



## Reducing Operational Time Complexity of k-NN Algorithms Using Clustering in Wrist-Activity Recognition

Sun-Taag Choe, We-Duke Cho, Jai-Hoon Kim, and Ki-Hyung Kim

Graduate School of Electrical and Computer Engineering, Ajou University  
Suwon 443-749, Republic of Korea

### ABSTRACT

Recent research on activity recognition in wearable devices has identified a key challenge: k-nearest neighbors (k-NN) algorithms have a high operational time complexity. Thus, these algorithms are difficult to utilize in embedded wearable devices. Herein, we propose a method for reducing this complexity. We apply a clustering algorithm for learning data and assign labels to each cluster according to the maximum likelihood. Experimental results show that the proposed method achieves effective operational levels for implementation in embedded devices; however, the accuracy is slightly lower than that of a traditional k-NN algorithm. Additionally, our method provides the advantage of controlling the computational burden, depending on the performance of the embedded device on which the algorithm is implemented.

**KEY WORDS:** Embedded Wearable Device, Human-Activity Recognition, Instance Reduction, k-Nearest Neighbors, k-Means Clustering, Triaxial Signal

### 1 INTRODUCTION

IN recent years, various wellness and care services have begun using wearable medical devices (often wristbands). Additionally, many services reward clients for having healthy lifestyles. For example, some insurance providers reduce premiums for customers who exercise regularly. These programs have been the subject of many pilot projects; however, the number of existing wearable devices that accurately analyze calorie expenditure is limited. In some cases, participants were able to produce false data that gave the appearance of actual exercise. Some common methods for generating these data are using a massage belt or putting a wristband in a spin-dryer (to mimic movements in high-intensity exercise).

Current calorie-expenditure estimations are based on wrist motion and convert a signal into a measure of intensity by analyzing the magnitude of the acceleration signal. Thus, many studies have attempted to calculate calorie expenditure using signal analysis, pattern recognition, and feature classification for determining activity intensity and duration. However, most pattern-recognition techniques are too complex to implement in embedded wearable devices,

leading to problems in the development of certain services.

### 2 RELATED WORKS

IN this domain, numerous studies have been conducted using probabilistic and statistical methods, and many of these utilized Bayesian network (BN) or naive Bayesian (NB) classifiers [Bao and Intille (2004), Maurer et al. (2006), Tapia et al. (2007), Jatoba et al. (2008), Pham and Abdelzaher (2008), Altun and Barshan (2010), Lara et al. (2012), Lara and Labrador (2012, Jan)]. These statistical methods require large amounts of learning data, which increases the complexity of the statistical models. Research has been conducted using linear discriminant analysis (LDA) [Cheng et al. (2010)]; however, LDA cannot adequately handle multimodal data. Additionally, several studies employed regression analysis [Riboni and Bettini (2011), Lara et al. (2012), Zhu and Sheng (2009), Pham and Abdelzaher (2008), Lee et al. (2011)]; however, a similar problem was encountered in these studies. Researchers applied a hidden Markov model (HMM) to analyze the continuity of certain behaviors. While this was a novel approach, it increased the complexity of the probability models.

In other studies [He and Jin (2008, Jul), He et al. (2008, Dec), He and Jin (2009)], a support vector machine (SVM) was used to transform the feature space and domain for activity recognition. Additionally, multilayer perceptron (MLP) has been applied as an artificial neural network model [Altun and Barshan (2010), Khan et al. (2010), Lara and Labrador (2012, Jan)]. These solutions are widely used and have a high accuracy; however, they lack explanatory capabilities. Furthermore, these methods require many feature space dimensions, and identifying reasons for the occurrence of errors is difficult. Finally, when the model is updated, it can only be modified by repeating the learning process.

Triaxial sensors are optimal for determining physical properties during activity-recognition studies. Hence, in previous studies [Chen et al. (2008), Kao et al. (2009), Berchtold et al. (2010, Sep), Berchtold et al. (2010, Oct)], activity-recognition models employing a fuzzy-inference system were designed. In human behavior, walking, running, and rest states each have obvious value gaps; thus, low-complexity, feature-based activity-recognition models can be easily designed with minimal learning data. However, these systems experience difficulty in choosing correct physical features and setting decision boundaries between real and fake physical activities.

Other studies [Bao and Intille (2004), Ravi et al. (2005), Maurer et al. (2006), Jatoba et al. (2008), Alttal et al. (2015), Altun and Barshan (2010)] employed the k-nearest neighbors (k-NN) algorithm, which is one mode of instance-based learning. This method resulted in excellent classification and pattern-recognition performance. Additionally, it is effective for regression analysis, owing to its high explanatory power. Nevertheless, the method has high computational complexity, because the model constitutes an entire instance during the learning process.

Decision-tree methods (ID3, C4.5 CART) [Bao and Intille (2004), Hanai et al. (2004), Maurer et al. (2006), Tapia et al. (2007), Ernes (2008), Jatoba et al. (2008), Altun and Barshan (2010), Lara et al. (2012), Lara and Labrador (2012, Jan)] also have considerable explanatory power; thus, they are widely used in action-recognition research. These methods have high computational complexity; however, this can be controlled to some degree. Nevertheless, if the decision boundaries are not rectangular, the data distribution fails, and the overall results are poor.

Bagging and boosting—ensemble algorithms that use decision trees—are voting-based (weighted) metaclassifiers. These metaclassifiers are produced as a result of many parallel sub-classifiers or a strong classifier connecting weak classifiers in a series [Minnen et al. (2007), McGlynn and Madden (2011), Lara et al. (2012)]. Such solutions require combining existing algorithms to ensure stable performance;

however, their computational complexity is too high to implement in small embedded devices.

**Table 1. Related works for activity recognition**

| Algorithm              | Related Works  |
|------------------------|--|
| Bayesian (BN, NB)      | [Bao and Intille (2004)],<br>[Maurer et al. (2006)],<br>[Tapia et al. (2007)],<br>[Jatoba et al. (2008)],<br>[Pham and Abdelzaher (2008)],<br>[Altun and Barshan (2010)],<br>[Lara et al. (2012)],<br>[Lara and Labrador (2012, Jan)]          |
| LDA                    | [Cheng et al. (2010)]  |
| Regression analysis    | [Riboni and Bettini (2011)],<br>[Lara et al. (2012)]   |
| HMM                    | [Zhu and Sheng (2009)],<br>[Pham and Abdelzaher (2008)],<br>[Lee et al. (2011)]  |
| SVM                    | [He and Jin (2008, Jul)],<br>[He et al. (2008, Dec)],<br>[He and Jin (2009)]   |
| MLP                    | [Altun and Barshan (2010)],<br>[Khan et al. (2010)],<br>[Lara and Labrador (2012, Jan)]  |
| Fuzzy-inference system | [Chen et al. (2008)],<br>[Kao et al. (2009)],<br>[Berchtold et al. (2010, Sep)],<br>[Berchtold et al. (2010, Oct)]   |
| k-NN                   | [Bao and Intille (2004)],<br>[Ravi et al. (2005)],<br>[Maurer et al. (2006)],<br>[Jatoba et al. (2008)],<br>[Ravi et al. (2005)],<br>[Alttal et al. (2015)]  |
| ID3, C4.5, CART        | [Bao and Intille (2004)],<br>[Hanai et al. (2004)],<br>[Maurer et al. (2006)],<br>[Tapia et al. (2007)],<br>[Ernes2008],<br>[Jatoba et al. (2008)],<br>[Altun and Barshan (2010)],<br>[Lara et al. (2012)],<br>[Lara and Labrador (2012, Jan)] |
| Boosting/bagging       | [Minnen et al. (2007)],<br>[McGlynn and Madden (2011)],<br>[Lara et al. (2012)]  |

The methods used in the aforementioned studies all yield a high recognition performance. However, owing to development issues, the latest wearable devices lack sufficient resources for these algorithms.

Put differently, the computational complexity of these methods is too high for use in small embedded devices. Therefore, we propose an activity-recognition algorithm with low computational complexity, which can operate in small embedded processors.

### 3 PROPOSED METHOD

#### 3.1 Overview

THE proposed method reduces the number of k-NN instances through k-means clustering. The k-NN algorithm has fewer matrix calculations than other algorithms. Thus, it is easy to implement, is relatively stable, and has satisfactory explanatory power. However, as noted in previous studies, this algorithm faces the critical challenge of performing many comparison calculations during operation. In the k-NN algorithm, the best k value should be empirically chosen, and the algorithm generally performs best when  $k = 1$ . In [Ravi et al. (2005)], the authors chose  $k = 7$  empirically; however, the k value in other studies was either 1 or was not mentioned because it was empirical. Thus, in the existing case studies, a characteristic was simply extracted and applied as a comparison target, rather than effectively applying the k-NN algorithm [Bao and Intille (2004), Parkka (2006), Maurer et al. (2006), Jatoba et al. (2008), Ravi et al. (2005), Alttal et al. (2015)]. Although unrelated to human-activity recognition in embedded devices, research has been conducted on fast searching for nearest neighbors through parallel processing using hardware processors [Li et al. (2011), Gracia et al. (2008), Gracia et al. (2010)].

To overcome the aforementioned challenge, we propose reducing the number of k-NN instances by dividing groups using a k-means clustering algorithm to extract the centroid of each group. All the extracted instances may not have the same label; thus, we assign a label according to the maximum likelihood (ML) in a particular group. The proposed classification method is shown in Figure 1.

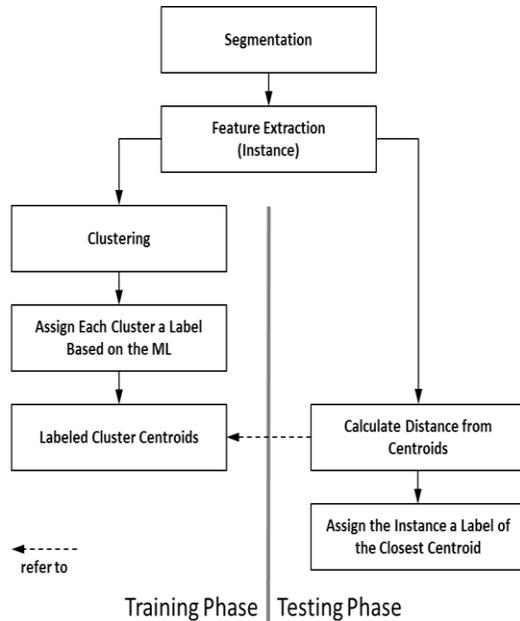


Figure 1. Overall procedure of the proposed method

#### 3.2 Feature Extraction

In previous triaxial sensor-based activity-recognition studies, various features were calculated to increase the accuracy, for example, the tendency (mean, median) in the time domain, the dispersion (standard deviation, mean, absolute deviation, entropy, inter-quartile range), histograms (cumulative, percentile), representative frequency, and the power in the frequency domain [Lara and Labrador (2013)].

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (1)$$

$$f_{S,mean} = \frac{1}{N} \sum_{i=1}^N a_{S,i} \quad (2)$$

where  $S = \{x, y, z, m\}$

$$f_{S,std} = \sqrt{\frac{1}{N} \sum_{i=1}^N (a_{S,i} - f_{S,mean})^2} \quad (3)$$

Resources are limited in miniaturized embedded environments. Thus, filters are not applied to reduce the computational complexity. Additionally, most previous studies extracted numerous characteristics to use, resulting in the calculation of characteristics that duplicated both the meaning and the distribution of values. For example, the meaning of the standard deviation is duplicated by the mean absolute deviation, entropy, and interquartile range mean dispersion, which have high correlation and similar distributions of calculated characteristic values. In the frequency-conversion operation and characteristic extraction procedures, the characteristics of the frequency domains were not calculated to reduce the computational complexity.

Thus, in the present study, we extracted the arithmetic mean, standard deviation, and maximum value, with the specific segment length (SL) on each axis of the basic time domain, without preprocessing. In addition, considering differences in sensor direction (on the wrist), we calculated the triaxial signal momentum and extracted the same three features for each subject. Equation 1 describes the calculation of the momentum signal from the triaxial signal. Here, “ $a$ ” represents the acceleration value, and the subscript indicates the axis. Equations 2 and 3 give the arithmetic mean and standard deviation, respectively, derived from each axis signal. The set  $S$  in these two equations includes each axis component ( $x, y, z$ ) and the momentum ( $m$ ), and  $N$  is the SL in the array, which refers to the sampling rate (SR) multiplied by unit time. Various results were obtained by changing the SR and SL (time, seconds). The feature space for each of the four signals (including momentum) had 12 dimensions. Figure 2 shows the acquired instances, with their labels, in three dimensions.

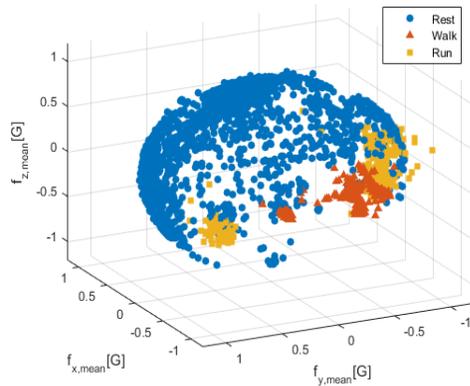


Figure 2. Acquired instances with their labels

### 3.3 Extracting Centroid Using k-Means Clustering

K-means [MacQueen (1967)] is a clustering algorithm (described in the flowchart of Figure 3) that first determines the number of clusters ( $k$ ) in the input data and then sets the number of centroids at an initial random location. Subsequently, i) the distances between all instances and centroids are calculated and ii) allocated to the same group. Finally, iii) the nearest centroid in each instance is individually selected. Next, if an allocated instance is changed from one group to another, the centroid now belongs to that same group. If there is no movement, the process is terminated. These processes are presented in the flowchart of Figure 3. In general, sound methods for determining the optimum cluster number for  $k$  must consider all options and should apply the Davies–Bouldin index [Davies and Bouldin (1979)].

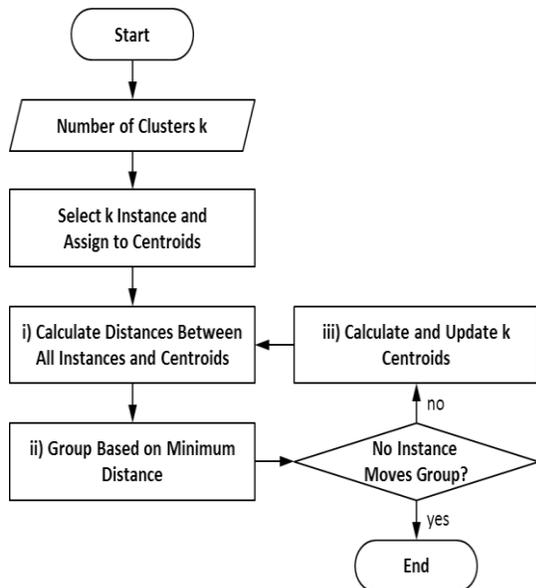


Figure 3. Flowchart of k-means clustering

The following equations correspond to key processes of the k-means clustering algorithm. Equation 4 expresses the calculation of the distances ( $D_{p,i}$ ) between all centroids ( $\mu_i$ ) and all instances ( $x_p$ ). We used the Euclidian distance for all equations. Process ii) in Figure 3 is represented by Equation 5, where all the  $D_{p,i}$  values obtained using  $x_p$  allocate the minimum  $i^{\text{th}}$  centroid ( $\mu_i$ ) of group ( $G_i$ ). Equation 6 represents process iii) and is a means of calculating the centroid ( $\mu_i$ ) from the renewed group ( $G_i$ ), which determines the numerical average for all instances ( $x_p$ ) in group ( $G_i$ ).

$$D_{p,i} = \sqrt{(x_p - \mu_i)^2}, \forall i, \forall p \quad (4)$$

$$G_i = \{x_p | p = \arg \min(D_{p,i})\}, \forall i, \forall p \quad (5)$$

$$\mu_i = \frac{1}{|G_i|} \sum_{x_p \in G_i} x_p, \forall i \quad (6)$$

In this study, we used the k-means algorithm to find the representative value of many acquired instances. This algorithm is similar to expectation maximization (EM), i.e., a clustering algorithm that is based on a probabilistic model and is highly effective. However, the EM algorithm can cause duplicate cluster detection depending on the probability distribution. Thus, it encounters difficulty in estimating computational complexity. For this reason, we used the k-means method, which reliably avoids duplicate clusters. This algorithm uses centroids extracted from each calculated cluster, which later replace instances. Figure 4 shows the instances with allocated acquired data, which were grouped using the k-means algorithm. Figure 5 shows all the centroids. To search for nearest neighbors in the embedded environment, we adjusted the number of clusters from 30 to 100, as experimental results revealed the recognition rate to be <90% with <30 clusters.

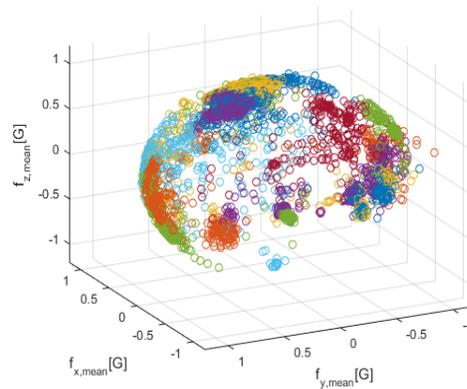
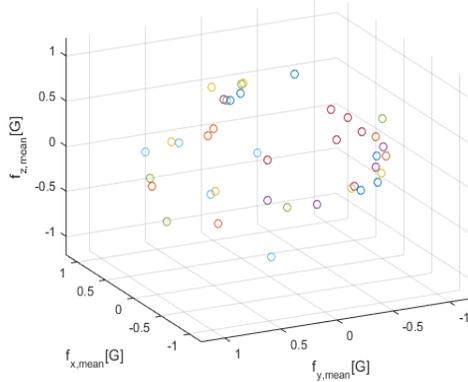


Figure 4. Instances grouped using the k-means algorithm



**Figure 5.** Extracted centroid of each group by k-means clustering

### 3.4 Assigning Label According to ML

Each extracted centroid represents one cluster, and each cluster contains several instances. These instances are each labeled according to their activity and time segment. Thus, one cluster is a gathering of similar activity instances, but each can have a different label. Therefore, in this study, ML was applied to the labels of many instances within a cluster, and these labels were assigned to a centroid. For a specific label (e.g., in Equations 7 and 8), the labels with the highest probability of instances belonging to that cluster were assigned. In Equation 7, the set  $H$  represents the collected behavior labels and has three elements: "Rest" =  $h_1$ , "Walk" =  $h_2$ , and "Run" =  $h_3$ . Equation 8 describes the process of allocating each group ( $G_i$ ) of the centroid ( $C_i$ ), where the most possible labels among each label ( $h_j$ ) in the group ( $G_i$ ) are assigned.

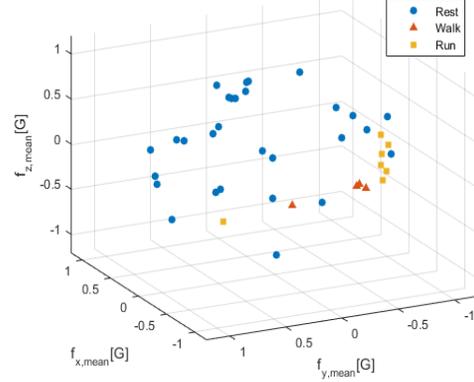
$$H = \left\{ \begin{array}{l} \text{"Rest"} = h_1, \text{"Walk"} = h_2, \\ \text{"Run"} = h_3 \end{array} \right\} \quad (7)$$

$$C_i = \arg \max_j \left( P(G_i | h_j) \right) \quad (8)$$

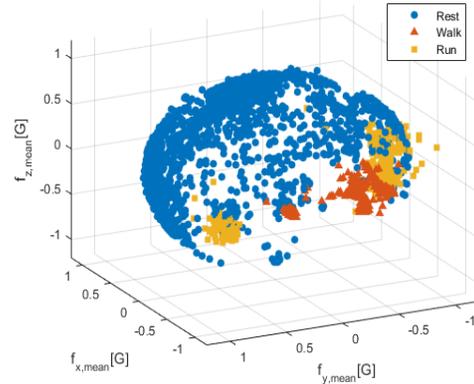
$$\forall i, 1 \leq j \leq |H|$$

Figure 6 represents centroids that were labeled by applying the ML method to each centroid through instances belonging to the group in Figure 4. Figure 7 shows the result of relabeling each instance according to calculated labeled centroids. There appears to be no difference from the original distributions (Figure 2); however, there is a small discernable change.

$$D_{X,Y} = \left( \sum_{i=1}^d |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (9)$$



**Figure 6.** Labeled centroid of each group by ML



**Figure 7.** Re-labeled instances in each group

### 3.5 k-NN Classification

The k-NN [Cover and Hart (1967)] algorithm is easy to implement and understand owing to its simple principles and high performance. This algorithm involves calculating the distances (and similarities) between an input target instance and all existing instances and then extracting the nearest previously set  $k$  neighbor instances. In the extracted nearest-neighbor instances, the highest frequency label besides the label of the input instance is identified. The  $k$  value is set to obtain the highest measurable performance. The distances between the instances are calculated using Equation 9 ( $d$  = dimension), which uses the  $X, Y$  vector to obtain the distance  $D_{X,Y}$  according to the Minkowski distance metric. Generally,  $p$  is 1 (Manhattan) or 2 (Euclidean). In this study, we set the value as 2.

Because we set the centroid to the representative value of the instance group, as a standard instance of activity recognition, the  $k$  value of the k-NN algorithm is set as 1. Figure 8 presents the distribution of the target instance for the tests, and Figure 9 shows the predicted result based on the centroid shown in Figure 6.

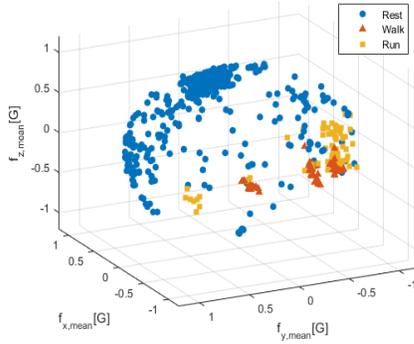


Figure 8. Target instance

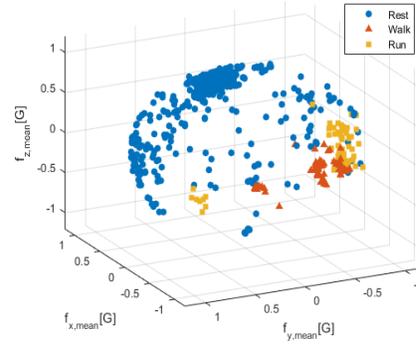


Figure 9. Predicted instance

## 4 PERFORMANCE EVALUATION

Table 2. Acquired signal lengths (in minutes) in the experiment

| Subjects     | 1               | 2   | 3  | 4  | 5   | Total      | Rate [%]      |              |
|--------------|-----------------|-----|----|----|-----|------------|---------------|--------------|
| Rest         | Working on Desk | 113 | 62 | 28 | 0   | 80         | 283           | 45.57        |
|              | Ridding Bus     | 36  | 0  | 0  | 0   | 62         | 98            | 15.78        |
|              | Belt Massager   | 5   | 5  | 0  | 0   | 5          | 15            | 2.42         |
|              | <b>Subtotal</b> | 154 | 67 | 28 | 0   | 147        | <b>396</b>    | <b>63.77</b> |
| Walk         | 24              | 28  | 24 | 23 | 24  | <b>123</b> | <b>19.81</b>  |              |
| Run          | 12              | 24  | 22 | 22 | 22  | <b>102</b> | <b>16.43</b>  |              |
| <b>Total</b> | 190             | 119 | 74 | 45 | 193 | <b>621</b> | <b>100.00</b> |              |

### 4.1 Experimental Scenario

TO evaluate the efficacy of the proposed algorithm, we measured the wrist-acceleration signals of five subjects, with acquired activity labels, over 621 min. The activities were walking in a field with various items (clothing, umbrella, and luggage), working at a desk, resting while using a smartphone, riding a bus, using a belt massager, and running on a treadmill. The measurable range of acceleration was  $\pm 2G$ , at an SR of 32 Hz. Table 2 presents the statuses of the collected data.

### 4.2 Performance Metrics

To compare the efficacy of k-NN with that of other algorithms, we calculated the number of instances used in recognition after reduction and measured the accuracy of each algorithm. The accuracy was defined as the probability that corresponded to each real and expected activity label in the entire test sample. When examining the behavioral cognition study using the k-NN algorithm, P-fold cross-validation was used with shuffled instances to evaluate the performance [Maurer et al. (2006), Altun and Barshan (2010), Ravi et al. (2005), Alttal et al. (2015)]. The P values employed in previous studies were as follows: 5 [Maurer et al. (2006)], 10 [Altun and Barshan (2010), Alttal et al. (2015)], and 12 [Ravi et al. (2005)]. In all instances, we applied P-fold cross-validation with P = 10, and all algorithms were applied to the same datasets. In the equally shuffled dataset, the proposed

algorithm applied k-means clustering while setting the first to  $k^{\text{th}}$  instances to the initial centroid.

$$R_{aa} = \frac{Accuracy_{knn}}{Accuracy_{algorithm}} \quad (10)$$

$$R_{ir} = 1 - \frac{N_{algorithm}}{N_{knn}} \quad (11)$$

In addition, we calculated the values of two evaluation factors using the k-NN algorithm that resulted from common target light-weighting and multiplied them by two additional factors. The first factor shows how each algorithm maintained the k-NN recognition rate by using the ratio of the achievement accuracy ( $R_{aa}$ , as shown in Equation 10). In Equation 10,  $Accuracy_{knn}$  represents the testing accuracy obtained from the current k-NN algorithm, and  $Accuracy_{algorithm}$  represents the testing accuracy after the application of the proposed instance-reduction method. The second factor shows the degree of reduction in the number of instances used in the k-NN algorithm relative to the ratio of overall instance reduction ( $R_{ir}$ , as shown in Equation 11). In Equation 11,  $N_{knn}$  represents the number of instances used for training in the current k-NN algorithm, and  $N_{algorithm}$  represents the reduced number of instances for the proposed method. These two factors become 0 in the worst case and 1 in the best case. Increasing the ratio of the achievement accuracy can improve recognition rate, thereby producing values greater than 1.

### 4.3 Results

**Table 3.** Accuracy (Acc.) and instance count (Inst. #) for different algorithms

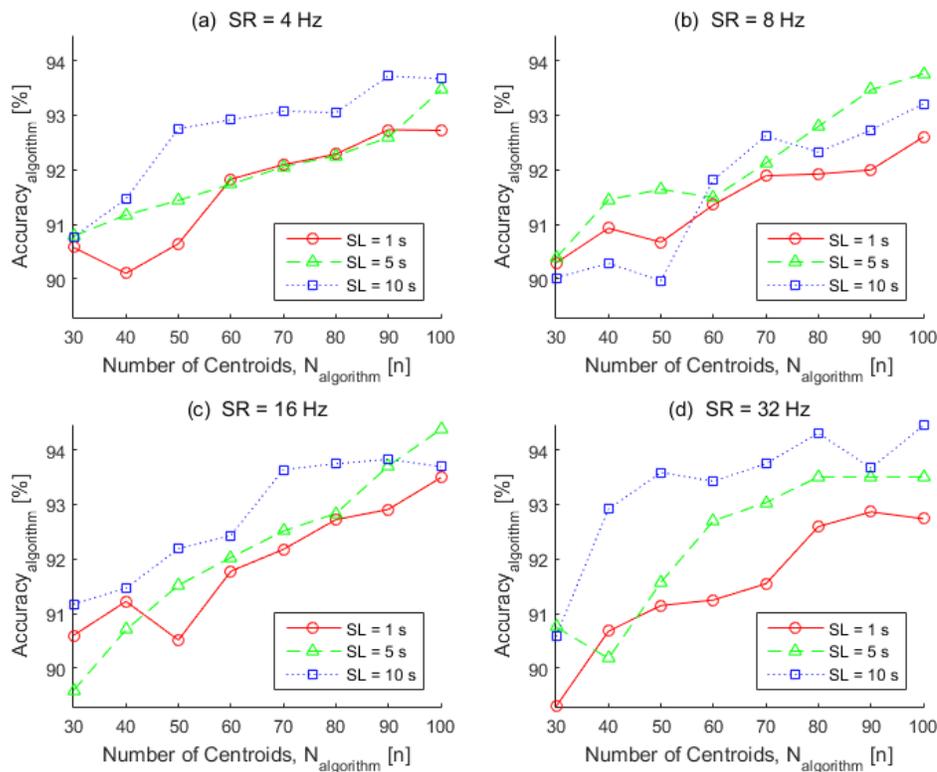
| SL [s] | Algorithm | Inst.#<br>[N] | SR [Hz], Accuracy with Standard Deviation [%] |              |              |              |
|--------|-----------|---------------|---|--------------|--------------|--------------|
|        |           |               | 4 Hz  | 8 Hz         | 16 Hz        | 32 Hz        |
| 1 s    | 1-NN      | 33534         | 96.94 ± 0.17                                  | 97.38 ± 0.24 | 97.62 ± 0.18 | 97.66 ± 0.17 |
|        | P30       | 30            | 90.57 ± 0.64                                  | 90.29 ± 0.42 | 90.60 ± 0.55 | 89.29 ± 0.55 |
|        | P40       | 40            | 90.10 ± 0.72                                  | 90.93 ± 0.24 | 91.22 ± 0.49 | 90.68 ± 1.01 |
|        | P50       | 50            | 90.64 ± 0.39                                  | 90.67 ± 0.39 | 90.51 ± 0.26 | 91.15 ± 0.39 |
|        | P60       | 60            | 91.83 ± 0.71                                  | 91.36 ± 0.40 | 91.78 ± 0.45 | 91.25 ± 0.38 |
|        | P70       | 70            | 92.09 ± 0.67                                  | 91.89 ± 0.35 | 92.18 ± 0.51 | 91.55 ± 0.72 |
|        | P80       | 80            | 92.29 ± 0.54                                  | 91.92 ± 0.37 | 92.72 ± 0.37 | 92.60 ± 0.41 |
|        | P90       | 90            | 92.73 ± 0.47                                  | 92.00 ± 0.28 | 92.91 ± 0.41 | 92.86 ± 0.56 |
|        | P100      | 100           | 92.72 ± 0.52                                  | 92.60 ± 0.32 | 93.49 ± 0.44 | 92.74 ± 0.62 |
|        | 5 s       | 1-NN          | 6706  | 97.61 ± 0.48 | 97.49 ± 0.63 | 97.50 ± 0.49 |
| P30    |           | 30            | 90.78 ± 0.97                                  | 90.38 ± 0.95 | 89.59 ± 1.17 | 90.75 ± 0.98 |
| P40    |           | 40            | 91.17 ± 1.09                                  | 91.45 ± 0.85 | 90.70 ± 1.16 | 90.18 ± 0.99 |
| P50    |           | 50            | 91.44 ± 1.05                                  | 91.64 ± 0.84 | 91.52 ± 1.23 | 91.57 ± 1.01 |
| P60    |           | 60            | 91.73 ± 0.98                                  | 91.49 ± 1.06 | 92.03 ± 1.13 | 92.70 ± 1.35 |
| P70    |           | 70            | 92.06 ± 0.84                                  | 92.12 ± 1.02 | 92.51 ± 1.06 | 93.04 ± 0.67 |
| P80    |           | 80            | 92.26 ± 0.77                                  | 92.79 ± 0.98 | 92.83 ± 0.60 | 93.51 ± 0.85 |
| P90    |           | 90            | 92.59 ± 1.05                                  | 93.46 ± 0.48 | 93.71 ± 0.65 | 93.51 ± 0.75 |
| P100   |           | 100           | 93.46 ± 0.89                                  | 93.76 ± 0.84 | 94.38 ± 0.38 | 93.51 ± 0.72 |
| 10 s   |           | 1-NN          | 3353  | 97.50 ± 1.09 | 97.56 ± 0.92 | 97.69 ± 0.48 |
|        | P30       | 30            | 90.77 ± 1.33                                  | 90.02 ± 1.93 | 91.17 ± 1.06 | 90.58 ± 1.81 |
|        | P40       | 40            | 91.47 ± 1.46                                  | 90.28 ± 2.10 | 91.47 ± 1.20 | 92.91 ± 1.68 |
|        | P50       | 50            | 92.75 ± 0.85                                  | 89.96 ± 1.63 | 92.19 ± 1.43 | 93.59 ± 1.39 |
|        | P60       | 60            | 92.91 ± 0.91                                  | 91.81 ± 1.58 | 92.43 ± 1.14 | 93.42 ± 1.32 |
|        | P70       | 70            | 93.08 ± 1.08                                  | 92.62 ± 1.68 | 93.64 ± 1.40 | 93.75 ± 1.44 |
|        | P80       | 80            | 93.05 ± 1.14                                  | 92.32 ± 1.55 | 93.75 ± 1.67 | 94.31 ± 1.49 |
|        | P90       | 90            | 93.72 ± 1.16                                  | 92.73 ± 1.15 | 93.83 ± 1.42 | 93.67 ± 1.45 |
|        | P100      | 100           | 93.67 ± 1.18                                  | 93.21 ± 1.05 | 93.69 ± 1.51 | 94.44 ± 1.24 |

Table 3 presents the accuracy (Acc.) and training instance count (Inst. #) of each algorithm. The SRs were 4, 8, 16, and 32 Hz, and the SLs were 1, 5, and 10 s. For these SLs, total number of extracted instances for each test was 37260, 7452, and 3726, respectively. Of these instances, 90% were used for training (33534, 6706, 3353), and 10% were used for testing (3726, 746, 37). In the performance-evaluation simulations, one instance comprised a minimum of four samples and a maximum of 320 samples, for each axis. In Table 3, “1-NN” represents the original k-NN algorithm, which did not apply a data-reduction method. In the range of P30 to P100, “k = 1” refers to the number of applied centroids (30–100) for the proposed method.

Figures 10(a)–(d) show the accuracy graphs for each applied SR. For each graph, the x-axis indicates the number of applied centroids, and the y-axis

indicates the estimated accuracy as a percentage. In each figure, one line represents an SR. Depending on the number of applied centroids, the accuracy increases with the SL. Depending on the SL, the number of k-NN instances (the object of reduction) and  $R_{lr}$  values increase.

Table 4 presents the performance of the proposed method. The results indicate that the proposed algorithm exhibited large differences in accuracy, corresponding to the number of 1-NN training instances and samples (sampling rate × segment time) comprising each instance. Thus, if our proposed algorithm has a high SR, it functions with high accuracy. Additionally, our method can clearly reduce the number of instances needed for classification. Moreover, our method is expected to solve the overfitting problems that arise from cluster-based classification. However, our experiments were



**Figure 10.** Accuracy<sub>algorithm</sub> for each SL and SR

conducted using a fixed number of centroids; thus, the reduction ratios increased with the number of 1-NN instances. Therefore, we were unable to reproduce a sufficient number of detailed decision boundaries; however, when the SR and SL were sufficiently high, the proposed method achieved satisfactory classification performance.

The graphs of each applied SR are shown in Figures 11 (a)–(d). The x-axis is same as that in Figure 10, and the y-axis indicates the  $R_{aa}$  value, which was between 0 and 1. Accuracy<sub>algorithm</sub> exhibited a different recognition rate according to the SR; however, there was no significant change in the  $R_{aa}$  of the relative difference with k-NN before instance reduction was applied. The recognition rate differed depending on the selected SL, as the number of instances used in the initial k-NN algorithm changed according to the SL. Thus, the effect of instance reduction using the proposed method that directly specifying the number of clusters for comparison was altered. Additionally, changing the calculated  $R_{ir}$  value to a percentage had the effect of further reducing the instance rate to 99.9%–97.0%, depending on the SL.

Figures 12(a) and (b) show the average Accuracy<sub>algorithm</sub> for each SL and SR. Of course, the SL and SR are proportional to the accuracy. The

algorithm can more clearly reflect representative features of behavior when the SL is larger, owing to the continuity of behavior in perception. Additionally, a higher SR produces greater accuracy, as the resolution of the signal for calculating the characteristics in the unit time is high.

The points in Figures 12(c) and (d) represent the average values of the Accuracy<sub>algorithm</sub> for all values of  $N_{algorithm}$ . The graphs in these figures show the average accuracy with respect to the SL at different SRs and with respect to the SR at different SLs, respectively. Overall, the correlation coefficient between each SL and the average value of all the accuracies was 0.995, and that between each SR and the average value of all the accuracies was 0.744. Both variables had strong correlations but different meanings. When both variables increased, with the accuracy increasing slightly, the number of operations increased linearly in the feature extraction. In the case where the SL increased, the execution cycle of the activity-recognition algorithm decreased, which slightly reduced the computational complexity. However, when the SR increased, the number of operations of the feature extraction increased. Thus, the SL should be considered over the SR in the implementation of the algorithm.

**Table 4.**  $R_{ir}$  and  $R_{aa}$  for each algorithm

| SL [s] | Algorithm | $R_{ir}$<br>[0 1] | SR [Hz], $R_{aa}$ [0 1] |        |        |        |
|--------|-----------|-------------------|-------------------------|--------|--------|--------|
|        |           |                   | 4 Hz                    | 8 Hz   | 16 Hz  | 32 Hz  |
| 1 s    | P30       | 0.9991            | 0.9343                  | 0.9272 | 0.9281 | 0.9143 |
|        | P40       | 0.9988            | 0.9294                  | 0.9338 | 0.9344 | 0.9285 |
|        | P50       | 0.9985            | 0.9350                  | 0.9311 | 0.9272 | 0.9333 |
|        | P60       | 0.9982            | 0.9473                  | 0.9382 | 0.9402 | 0.9344 |
|        | P70       | 0.9979            | 0.9500                  | 0.9436 | 0.9443 | 0.9374 |
|        | P80       | 0.9976            | 0.9520                  | 0.9439 | 0.9498 | 0.9482 |
|        | P90       | 0.9973            | 0.9566                  | 0.9448 | 0.9518 | 0.9508 |
|        | P100      | 0.9970            | 0.9565                  | 0.9509 | 0.9577 | 0.9496 |
| 5 s    | P30       | 0.9955            | 0.9300                  | 0.9271 | 0.9189 | 0.9316 |
|        | P40       | 0.9940            | 0.9340                  | 0.9380 | 0.9303 | 0.9258 |
|        | P50       | 0.9925            | 0.9368                  | 0.9400 | 0.9387 | 0.9400 |
|        | P60       | 0.9911            | 0.9398                  | 0.9385 | 0.9439 | 0.9516 |
|        | P70       | 0.9896            | 0.9431                  | 0.9449 | 0.9488 | 0.9551 |
|        | P80       | 0.9881            | 0.9452                  | 0.9518 | 0.9521 | 0.9600 |
|        | P90       | 0.9866            | 0.9486                  | 0.9587 | 0.9611 | 0.9600 |
|        | P100      | 0.9851            | 0.9575                  | 0.9617 | 0.9680 | 0.9600 |
| 10 s   | P30       | 0.9911            | 0.9310                  | 0.9227 | 0.9333 | 0.9303 |
|        | P40       | 0.9881            | 0.9382                  | 0.9254 | 0.9363 | 0.9542 |
|        | P50       | 0.9851            | 0.9513                  | 0.9221 | 0.9437 | 0.9612 |
|        | P60       | 0.9821            | 0.9529                  | 0.9411 | 0.9462 | 0.9594 |
|        | P70       | 0.9791            | 0.9547                  | 0.9494 | 0.9585 | 0.9628 |
|        | P80       | 0.9761            | 0.9544                  | 0.9463 | 0.9597 | 0.9686 |
|        | P90       | 0.9732            | 0.9612                  | 0.9505 | 0.9605 | 0.9620 |
|        | P100      | 0.9702            | 0.9607                  | 0.9554 | 0.9591 | 0.9699 |

#### 4.4 Time Complexity

The k-NN method had a time complexity of  $O(1)$  for training and  $O(N_{knn}dk)$  for testing (classification). Here,  $N_{knn}$  represents the number of instances for classification,  $d$  represents the dimensionality of the feature vectors, and  $k(1)$  represents the number of nearest instances. To reduce  $N_{knn}$ , the applied clustering method had a time complexity of  $O(N_{knn}N_{algorithm} Id)$  for the training process, where  $I$  represents the maximum number of iterations of the clustering operations, which was set as 1000.  $N_{algorithm}$  represents the number of target centroids in the clusters and the centroid model, which were the classification standards. After clustering, each centroid was labeled using the ML, which had a time complexity of  $O(N_{knn}N_{algorithm}d)$ . Thus, the entire time complexity of the training process was  $O(N_{knn}N_{algorithm} Id)$ , indicating that the proposed method had a time complexity of  $O(N_{algorithm}dk)$  for testing (classification). Compared with the k-NN algorithm, the proposed method can reduce the time complexity for classification, and the rate of instance reduction is  $R_{ir}$ .

## 5 CONCLUSION

THE proposed method is an effective data-distribution model owing to its efficacy for calculating cluster centroids. However, in some cases, the distributions of different classes slightly overlapped, and certain instances were interpreted as noise at the center of the data distribution in some classes. Nevertheless, our method can provide more stable performance by simplifying decision boundaries and overriding this noise component.

A data-reduction method was proposed for k-NN comparison operations. This method employs the k-means clustering algorithm and can be implemented in low-performance embedded mobile devices. The method facilitated controllability for accurate target time complexity. In our simulations, we fixed the initial values in the k-means clustering process; however, when the initial value was randomly selected and the clustering process was repeated, the performance improved. In an actual embedded processing environment, multiplex and memory-access operations are very complex; therefore, a

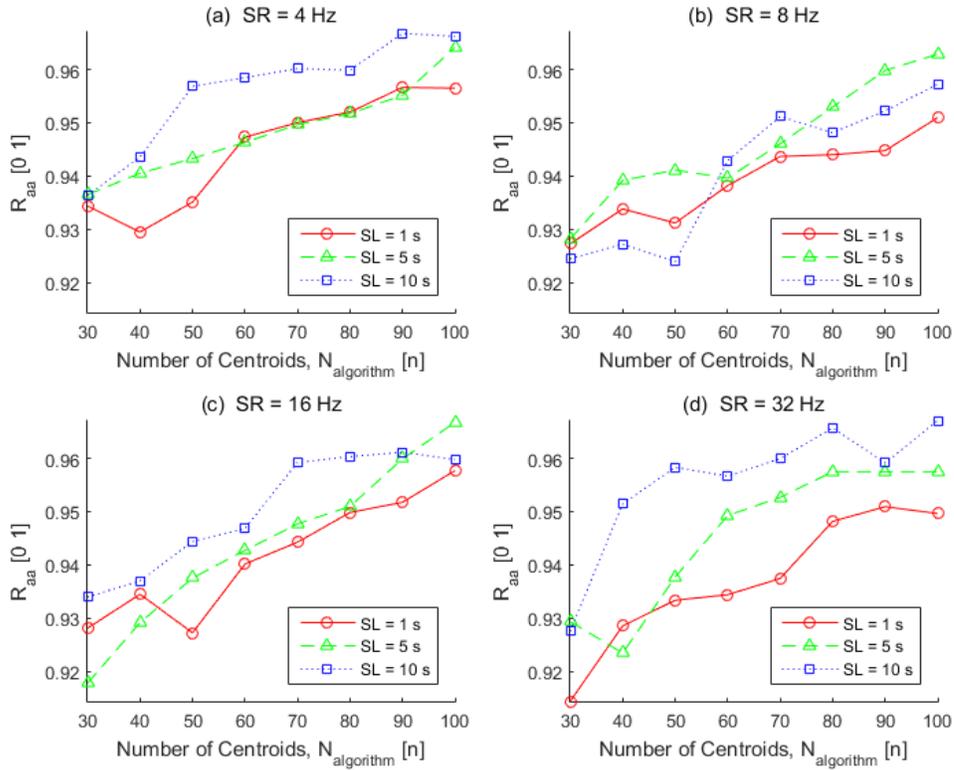


Figure 11.  $R_{aa}$  for each SL and SR

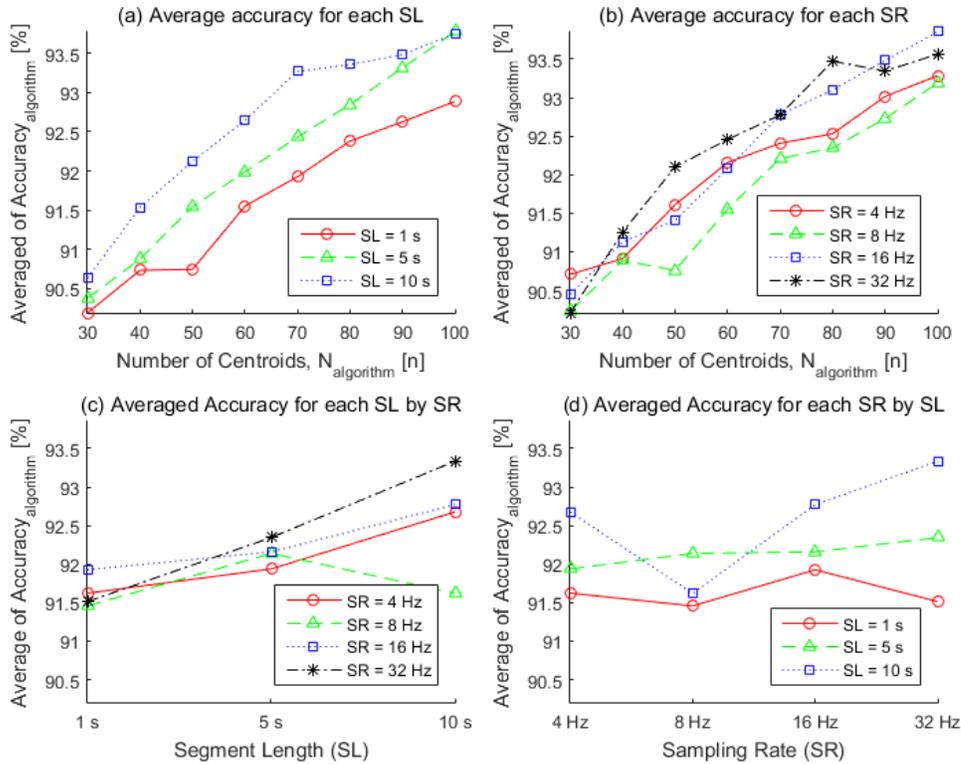


Figure 12. Average Accuracy<sub>algorithm</sub> for each SL and SR

lightweight method with reduced comparing instances (and time complexity) is essential.

In future research, we will attempt to reduce the operational time complexity of the proposed method. An improved centroid-extraction method combining k-means or k-NN with bootstrapping may be developed, and a tree structure may be employed to reduce the time complexity of the comparison operations for the k-NN algorithm.

## 6 ACKNOWLEDGMENTS

THIS research was supported by the NRF (National Research Foundation of Korea) (2019R1F1A1063128) and KEIT (Korea Evaluation Institute of Industrial Technology) (Project No. 20004372).

## 7 REFERENCES

- Altun, K., & Barshan, B. (2010, August). Human activity recognition using inertial/magnetic sensor units. In *International workshop on human behavior understanding* (pp. 38-51). Springer, Berlin, Heidelberg.
- Attal, F., Mohammed, S., Dedabrishvili, M., Chamroukhi, F., Oukhellou, L., & Amirat, Y. (2015). Physical human activity recognition using wearable sensors. *Sensors*, 15(12), 31314-31338.
- Bao, L., & Intille, S. S. (2004, April). Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing* (pp. 1-17). Springer, Berlin, Heidelberg.
- Berchtold, M., Budde, M., Gordon, D., Schmidtke, H. R., & Beigl, M. (2010, October). Actiserv: Activity recognition service for mobile phones. In *Wearable Computers (ISWC), 2010 International Symposium on* (pp. 1-8). IEEE.
- Berchtold, M., Budde, M., Schmidtke, H. R., & Beigl, M. (2010, September). An extensible modular recognition concept that makes activity recognition practical. In *Annual Conference on Artificial Intelligence* (pp. 400-409). Springer, Berlin, Heidelberg.
- Chen, Y. P., Yang, J. Y., Liou, S. N., Lee, G. Y., & Wang, J. S. (2008). Online classifier construction algorithm for human activity detection using a tri-axial accelerometer. *Applied Mathematics and Computation*, 205(2), 849-860.
- Cheng, J., Amft, O., & Lukowicz, P. (2010, May). Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition. In *International conference on pervasive computing* (pp. 319-336). Springer, Berlin, Heidelberg.
- Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), 21-27.
- Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2), 224-227.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1-38.
- Ermes, M., Parkka, J., & Cluitmans, L. (2008, August). Advancing from offline to online activity recognition with wearable sensors. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE* (pp. 4451-4454). IEEE.
- Garcia, V., Debreuve, E., & Barlaud, M. (2008). Fast k nearest neighbor search using GPU. *arXiv preprint arXiv:0804.1448*.
- Garcia, V., Debreuve, E., Nielsen, F., & Barlaud, M. (2010, September). K-nearest neighbor search: Fast GPU-based implementations and application to high-dimensional feature matching. In *Image Processing (ICIP), 2010 17th IEEE International Conference on* (pp. 3757-3760). IEEE.
- Hanai, Y., Nishimura, J., & Kuroda, T. (2009, January). Haar-like filtering for human activity recognition using 3d accelerometer. In *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, 2009. DSP/SPE 2009. IEEE 13th* (pp. 675-678). IEEE.
- He, Z. Y., & Jin, L. W. (2008, July). Activity recognition from acceleration data using AR model representation and SVM. In *Machine Learning and Cybernetics, 2008 International Conference on* (Vol. 4, pp. 2245-2250). IEEE.
- He, Z., & Jin, L. (2009, October). Activity recognition from acceleration data based on discrete cosine transform and SVM. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on* (pp. 5041-5044). IEEE.
- He, Z., Liu, Z., Jin, L., Zhen, L. X., & Huang, J. C. (2008, December). Weightlessness feature—a novel feature for single tri-axial accelerometer based activity recognition. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (pp. 1-4). IEEE.
- Jatoba, L. C., Grossmann, U., Kunze, C., Ottenbacher, J., & Stork, W. (2008, August). Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE* (pp. 5250-5253). IEEE.
- Kao, T. P., Lin, C. W., & Wang, J. S. (2009, July). Development of a portable activity detector for daily activity recognition. In *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on* (pp. 115-120). IEEE.
- Khan, A. M., Lee, Y. K., Lee, S. Y., & Kim, T. S. (2010). A triaxial accelerometer-based physical-

- activity recognition via augmented-signal features and a hierarchical recognizer. *IEEE transactions on information technology in biomedicine*, 14(5), 1166-1172.
- Lara, O. D., & Labrador, M. A. (2012, January). A mobile platform for real-time human activity recognition. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE* (pp. 667-671). IEEE.
- Lara, O. D., & Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3), 1192-1209.
- Lara, O. D., Pérez, A. J., Labrador, M. A., & Posada, J. D. (2012). Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and mobile computing*, 8(5), 717-729.
- Lee, S., Le, H. X., Ngo, H. Q., Kim, H. I., Han, M., & Lee, Y. K. (2011). Semi-Markov conditional random fields for accelerometer-based activity recognition. *Applied Intelligence*, 35(2), 226-241.
- Li, H. Y., Yeh, Y. J., & Hwang, W. J. (2011). Fast kNN classification based on softcore CPU and reconfigurable hardware. *Intelligent Automation & Soft Computing*, 17(4), 431-446.
- MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, No. 14, pp. 281-297).
- Maurer, U., Smailagic, A., Siewiorek, D. P., & Deisher, M. (2006, April). Activity recognition and monitoring using multiple sensors on different body positions. In *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on* (pp. 4-pp). IEEE.
- McGlynn, D., & Madden, M. G. (2011). An ensemble dynamic time warping classifier with application to activity recognition. In *Research and Development in Intelligent Systems XXVII* (pp. 339-352). Springer, London.
- Minnen, D., Westeyn, T., Ashbrook, D., Presti, P., & Starner, T. (2007). Recognizing soldier activities in the field. In *4th International workshop on wearable and implantable body sensor networks (BSN 2007)* (pp. 236-241). Springer, Berlin, Heidelberg.
- Pham, N., & Abdelzaher, T. (2008, June). Robust dynamic human activity recognition based on relative energy allocation. In *International Conference on Distributed Computing in Sensor Systems* (pp. 525-530). Springer, Berlin, Heidelberg.
- Ravi, N., Dandekar, N., Mysore, P., & Littman, M. L. (2005, July). Activity recognition from accelerometer data. In *Aaai* (Vol. 5, No. 2005, pp. 1541-1546).
- Riboni, D., & Bettini, C. (2011). COSAR: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3), 271-289.
- Tapia, E. M., Intille, S. S., Haskell, W., Larson, K., Wright, J., King, A., & Friedman, R. (2007, October). Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *Wearable Computers, 2007 11th IEEE International Symposium on* (pp. 37-40). IEEE.
- Zhu, C., & Sheng, W. (2010, July). Recognizing human daily activity using a single inertial sensor. In *Intelligent Control and Automation (WCICA), 2010 8th World Congress on* (pp. 282-287). IEEE.

## 8 AUTHORS



**Sun-Taag Choe** received a B.S. a integrated Ph.D in Electrical and Computer Engineering from Ajou University in Korea. He is currently an associate research engineer in Engineering Research Institute in Ajou University. His research areas include quantified self, biosignal processing, human-activity recognition, and embedded machine-learning optimization.  
Email: suntaag62@gmail.com



**We-Duke Cho** received an M.S. and a Ph.D. in Electrical and Electronic Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1983 and 1987, respectively. He is currently a professor in the Department of Electronics Engineering at the College of Information Technology and the director of the Ubiquitous Convergence Research Institute (UCRI) and the life-care science lab at Ajou University in Korea. Additionally, he is the director of the Korea Association of Smart Home (KASH) and an adjunct professor at Stony Brook University (SUNY) in New York, USA. He presented a talk entitled "A design for lifestyle analyzer based on IoT Big data with human behavior recognition". His research areas include signal processing, IoT data pattern modeling, and care system design.  
Email: wdukecho@gmail.com



**Jai-Hoon Kim** received a B.S. in Control and Instrumentation Engineering from Seoul National University, Seoul, South Korea, in 1984; an M.S. in Computer Science from Indiana University, USA, in 1993; and a Ph.D. in Computer Science, Texas A&M University, USA, in 1997. He is currently a professor in the Cyber Security department at Ajou University, South Korea. His research interests include distributed systems, cyber-physical systems, and mobile computing.  
Email: [jaikim@ajou.ac.kr](mailto:jaikim@ajou.ac.kr)



**Ki-Hyung Kim** is a professor in the Department of Cyber Security at Ajou University. He received an M.S. and a Ph.D. from KAIST in 1996 and 1992, respectively. His research interests include wireless sensor networks, Internet of Things security, network security, embedded software, and blockchain. He is currently leading a research lab called the Internet Lab in the graduate school of Ajou University.  
Email: [kim86@ajou.ac.kr](mailto:kim86@ajou.ac.kr)