of confusing activities of L_A
5. **for** each activity L_A of L **do**
   5.1) **for** each activity L_B of L **do**
      calculate $\eta(L_B, L_A)$ using (3);
      if L_A != L_B and $\eta(L_B, L_A) \geq \theta$ **do**
         $S(L_A)$.add(L_B); // save the confusing activity L_B of L_A to $S(L_A)$
   5.2) **if** not_empty($S(L_A)$) **do**
         train a classifier cls_L_A to distinguish between L_A and $S(L_A)$;

// PREDICTION STAGE
6. $L_A = cls\_all\ (tx)$; // infer the label of tx using the top-level classifier
7. **if** not_empty($S(L_A)$) **do**
   $L_A = cls\_L_A(tx)$; // infer the label of tx using the second-level classifier
8. **return** L_A // return the prediction

Experimental Setup & Results

- UCI-HAR consists of six human activities performed by thirty volunteers with a smartphone attached to their waist
  - ✓ *walking, standing going downstairs, going upstairs, sitting, lying*
  - ✓ smartphone was embedded with a 3-axis accelerometer and a 3-axis gyroscope and worked at a 50 Hz sample rate
  - ✓ The streaming sensor readings were divided into segments with a 2.56s half-overlap sliding window
- *SkodaMiCP* contains the sensor signals of ten manipulative gestures performed by the assembly-line worker in a car maintenance environment
  - ✓ *write on notepad (WN), open hood (OH), close hood (CH), check gaps on the front (CG), open left front door (OL), close left front door (CL), close both left door (CB), check trunk gaps(CT), open and close trunk (OCT), checking steering wheel (CSW)*
  - ✓ collected for about three hours with USB sensors placed on the right and left lower and upper arm
  - ✓ each USB sensor is a 3-axis accelerometer working at a 64 Hz
  - ✓ the data were divided into 1s segments with 50% overlap between two adjacent windows

- Tree-based model and graph-based model are general frameworks that can take as the building blocks various classification models
  - ✓ homogeneous mode: use the same classification model at the top level and the second level
    - homogeneous tree-based model (HoT)
    - homogeneous graph-based model (HoG)
  - ✓ heterogeneous mode: use different classification models in the two levels
    - heterogeneous tree-based model (HeT)
    - heterogeneous graph-based model (HeG)

- Use four classification models that have different metrics
  - ✓ naïve Bayes (NB), k nearest neighbor with k = 1 (KNN), decision tree (DT), support vector machine (SVM)

- Performance metrics
  - ✓ Precision, recall
  - ✓ F1, g-mean

$$F1 = \frac{2 * precision * recall}{precision + recall} \qquad G - mean = \sqrt{\prod_{i=1}^{C} recall_i}$$

Recognition Performance (Confusion matrix-based model)

- Tree-based model has mixed results. Specifically, HeT outperforms HoT on UCI-HAR, while HoTperforms better than HeT on SkodaMiCP
- Graph-based model, HeG consistently performs better than HoG
- In terms of the tree-based model and flat model, the flat model achieves a higher recognition rate in some cases. **The main reason is that tree-based model probably induces compounding errors**
- **Graph-based model obtains consistently better generalization ability**

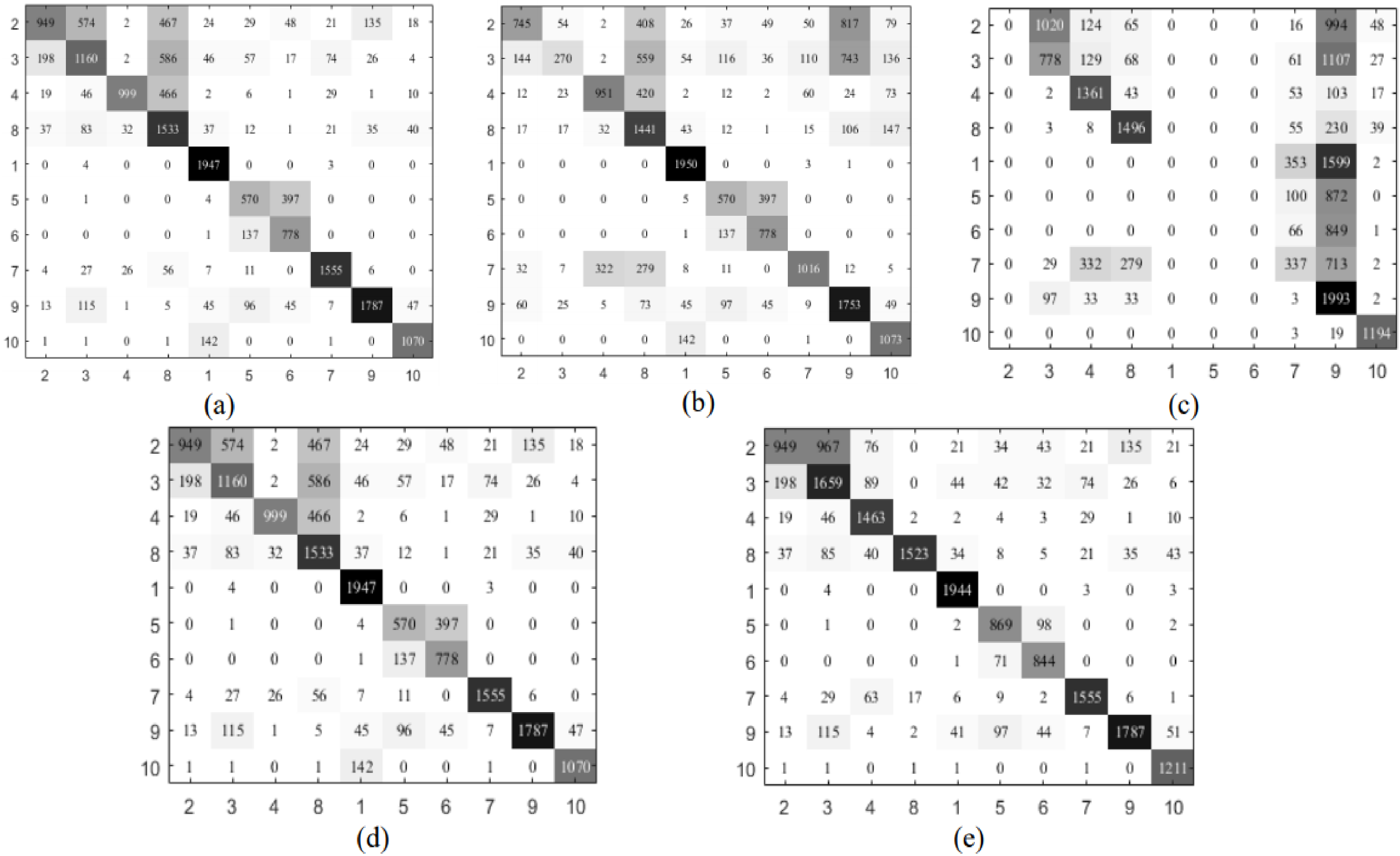
TABLE III
RECOGNITION PERFORMANCE ON UCI-HAR OF FLAT, TREE-, AND GRAPH-BASED MODELS.

Classifier		**NB**				**KNN**				**DT**				**SVM**			
Metrics (%)		*Acc*	*Prec*	*F1*	*Gm*	*Acc*	*Prec*	*F1*	*Gm*	*Acc*	*Prec*	*F1*	*Gm*	*Acc*	*Prec*	*F1*	*Gm*
	Flat	76.99	41.03	55.10	79.63	87.85	58.54	72.55	90.74	86.36	55.82	69.17	88.14	96.40	82.83	82.83	82.83
	HoT	76.86	40.86	54.95	79.54	87.85	58.54	72.55	90.74	86.29	55.71	68.97	87.94	96.44	82.97	90.36	97.52
AR	HeT	**95.69**	**80.00**	**88.57**	**97.07**	**96.40**	**82.83**	**90.28**	**97.50**	**96.44**	**82.97**	**90.36**	**97.52**	**96.44**	**82.97**	**90.36**	**97.52**
	HoG	76.99	41.03	55.10	79.63	85.34	53.62	68.70	89.21	83.88	51.55	59.57	78.16	96.40	82.83	90.28	97.50
	HeG	95.11	77.76	87.26	96.79	89.72	62.78	75.78	91.98	95.42	80.13	87.67	95.96	96.40	82.83	90.27	97.50

TABLE IV
RECOGNITION PERFORMANCE ON SKODAMICP OF FLAT, TREE-, AND GRAPH-BASED MODELS.

Classifier		**NB**				**KNN**				**DT**				**SVM**			
Metrics (%)		*Acc*	*Prec*	*F1*	*Gm*	*Acc*	*Prec*	*F1*	*Gm*	*Acc*	*Prec*	*F1*	*Gm*	*Acc*	*Prec*	*F1*	*Gm*
	Flat	73.68	30.66	46.90	83.67	78.83	34.05	48.90	82.16	92.91	62.37	76.54	95.52	25.52	0	-	0
	HoT	62.94	23.91	38.57	76.13	72.49	28.23	42.77	78.79	92.47	60.95	75.38	95.14	12.79	0	-	0
AR	HeT	42.72	0	-	0	12.86	0	-	0	35.76	0	-	0	12.79	0	-	0
	HoG	73.68	30.66	46.90	83.67	79.25	34.53	49.40	82.41	93.08	62.93	76.96	95.62	25.52	0	-	0
	HeG	**82.37**	**39.78**	**56.84**	**89.28**	**82.24**	**38.39**	**53.10**	**83.89**	**93.23**	**63.46**	**77.35**	**95.71**	**25.52**	0	-	0

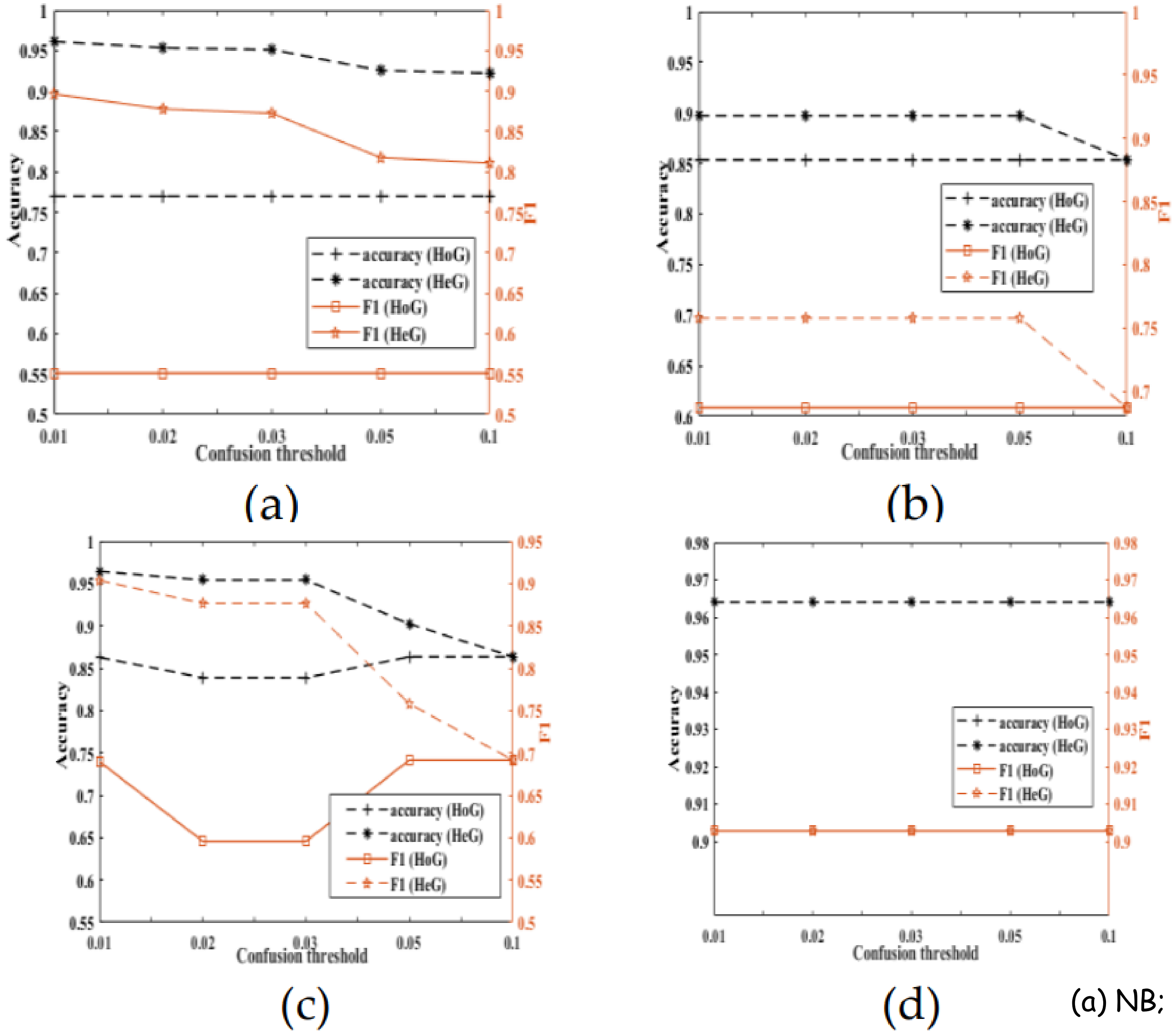
Confusion Matrix



Confusion matrix on SkodaMiCP with NB used at the top-level. (a) Flat; (b) HoT; (c) HeT; (d) HoG; (e) HeG.

Evaluation of Hyperparameter (UCI-HAR)

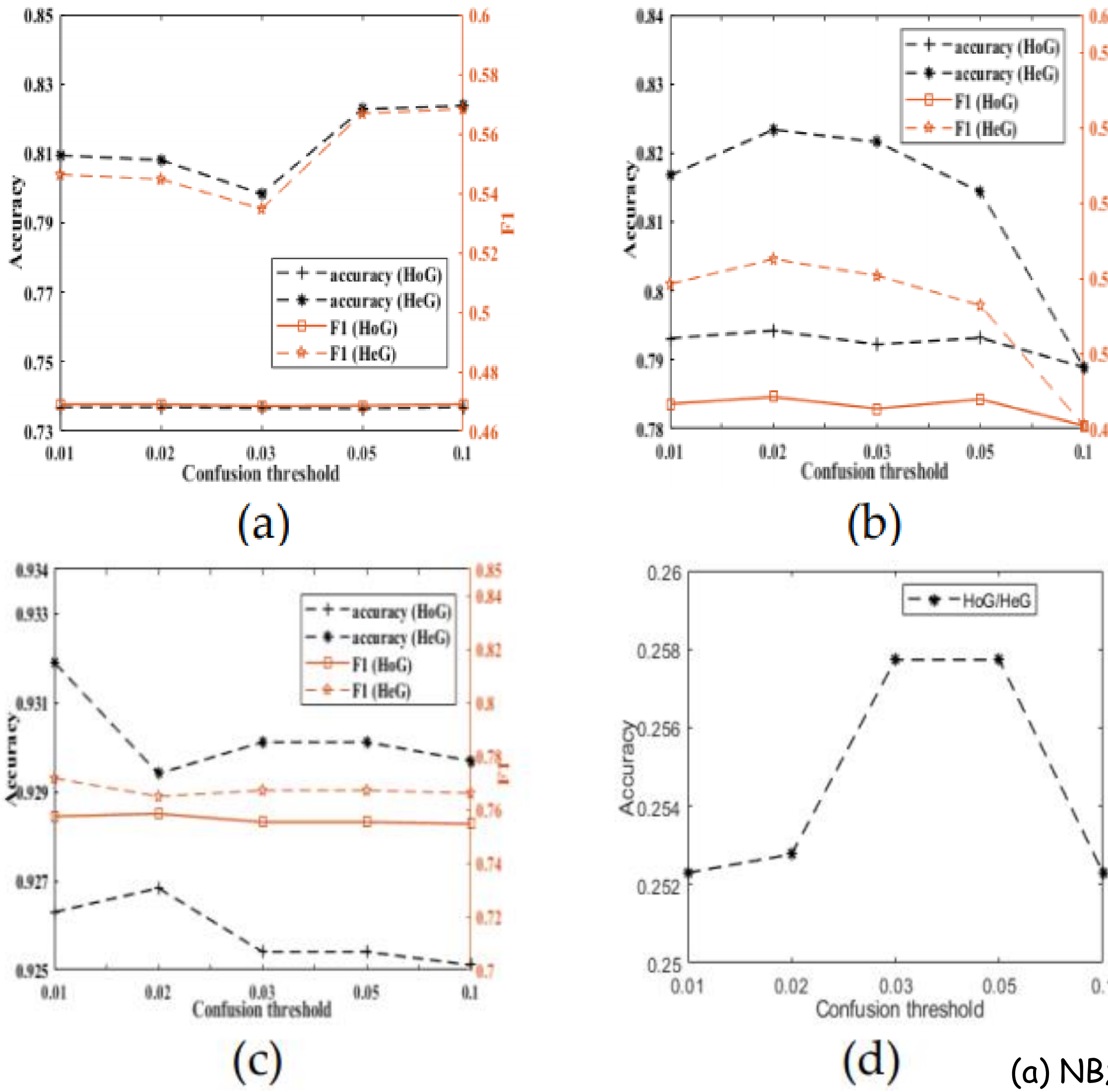
- The candidate values of θ include 0.01, 0.02, 0.03, 0.05, and 0.1
- 3% is a reasonable choice and the graph-based model works well in the majority of cases



(a) NB; (b) KNN; (c) DT; (d) SVM.

Evaluation of Hyperparameter (SkodaMiCP)

- The candidate values of θ include 0.01, 0.02, 0.03, 0.05, and 0.1
- 3% is a reasonable choice and the graph-based model works well in the majority of cases



(a) NB; (b) KNN; (c) DT; (d) SVM.

Evaluation of the Combination of Classifiers

- Use NB, KNN, DT, or SVM at the top level and use NB, KNN, DT or SVM at the second level
- For UCI-HAR, we observe that the use of SVM at the second level generally outperforms its competitors. For SkodaMiCP, the homogeneous model is inferior to that of the heterogeneous model that uses SVM at the second level

PERFORMANCE ON UCI-HAR WITH THE COMBINATION OF DIFFERENT CLASSIFIERS.

Classifier	NB-NB		NB-KNN		NB-DT		NB-SVM		KNN-NB		KNN-KNN		KNN-DT		KNN-SVM	
Metrics (%)	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
HoT/HeT	76.86	54.95	87.17	71.45	85.48	67.71	**95.69**	88.57	77.30	55.43	87.85	72.55	86.22	68.87	**96.40**	90.28
HoG/HeG	76.99	55.10	86.87	71.10	85.34	67.81	95.11	87.26	86.39	70.27	85.34	68.70	85.95	69.60	89.72	75.78
Classifier	DT-NB		DT-KNN		DT-DT		DT-SVM		SVM-NB		SVM-KNN		SVM-DT		SVM-SVM	
Metrics (%)	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
HoT/HeT	77.30	55.43	87.85	72.55	86.29	68.97	**96.44**	90.36	77.30	55.43	87.85	72.55	86.29	68.97	**96.44**	90.36
HoG/HeG	86.70	68.49	88.60	73.63	83.88	59.57	95.42	87.67	92.78	82.62	92.64	81.93	92.50	81.66	96.40	90.28

PERFORMANCE ON SKODAMICP WITH THE COMBINATION OF DIFFERENT CLASSIFIERS.

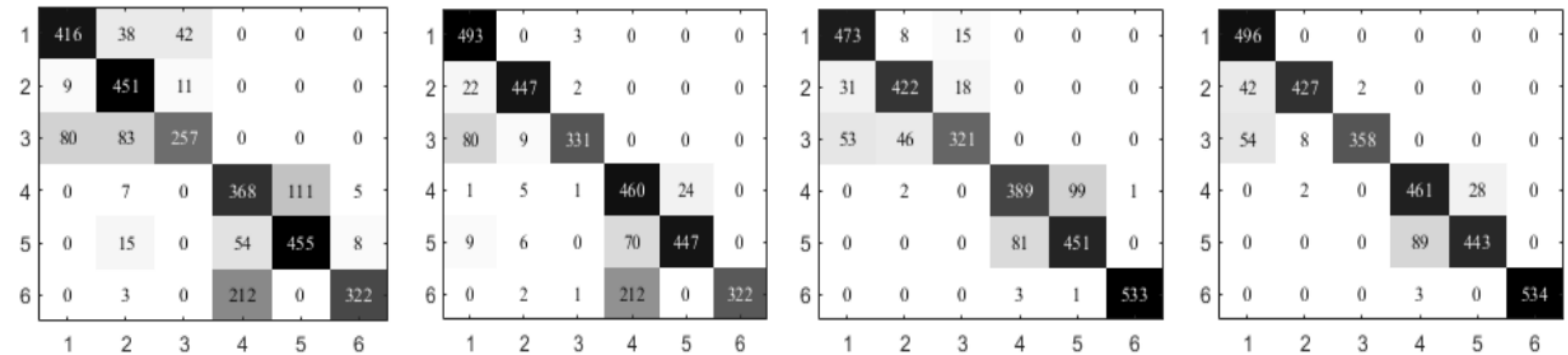
Classifier	NB-NB		NB-KNN		NB-DT		NB-SVM		KNN-NB		KNN-KNN		KNN-DT		KNN-SVM	
Metrics (%)	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
HoT/HeT	62.94	38.57	66.80	38.84	74.82	47.72	42.72	-	67.28	41.09	72.60	42.74	85.42	60.80	12.86	-
HoG/HeG	73.68	46.90	85.58	61.08	**89.93**	69.75	79.97	53.69	74.93	45.00	79.30	49.82	**87.01**	60.98	82.24	53.10
Classifier	DT-NB		DT-KNN		DT-DT		DT-SVM		SVM-NB		SVM-KNN		SVM-DT		SVM-SVM	
Metrics (%)	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
HoT/HeT	73.92	47.05	82.08	53.48	92.53	75.49	35.76	-	66.73	41.10	72.62	42.92	84.61	59.89	12.79	-
HoG/HeG	84.72	60.05	91.55	73.11	**92.73**	75.97	93.23	77.35	30.40	-	25.23	-	**92.43**	75.31	43.32	-

Recognition Performance (Clustering guided model)

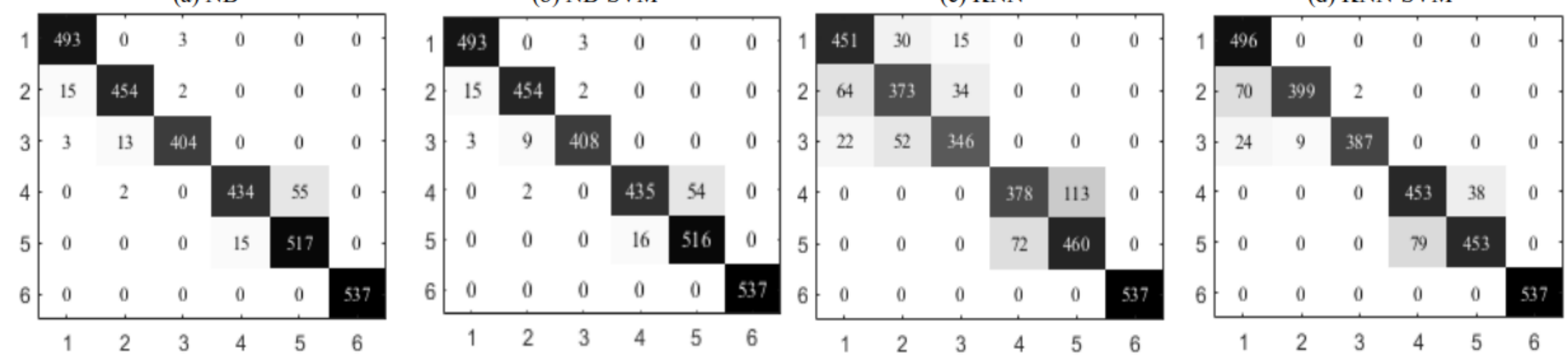
- Use k-means with Euclidean distance

RECOGNITION PERFORMANCE ON THE UCI-HAR DATASET.

Classifier	NB	NB-NB	NB-KNN	NB-SVM	NB-DT	KNN	KNN-NB	KNN-KNN	KNN-SVM	KNN-DT
Accuracy	76.99	76.99	78.86	**84.83**	77.03	87.85	78.72	87.85	**92.26**	85.31
Precision	76.88	76.88	78.90	**85.13**	77.08	87.44	78.63	87.44	**92.08**	84.80
Recall	79.23	79.23	83.48	**88.60**	81.09	87.96	82.86	87.96	**93.06**	85.98
F1	78.04	78.04	81.12	**86.83**	79.03	87.70	80.69	87.7	**92.57**	85.39
Classifier	**SVM**	**SVM-NB**	**SVM-KNN**	**SVM-SVM**	**SVM-DT**	**DT**	**DT-NB**	**DT-KNN**	**DT-SVM**	**DT-DT**
Accuracy	96.34	82.32	90.43	**96.47**	89.79	86.36	77.10	85.88	**92.47**	86.36
Precision	96.26	82.16	90.01	**96.42**	89.40	85.99	76.95	85.40	**92.38**	85.99
Recall	96.52	85.58	90.91	**96.65**	89.80	86.31	80.78	86.85	**93.13**	86.31
F1	96.39	83.84	90.46	**96.53**	89.60	86.15	78.82	86.12	**92.75**	86.15



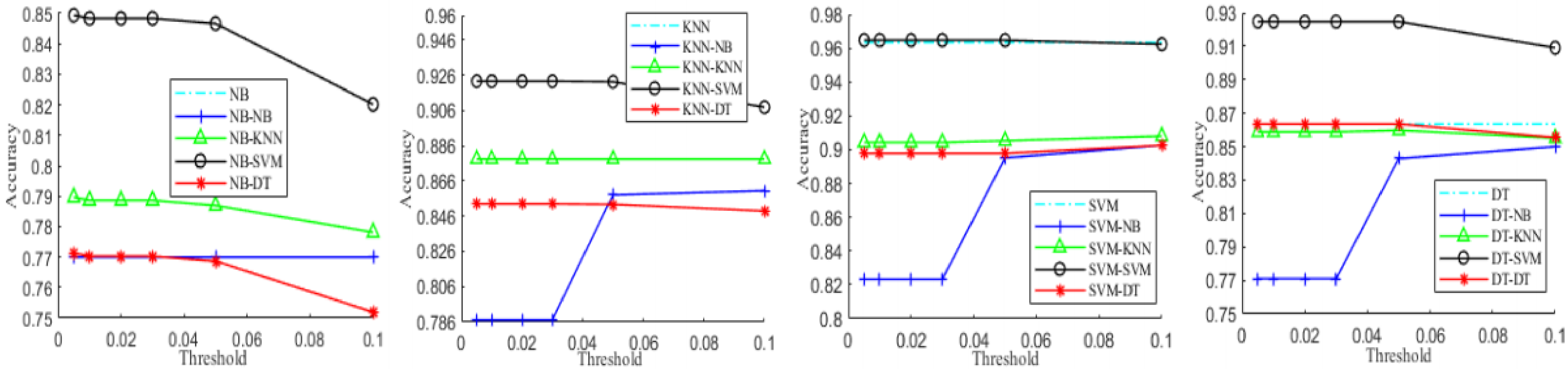
(a) NB (b) NB-SVM (c) KNN (d) KNN-SVM



(e) SVM (f) SVM-SVM (g) DT (h) DT-SVM

Evaluation of Hyperparameter

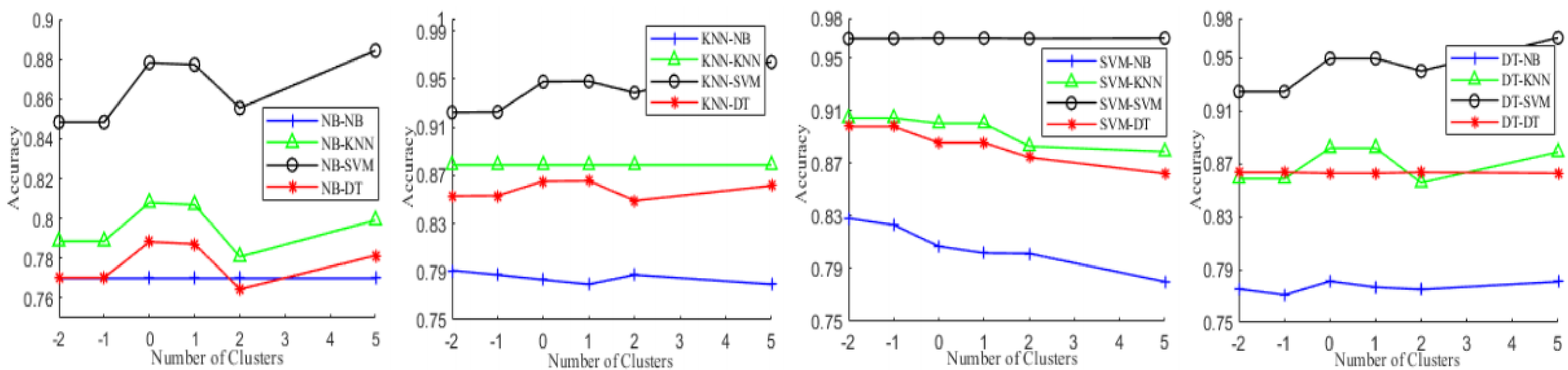
- Confusion threshold



(a) (b) (c) (d)

- Number of clusters

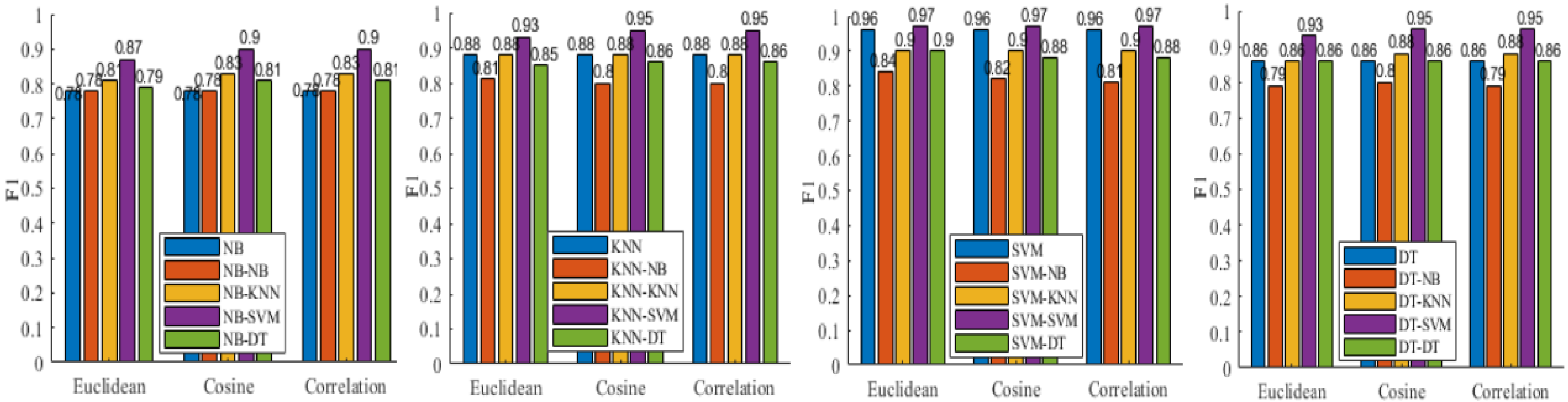
(a) NB; (b) KNN; (c) DT; (d) SVM.



(a) (b) (c) (d)

Evaluation of Hyperparameter

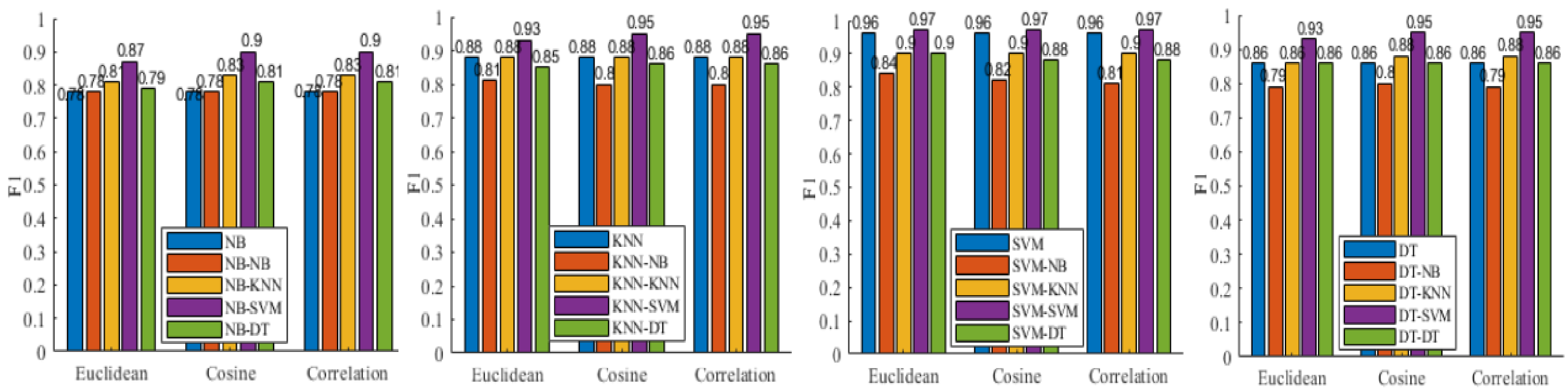
- Different distance metrics



(a) (b) (c) (d)

- Different clustering algorithms

(a) NB; (b) KNN; (c) DT; (d) SVM.



(a) (b) (c) (d)

Conclusion and Future Work

- Conclusion
  - How to better discriminate activities with (triggered) similar sensor readings
  - Present two different data-driven methods to build hierarchical human activity recognition model, i.e., confusion matrix-based method & clustering guided method
  - Tree-based model and graph-based model are presented
  - Conduct extensive comparative experiments
- Future work
  - Human behavior itself driven research work

References

- A. Wang, et al. “Towards human activity recognition: A hierarchical feature election framework,” Sensors, vol. 18, p. 3629, 2018.
- A. Wang, et al. "HierHAR: Sensor-based data-driven hierarchical human activity recognition." IEEE Sensors J., vol. 21, pp. 3353-3365, 2021.
- A. Wang, et al. “Towards a clustering guided hierarchical framework for sensor-based activity recognition." Sensors, vol. 21, p. 6962, 2021.
- A. Wang, et al. "Activities of daily living recognition with binary environment sensors using deep learning: A comparative study." IEEE Sensors J., vol. 21, pp. 5423-5433, 2021.
- P. Dawadi, D. Cook, and M. Schmitter-Edgecombe, “Automated cognitive health assessment from smart home-based behavior data,” IEEE J. Biomed. Health. Inform., vol. 20, pp. 1188-1194, 2015.
- O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” IEEE Commun. Surv. Tut., vol. 15, pp. 1192-1209, 2012.
- J. H. Hong, J. Ramos, and A. K. Dey, “Toward personalized activity recognition systems with a semipopulation approach,” IEEE Trans. Human Mach. Syst., vol. 46, pp. 101-112, 2015.
- J. Rafferty, C. D. Nugent, J. Liu, and L. Chen, “From activity recognition to intention recognition for assisted living within smart homes,” IEEE Trans. Human Mach. Syst., vol. 47, pp. 368-379, 2017.
- L. Chen, C. D. Nugent, and H. Wang, “A knowledge-driven approach to activity recognition in smart homes,” IEEE Trans. Knowl. Data Eng., vol. 24, pp. 961–974, 2012.
- N. C. Krishnan and D. J. Cook, “Activity recognition on streaming sensor data,” Pervasive Mob. Comput., vol. 10, pp. 138-154, 2014.
- J. Yang, M. Nguyen, P. San, X. L. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” IJCAI, 2015.
- H. Cho and S. Yoon, “Divide and conquer-based 1D CNN human activity recognition using test data sharpening,” Sensors, vol. 18, p. 1055, 2018.
- H. Ma, et al. "AttnSense: Multi-level attention mechanism for multimodal human activity recognition," IJCAI. 2019.

Thank you

For Your Attention