Comparison of Feature Extraction Techniques for Ambient Sensor-based In-home Activity Recognition

Aiguo Wang
School of Electronic Information Engineering
Foshan University
Foshan, China
wangaiguo2546@163.com

Yue Meng
School of Computer Science and Technology
Donghua University
Shanghai, China
meng_yue97@163.com

Liang Zhao, Jinjun Liu*, Guilin Chen
School of Computer and Information Engineering
Chuzhou University
Chuzhou, China
{leon, jinjunl, glchen}@chzu.edu.cn

Abstract—**Ambient sensor-based in-home activity recognition plays a crucial role in the design and development of a smart home to better and actively respond to population aging. From the perspective of machine learning, how to extract features from sensor data largely determines the power of a data-driven human activity recognizer. However, few studies systematically investigate how to encode streaming sensor events. To this end, we herein conduct a comparison of different feature extraction techniques for activity recognition. Specifically, we explore two types of feature representations (i.e., *statistical* features and *structural* features) and evaluate their single use and joint use. Besides, we experimentally analyze the impact of window size on prediction accuracy. Finally, we perform experiments on three public datasets with 15 different feature encodings and 6 classifiers. Results show that the joint use of different features generally obtains enhanced accuracy and that the interval 60s of window size achieves a better accuracy-speed tradeoff.**

Keywords-Smart home; human activity recognition; feature encoding

I. Introduction

Smart homes advanced by the development of pervasive computing, senor technology, artificial intelligence, internet of things, and edge computing provide a feasible solution to the aging society in helping the elderly to live independently and maintain quality of life [1,2]. The key to the success of various smart home applications such as wellness evaluation, behaviour analysis, ambient assisted living tools, abnormal detection, and chronic disease management is to perceive the user states and environments, where activity recognition plays an important role in bridging the gap [3,4,5].

The primary task of activity recognition is to train an activity recognizer for automatic and accurate prediction of on-going human activities with sensors [3,6,7,8]. Different from traditional tasks, activity recognition remains a more challenging topic, as human behaviours are characterized by inherent complexity. For example, different individuals may perform the same activity differently and even the same people can perform an activity differently for different time intervals and locations. Besides, interleaved, concurrent, and similar activities also exist [9,10].

According to the used underlying sensing units, existing activity recognition methods can be broadly categorized into three groups: vision-, wearable-, and ambient sensor-based methods [7,11]. The vision-based methods utilize computer vision techniques to analyze the image captured by a camera or video, which is often limited to fixed locations and easily influenced by the variations of background, occlusion, and light. Wearable-based methods use the data of sensors worn by the user to train an activity recognizer and then use it to infer the on-going human activities [11]. Such methods have the advantage of low costs and easy configuration and are suitable for indoor and outdoor scenarios. One drawback is that they require users to wear a device, which would bring inconvenience to users, especially the elderly. In contrast, ambient sensor-based methods recognize human activities by capturing the interactions between an individual and the household objects and training an activity recognizer [12]. Such methods remain a priority in smart home environments due to their non-intrusiveness.

With an aim to improve the generalization ability of an activity recognizer, researchers have conducted a wealth of studies on the components of the activity recognition chain, mainly including the segmentation of sensor data, feature extraction, and the choice of a classification model [13,14], among which how to encode time-series sensor data largely determines the recognition performance and has also drawn researcher's attention [15]. For example, Tapia et al. used temporal features, such as the duration of a sensor event and the order of two sensors, for activity recognition [3]. Van Kasteren et al. proposed to encode sensor events with *binary*, *change point*, and *last* representations, respectively [12]. Yatbaz et al. used the duration of firing for each sensor to encode an instance and proposed a scanpath trend analysis method for activity recognition [16]. Wang et al. proposed three different feature encodings (i.e., *binary*, *numerical*, and *probability* representations) and then evaluated their combinations with different classification models [17]. The features used by Yan et al. include the ID and time of the first and last sensor events, the window size, and the number of firings of different sensors in a window [18]. Although researches have reported promising results, few studies conduct a systematic study on how to encode the streaming sensor data in ambient sensor-based activity recognition. To this end, we investigate different ways of extracting features from time-series sensor events. The main contributions are as follows. (1) We explore two types of features (i.e., *statistical* features and *structural* features) to encode sensor events. Specifically, we investigate three specific forms for

978-1-6654-6131-3/22/$31.00 ©2022 IEEE
DOI 10.1109/NaNA56854.2022.00072

the former and two specific forms for the latter. For the five groups of features, besides their single use, we evaluate their joint use. (2) We preliminarily conduct parameter sensitivity analysis of the sliding-window size in segmenting streaming sensor events. (3) Comparative experiments are conducted on activity recognition datasets collected by ambient sensors in smart homes. Results show that the joint use of different features generally obtains enhanced prediction accuracy and that the window size has an impact on the performance of an activity recognizer.

The rest of this paper is organized as follows. Second II illustrates the activity recognition chain and different ways of feature extraction. Experimental setting and results are shown in section III, followed by the conclusion section.

II. ACTIVITY RECOGNITION MODEL

A. Activity Recognition Chain

Sensor-based activity recognition chain mainly consists of the following steps. First, the time-series data of the used sensors are collected and annotated, where the starting time and ending time of an activity and associated sensor events are recorded. Commonly used annotation methods include self-recall, audio and video, and experience sampling. Then, the streaming sensor data are divided into segments with the sliding window technique. Explicit segmentation and time-based segmentation are two representatives. The former is more suitable for the offline setting, while the latter is more appropriate to practical use. Obviously, the size of the used sliding window is a crucial factor. Third, we extract various features from the segment to form a feature vector. Finally, we train an activity recognizer with the feature vectors and a classification model. There are a number of classification models available that range from discriminant models to generative models. During the prediction phase, the test data are first segmented and encoded into feature vectors and then the activity recognizer infers on-going activities. Figure 1 presents the overall framework.

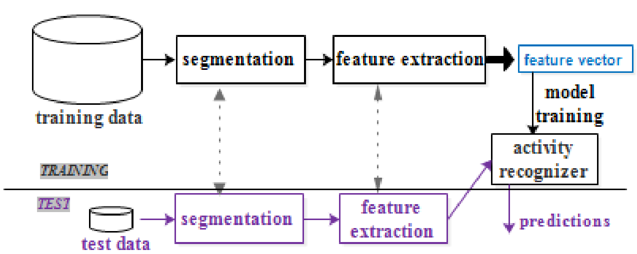

Figure 1. Activity recognition chain.

Figure 1 clearly shows the key role of feature extraction in building a powerful activity recognizer. In the following subsection, we will introduce how to encode the time-series sensor data.

B. Feature Extraction

After dividing time-series sensor data into segments, we extract features from each segment. Different from wearable sensors that have constant sampling rates, ambient sensors often work in an event-triggering scheme. We in this section present two ways of extracting features from one segment t to get a feature vector $x_t = (x_1, x_2, \ldots, x_N)$, where N equals the number of sensors deployed in a smart home and x_i corresponds to the i^{th} sensor. First, we can extract *statistical* features to represent a segment.

1) *binary representation*. x_i shows whether sensor i fired in t. x_i equals 1 if it fired; otherwise, 0.

2) *numerical representation*. x_i records the number of firings of the i^{th} sensor in t.

3) *probability representation*. x_i is a normalized version of numerical representation and indicates the percentage of firings.

For example, for a smart home with five sensors, during a specific time interval, the first sensor was triggered twice and the third sensor was triggered three times, then the vectors of the three representations are $x_b = (1, 0, 1, 0, 0)$, $x_b = (2, 0, 3, 0, 0)$, and $x_b = (0.4, 0, 0.6, 0, 0)$, respectively.

Second, rather than use the number of occurrences, we can extract *structural* information.

4) *change point representation*. We set the value of 1 to time-slices where the sensor reading changes; otherwise, 0. This indicates the starting and ending times of the firings of a sensor.

5) *last fired representation*. The last sensor that changed state (i.e., corresponding to the starting and ending times) continues to be given 1 until a different sensor changes state.

Furthermore, we can jointly use the feature encodings to form a concatenated feature vector. For example, we can use *binary* and *numerical representations* or use *binary*, *change point* and *last fired representations*.

III. EXPERIMENTAL SETTING AND RESULTS

A. Experimental Setting

Comparative experiments are performed on three public activity recognition datasets as shown in Table I [19], where activities of interest are sensed and recorded by a collection of ambient sensors when a resident living in a smart home performed activities of daily living. The used sensors are attached to or placed on the ambient objects and idle activity is not considered [17]. We use a time-based sliding window with a size of 60s to segment time-series sensor events and then extract features from each segment to get a feature vector. To compare the power of different types of features, six widely used classification models with different metrics, including naïve Bayes (NB), k nearest-neighbor with k = 1 (1NN), support vector machine with linear kernel (SVM), decision tree (DT), hidden Markov model (HMM), and hidden semi-Markov model (HSMM) are employed to train activity recognizers towards unbiased evaluation [17].

TABLE I. Description of the experimental datasets

Dataset	SH1	SH2	SH3
Number of sensors	14	23	21
Number of residents	1	1	1
Number of activities	9	12	15
Number of sensor events	1229	19075	22700
Number of days monitored	25	14	19

B. Performance Metrics

To generate independent training and test sets, leave one day cross validation scheme is adopted. That is, sensor data of one full day are used as the test set and sensor data of the remaining days are used as the training set. Afterwards, an activity recognizer is trained on the training set and makes predictions on the test set. The final results are the average of the above processes and we herein report both class accuracy and time-slice accuracy.

Given $L = \{L_1, L_2, \ldots, L_{|L|}\}$ to indicate a class space with $|L|$ classes, for a test sample x with true label y_x and predicted label $prd(x)$, the performance of a classifier associated with a test set X is defined as:

$$accuracy_{class} = \frac{1}{|L|}\sum_{l=1}^{|L|}\left\{\frac{\sum_{x\in L_l} I(prd(x)=y_x)}{N_l}\right\} \quad (1)$$

$$accuracy_{timeslice} = \frac{\sum_{x\in X} I(prd(x)=y_x)}{|X|} \quad (2)$$

, where I(a=b) is an indicator function, N_l is the number of samples from class L_l, and $|X|$ is the number of samples of X.

Besides time-slice accuracy, we report F1 to account for the class imbalance problem.

C. Experimental Results

Tables II-IV present the experimental results of different types of features and their combinations on the datasets, respectively. *Bin*, *Nume*, and *Prob* denote *binary*, *numerical*, and *probability* representations, respectively. For illustration purpose, we divide the results in three groups based on the use of feature representations. For each classification model, the best F1 is underlined and the best result in each group is shown in bold. *Class* and *slice* in the second row correspond to the class accuracy and time-slice accuracy, respectively. From Tables II-IV, we see that the use of *last* representation generally has better accuracy than that of *binary*, *numerical*, *probability*, and *change point* encodings. This is possibly because that the *last* representation can better characterize human activities. Second, as for the use of two types of features, we observe that the joint use of *change point* and *last* representations performs better in the majority of cases. Particularly, it achieves higher accuracy than the single use

TABLE II. Experimental results of different feature sets on SH1

Features	**NB**			**1NN**			**SVM**			**DT**			**HMM**			**HSMM**		
	class	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*
Bin	45.45	86.53	48.74	35.92	37.92	37.91	51.28	93.69	53.39	51.39	93.71	53.96	54.17	61.99	49.90	62.16	67.51	55.29
Nume	44.81	86.34	47.80	39.21	38.30	39.86	56.69	93.74	59.16	58.29	93.90	60.56	47.69	61.71	44.81	59.94	73.38	55.16
Prob	31.80	83.36	30.28	38.86	38.28	39.73	48.16	85.46	50.72	58.04	93.72	59.04	24.31	43.51	20.47	31.82	59.80	29.53
Change	49.14	62.28	53.22	44.33	37.79	49.11	50.96	62.34	55.41	49.21	62.31	53.40	79.65	93.83	76.16	81.35	93.82	77.80
Last	**76.19**	**98.80**	**72.07**	**75.08**	**98.25**	**70.20**	**75.91**	**98.79**	**72.83**	**75.68**	**98.83**	**71.83**	**81.69**	**98.69**	**76.45**	**83.42**	**98.71**	**77.80**
Bin&Change	59.06	87.15	59.61	42.97	45.43	44.24	63.49	95.24	65.15	62.78	94.39	65.41	64.71	62.80	56.65	74.82	81.01	66.74
Nume&Change	58.73	87.18	59.93	42.62	45.42	42.43	62.98	95.27	64.83	62.20	94.35	64.20	65.73	70.82	58.14	74.68	85.30	67.59
Prob&Change	56.21	85.18	57.99	43.22	45.44	43.68	58.97	88.23	61.15	62.92	94.26	62.70	60.18	56.07	54.34	67.83	70.35	62.20
Bin&Last	76.58	98.75	71.55	72.40	98.33	68.69	76.13	98.66	72.44	75.20	98.69	69.88	84.36	98.80	**80.14**	84.44	98.82	**80.68**
Nume&Last	77.68	98.86	**73.39**	74.14	98.47	68.80	76.46	98.77	71.77	76.17	98.68	71.92	82.85	98.87	78.67	83.90	98.86	78.95
Prob&Last	72.39	98.74	70.93	75.90	98.52	70.83	75.77	98.77	71.61	75.48	98.63	71.28	81.65	98.91	78.38	82.90	98.96	79.73
Change&Last	**79.82**	**98.95**	73.24	**77.08**	**98.47**	**72.44**	**78.40**	**98.92**	**73.31**	**77.66**	**98.91**	**74.37**	**85.48**	**98.98**	80.02	**85.59**	**98.98**	79.60
Bin&Change&Last	79.57	98.87	73.34	74.90	98.58	70.11	77.07	98.78	**72.57**	75.87	98.59	70.78	**85.08**	98.82	**80.24**	85.36	98.86	**80.15**
Nume&Change&Last	**79.92**	98.87	**73.61**	75.80	98.56	69.74	76.51	98.79	71.69	**76.48**	**98.68**	**72.46**	84.81	99.01	79.62	**85.25**	98.98	79.65
Prob&Change&Last	77.13	**98.88**	71.89	**76.22**	**98.60**	**71.64**	**77.48**	**98.82**	71.92	76.03	98.63	71.76	83.66	**99.02**	78.66	84.17	**99.07**	79.08

TABLE III. Experimental results of different feature sets on SH2

Features	**NB**			**1NN**			**SVM**			**DT**			**HMM**			**HSMM**		
	class	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*
Bin	40.44	89.17	39.09	35.86	64.90	33.85	47.52	81.45	46.90	47.81	81.70	44.28	46.97	62.24	42.41	48.59	65.53	44.39
Nume	32.48	88.44	31.98	48.70	71.35	42.57	**56.16**	86.31	**53.83**	**56.44**	84.33	50.16	39.49	71.41	36.49	41.54	71.54	38.44
Prob	25.28	68.62	24.25	48.09	70.98	42.69	45.27	**92.53**	43.04	51.84	80.38	48.64	30.58	**89.05**	30.23	30.58	**89.05**	30.23
Change	47.80	71.03	46.81	38.19	39.30	34.89	47.51	70.57	48.13	50.08	71.19	50.16	**67.91**	82.80	**57.41**	**67.67**	86.08	**55.68**
Last	**51.78**	**89.87**	**49.11**	**50.54**	**84.58**	**46.83**	53.57	88.54	50.25	56.20	**88.51**	**54.13**	50.84	48.47	42.10	64.11	67.23	51.69
Bin&Change	56.46	92.04	53.35	43.79	69.53	40.26	55.22	89.09	52.56	54.48	**88.84**	50.18	61.97	78.25	56.19	58.45	79.19	52.13
Nume&Change	52.57	**92.08**	51.22	48.18	69.60	42.18	59.06	88.57	56.99	57.91	87.06	52.46	59.86	81.20	53.53	61.38	81.33	55.60
Prob&Change	46.02	74.13	46.06	49.78	72.22	44.67	60.72	**92.34**	57.79	53.39	84.17	50.64	**62.41**	**90.82**	**56.56**	63.12	91.23	58.50
Bin&Last	57.78	87.27	**54.76**	46.39	78.23	43.22	58.29	88.14	53.96	57.77	85.57	51.06	60.90	73.33	54.56	65.32	75.17	57.67
Nume&Last	53.41	89.99	51.79	52.07	78.90	**47.63**	**63.75**	88.89	57.39	57.02	86.31	51.81	60.93	79.17	54.17	61.13	82.15	54.37
Prob&Last	49.23	88.70	49.46	51.78	79.04	45.78	62.27	90.48	**58.13**	57.16	83.18	50.87	61.45	80.85	56.76	66.35	**90.23**	**61.03**
Change&Last	**59.32**	89.60	52.10	**53.69**	**86.74**	46.60	59.96	89.76	57.19	**58.82**	88.72	**54.22**	55.41	51.20	45.46	**69.22**	85.82	53.51
Bin&Change&Last	**60.85**	90.25	**54.88**	48.93	79.01	44.33	59.09	88.59	56.13	53.06	**88.39**	47.38	62.88	73.71	53.69	66.86	82.78	59.67
Nume&Change&Last	60.29	**90.70**	54.45	**52.69**	**80.98**	**46.19**	59.97	88.62	56.30	57.13	86.30	51.87	**66.25**	84.25	58.32	67.36	84.29	59.06
Prob&Change&Last	57.83	89.75	52.69	52.11	80.28	46.12	**62.28**	**91.81**	**59.26**	**57.34**	86.04	**51.92**	65.09	**86.69**	**58.32**	**69.76**	**90.18**	**62.23**

TABLE IV. Experimental results of different feature sets on SH3

Features	NB			1NN			SVM			DT			HMM			HSMM		
	class	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*	*class*	*slice*	*F1*
Bin	14.82	49.17	14.93	17.14	26.91	15.77	21.19	54.21	18.96	18.06	34.21	16.17	16.77	24.79	14.60	20.10	27.00	15.70
Nume	11.96	47.35	12.74	26.73	33.06	26.41	30.83	52.88	30.96	23.55	40.91	21.32	11.22	27.04	10.86	16.70	38.53	14.73
Prob	11.44	49.88	9.57	27.93	34.03	27.08	23.37	61.18	23.20	28.10	43.21	27.55	12.39	34.53	10.31	12.14	40.03	9.66
Change	34.52	69.08	38.70	36.36	41.49	36.25	34.39	68.90	37.49	34.91	69.08	39.50	55.03	87.30	48.48	55.49	87.99	50.07
Last	**50.23**	**89.15**	**45.62**	**51.18**	**88.36**	**48.06**	**50.08**	**94.32**	**45.75**	**51.65**	**94.51**	**48.86**	**60.57**	**88.80**	**53.79**	**62.44**	**88.89**	**55.75**
Bin&Change	35.12	60.40	36.20	27.33	33.07	28.81	34.44	60.76	36.17	26.14	40.96	25.48	36.84	46.18	32.32	37.81	49.12	33.55
Nume&Change	32.99	60.02	35.12	30.10	34.01	29.68	36.33	60.61	37.84	26.72	41.84	27.35	40.06	60.02	35.99	42.75	66.71	39.83
Prob&Change	31.57	60.72	33.58	33.01	35.90	31.40	34.06	54.88	36.93	29.11	41.81	27.28	41.50	63.26	37.18	41.89	63.44	37.20
Bin&Last	51.86	**92.89**	46.37	43.22	75.61	40.95	47.93	82.66	44.75	44.49	84.04	40.57	50.12	81.80	44.87	55.02	87.11	47.98
Nume&Last	50.86	92.87	47.26	43.35	77.37	40.45	48.10	84.03	44.82	44.39	82.78	40.11	50.28	82.10	45.69	56.24	88.09	50.08
Prob&Last	49.20	93.13	44.05	47.86	86.61	43.94	**52.08**	94.21	47.05	43.02	81.36	38.51	48.01	81.34	43.72	53.92	86.62	48.27
Change&Last	**53.34**	92.01	**47.61**	**52.54**	**88.61**	**49.44**	50.66	**94.47**	**47.55**	**50.57**	**94.44**	**47.81**	**60.71**	**88.80**	**53.20**	**61.65**	**88.89**	**54.21**
Bin&Change&Last	**53.14**	92.73	47.23	45.11	79.44	44.06	47.54	82.69	45.66	**44.66**	84.04	39.99	57.19	87.45	50.25	57.63	88.20	51.41
Nume&Change&Last	52.40	**93.45**	**47.63**	44.86	79.20	40.98	48.39	85.15	44.89	44.31	**84.06**	**40.25**	**57.61**	**88.19**	**51.08**	**58.80**	**88.43**	**52.10**
Prob&Change&Last	50.66	92.96	44.43	**49.10**	**88.02**	**45.04**	**53.36**	**94.51**	**49.86**	43.55	83.25	39.02	55.80	86.96	49.18	56.79	87.26	50.72

TABLE V. Time-slice accuracy of different window sizes

	Features	NB			1NN			SVM			DT			HMM			HSMM		
		30s	*60s*	*90s*	*30s*	*60s*	*90s*	*30s*	*60s*	*90s*	*30s*	*60s*	*90s*	*30s*	*60s*	*90s*	*30s*	*60s*	*90s*
SH1	*Last*	98.95	98.80	97.63	98.55	98.25	97.27	99.01	98.79	97.66	99.06	98.83	97.53	98.91	98.69	97.81	98.95	98.71	97.85
	Change&Last	99.16	98.95	97.86	98.59	98.47	97.62	99.07	98.92	97.84	99.10	98.91	97.74	99.10	98.98	97.92	99.11	98.98	97.90
	Bin&Change&Last	99.09	98.87	97.75	98.76	98.58	96.85	99.03	98.78	97.70	98.83	98.59	96.59	99.00	98.82	97.80	99.03	98.86	97.82
	Nume&Change&Last	99.06	98.87	97.73	98.73	98.56	96.84	99.02	98.79	97.70	98.82	98.68	96.68	99.07	99.01	97.86	99.12	98.98	97.87
	Prob&Change&Last	99.17	98.88	97.81	98.70	98.60	96.88	99.08	98.82	97.79	98.70	98.63	96.67	99.21	99.02	97.98	99.22	99.07	98.01
SH2	*Last*	90.18	89.87	88.54	87.65	84.58	85.49	90.29	88.54	88.87	90.27	88.51	87.25	51.26	48.47	48.96	65.42	67.23	80.45
	Change&Last	90.05	89.60	88.86	87.70	86.74	89.14	90.19	89.76	89.74	90.12	88.72	89.36	51.49	51.20	74.19	69.97	85.82	89.83
	Bin&Change&Last	90.52	90.25	89.32	81.52	79.01	82.02	87.49	88.59	90.72	90.56	88.39	90.85	68.69	73.71	80.41	79.65	82.78	82.64
	Nume&Change&Last	90.76	90.70	89.77	81.64	80.98	81.78	86.26	88.62	90.46	92.02	86.30	87.73	71.14	84.25	83.72	80.38	84.29	83.72
	Prob&Change&Last	89.62	89.75	88.80	80.12	80.28	78.73	89.72	91.81	95.67	84.21	86.04	85.07	73.63	86.69	89.27	81.33	90.18	89.85
SH3	*Last*	88.22	89.15	91.54	88.28	88.36	87.05	94.87	94.32	94.00	94.92	94.51	93.71	89.06	88.80	88.04	89.22	88.89	87.98
	Change&Last	89.62	92.01	93.75	88.32	88.61	87.37	94.62	94.47	93.71	94.74	94.44	93.36	88.98	88.80	87.90	89.04	88.89	88.11
	Bin&Change&Last	93.28	92.73	92.76	76.77	79.44	77.35	81.79	82.69	81.04	85.25	84.04	84.80	82.34	87.45	87.24	87.18	88.20	87.38
	Nume&Change&Last	93.17	93.45	93.93	76.92	79.20	77.50	79.92	85.15	86.52	85.12	84.06	84.97	85.13	88.19	87.52	87.58	88.43	88.97
	Prob&Change&Last	93.39	92.96	92.53	88.06	88.02	86.68	94.89	94.51	93.45	82.10	83.25	80.38	81.77	86.96	86.62	87.00	87.26	88.27

of *last* representation, which, to a certain extent, indicates the information complementary of different types of features. Third, in terms of the combination of three different types of features, we observe that they achieve comparable results. Compared with *change point* and *last* representations, we observe that they generally obtain slightly worse accuracy. The main reason is that there probably exist redundant and irrelevant features [20,21].

Besides, to further analyze the recognition results, we investigate the confusion matrix and only present the results of using NB on SH1 in Figure 2 due to the limited space. The rows (columns) denote the true (predicted) labels. We observe that the use of *last* representation better recognizes activities 1 and 5 compared to the use of *binary*, *numerical*, *probability*, and *change point* representations. For example, 1785 samples of activity 5 are misclassified into activity 1 for *binary* representation, the use of *probability* encoding makes 89 errors, and the *last* representation does not classify activity 5 into activity 1. Second, compared with the *last* representation, the joint use of *change point* and *last* encodings further enhances the overall recognition accuracy. Last, we observe that activities 1 and 5 have much more samples. This demonstrates the necessity of using F1 and reminds us of the class imbalance problem.

The sliding-window size is an important parameter in determining the predictive ability of an activity recognizer, where a small size probably misses the sensor readings of an activity and a large size may involve multiple activities. We here experimentally evaluate its effects with three different values (i.e., 30s, 60s, and 90s) on the time-slice accuracy. Table V gives the experimental results. We only present the results of five types of features (i.e., *last*, *change* & *last*, *bin* & *change* & *last*, *nume* & *change* & *last*, and *prob* & *change* & *last*), since they performed better than others as shown in Tables II-IV. From Table V, we observe that the sliding-window size has an impact on the recognition rate and that a larger size tends to get degraded accuracy. This is mainly because the segment of a larger size contains sensor data associated with multiple activities. We also observe that the use of 60s achieves comparable performance to that of 30s. In view of the time costs in marking predictions, the use of 60s achieves a better accuracy-speed tradeoff.

	1	2	3	4	5	6	7	8	9
1	17638	1	0	0	1681	1	0	0	0
2	180	92	0	0	5	2	10	0	0
3	214	3	0	0	13	0	0	0	0
4	27	2	0	0	0	0	2	0	0
5	1785	7	0	0	9043	0	4	0	0
6	17	0	0	0	2	24	16	3	6
7	213	2	0	0	0	21	41	6	2
8	8	0	0	0	0	9	10	3	2
9	5	0	0	0	0	5	8	1	13

(a) *Bin*

	1	2	3	4	5	6	7	8	9
1	17653	1	0	0	1681	0	0	2	0
2	199	134	0	0	22	2	8	0	0
3	234	4	0	0	13	0	0	0	0
4	26	5	0	0	0	0	1	0	0
5	1785	12	0	0	9030	0	2	0	0
6	23	0	0	0	3	25	29	4	3
7	190	1	0	0	0	11	71	11	3
8	12	0	0	0	1	7	16	4	2
9	14	0	0	0	0	4	11	2	18

(b) *Nume*

	1	2	3	4	5	6	7	8	9
1	15117	0	0	0	4220	0	0	0	0
2	145	0	0	0	220	0	0	0	0
3	195	0	0	0	56	0	0	0	0
4	30	0	0	0	2	0	0	0	0
5	89	0	0	0	10740	0	0	0	0
6	27	0	0	0	60	0	0	0	0
7	124	0	0	0	163	0	0	0	0
8	11	0	0	0	31	0	0	0	0
9	8	0	0	0	41	0	0	0	0

(c) *Prob*

	1	2	3	4	5	6	7	8	9
1	19322	6	5	0	0	0	1	3	0
2	48	291	14	1	8	1	1	0	1
3	161	11	79	0	0	0	0	0	0
4	10	11	11	0	0	0	0	0	0
5	10741	32	4	0	52	0	0	0	0
6	16	2	0	0	0	35	23	10	1
7	181	0	0	0	0	19	72	9	6
8	2	0	0	0	1	13	20	6	0
9	4	0	0	0	0	6	21	0	18

(d) *Change*

	1	2	3	4	5	6	7	8	9
1	19318	9	9	0	0	0	1	0	0
2	17	293	19	1	25	1	9	0	0
3	3	20	226	0	0	0	2	0	0
4	3	14	11	1	0	0	3	0	0
5	0	32	7	0	10789	1	0	0	0
6	0	3	0	0	1	39	33	9	2
7	0	2	2	0	0	17	229	20	17
8	1	0	1	0	1	6	23	10	0
9	1	5	1	0	0	5	18	1	18

(e) *Last*

	1	2	3	4	5	6	7	8	9
1	19310	9	13	0	0	0	1	3	1
2	4	312	16	7	15	1	9	0	1
3	1	15	232	2	0	0	1	0	0
4	3	13	12	0	0	0	4	0	0
5	0	43	6	2	10778	0	0	0	0
6	0	3	0	0	1	42	23	11	7
7	0	0	2	0	0	24	239	11	11
8	0	0	0	0	1	13	12	15	1
9	0	0	0	0	0	8	6	0	35

(f) *Change&Last*

	1	2	3	4	5	6	7	8	9
1	19307	8	13	4	0	0	1	3	1
2	4	299	18	11	13	2	17	0	1
3	1	15	233	1	0	0	1	0	0
4	3	12	11	3	0	0	3	0	0
5	0	44	6	2	10776	0	1	0	0
6	0	2	1	0	1	44	25	7	7
7	0	0	2	0	0	26	235	13	11
8	0	0	0	0	1	12	15	12	2
9	0	0	0	0	0	7	11	1	30

(g) *Nume&Change&Last*

	1	2	3	4	5	6	7	8	9
1	19310	8	14	0	0	0	1	2	2
2	4	288	36	8	14	1	12	0	2
3	1	15	233	1	0	0	1	0	0
4	3	13	12	0	0	0	4	0	0
5	0	36	10	3	10780	0	0	0	0
6	0	3	0	0	2	37	29	11	5
7	0	1	2	0	0	11	257	11	5
8	0	0	0	0	1	10	20	11	0
9	0	0	0	0	0	7	17	0	25

(g) *Prob&Change&Last*

Figure 2. Confusion matrix on SH1 of different feature representations using NB.

IV. CONCLUSION

Activity recognition in smart homes greatly advances the design and implementation of ambient assisted living tools and applications towards healthy aging, in which how to extract features from time-series sensor events largely determines the performance and applicability of an activity recognizer. We in this study conduct a comparison of feature extraction methods for ambient sensor-based human activity recognition. Specifically, we extract and evaluate five types of feature encodings and their different combinations. We also evaluate the impact of sliding-window size on prediction accuracy. Experiments are conducted on three public datasets in terms of fifteen feature encodings and six classification models. Results show that the joint use of different features gets enhanced performance and that the use of *change point* and *last* representations generally works better. Besides, preliminary results indicate that the sliding-window size is an important factor that needs further investigation.

ACKNOWLEDGMENT

This work was supported by the Featured Innovation Project of the Department of Education of Guangdong Province (No. 2021KTSCX117), Anhui Provincial Major Scientific and Technological Special Project (No. 201903A06020026), Natural Science Research Project of the Anhui Higher Education Institution of China (No. KJ2020A0725), and Natural Science Foundation of Anhui Province (No. 2108085MF208).

REFERENCES

[1] Dahmen, Jessamyn, and Diane J. Cook. "Indirectly supervised anomaly detection of clinically meaningful health events from smart home data." *ACM Transactions on Intelligent Systems and Technology* (TIST) 12, no. 2 (2021): 1-18.

[2] Acampora, Giovanni, Diane J. Cook, Parisa Rashidi, and Athanasios V. Vasilakos. "A survey on ambient intelligence in healthcare." *Proceedings of the IEEE* 101, no. 12 (2013): 2470-2494.

[3] Tapia, Emmanuel Munguia, Stephen S. Intille, and Kent Larson. "Activity recognition in the home using simple and ubiquitous sensors." In International Conference on Pervasive Computing, pp. 158-175. Springer, Berlin, Heidelberg, 2004.

[4] Guo, Jinghuan, Yiming Li, Mengnan Hou, Shuo Han, and Jianxun Ren. "Recognition of daily activities of two residents in a smart home based on time clustering." *Sensors* 20, no. 5 (2020): 1457.

[5] Avci, Akin, Stephan Bosch, Mihai Marin-Perianu, Raluca Marin-Perianu, and Paul Havinga. "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey." In 23th International Conference on Architecture of Computing Systems 2010, pp. 1-10. VDE, 2010.

[6] Wang, Aiguo, Shenghui Zhao, Chundi Zheng, Huihui Chen, Li Liu, and Guilin Chen. "HierHAR: Sensor-based data-driven hierarchical human activity recognition." *IEEE Sensors Journal* 21, no. 3 (2020): 3353-3365.

[7] Dang, L. Minh, Kyungbok Min, Hanxiang Wang, Md Jalil Piran, Cheol Hee Lee, and Hyeonjoon Moon. "Sensor-based and vision-based human activity recognition: A comprehensive survey." *Pattern Recognition* 108 (2020): 107561.

[8] Nweke, Henry Friday, Ying Wah Teh, Mohammed Ali Al-Garadi, and Uzoma Rita Alo. "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges." *Expert Systems with Applications* 105 (2018): 233-261.

[9] Krishnan, Narayanan C., and Diane J. Cook. "Activity recognition on streaming sensor data." *Pervasive and Mobile Computing* 10 (2014): 138-154.

[10] Thapa, Keshav, Abdullah Al, Zubaer Md, Barsha Lamichhane, and Sung-Hyun Yang. "A deep machine learning method for concurrent and interleaved human activity recognition." *Sensors* 20, no. 20 (2020): 5770.

[11] Wang, Aiguo, Guilin Chen, Jing Yang, Shenghui Zhao, and Chih-Yung Chang. "A comparative study on human activity recognition using inertial sensors in a smartphone." *IEEE Sensors Journal* 16, no. 11 (2016): 4566-4578.

[12] Van Kasteren, Tim, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. "Accurate activity recognition in a home setting." In Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 1-9. 2008.

[13] Hamad, Rebeen Ali, Alberto Salguero Hidalgo, Mohamed-Rafik Bouguelia, Macarena Espinilla Estevez, and Javier Medina Quero. "Efficient activity recognition in smart homes using delayed fuzzy temporal windows on binary sensors." *IEEE Journal of Biomedical and Health Informatics* 24, no. 2 (2019): 387-395.

[14] Ramos, Raúl Gómez, Jaime Duque Domingo, Eduardo Zalama, and Jaime Gómez-García-Bermejo. "Daily human activity recognition using non-intrusive sensors." *Sensors* 21, no. 16 (2021): 5270.

[15] Liciotti, Daniele, Michele Bernardini, Luca Romeo, and Emanuele Frontoni. "A sequential deep learning application for recognising human activities in smart homes." *Neurocomputing* 396 (2020): 501-513.

[16] Yatbaz, Hakan Yekta, Sukru Eraslan, Yeliz Yesilada, and Enver Ever. "Activity recognition using binary sensors for elderly people living alone: Scanpath trend analysis approach." *IEEE Sensors Journal* 19, no. 17 (2019): 7575-7582.

[17] Wang, Aiguo, Shenghui Zhao, Chundi Zheng, Jing Yang, Guilin Chen, and Chih-Yung Chang. "Activities of daily living recognition with binary environment sensors using deep learning: A comparative study." *IEEE Sensors Journal* 21, no. 4 (2020): 5423-5433.

[18] Yan, Surong, Kwei-Jay Lin, Xiaolin Zheng, and Wenyu Zhang. "Using latent knowledge to improve real-time activity recognition for smart IoT." *IEEE Transactions on Knowledge and Data Engineering* 32, no. 3 (2019): 574-587.

[19] van Kasteren, Timotheus Leonhard Martinus. "Activity recognition for health monitoring elderly using temporal probabilistic models." ASCI, 2011.

[20] Wang, Aiguo, Guilin Chen, Xi Wu, Li Liu, Ning An, and Chih-Yung Chang. "Towards human activity recognition: A hierarchical feature selection framework." *Sensors* 18, no. 11 (2018): 3629.

[21] Wang, Aiguo, Ning An, Guilin Chen, Lian Li, and Gil Alterovitz. "Accelerating wrapper-based feature selection with K-nearest-neighbor." *Knowledge-Based Systems* 83 (2015): 81-91.