

Effect of Combinations of Sensor Positions on Wearable-sensor-based Human Activity Recognition

Yuhao Duan¹ and Kaori Fujinami^{1,2*}

¹Graduate School of Bio-Applications and Systems Engineering,

Tokyo University of Agriculture and Technology, 2-24-16 Naka-cho, Koganei, Tokyo 184-8588, Japan

²Division of Advanced Information Technology and Computer Science, Institute of Engineering, Tokyo University
of Agriculture and Technology, 2-24-16 Naka-cho, Koganei, Tokyo 184-8588, Japan

(Received April 7, 2023; accepted June 19, 2023)

Keywords: activity recognition, wearable sensors, accelerometer, on-body sensor position

Human activity recognition (HAR) has attracted widespread attention in areas such as human–computer interaction, work performance management, and healthcare. Owing to advantages such as continuous monitoring, reduced cost of deployment, and ease of privacy protection, wearable-sensor-based HAR is preferred over the traditional approach of using external sensors. In this study, the influence of different combinations of seven body-worn accelerometer positions on the classification of 23 complex daily activities was examined. A conventional machine learning model, namely, RandomForest (RF), and two deep-learning (DL) models, convolutional neural network (CNN)-long short-time memory (LSTM) and CNN-transformer, were used to understand the impact of using different models on the classification performance. The results showed a strong correlation between the classification models regarding the combinations of sensor positions and classification performance (F1-score). Additionally, the combination of the four sensors from the left and right wrists, right upper arm, and right thigh was determined to be the best. This study also showed that, owing to feature calculation, the RF model took a longer processing time than the DL-based models and that the CNN-LSTM model would be preferable to RF if plenty of data were available for training it. The results can provide a reference for application designers in choosing appropriate combinations of sensor positions based on requirements for wearability and classification performance.

1. Introduction

In recent years, several studies have been conducted in the field of human activity recognition (HAR), which has been widely used in human–computer interaction,⁽¹⁾ work performance management,⁽²⁾ and healthcare.⁽³⁾ Data acquisition methods in HAR can be divided into two types: external-sensor-based HAR and wearable-sensor-based HAR (WHAR).⁽⁴⁾ In external-sensor-based HAR, system designers arrange cameras in the locations where users perform activities⁽⁵⁾ or sensors in the environment,⁽⁶⁾ for example, on furniture and floors. However, the approach inherently has three drawbacks: 1) external sensors are usually large and power-consuming, which may incur high costs for installation and maintenance; 2) external sensors are

*Corresponding author: e-mail: fujinami@cc.tuat.ac.jp

<https://doi.org/10.18494/SAM4421>

not suitable for long-term, continuous recording of human activities, because the activities cannot be recognized once the users leave the place where the sensors are installed; and 3) devices such as cameras and microphones can infringe on the user's privacy. In WHAR, sensors are attached to the user or carried by the user, which allows for the continuous recording of human activity. In addition, with the popularity of smartphones and smartwatches equipped with inertial measurement units in daily life, WHAR system designers can use user-owned devices in their systems, which reduces the cost of system deployment. The WHAR system also has little impact on the feeling of privacy violation because the user can take control of the sensors and applications that use their data. The user can remove the sensor or turn off the application if they do not want their activities recorded. Owing to these advantages, WHAR systems have been favored by many researchers in recent years.

In recent studies, in the field of WHAR, accelerometers have been shown to be effective in determining behavioral characteristics.⁽⁷⁾ Therefore, accelerometers have been used in many WHAR systems. However, the positioning of an accelerometer on the user's body remains unresolved. As mentioned in Ref. 8, significant differences exist in the amplitudes of the acceleration signals at different positions of the body, even in the same activity. Wearable-sensor-based application designers often follow experience or subjective judgments to decide sensor placement locations for a particular set of activities. However, this approach may fail if ineffective positions are selected, in which case, ineffective motion or posture signatures might be recorded, resulting in poor system performance. The difficulty in sensor position selection is that the best placement of the sensor is not necessarily where the movement is most apparent, as discussed in Ref. 9. In a study on gait detection in limb injuries, the results showed that the head provided the best classification feature for gait rather than the legs,^(10,11) which demonstrates the difficulty for a system designer to find the best location for the sensor based on subjective judgment. For activity recognition, the selection of number of sensors and their positions remains unresolved and requires further research.

In addition, almost all studies have considered only conventional machine learning (ML)-based sensor position placement strategies. The classification accuracy of each position is highly dependent on the features employed by the researchers, and the manually determined features do not generalize the classification performance of each position, which may affect one's judgment of the importance of the sensor position. Deep learning (DL) can extract the deep features of a sensor, which can reduce classification inaccuracies caused by insufficient information from manually designed features and better reflect the differences in information between the positions themselves. However, a performance comparison between DL and conventional ML using the same position may be worth exploring.

In this study, we evaluated the classification performance of different combinations of sensor positions by conducting experiments using daily life activity data. We applied and compared three types of classification model: a conventional ML-based model with classification feature engineering and two DL-based models with feature learning. The processing performance and processing time were compared. The results are expected to contribute to the determination of appropriate positions and combinations of sensors and to the selection of a classification model for the complex activities of daily life.

2. Materials and Methods

2.1 Overview of experiment

We performed an offline experiment that aims to provide wearable-sensor-based application designers with useful information to choose an appropriate classification method and specify both desirable and undesirable sensor positions for complex daily activity recognition. In Sect. 2.2, a dataset consisting of 23 complex activities of daily life (CADL) collected from 14 young adults who wore seven accelerometers is described. Three classification models were used: a conventional ML-based model (Sect. 2.3.1) and two DL-based models (Sect. 2.3.2). These three models were compared in terms of their tendency toward effective sensor-position combinations, classification performance, and processing time per window.

The classification performances for all the sensor combinations were obtained, in which 127 ($= \sum_{i=1}^7 C_i$) combinations of sensor positioning were tested. For the performance measure, we used the F1-score, which is the harmonic mean between recall and precision.

To implement the conventional ML-based method, we utilized the Weka 3.10 machine learning toolkit. In contrast, scikit-learn 0.24.2 and PyTorch 1.10.1 were used to implement the DL-based methods. The evaluation was run on an 11th generation Intel Core i9-11900K CPU with an NVIDIA GeForce RTX3080Ti GPU.

2.2 Dataset

A dataset collected from the laboratory of the authors was used. The dataset consists of three-axis acceleration data for 23 daily life activities from seven positions on the bodies of 14 volunteers (five females and nine males between the ages of 22 and 25 years, all right-handed). Figure 1 shows (a) the sensor placement and (b)–(x) snapshots of the activities. Six of the seven sensor nodes (ATR Promotions Inc., TSND151) were attached to the upper arms, wrists, and thighs for symmetry, whereas one node (TSND121) was placed on the chest. All the sensor nodes were securely attached to the body with a band. The sensor nodes on the upper arms and wrists were worn such that they could be on the outside of the body. Each sensor node has a real-time clock (RTC) synchronized using the clock on a data-collection personal computer. The major difference between TSND121 and TSND151 is the six-axis motion (accelerometer and gyro) sensing unit, that is, the Invensense MPU-6050 and MPU-9250 for TSND121 and TSND151, respectively. In this study, we only used accelerometers by setting the measurement range to $\pm 19.62 \text{ m/s}^2$ and believe that the effect of this difference is minimized by the placement of the same sensor, TSND151, in symmetrical positions on the body. Note that the main reason for using the TSND series is that the data collection experiments, including synchronization of time between sensor nodes, can be managed on a single personal computer using dedicated data recording software. This allowed rapid data collection and subsequent analysis.

The activities included not only simple activities such as walking and running but also complex upper-limb activities such as making coffee and vacuum cleaning, which are frequently performed in daily life. The subjects performed various activities for approximately 12 min each

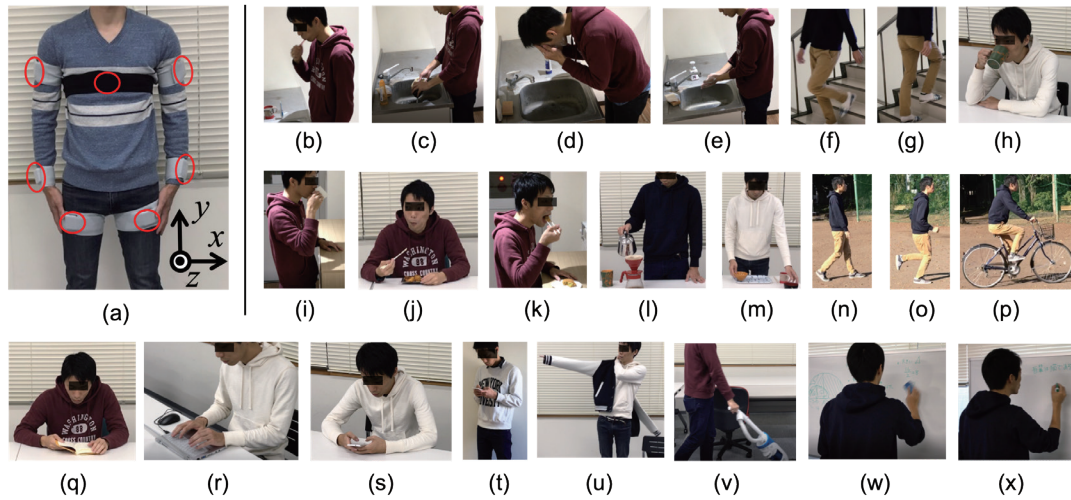


Fig. 1. (Color online) Sensor placement and activities in the dataset: (a) placement of three-axis accelerometer on the body, (b) brushing teeth (BT), (c) washing dishes (WD), (d) washing face (WF), (e) washing hands (WH), (f) going down stairs (DS), (g) going up stairs (US), (h) having a drink while sitting (DK_SIT), (i) having a drink while standing (DK_STD), (j) eating food while sitting (ET_SIT), (k) eating food while standing (ET_STD), (l) making coffee (MC), (m) setting table (ST), (n) walking (WK), (o) running (RN), (p) riding a bike (BK), (q) reading a book (RB), (r) typing on a keyboard while sitting (TY), (s) using a smartphone while sitting (SP_SIT), (t) using a smartphone while standing (SP_STD), (u) wearing and taking off a jacket (WJ), (v) vacuum cleaning (VC), (w) erasing figures on a whiteboard (EW), and (x) writing figures on a whiteboard (WW).

in the way they usually do. Note that this does not indicate a continuous 12-min session but the total time of several separate sessions. The acceleration signal sampling rate was 50 Hz. Data were collected for approximately 64 h (12 min \times 23 activities \times 14 persons). Notably, the dataset was balanced for both activity classes and individuals, with 10268.3 s [standard deviation (*SD*): 333.9 s] per activity and 16871.1 s (*SD*: 332.9 s) per individual. Therefore, the training of a classifier is less likely to be affected by specific activities or individuals regarding the bias in the number of data sets.

2.3 Experimental methods

2.3.1 Conventional ML-based method

Conventional ML comprises two parts: feature extraction and a classification model. The features that could characterize the motions of various activities were calculated from the raw acceleration signals. In addition to the three axes of an accelerometer, i.e., x , y , and z , we introduced the magnitude of the acceleration signal (m) as the fourth dimension [Eq. (1)], where $i \in \{1, \dots, N\}$, and N indicates the number of samples in a calculation window.

$$m_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \quad (1)$$

A total of 39 features were defined by the four axes of the acceleration signal in the time and frequency domains of each sensor, as summarized in Table 1. These features are frequently used for on-body HAR and device localization.^(9,12–15) A window size of 256 ($N = 256$) was selected. The importance of the features at each position was then evaluated using Relief⁽¹⁶⁾ that evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same or different classes. We confirmed that, for each position, adding more features did not significantly improve classification performance when the number of features exceeded 19. Thus, 19 features were used for each position to examine the combination of positions, as listed in Table 2. The table also shows the effective features for each position. Among them, F3, F10, F15, F19, F30, and F38 were selected at any position and are thus effective features regardless of the position.

For each sensor combination, an activity was characterized by $19 \times K$ features when K sensors were used. We used RandomForest (RF) as the classification algorithm for the conventional method because RF has been shown to exhibit good classification performance in WHAR tasks.^(17,18)

2.3.2 Deep-learning-based methods: CNN-LSTM

Several WHAR studies based on DL models have emerged in recent years. The most frequently used network layers are convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM). Recently, hybrid models have also been used. CNN and LSTM have been used in combination to outperform CNN alone.^(19–22) In a

Table 1
Candidates of classification features.

Feature	Description and definition
F1, F2, F3, F4	Mean value in time domain data, i.e., $\bar{a} = \frac{1}{N} \sum_{i=1}^N a_i$
F5, F6, F7, F8	Variance in time domain data, i.e., $v_a = \frac{1}{N-1} \sum_{i=1}^N (a_i - \bar{a})^2$
F9, F10, F11, F12	Standard deviation in time domain data, i.e., $\sqrt{v_a}$
F13, F14, F15, F16	1st quartile (1/4 smallest value) in time domain data
F17, F18, F19, F20	3rd quartile (3/4 smallest value) in time domain data
F21, F22, F23, F24	1st quartile (1/4 smallest value) in frequency spectrum data
F25, F26, F27, F28	3rd quartile (3/4 smallest value) in frequency spectrum data
F29, F30, F31, F32	Frequency entropy, i.e., $-\sum_{i=1}^{N/2} p_{a,i} \times \log_2 p_{a,i}$, where $p_{a,i} = f_{a,i}^2 / \sum_{i=1}^{N/2} f_{a,i}^2$
F33, F34, F35, F36	Sum of energy spectrum, i.e., $\sum_{i=1}^{N/2} f_{a,i}^2$
F37, F38, F39	Pearson's correlation coefficient of signal from two axes s and t , i.e., $cov_{st}/v_s v_t$, where F37, F38, and F39 correspond to the values between axes x and y , x and z , and y and z , respectively.

Note: $a \in \{x, y, z, m\}$. $f_{a,i}$ indicates the value of the i th smallest frequency component for axis a . The four features in each feature category, e.g., F1, F2, F3, and F4, correspond to x , y , z , and m in this order. The term cov_{st} indicates the covariance between signals from axes s and t .

Table 2

Selected 19 features in each position; the number indicates the order (relevance) of adding to feature subset.

	LT	LW	LU	C	RU	RW	RT		LT	LW	LU	C	RU	RW	RT
F1		11	6		11	9		F21	19						
F2	5	1				1	7	F22	16						14
F3	8	5	7	4	3	3	1	F23							16
F4								F24	13						
F5								F25	14			10	10		19
F6								F26	7	19	18	6	9		5
F7								F27	18			16	18		8
F8								F28	6		17	5	8		10
F9			9	13	14			F29	12	18	19	2		17	17
F10	9	17	16	12	15	15	9	F30	15	13	11	3	17	18	18
F11		16	14	17	19	14		F31	3		15	14	13	19	15
F12		10	12	9	16	7		F32	17	12	8	1		16	
F13		15	10		7	10		F33							
F14	1	4		19		5	4	F34							
F15	2	9	5	11	4	11	2	F35							
F16								F36							
F17		14	4		12	13		F37		3	2	7	6	4	13
F18	4	6				2	6	F38	11	7	3	18	2	8	11
F19	10	8	13	15	5	12	3	F39		2	1	8	1	6	12
F20															

C: chest, LT: left thigh, LU: left upper arm, LW: left wrist, RT: right thigh, RU: right upper arm, RW: right wrist

study using multiple sensors,⁽²²⁾ multiple convolutional subnetworks were used to collect features from each sensor, which were then integrated into the depth concatenation layer. Finally, classification was performed in the output layer after collecting the temporal features through the LSTM layer. A similar CNN subnetwork design was also applied in Ref. 23 and was shown to be effective in separately extracting the information provided by multiple sensors. Because the DEBONAIR model proposed in Ref. 22 achieved a higher accuracy of 83% on the CADL dataset, the CNN-LSTM model in this study was built with reference to the architecture of DEBONAIR, as shown in Fig. 2. For each sensor, a convolutional subnet containing three convolutional and three pooling layers was used to extract information, which was integrated into the depth concatenation layer and subjected to a convolution operation. Then, the time features were extracted from the data using a two-layer LSTM, and finally, classification was performed using a softmax function on a fully connected layer.

A preliminary experiment showed that the optimizer and learning rate affected the classification performance among the other hyperparameters. These hyperparameters were tuned for each sensor combination. The hyperparameters considered are listed in Table 3.

2.3.3 Deep-learning-based methods: CNN-transformer-based method

The transformer model based on multihead attention has been proven to be highly advantageous in recent years for handling sequence analysis tasks. Shavit and Klein used the transformer encoder for WHAR tasks⁽²⁴⁾ for the first time, and the results showed that the

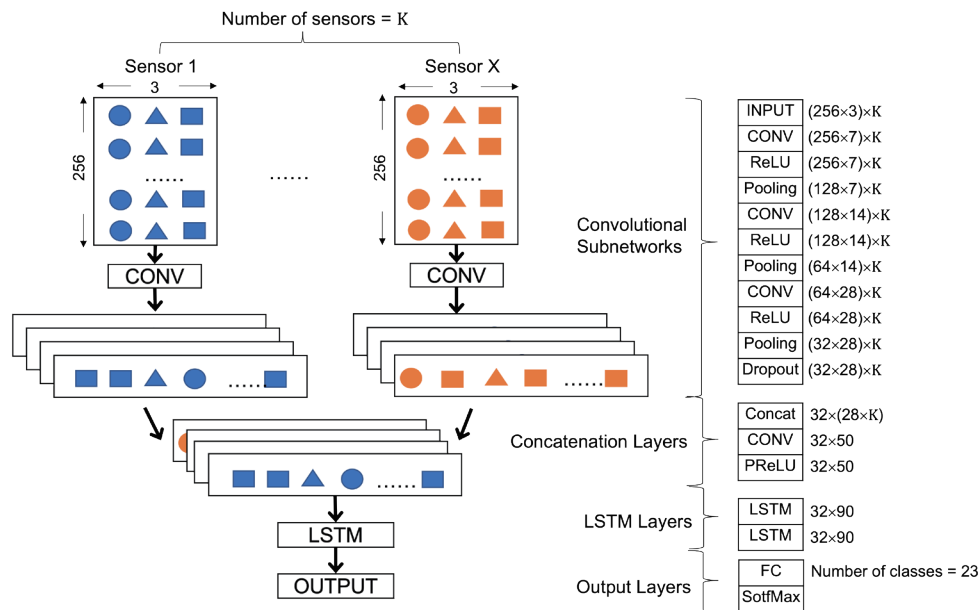


Fig. 2. (Color online) Architecture of CNN-LSTM-based classification.

Table 3

Hyperparameters considered and used in the implementation of CNN-LSTM classifier. The hyperparameter names with ‘_’ are the ones used in PyTorch. The underlined values represent the chosen values in a preliminary experiment.

Layer	Hyperparameter name	Range or candidates value(s)
Convolutional Subnetworks	First CNN out_channels	5, 6, <u>7</u>
	First CNN kernel_size	<u>7</u> , 9, 11
	First CNN activation function	ReLU
	First CNN Max pooling kernel_size	2
	Second CNN out_channels	14
	Second CNN kernel_size	5, <u>7</u> , 9
	Second CNN activation function	ReLU
	Second CNN Max pooling kernel_size	2
	Third CNN out_channels	28
	Third CNN kernel_size	<u>3</u> , 5, 7
	Third CNN activation function	ReLU
	Third CNN Max pooling kernel_size	2
	CNN dropout rate	<u>0.1</u> , 0.2, 0.3, 0.4, 0.5
Concatenation	Concatenation CNN out_channels	40, <u>50</u> , 60
	Activation function	PReLU
LSTM	hidden_size	30, 40, 50, 60, 70, 80, <u>90</u>
	Optimizer	SGD, Adam, RMSprop
General	Learning rate	Log domain in ranging from 0.00001 to 0.1
	Batch size	64, <u>128</u> , 192, 256

transformer encoder improved classification performance. In this work, the CNN-transformer model was adopted to examine the sensor position combination with reference to the model proposed by Shavit and Klein.

Figure 3 illustrates the architecture. First, each sensor data point was integrated using the time dimension, and then token embedding and position embedding operations were performed on the data. Subsequently, the self-attention value was calculated for each vector using the transformer encoder, and the class token embedded in the token embedding was used to classify the data using a softmax function on the fully connected layer. The hyperparameters are listed in Table 4 and are based on the specific values or calculation methods used in the model of Shavit and Klein.⁽²⁴⁾

2.4 Evaluation method

Classification performance was evaluated by cross-validation (CV) of the training and test data. We chose leave-one-person-out (LOPO) CV as the primary CV method, which was performed by testing a dataset from a particular person with a classifier that was trained without

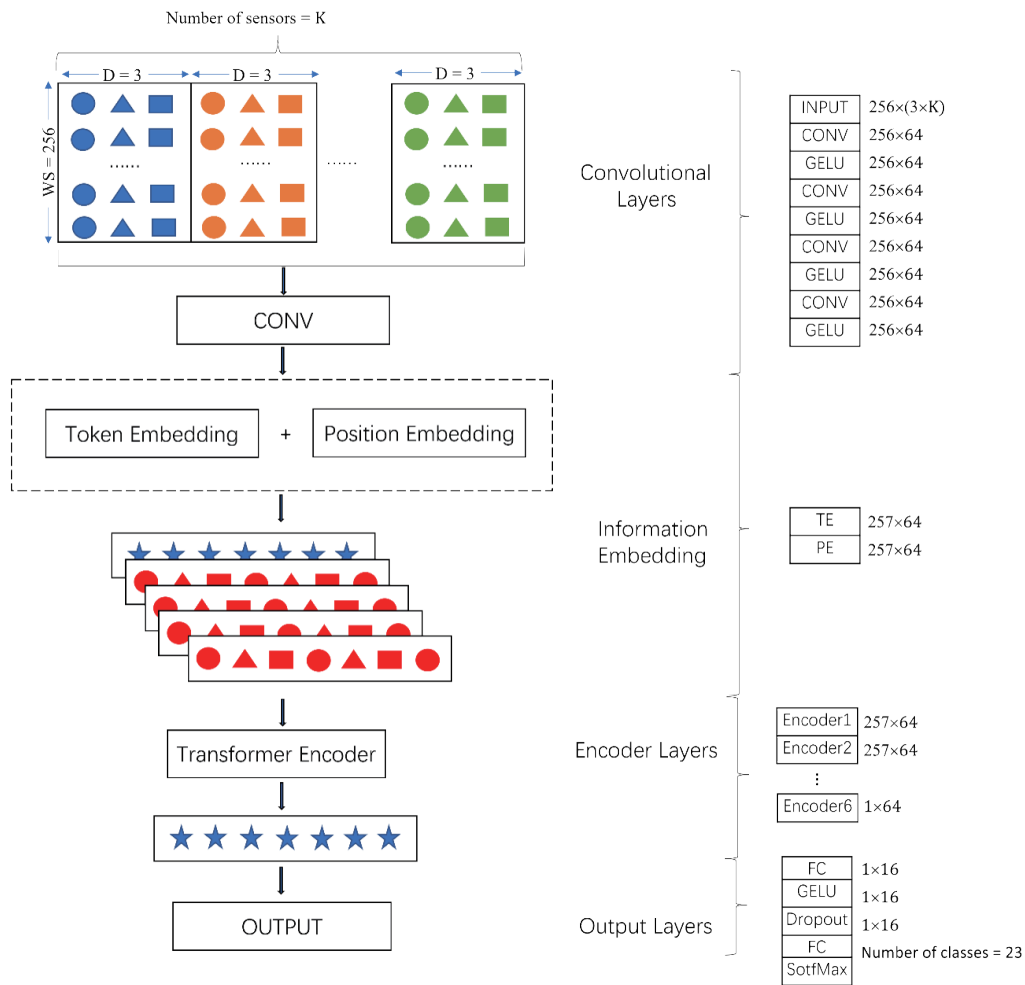


Fig. 3. (Color online) Architecture of CNN-transformer-based classification.

Table 4

Hyperparameters used in the implementation of CNN-transformer classifier. The hyperparameter names with ‘_’ are the ones used in PyTorch. The values are the same as those used in Ref. 24.

Layer	Hyperparameter name	Value
Convolution	All CNNs <code>in_channels</code>	64 (= latent dimension “ <i>d</i> ” in Ref. 24)
	All CNNs <code>out_channels</code>	64 (= <i>d</i>)
Information Embedding	Position embedding	256 (= window size “ <i>k</i> ” in Ref. 24) + 1
Encoder	<code>num_layer</code>	6
	<code>n_head</code>	8
Output	First fully connected (FC) <code>in_features</code>	64
	First FC <code>out_features</code>	16 (= <i>d</i> /4)
	Dropout ratio	0.1
	Second FC <code>in_features</code>	16 (= <i>d</i> /4)
	Second FC <code>out_features</code>	23

the data from that person. Training and testing were repeated with different combinations of participants. Because the trained classifier did not contain data from the test participants, LOPO-CV was regarded as a fairer and more practical test method.

In addition, *n*-fold CV was applied in two ways: against a dataset containing data of all participants (*n*-fold CV_all), and averaging the results from *n*-fold CV against datasets consisting of each participant’s data (*n*-fold CV_each). The *n*-fold CV utilizes (*n*−1)/*n* of the dataset for training a classifier and 1/*n* for testing the classifier. The *n*-fold CV_all represents the average classification performance because the classifier knows the participants from (*n*−1)/*n* of their data. In contrast, *n*-fold CV_each has an optimistic performance because each classifier knows nothing except the test participant. We set *n* to 10. Section 3.3 utilizes three evaluation methods. Otherwise, LOPO-CV is used to understand the lower bound of the performance.

The classification performance is evaluated using a macro-average F1-score. An F1-score is a harmonic mean between recall and precision. Equations (2), (3), and (4) define these metrics for class *i*, respectively, where $N_{correct_i}$, N_{tested_i} , and N_{judged_i} represent the number of cases correctly classified into class *i*, the number of test cases in class *i*, and the number of cases classified into class *i*, respectively. A macro-average F1-score is an average of F1-score over 23 classes. Hereinafter, we simply refer to a macro-average F1-score as an F1-score.

$$F1\text{-score}_i = \frac{2}{\frac{1}{recall_i} + \frac{1}{precision_i}} \quad (2)$$

$$recall_i = N_{correct_i} / N_{tested_i} \quad (3)$$

$$precision_i = N_{correct_i} / N_{judged_i} \quad (4)$$

3. Results

3.1 Classification performance comparison in the three methods

The differences among the three classification algorithms were analyzed. Figure 4 shows the maximum classification performance (F1-score) for different numbers of sensors in the three classification models; the sensor combination is also presented. The CNN-LSTM model obtained the highest score for single-sensor usage. When the number of sensors was greater than one, the RF outperformed the two DL models. Comparing the three algorithms for the 127 sensor combinations, we found that the RF model achieved the highest F1-score for 119 sensor combinations. The CNN-LSTM model achieved the highest F1-score for the remaining eight combinations, and in seven of these eight sensor combinations, one sensor was used. Although the CNN-LSTM model outperformed the RF model when only one sensor was used, the degradation in the classification performance of the CNN-LSTM model was most pronounced when the number of sensors exceeded four. The number of trainable hyperparameters of the model increased by 1400 per sensor because the CNN subnetwork structure was used to extract the features of data from each sensor. In contrast, we did not use a subnetwork structure in the CNN-transformer model but instead integrated different sensors into one dimension; this increased the number of hyperparameters by only 192 per sensor. Such a difference in the number of tunable hyperparameters made the CNN-LSTM model susceptible to overfitting

K	Model	LT	LW	LU	C	RU	RW	RT	F1-score
1	RF						✓		0.567
	CNN-LSTM						✓		0.616
	CNN-Transformer						✓		0.570
2	RF						✓	✓	0.660
	CNN-LSTM	✓					✓		0.657
	CNN-Transformer						✓	✓	0.649
3	RF	✓	✓				✓		0.718
	CNN-LSTM		✓				✓	✓	0.702
	CNN-Transformer	✓	✓				✓		0.696
4	RF		✓			✓	✓	✓	0.744
	CNN-LSTM		✓			✓	✓	✓	0.721
	CNN-Transformer		✓			✓	✓	✓	0.708
5	RF		✓	✓		✓	✓	✓	0.743
	CNN-LSTM	✓	✓			✓	✓	✓	0.694
	CNN-Transformer	✓	✓			✓	✓	✓	0.693
6	RF		✓	✓	✓	✓	✓	✓	0.739
	CNN-LSTM	✓	✓	✓	✓	✓	✓		0.685
	CNN-Transformer	✓	✓	✓		✓	✓	✓	0.689
7	RF	✓	✓	✓	✓	✓	✓	✓	0.735
	CNN-LSTM	✓	✓	✓	✓	✓	✓	✓	0.668
	CNN-Transformer	✓	✓	✓	✓	✓	✓	✓	0.698

Fig. 4. (Color online) Highest F1-scores for three classification models on one to seven sensors in LOPO-CV (LT: left thigh, LW: left wrist, LU: left upper arm, C: chest, RU: right upper arm, RW: right wrist, RT: right thigh).

when noisy data were passed in, which led to a decrease in the F1-score. We believe this is especially applicable to LOPO-CV, where the providers of training and test data are different. In contrast, in the RF model, we performed feature selection for each position, i.e., dimensionality reduction, such that only valid features would be used in the training for each position. This did not significantly degrade classification performance, even when useless sensors were added.

The CNN-transformer model did not achieve high F1-scores. In this model, we used the hyperparameters provided in Ref. 24, which might have led to such results. Another reason could have been the lack of data. The multihead self-attention mechanism in the transformer encoder can focus on the information at any one position in the data; however, this also requires an extensive dataset for support. As mentioned in Ref. 25, where the transformer structure was applied for image processing, transformers lack some of the inductive biases inherent to CNNs, such as translation equivariance and locality, and therefore do not generalize well when trained on insufficient amounts of data. In WHAR, collecting large amounts of data with labels is challenging. Although the sample size of our dataset exceeded that of most large publicly available datasets,^(26–28) the results showed that conventional ML still had an advantage. The collection of large amounts of high-quality ADL data is a major future challenge.

To determine whether classification performance for a combination of sensors varied with the classification model, we examined the strength of the relationship using the F1-scores of all 127 combinations under the three classification models. Table 5 shows the Pearson's correlation coefficients between different pairs of the three models, where values closer to 1.0 indicate a stronger relationship between the pairs of classification models. The table shows strong correlations among the three pairs, indicating that trends in the effectiveness of the sensor combinations were stable against changes in the classification models. Therefore, the averages of the F1-scores of the three classification models per sensor-position combination are presented in the following sections, unless otherwise noted.

3.2 Effect of sensors' positions on classification

Figure 5 shows a heatmap representing the overall trend of the F1-scores per combination of sensors, grouped by the number of sensors and sorted in ascending order. In the figure, the check marks in the sensor position columns indicate the use of the sensor position. In the activity columns, the darker cells indicate higher F1-scores. In the rightmost columns, the macro averages are presented as bar charts and numbers. The figure suggests that the sensors' positions and combinations affect not only the average classification performance but also the classification per activity. Note that Table A1 in Appendix shows concrete values.

Table 5
Processing speed by the number of sensors (ms/window).

	RF	CNN-LSTM	CNN-Transformer
RF	1.0	0.924	0.937
CNN-LSTM	—	1.0	0.978
CNN-Transformer	—	—	1.0

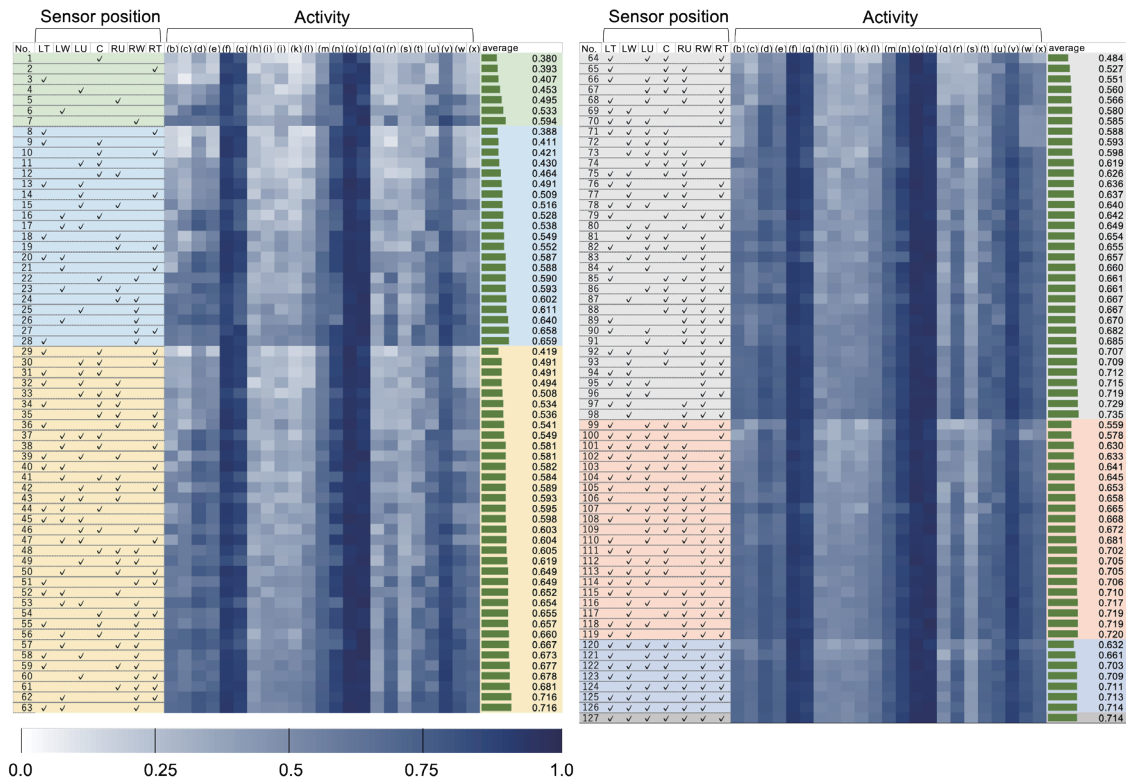


Fig. 5. (Color online) Heatmap representing the trend of F1-scores in LOPO-CV, grouped by the number of sensors and sorted in an ascending order per activity. The abbreviations for the columns of sensor position correspond to left upper arm, left wrist, left thigh, chest, right upper arm, right wrist, and right thigh, respectively. The symbols for the columns of activity correspond to the ones in Fig. 1. The concrete values are shown in Table A1 in Appendix.

A larger number of sensors was not necessarily better, as F1-scores lower than the largest value in the group with a smaller number of sensors were obtained with a larger number of sensors. For example, the highest value was obtained from one sensor (No. 7) worn on the right wrist, whereas only five of the 21 combinations yielded higher values when two sensors were used: Nos. 24, 25, 26, 27, and 28. This is not surprising because activities that show differences in the upper body, such as brushing teeth (b) or washing dishes (c), cannot be distinguished using sensors attached to the left and right thighs. Furthermore, in several cases, the use of fewer sensors was better than the use of all seven sensors. These were Nos. 62, 63, 95, 96, 97, 98, 116, 117, 118, and 119, among which the sensor combination of the left wrist, right upper arm, right wrist, and right thigh (No. 98) was the best. In all cases, except for case 117, no sensor was worn on the chest. The differences in the posture and movement of the chest showed little difference between activities with different hand use, which could be seen from the fact that the value obtained from the chest was the lowest in the case of one-sensor use. This suggests that the information obtained from the chest-mounted sensor was noisy when discriminating between activities that differed in hand or arm movements. In fact, comparing the intensity of the heatmap for each activity in combination Nos. 98 and 127, we found that the cells for standing tasks such as washing dishes (c), washing hands (e), and eating food while sitting (j) appeared

darker in No. 98 than in No. 127. Table A1 in the Appendix concretely shows this fact by indicating a higher F1-score of No. 98.

The right wrist made the highest contribution among the seven positions. For each sensor group, the combination that included the right wrist was the top-ranking. This may be because all of the subjects were right-handed, although the participants in the data collection were not instructed to hold objects such as toothbrushes with their dominant hand. As watches are often worn on the opposite side of the dominant hand, the usefulness of the left wrist must be verified when considering a smartwatch as a practical implementation of the sensor. In the case of single-sensor use, the left wrist ranked second in usefulness, behind the right wrist (No. 6). When two sensors were used, the left wrist appeared first with the right upper arm (No. 23), followed by the right thigh (No. 21) and left thigh (No. 20), with the exception of the right wrist (No. 26). During yoga, a smartphone can be “worn” in a holder attached to the upper arm. Although the degrees of freedom of movement are greater than those under the current data collection conditions, a sensor can be attached to the thigh by keeping a smartphone in the front pocket of the pants. Therefore, we believe that these three pairs represent the classification performance for activity recognition under practical conditions; however, they were 0.066, 0.071, and 0.072 lower than the best pair (No. 28).

3.3 Individual user differences

The relationship between the number of sensors and classification performance under different user data distributions is discussed next. Here, we focus on the RF model because it proved to be the most effective, as discussed in Sect. 3.1. The three evaluation methods presented in Sect. 2.4 were used. Figure 6 shows the relationship between classification performance and number of sensors under the three evaluation methods. Each bar indicates the average F1-score

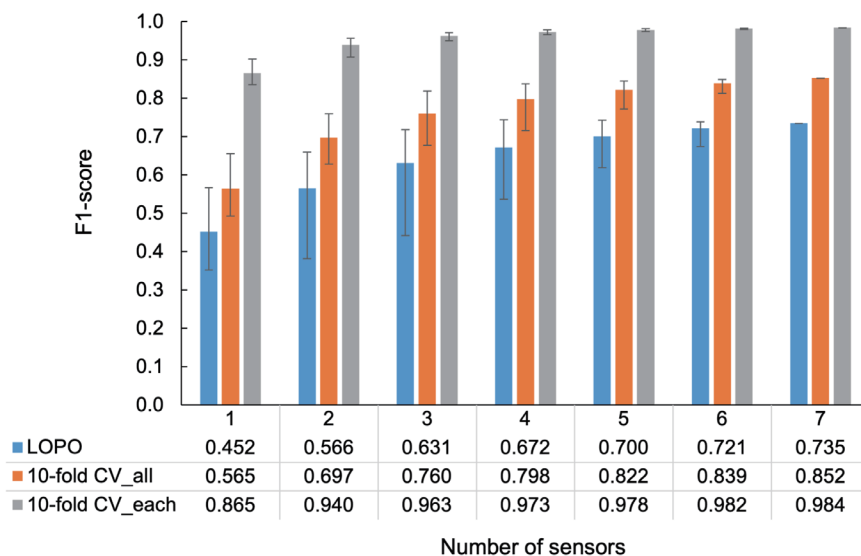


Fig. 6. (Color online) Average F1-scores per number of sensors on the three evaluation methods.

of the results of ${}_7C_m$ combinations in the case of m sensors, and the error bar represents the range between the highest and lowest values.

With respect to the maximum value, the three evaluation methods appeared to be saturated with four sensors. The trend in the mean values was LOPO-CV < 10-fold CV_all < 10-fold CV_each with LOPO. As expected, LOPO-CV and 10-fold CV_each indicate the lower and upper bounds of the classification performance, respectively. The F1-score of LOPO-CV was much lower than that of 10-fold CV_all and 10-fold CV_each because the methods used to perform CADL can vary considerably among individuals. Thus, misclassifications can occur. The results of 10-fold CV_all showed that an average F1-score of more than 0.82 could be achieved using more than three sensors if even a small amount of the user's own activity data was included in the training data. Furthermore, an average F1-score of more than 0.95 was achieved for 23 CADL using only two sensors if the training data were obtained exclusively from a particular user (10-fold CV_each). To improve the classification performance in the LOPO-CV, that is, when testing on data from an unknown user, data should be collected from more participants to increase the heterogeneity of the training data, which would increase the possibility of including people whose data are comparable to those of the unknown user. In other words, it creates a situation that is similar to including the user's own data.

3.4 Processing speed comparison in the three models

Table 6 summarizes the processing speeds in milliseconds per window, in which the DL-based models were evaluated with and without the GPU (using only the CPU). This table presents the following three facts: First, the RF model required a much longer time than the two DL-based models. This is because the processing time in the RF model includes feature calculations requiring approximately 2.70 ms/window. Second, the processing times of the RF and CNN-LSTM models increased linearly, whereas that of the CNN-transformer model was nearly constant. In the RF model, because the feature calculation time with K sensors was almost K times longer, even though the optimal feature subset varied by position, the time required for feature calculation had a greater impact on the overall processing time than the classification time (0.015 ms/window per sensor). In the CNN-LSTM model, as shown in Fig. 2, the number of sensors, K , affects even the concatenation layers, which we considered increased the processing time linearly, although not significantly. By contrast, because K appeared only at the input of the convolutional layers, as shown in Fig. 3, the computational cost for subsequent processing is

Table 6
Processing speed by the number of sensors (ms/window).

Classification model	Number of sensors						
	1	2	3	4	5	6	7
RF (CPU)	2.711	5.213	7.713	10.178	12.721	15.127	17.627
CNN-LSTM (CPU)	0.222	0.266	0.288	0.314	0.309	0.361	0.348
CNN-Transformer (CPU)	9.200	9.195	9.204	9.078	9.139	9.245	9.191
CNN-LSTM (GPU)	0.019	0.024	0.024	0.024	0.029	0.034	0.036
CNN-Transformer (GPU)	0.170	0.169	0.169	0.174	0.169	0.174	0.174

independent of the number of sensors. Therefore, the processing time in the CNN-transformer model was almost constant. Third, GPUs were more than 10 times faster than CPUs when the CNN-LSTM model was processed and 50 times faster for the CNN-transformer model, as expected. In Sect. 3.1, the RF model exhibited the best classification performance; however, its processing speed was the lowest among the three models. Thus, overall, the CNN-LSTM model is the best classification model for both classification performance and processing speed if a large amount of labeled data can be obtained.

4. Conclusion

In this study, we examined the effect of different combinations of seven body-worn accelerometer positions on the classification of 23 CADL. One conventional ML model (RF) and two DL models (CNN-LSTM and CNN-transformer) were used to understand the differences between the classification models. A total of 127 combinations using the three classification models were tested. The findings are as follows:

- A strong correlation between combinations of sensor positions and classification performance was found.
- A larger number of sensors did not necessarily yield better classification performance.
- The sensors placed on the right sides of the subjects exhibited better classification performance than those on the left side and center because of the effect of the dominant hand (all participants were right-handed).
- The combination of four sensors placed on the left and right wrists, right upper arm, and right thigh was the best.
- Assuming that sensors could be integrated with smartwatches and smartphones, practical combinations where a smart watch was worn on the nondominant wrist (left) and a smartphone was kept in the left or right trouser pocket were ranked 85th and 87th performance-wise in the 127 combinations, lower than the best combination by 0.147 and 0.149, respectively.
- A comparison of the three evaluation methods showed the lower, average, and upper bounds of classification performance. Training a classifier using a small amount of data from the test participants significantly improved classification performance.
- The RF model required processing time for feature calculation, which caused a significantly longer processing time per window than DL-based models. Thus, the CNN-LSTM model would be a better choice than RF if a large amount of data is used for training the model.

The findings of this study enable application designers who use activity information to choose a combination of the sensor positions based on the requirements for the wearability of sensors and classification performance of activities according to their interests. In the future, we plan to apply active learning,⁽²⁹⁾ a machine-learning method that engages the user in the labeling process, to adapt the decision boundary of a classifier to the data distribution of a particular user. Furthermore, we will investigate a method to determine the best combination for a new set of activities without evaluating all the combinations.

Acknowledgments

This work was supported by the Japan Society for the Promotion of Science (JSPS) (Grant Nos. 18H03228 and 21K11992).

References

- 1 A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, and J. S. Nayak: *Procedia Comput. Sci.* **115** (2017) 367. <https://doi.org/10.1016/j.procs.2017.09.092>
- 2 M. Zhang, S. Chen, X. Zhao, and Z. Yang: *Sensors* **18** (2018) 2667. <https://doi.org/10.3390/s18082667>
- 3 Y. Wang, S. Cang, and H. Yu: *Expert Syst. Appl.* **137** (2019) 167. <https://doi.org/10.1016/j.eswa.2019.04.057>
- 4 O. D. Lara and M. A. Labrador: *IEEE Commun. Surv. Tutor.* **15** (2013) 1192. <https://doi.org/10.1109/SURV.2012.110112.00192>
- 5 K. Kim, A. Jalal, and M. Mahmood: *J. Electr. Eng. Technol.* **14** (2019) 2567. <https://doi.org/10.1007/s42835-019-00278-8>
- 6 A. Tolstikov, X. Hong, J. Biswas, C. Nugent, L. Chen, and G. Parente: *J. Control Theory Appl.* **9** (2011) 18. <https://doi.org/10.1007/s11768-011-0260-7>
- 7 M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga: *Sensors* **16** (2016) 426. <https://doi.org/10.3390/s16040426>
- 8 H. Gjoreski, M. Lustrek, and M. Gams: *Proc. 2011 7th Int. Conf. Intelligent Environments (IEEE, 2011)* 47. <https://doi.org/10.1109/IE.2011.11>
- 9 L. Atallah, B. Lo, R. King, and G.-Z. Yang: *IEEE Trans. Biomed. Circuits Syst.* **5** (2011) 320. <https://doi.org/10.1109/TBCAS.2011.2160540>
- 10 J. J. Kavanagh, S. Morrison, and R. S. Barrett: *Eur. J. Appl. Physiol.* **94** (2005) 468. <https://doi.org/10.1007/s00421-005-1328-1>
- 11 L. Atallah, O. Aziz, B. Lo, and G.-Z. Yang: *Proc. 2009 6th Int. Workshop on Wearable and Implantable Body Sensor Networks (IEEE, 2009)* 175. <https://doi.org/10.1109/BSN.2009.41>
- 12 I. Cleland, B. Kikhia, C. Nugent, A. Boytsov, J. Hallberg, K. Synnes, S. McClean, and D. Finlay: *Sensors* **13** (2013) 9183. <https://doi.org/10.3390/s130709183>
- 13 N. Pannurat, S. Thiemjarus, E. Nantajeewarawat, and I. Anantavasilp: *Sensors* **17** (2017) 774. <https://doi.org/10.3390/s17040774>
- 14 J. Wang, Y. Chen, L. Hu, X. Peng, and P. S. Yu: *Proc. 2018 IEEE Int. Conf. Pervasive Computing and Communications (IEEE, 2018)* 1. <https://doi.org/10.1109/PERCOM.2018.8444572>
- 15 K. Fujinami, T. Saeki, Y. Li, T. Ishikawa, T. Jumbo, D. Nagase, and K. Sato: *Int. J. Adv. Comput. Sci. Appl.* **8** (2017) 8. <https://doi.org/10.14569/IJACSA.2017.080858>
- 16 I. Kononenko: *Machine Learning: ECML-94*, F. Bergadano and L. De Raedt Eds, (Springer, Berlin, Heidelberg, 1994) p. 171. https://doi.org/10.1007/3-540-57868-4_57
- 17 T. Szttyler and H. Stuckenschmidt: *Proc. 2016 IEEE Int. Conf. Pervasive Computing and Communications (IEEE, 2016)* 1. <https://doi.org/10.1109/PERCOM.2016.7456521>
- 18 B. Kikhia, M. Gomez, L. L. Jiménez, J. Hallberg, N. Karvonen, and K. Synnes: *Sensors* **14** (2014) 3. <https://doi.org/10.3390/s140305725>
- 19 F. J. Ordóñez and D. Roggen: *Sensors* **16** (2016) 1. <https://doi.org/10.3390/s16010115>
- 20 K. Xia, J. Huang, and H. Wang: *IEEE Access* **8** (2020) 56855. <https://doi.org/10.1109/ACCESS.2020.2982225>
- 21 M. Bock, A. Hoelzemann, M. Moeller, and K. Van Laerhoven: *ArXiv210800702* (2021). <http://arxiv.org/abs/2108.00702>
- 22 L. Chen, X. Liu, L. Peng, and M. Wu: *Appl. Intell.* **51** (2021) 4029. <https://doi.org/10.1007/s10489-020-02005-7>
- 23 I. A. Lawal and S. Bano: *IEEE Access* **8** (2020) 155060. <https://doi.org/10.1109/ACCESS.2020.3017681>
- 24 Y. Shavit and I. Klein: *IEEE Access* **9** (2021) 53540. <https://doi.org/10.1109/ACCESS.2021.3070646>
- 25 A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby: *ArXiv201011929* (2021). <http://arxiv.org/abs/2010.11929>
- 26 A. Reiss and D. Stricker: *Proc. 2012 16th Int. Symp. Wearable Computers (IEEE, 2012)* 108. <https://doi.org/10.1109/ISWC.2012.13>
- 27 R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. R. Millán, and D. Roggen: *Pattern Recognit. Lett.* **34** (2013) 2033. <https://doi.org/10.1016/j.patrec.2012.12.014>

- 28 M. Zhang and A. A. Sawchuk: Proc. 2012 ACM Conf. Ubiquitous Computing (ACM, 2012) 1036. <https://doi.org/10.1145/2370216.2370438>
- 29 B. Settles: Technical Report #1648 (Computer Science Department, University of Wisconsin Madison, 2009). <http://digital.library.wisc.edu/1793/60660> (accessed June 1, 2023).

About the Authors

Yuhao Duan received his B.E. degree in software engineering from Century College, Beijing University of Posts and Telecommunications, China, in 2017, and his M.E. degree in computer and information sciences from Tokyo University of Agriculture and Technology (TUAT), Japan, in 2022. His research interests are in activity recognition, artificial intelligence, and wearable computing.

Kaori Fujinami received his B.E. and M.E. degrees in electrical engineering and his Ph.D. degree in computer science from Waseda University, Japan, in 1993, 1995, and 2005, respectively. From 2005 to 2006, he was a visiting lecturer at Waseda University. From 2007 to 2017, he was an associate professor in the Department of Computer and Information Sciences at TUAT. Since 2018, he has been a professor at TUAT. His research interests are in machine learning, activity recognition, human–computer interaction, and ubiquitous computing. (fujinami@cc.tuat.ac.jp).

Appendix

Table A1

(Color online) Concrete values (F1-scores in LOPO-CV) of the heatmap presented in Fig. 5. Unlike other tables, the F1-scores are rounded off to two decimal places for readability. The numbers in the leftmost column correspond to the ones in Fig. 5, representing combinations of sensor positions. The symbols in the first row correspond to the activities in Fig. 1.

No.	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)	(p)	(q)	(r)	(s)	(t)	(u)	(v)	(w)	(x)
1	0.25	0.16	0.38	0.21	0.88	0.78	0.19	0.14	0.11	0.07	0.24	0.42	0.76	0.93	0.79	0.21	0.20	0.26	0.25	0.37	0.59	0.32	0.20
2	0.20	0.15	0.28	0.17	0.85	0.86	0.14	0.24	0.26	0.10	0.16	0.46	0.82	0.88	0.93	0.26	0.22	0.33	0.41	0.21	0.60	0.29	0.21
3	0.22	0.06	0.23	0.24	0.91	0.86	0.29	0.20	0.33	0.13	0.19	0.45	0.84	0.93	0.92	0.28	0.21	0.34	0.39	0.24	0.58	0.29	0.24
4	0.40	0.34	0.41	0.53	0.87	0.77	0.31	0.25	0.12	0.16	0.27	0.59	0.77	0.95	0.88	0.21	0.24	0.27	0.27	0.60	0.60	0.34	0.25
5	0.45	0.33	0.46	0.59	0.86	0.80	0.24	0.24	0.24	0.27	0.39	0.58	0.79	0.94	0.81	0.31	0.17	0.36	0.34	0.42	0.70	0.64	0.45
6	0.33	0.57	0.75	0.71	0.88	0.73	0.26	0.21	0.35	0.18	0.39	0.61	0.63	0.92	0.94	0.40	0.40	0.37	0.37	0.70	0.67	0.46	0.42
7	0.62	0.63	0.61	0.64	0.81	0.77	0.34	0.32	0.38	0.38	0.53	0.66	0.64	0.90	0.95	0.31	0.65	0.33	0.41	0.62	0.86	0.70	0.61
8	0.16	0.11	0.29	0.22	0.89	0.90	0.18	0.20	0.39	0.12	0.11	0.47	0.83	0.92	0.90	0.17	0.15	0.31	0.33	0.21	0.60	0.25	0.22
9	0.17	0.09	0.47	0.21	0.93	0.88	0.26	0.17	0.30	0.13	0.18	0.49	0.85	0.95	0.91	0.19	0.10	0.32	0.29	0.40	0.66	0.32	0.20
10	0.17	0.14	0.42	0.19	0.93	0.88	0.26	0.21	0.20	0.15	0.22	0.52	0.85	0.94	0.85	0.22	0.26	0.31	0.35	0.40	0.67	0.33	0.23
11	0.29	0.36	0.49	0.41	0.88	0.76	0.28	0.21	0.16	0.13	0.19	0.49	0.76	0.92	0.83	0.19	0.20	0.26	0.26	0.64	0.63	0.35	0.19
12	0.38	0.33	0.50	0.52	0.89	0.78	0.19	0.25	0.19	0.23	0.36	0.52	0.77	0.93	0.84	0.23	0.15	0.18	0.26	0.46	0.70	0.57	0.44
13	0.38	0.29	0.47	0.49	0.91	0.85	0.25	0.29	0.27	0.17	0.25	0.58	0.86	0.96	0.92	0.24	0.33	0.28	0.41	0.63	0.71	0.42	0.34
14	0.41	0.31	0.48	0.52	0.91	0.90	0.20	0.38	0.33	0.21	0.25	0.60	0.88	0.95	0.93	0.25	0.37	0.32	0.47	0.59	0.71	0.41	0.34
15	0.53	0.34	0.51	0.54	0.84	0.75	0.35	0.28	0.20	0.23	0.42	0.55	0.74	0.93	0.86	0.32	0.29	0.32	0.39	0.61	0.69	0.64	0.53
16	0.30	0.53	0.73	0.66	0.89	0.75	0.25	0.17	0.33	0.17	0.38	0.63	0.72	0.94	0.89	0.43	0.50	0.29	0.30	0.70	0.71	0.44	0.44
17	0.43	0.57	0.77	0.67	0.87	0.72	0.28	0.32	0.27	0.24	0.39	0.61	0.70	0.94	0.93	0.36	0.41	0.30	0.28	0.76	0.71	0.45	0.40
18	0.42	0.22	0.45	0.55	0.91	0.88	0.44	0.31	0.44	0.34	0.42	0.61	0.87	0.93	0.92	0.27	0.38	0.33	0.52	0.47	0.80	0.61	0.54
19	0.44	0.31	0.48	0.59	0.93	0.90	0.31	0.35	0.46	0.35	0.44	0.60	0.86	0.94	0.93	0.33	0.27	0.24	0.57	0.42	0.80	0.63	0.54
20	0.34	0.52	0.75	0.67	0.92	0.87	0.41	0.26	0.39	0.36	0.39	0.70	0.86	0.94	0.92	0.48	0.48	0.45	0.45	0.68	0.72	0.47	0.47
21	0.28	0.57	0.76	0.70	0.92	0.88	0.32	0.28	0.42	0.32	0.40	0.70	0.86	0.95	0.92	0.52	0.47	0.38	0.52	0.65	0.73	0.46	0.52
22	0.62	0.61	0.62	0.61	0.90	0.79	0.27	0.34	0.32	0.29	0.52	0.66	0.73	0.94	0.94	0.30	0.69	0.31	0.42	0.67	0.83	0.66	0.55
23	0.51	0.57	0.75	0.72	0.86	0.75	0.30	0.32	0.33	0.27	0.45	0.65	0.71	0.93	0.94	0.44	0.50	0.35	0.52	0.67	0.80	0.74	0.56
24	0.64	0.61	0.64	0.70	0.84	0.74	0.34	0.39	0.38	0.40	0.58	0.67	0.67	0.91	0.96	0.25	0.63	0.26	0.40	0.59	0.87	0.73	0.64
25	0.66	0.59	0.71	0.64	0.86	0.74	0.42	0.51	0.32	0.38	0.53	0.65	0.68	0.94	0.95	0.26	0.59	0.32	0.42	0.73	0.87	0.69	0.60
26	0.64	0.71	0.76	0.74	0.86	0.71	0.30	0.46	0.45	0.39	0.56	0.67	0.57	0.93	0.95	0.41	0.74	0.39	0.49	0.71	0.83	0.79	0.63
27	0.61	0.56	0.65	0.63	0.84	0.91	0.35	0.51	0.51	0.54	0.58	0.74	0.87	0.93	0.95	0.38	0.70	0.31	0.72	0.59	0.89	0.70	0.67
28	0.59	0.55	0.64	0.63	0.91	0.88	0.44	0.48	0.54	0.49	0.55	0.73	0.86	0.94	0.93	0.37	0.72	0.39	0.69	0.59	0.87	0.69	0.68
29	0.21	0.11	0.44	0.21	0.92	0.89	0.32	0.18	0.24	0.15	0.19	0.51	0.85	0.96	0.88	0.21	0.16	0.33	0.36	0.34	0.66	0.29	0.24
30	0.35	0.35	0.51	0.47	0.91	0.88	0.29	0.27	0.28	0.19	0.25	0.58	0.86	0.95	0.90	0.28	0.30	0.28	0.39	0.57	0.72	0.40	0.30
31	0.34	0.33	0.52	0.45	0.90	0.87	0.36	0.27	0.26	0.17	0.22	0.57	0.86	0.95	0.93	0.25	0.29	0.36	0.36	0.66	0.71	0.37	0.29
32	0.36	0.32	0.50	0.50	0.91	0.88	0.16	0.26	0.35	0.21	0.22	0.59	0.88	0.94	0.93	0.29	0.37	0.26	0.45	0.54	0.71	0.41	0.31
33	0.49	0.39	0.53	0.50	0.86	0.76	0.30	0.27	0.18	0.21	0.41	0.58	0.75	0.93	0.88	0.30	0.28	0.31	0.32	0.59	0.72	0.64	0.48
34	0.40	0.30	0.52	0.51	0.90	0.86	0.38	0.31	0.35	0.28	0.41	0.59	0.84	0.95	0.92	0.30	0.30	0.33	0.44	0.50	0.75	0.61	0.53
35	0.39	0.34	0.48	0.54	0.93	0.87	0.40	0.34	0.33	0.29	0.41	0.60	0.83	0.94	0.89	0.27	0.31	0.30	0.50	0.48	0.78	0.58	0.53
36	0.40	0.25	0.44	0.57	0.93	0.90	0.30	0.34	0.49	0.33	0.42	0.62	0.89	0.95	0.91	0.26	0.29	0.26	0.50	0.39	0.80	0.61	0.56
37	0.36	0.54	0.74	0.65	0.88	0.77	0.30	0.30	0.35	0.22	0.37	0.64	0.77	0.94	0.90	0.40	0.45	0.35	0.32	0.76	0.75	0.43	0.43
38	0.30	0.56	0.77	0.66	0.94	0.88	0.37	0.25	0.35	0.33	0.41	0.70	0.85	0.96	0.91	0.47	0.54	0.30	0.47	0.68	0.74	0.47	0.47
39	0.55	0.33	0.57	0.56	0.92	0.87	0.45	0.33	0.40	0.35	0.43	0.61	0.87	0.95	0.96	0.31	0.35	0.39	0.51	0.63	0.79	0.67	0.57
40	0.34	0.50	0.75	0.68	0.94	0.90	0.37	0.23	0.40	0.35	0.37	0.69	0.90	0.95	0.92	0.46	0.47	0.38	0.48	0.63	0.72	0.45	0.48
41	0.51	0.55	0.72	0.69	0.88	0.78	0.28	0.30	0.31	0.24	0.44	0.69	0.73	0.93	0.91	0.45	0.49	0.34	0.40	0.72	0.79	0.71	0.56
42	0.56	0.36	0.57	0.60	0.90	0.88	0.43	0.35	0.41	0.31	0.45	0.66	0.90	0.93	0.94	0.36	0.37	0.33	0.58	0.61	0.80	0.69	0.56
43	0.57	0.56	0.74	0.71	0.84	0.76	0.31	0.34	0.29	0.28	0.49	0.63	0.75	0.93	0.94	0.43	0.44	0.36	0.39	0.74	0.81	0.73	0.61
44	0.38	0.51	0.77	0.65	0.91	0.87	0.44	0.28	0.41	0.35	0.38	0.71	0.85	0.94	0.92	0.46	0.61	0.44	0.42	0.70	0.74	0.47	0.48
45	0.44	0.52	0.78	0.65	0.91	0.87	0.41	0.33	0.37	0.41	0.40	0.67	0.90	0.96	0.94	0.49	0.45	0.34	0.43	0.74	0.76	0.50	0.46
46	0.65	0.56	0.68	0.61	0.88	0.80	0.37	0.43	0.31	0.33	0.49	0.67	0.74	0.94	0.95	0.27	0.58	0.34	0.44	0.74	0.86	0.70	0.54
47	0.40	0.57	0.79	0.69	0.92	0.89	0.34	0.37	0.39	0.40	0.41	0.67	0.90	0.96	0.94	0.46	0.49	0.31	0.51	0.74	0.77	0.48	0.46
48	0.62	0.58	0.61	0.66	0.89	0.81	0.30	0.38	0.35	0.36	0.55	0.68	0.77	0.92	0.94	0.35	0.61	0.29	0.46	0.61	0.84	0.72	0.61
49	0.67	0.58	0.70	0.68	0.83	0.73	0.43	0.49	0.34	0.38	0.58	0.66	0.72	0.93	0.95	0.28	0.57	0.37	0.39	0.69	0.86	0.77	0.66
50	0.58	0.59	0.77	0.73	0.93	0.90	0.38	0.37	0.45	0.43	0.50	0.70	0.89	0.94	0.94	0.48	0.51	0.36	0.62	0.70	0.83	0.70	0.61
51	0.59	0.53	0.61	0.60	0.90	0.89	0.38	0.47	0.48	0.47	0.52	0.74	0.88	0.95	0.94	0.37	0.70	0.41	0.73	0.53	0.88	0.69	0.67
52	0.56	0.54	0.74	0.71	0.92	0.88	0.49	0.35	0.44	0.44	0.51	0.72	0.89	0.95	0.95	0.53	0.52	0.40	0.60	0.70	0.84	0.73	0.61
53	0.68	0.68	0.78	0.72	0.86	0.74	0.39	0.49	0.41	0.40	0.58	0.71	0.66	0.94	0.97	0.45	0.65	0.39	0.51	0.75	0.85	0.78	0.65
54	0.62	0.59	0.64	0.60	0.92	0.89	0.42	0.49	0.47	0.45	0.57	0.73	0.84	0.95	0.93	0.40	0.69	0.37	0.70	0.62	0.85	0.69	0.63
55	0.63	0.54	0.65	0.61																			

Table A1 (Continued)

No.	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)	(p)	(q)	(r)	(s)	(t)	(u)	(v)	(w)	(x)
64	0.37	0.31	0.52	0.45	0.92	0.89	0.30	0.19	0.31	0.21	0.23	0.56	0.86	0.96	0.88	0.23	0.28	0.32	0.38	0.64	0.69	0.37	0.30
65	0.43	0.26	0.51	0.48	0.93	0.89	0.33	0.27	0.36	0.26	0.40	0.57	0.86	0.95	0.91	0.30	0.29	0.25	0.49	0.48	0.77	0.58	0.56
66	0.52	0.35	0.56	0.50	0.90	0.86	0.41	0.30	0.34	0.33	0.39	0.61	0.85	0.93	0.93	0.33	0.30	0.31	0.42	0.63	0.73	0.64	0.54
67	0.50	0.41	0.57	0.52	0.91	0.87	0.39	0.33	0.35	0.28	0.44	0.62	0.85	0.94	0.94	0.31	0.32	0.30	0.44	0.61	0.77	0.66	0.54
68	0.53	0.32	0.56	0.56	0.91	0.88	0.38	0.32	0.36	0.34	0.40	0.61	0.87	0.95	0.92	0.29	0.40	0.32	0.51	0.60	0.79	0.64	0.56
69	0.33	0.50	0.76	0.64	0.93	0.88	0.38	0.29	0.31	0.33	0.38	0.70	0.88	0.95	0.90	0.42	0.52	0.34	0.48	0.69	0.75	0.48	0.51
70	0.39	0.53	0.77	0.66	0.90	0.86	0.30	0.35	0.40	0.41	0.40	0.68	0.88	0.95	0.93	0.42	0.50	0.29	0.46	0.70	0.75	0.46	0.47
71	0.37	0.51	0.77	0.63	0.90	0.85	0.43	0.31	0.38	0.36	0.41	0.69	0.87	0.95	0.90	0.51	0.45	0.38	0.41	0.74	0.76	0.46	0.47
72	0.36	0.53	0.78	0.64	0.90	0.86	0.44	0.34	0.39	0.34	0.42	0.66	0.85	0.94	0.90	0.49	0.53	0.35	0.46	0.74	0.78	0.47	0.47
73	0.53	0.57	0.71	0.69	0.89	0.79	0.31	0.36	0.32	0.25	0.47	0.66	0.76	0.95	0.92	0.44	0.50	0.38	0.41	0.72	0.81	0.69	0.60
74	0.65	0.59	0.68	0.66	0.88	0.77	0.45	0.43	0.31	0.37	0.57	0.68	0.74	0.94	0.93	0.35	0.57	0.26	0.47	0.75	0.86	0.73	0.61
75	0.47	0.51	0.75	0.68	0.92	0.88	0.43	0.31	0.47	0.38	0.48	0.72	0.87	0.95	0.93	0.44	0.51	0.36	0.53	0.71	0.83	0.69	0.61
76	0.51	0.55	0.73	0.68	0.92	0.89	0.41	0.37	0.45	0.44	0.49	0.74	0.90	0.95	0.93	0.50	0.42	0.38	0.58	0.65	0.85	0.70	0.60
77	0.52	0.58	0.75	0.70	0.91	0.89	0.42	0.32	0.42	0.40	0.50	0.72	0.87	0.94	0.92	0.46	0.53	0.40	0.59	0.69	0.83	0.67	0.62
78	0.56	0.54	0.75	0.69	0.90	0.87	0.48	0.35	0.40	0.41	0.50	0.70	0.87	0.93	0.94	0.50	0.52	0.38	0.56	0.73	0.81	0.73	0.60
79	0.59	0.55	0.64	0.60	0.93	0.88	0.41	0.48	0.39	0.45	0.56	0.72	0.85	0.94	0.91	0.39	0.64	0.36	0.67	0.62	0.86	0.68	0.64
80	0.58	0.58	0.77	0.67	0.88	0.87	0.44	0.43	0.42	0.41	0.54	0.70	0.86	0.95	0.94	0.50	0.54	0.36	0.60	0.71	0.85	0.71	0.62
81	0.67	0.66	0.77	0.71	0.88	0.78	0.36	0.52	0.43	0.36	0.56	0.69	0.75	0.94	0.95	0.42	0.69	0.39	0.47	0.79	0.86	0.75	0.67
82	0.65	0.56	0.68	0.61	0.90	0.85	0.45	0.47	0.46	0.47	0.54	0.72	0.85	0.95	0.95	0.39	0.67	0.37	0.60	0.72	0.86	0.70	0.65
83	0.69	0.67	0.78	0.74	0.84	0.72	0.39	0.45	0.43	0.40	0.60	0.67	0.70	0.94	0.96	0.44	0.71	0.39	0.51	0.71	0.87	0.80	0.70
84	0.66	0.55	0.69	0.64	0.91	0.89	0.36	0.46	0.48	0.50	0.55	0.75	0.90	0.95	0.94	0.33	0.69	0.34	0.66	0.70	0.89	0.71	0.67
85	0.64	0.56	0.63	0.66	0.91	0.88	0.43	0.50	0.44	0.50	0.59	0.72	0.87	0.95	0.93	0.37	0.67	0.30	0.68	0.65	0.86	0.73	0.70
86	0.65	0.55	0.69	0.58	0.90	0.87	0.44	0.53	0.47	0.48	0.55	0.74	0.85	0.94	0.93	0.43	0.64	0.36	0.68	0.73	0.85	0.71	0.65
87	0.67	0.67	0.77	0.73	0.91	0.82	0.37	0.48	0.45	0.48	0.57	0.71	0.77	0.94	0.94	0.44	0.74	0.44	0.54	0.73	0.86	0.77	0.66
88	0.64	0.58	0.64	0.65	0.92	0.89	0.49	0.46	0.48	0.50	0.60	0.73	0.88	0.95	0.94	0.38	0.67	0.36	0.71	0.63	0.86	0.73	0.66
89	0.61	0.52	0.64	0.68	0.91	0.89	0.47	0.49	0.53	0.55	0.60	0.74	0.86	0.93	0.94	0.39	0.71	0.38	0.70	0.56	0.87	0.73	0.71
90	0.69	0.53	0.70	0.64	0.89	0.86	0.53	0.53	0.50	0.55	0.58	0.71	0.85	0.95	0.95	0.41	0.70	0.40	0.67	0.72	0.90	0.74	0.68
91	0.69	0.56	0.71	0.68	0.88	0.87	0.50	0.54	0.54	0.52	0.60	0.74	0.87	0.93	0.93	0.38	0.69	0.39	0.70	0.70	0.88	0.77	0.69
92	0.68	0.69	0.78	0.71	0.92	0.87	0.51	0.53	0.48	0.51	0.58	0.77	0.86	0.95	0.93	0.53	0.73	0.46	0.69	0.74	0.85	0.79	0.70
93	0.66	0.67	0.79	0.71	0.93	0.89	0.52	0.56	0.51	0.51	0.60	0.74	0.86	0.94	0.94	0.50	0.73	0.45	0.75	0.72	0.88	0.77	0.68
94	0.66	0.67	0.76	0.73	0.91	0.90	0.45	0.54	0.53	0.55	0.59	0.78	0.88	0.95	0.94	0.47	0.75	0.47	0.78	0.69	0.88	0.78	0.71
95	0.68	0.66	0.80	0.72	0.92	0.87	0.48	0.50	0.54	0.59	0.59	0.75	0.86	0.95	0.95	0.54	0.73	0.49	0.70	0.79	0.88	0.76	0.69
96	0.70	0.68	0.82	0.73	0.91	0.89	0.43	0.54	0.54	0.60	0.61	0.74	0.87	0.94	0.95	0.51	0.74	0.45	0.73	0.76	0.88	0.81	0.69
97	0.71	0.68	0.78	0.75	0.92	0.88	0.57	0.49	0.53	0.58	0.62	0.76	0.87	0.95	0.95	0.54	0.75	0.52	0.72	0.73	0.90	0.81	0.73
98	0.72	0.70	0.79	0.76	0.92	0.91	0.51	0.53	0.60	0.58	0.65	0.75	0.89	0.95	0.94	0.54	0.78	0.49	0.75	0.72	0.90	0.80	0.72
99	0.54	0.37	0.56	0.54	0.91	0.87	0.42	0.30	0.41	0.26	0.40	0.62	0.88	0.94	0.91	0.32	0.27	0.32	0.47	0.64	0.77	0.61	0.52
100	0.36	0.48	0.77	0.61	0.92	0.87	0.35	0.30	0.41	0.36	0.41	0.67	0.90	0.95	0.89	0.43	0.47	0.37	0.43	0.71	0.74	0.41	0.50
101	0.49	0.51	0.74	0.66	0.89	0.86	0.50	0.35	0.36	0.39	0.49	0.73	0.87	0.94	0.93	0.49	0.55	0.40	0.48	0.72	0.84	0.68	0.62
102	0.50	0.54	0.74	0.67	0.93	0.89	0.44	0.35	0.43	0.37	0.50	0.72	0.89	0.95	0.92	0.40	0.49	0.39	0.59	0.70	0.84	0.70	0.61
103	0.55	0.56	0.76	0.69	0.90	0.88	0.39	0.36	0.47	0.41	0.52	0.74	0.86	0.94	0.93	0.51	0.50	0.35	0.60	0.72	0.84	0.67	0.61
104	0.56	0.55	0.75	0.68	0.92	0.89	0.46	0.36	0.46	0.46	0.51	0.72	0.89	0.95	0.93	0.48	0.47	0.34	0.58	0.72	0.86	0.71	0.60
105	0.63	0.55	0.69	0.59	0.91	0.89	0.44	0.47	0.43	0.46	0.55	0.74	0.89	0.96	0.93	0.34	0.62	0.38	0.62	0.71	0.87	0.71	0.64
106	0.62	0.54	0.64	0.63	0.92	0.88	0.47	0.48	0.47	0.49	0.56	0.73	0.88	0.95	0.92	0.36	0.68	0.31	0.70	0.62	0.86	0.73	0.67
107	0.70	0.66	0.78	0.70	0.88	0.77	0.44	0.49	0.41	0.36	0.60	0.68	0.76	0.92	0.95	0.46	0.68	0.43	0.50	0.76	0.86	0.81	0.72
108	0.68	0.57	0.70	0.62	0.90	0.86	0.46	0.49	0.44	0.52	0.60	0.75	0.86	0.92	0.94	0.42	0.64	0.33	0.65	0.71	0.89	0.75	0.69
109	0.66	0.58	0.68	0.63	0.90	0.88	0.44	0.52	0.48	0.51	0.61	0.73	0.85	0.93	0.95	0.44	0.63	0.36	0.69	0.71	0.86	0.74	0.67
110	0.67	0.54	0.68	0.65	0.91	0.89	0.47	0.51	0.49	0.54	0.60	0.74	0.89	0.94	0.92	0.40	0.69	0.37	0.67	0.71	0.90	0.77	0.71
111	0.65	0.65	0.78	0.69	0.93	0.90	0.44	0.54	0.52	0.53	0.58	0.74	0.89	0.94	0.92	0.48	0.72	0.42	0.73	0.74	0.88	0.79	0.69
112	0.69	0.67	0.78	0.70	0.91	0.88	0.54	0.53	0.54	0.57	0.60	0.75	0.85	0.95	0.93	0.47	0.67	0.39	0.67	0.75	0.88	0.78	0.68
113	0.67	0.65	0.79	0.69	0.89	0.86	0.51	0.52	0.49	0.56	0.61	0.75	0.86	0.92	0.95	0.54	0.69	0.46	0.69	0.77	0.88	0.79	0.68
114	0.68	0.64	0.78	0.70	0.91	0.88	0.45	0.51	0.53	0.55	0.61	0.74	0.87	0.94	0.95	0.50	0.72	0.44	0.70	0.74	0.89	0.78	0.71
115	0.69	0.66	0.77	0.71	0.91	0.88	0.55	0.53	0.48	0.55	0.62	0.75	0.83	0.93	0.95	0.51	0.73	0.45	0.70	0.73	0.88	0.81	0.71
116	0.71	0.66	0.78	0.75	0.91	0.89	0.51	0.51	0.56	0.57	0.63	0.74	0.88	0.94	0.95	0.53	0.70	0.43	0.74	0.72	0.91	0.78	0.70
117	0.69	0.68	0.78	0.73	0.93	0.89	0.50	0.51	0.52	0.55	0.64	0.76	0.89	0.94	0.93	0.50	0.74	0.45	0.74	0.73	0.89	0.80	0.72
118	0.70	0.67	0.79	0.73	0.89	0.87	0.57	0.52	0.53	0.58	0.62	0.75	0.85	0.92	0.97	0.55	0.74	0.48	0.68	0.73	0.89		