New Class Candidate Generation applied to On-Body Smartphone Localization

Mitsuaki Saito and Kaori Fujinami

Abstract On-body device position-awareness plays an important role relative to providing smartphone-based services with high levels of usability and quality. Existing on-body device localization methods deal with a fixed number of positions. In contrast, we have proposed a framework to discover new positions that are not initially supported by the system and add them as recognition targets during use. In this paper, we focus on the task of detecting new class candidates in the framework, which consists of anomaly detection, dimension reduction, and clustering. Anomaly detection and dimension reduction are pre-processing to make clustering more effective. A preliminary experiment is carried out to prove the concept and find out that it is appropriate to implement the k-means clustering on the number of clusters estimated by X-means after performing anomaly detection by IForest and dimension reduction by t-distributed stochastic neighbor embedding (t-SNE).

1 Introduction

Currently, people carry their smartphones in several ways, such as carrying them in pockets or bags [1]. How a smartphone is carried is important to know its usability and the quality of sensor-dependent services, which serve to facilitate human–human communication, reduce unnecessary energy consumption, and enable automatic selection of an appropriate notification method [2, 3, 4]. Recently, on-body device state recognition has been gaining increasing attention in ubiquitous computing communities. In this task, multi-class classification techniques are used, and the

Mitsuaki Saito
Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology, 2-24-16 Naka-cho, Koganei, Tokyo 184-8588, Japan, e-mail: s193252s@st.go.tuat.ac.jp

Kaori Fujinami
Department of Computer and Information Sciences, Tokyo University of Agriculture and Technology, 2-24-16 Naka-cho, Koganei, Tokyo 184-8588, Japan, e-mail: fujinami@cc.tuat.ac.jp

number of positions is fixed during the period of use [2, 5, 6, 7, 8]. However, it is necessary to collect the data for training classifiers from all possible positions in advance, which requires significant efforts considering that people do not always use all possible positions and tend to have several preferred positions of storage instead. Generally, the recognition task becomes difficult to resolve with an increase in the number of classes. Therefore, the recognition system should first be tuned on the limited number of common classes for all people and should then be made extensible according to the usage patterns of a particular user.

Guo et al. have proposed a framework for incremental recognizable addition of daily activities as recognition targets, in which new activities are detected as the “unknown” class using a novelty detection technique. The unknown class elements are then clustered into one or more new class candidates, followed by labeling and retraining processes [9]. Saito and Fujinami have applied this framework to the on-body device localization problem and thoroughly have investigated the new position detection [10, 11]. In this study, we focus on the new class candidate generation components, which have not been considered in detail in [9]. We design the following three sub-components: anomaly sample removal, dimension reduction, and clustering. We then conduct experiments to identify the optimal combination of the sub-components and evaluate the feasibility of the clustering-based addition of new classes by simulation.

The remainder of this paper is organized as follows. Section 2 provides an overview on the research works related to position recognition and application examples of novelty detection. Section 3 describes the intended usage and flow of the proposed framework. Section 4 discusses the new class candidate generation methods. Section 5 outlines the methods, considered datasets, and results of the conducted experiments. In this study, we perform the experiments on the clustering method that specifies the number of clusters, the one that does not specify the number of clusters, and the estimation method of the appropriate number of clusters. On the basis of the experimental results, we analyze the efficient provisional cluster generation methods in each experiment. Moreover, we analyze changes in the experimental results corresponding to the number of new class candidates that are generated. Finally, Section 6 provides conclusions and specifies the future research directions.

2 Related Work

Table 1 summarizes the storage positions of mobile devices mentioned in the literature, including trouser pockets, chest pockets, shoulder bags, handbags, and hands. They all correspond to recognizing the position of a device in one of the predefined positions. Consequently, the system can inaccurately recognize the data from an unknown position as one of the known positions. However, once the system is aware that the data are obtained from an unknown position, it can take an appropriate action, such as discarding the result and asking a user to label the data accordingly with the purpose of registering the new detected position as a recognition target. Therefore,

Table 1 Examples of on-body position recognition

Literature	Positions supported in the work
Fujinami et al. [6]	neck (hanging), chest pocket, jacket pocket, trouser front/back pockets, bag (backpack, handbag, shoulder bag), hand (calling, watching the screen in the portrait direction, swinging during walking)
Fujinami [2]	neck (hanging), chest pocket, jacket pocket, trouser front/back pockets, bag (backpack, handbag, shoulder bag, messenger bag)
Alanezi and Mishra [7]	jacket pocket, trouser front/back pockets, desk, hand (calling, watching the screen in the portrait direction, swinging during walking)
Shi et al. [12]	chest pocket, trouser front/back pockets, hand
Sztyler and Stuckenschmidt [13]	head, chest, upper arm, waist, forearm, thigh, shin
Wiese et al. [14]	pocket, bag, hand, away from human
Yang and Wang [15]	jacket pocket, trouser pocket, bag, hand

distinguishing unknown classes from known classes is an important pre-processing step required to implement a reliable system.

The "unknown class problem" was also investigated in other research domains. AlSuwaidi et al. applied novelty detection to anomaly leaf one by using the images of leaves [16]. After analyzing images of normal leaves for a known class, they detected those with stress and diseases. Yin et al. studied human activity recognition, and unknown activities were detected using novelty detection [17]. In their work, the five activities—walking, running, sitting down, walking upstairs, and walking downstairs—were considered as known behaviors, and all other activities were classified as unknown behaviors. Similar to the abovementioned studies, the unknown class problem was addressed in various research fields; however, it was generally implied that the detected unknown class was often to be simply discarded, in other words, it was not used anymore. However, in several related studies, the detected unknown class was used. Guo et al. proposed a framework to address this issue [9]. Their research focused on daily activity recognition using the inertial sensor data obtained from an on-body device, in which unknown activities were detected as a single "unknown class" based on the novelty detection technique, and the unknown class was added to the recognition targets through clustering into *provisional new activities* and then were labeled by humans. They considered the six activities—walking, running, upstairs, downstairs, sitting, and standing. The framework used in this study refers to their framework, in which they mainly focused on the unknown activity detection using kernel null Foley-Sammon transformation (KNFST). By contrast, in our study, we focus on the new class candidate generation approach that implies applying anomaly sample removal, dimension reduction, and clustering after completing the novelty detection process. In addition, we evaluate the proposed framework using multiple datasets to ensure the applicability of the proposed method.

To achieve the main research goal, we apply clustering techniques to identify classes (storing positions) as new additional recognition targets. Previously, clustering was applied to various domains, such as text data, images, and sensor values. Specifically, concerning inertial sensor values, clustering was often used for unsupervised learning to reduce the burden caused by labeling training samples [18, 19]. Trabelsi et al. [18] applied clustering to unsupervised learning. They considered the four clustering methods: k-means, Gaussian mixture model (GMM), hidden Markov model (HMM), and multiple HMM regression (MHMMR) and tested them on the inertial sensor data gathering information about human activity. All of these clustering methods required specifying the number of clusters. Then, Kwon et al. [19] also applied a clustering technique to human activity recognition. In addition to the methods that required specifying the number of clusters, such as k-means, GMM, and hierarchical agglomerative clustering (HIER), they used density-based spatial clustering of applications with noise (DBSCAN) that did not require the number of clusters.

In this research, we denote the results of clustering as new class candidates. After being labeled by human, all samples in a particular cluster have the same label and are used to train or retrain the recognition model. In an extreme case, all samples belong to different clusters, meaning that each cluster has only one element. In this case, the human annotator has to label all the data even if only one new class exists in the target data, which is apparently difficult. Therefore, an appropriate number of new classes should be used in the clustering process. In this study, the k-mean was tested assuming the ideal situation wherein the number of classes is known; however, as the number of classes is actually unknown, we also verified the applicability of DBSCAN and X-means for which it was not necessary to specify the number of clusters.

3 Overview of the Incremental Recognizable Position Addition Framework

Fig. 1 illustrates the framework developed to add smartphone carrying positions as recognition targets incrementally. The framework is composed of the seven major components (marked 'A' to 'H' in the figure). We assume that the labeled data are collected considering the limited number of popular positions and that the recognition component is trained before actually using the device. The initial training data are supposed to be organized by those who do not correspond to the device users.

For each feature vector considered for position recognition, a process is performed to determine whether it can be classified into one of n-known positions through the *novelty detection* technique (Fig. 1 A). If the feature vector is evaluated as corresponding to one of the known classes, it is forwarded to the classifier component (Fig. 1 B). Otherwise, the vector is stored in the novelty sample pool (Fig. 1 C).

If a certain condition to trigger the next process is satisfied, the stored samples can be used to generate k new classes of candidates to be added as new positions. The number of new positions in the novelty sample pool is not always equal to one. Therefore, this step is performed to separate multiple new positions. The new class candidate generation consists of the three sub-components: anomaly sample removal, dimension reduction, and clustering. There are the two types of novelty samples: the real one that appears when a user holds it at a new position, and the missing sample that appears by accident such as when the user fell and thus does not belong to known and new position. When performing the step of clustering, it is checked whether the identified samples include any missing samples and whether the result of clustering may be adversely affected. Therefore, any missing samples in the stored samples are rejected using the anomaly detection technique (Fig. 1 D). Applying clustering in higher dimensions may produce incorrect outcomes. Therefore, the dimension needs to be reduced before clustering (Fig. 1 E). Hence, the anomaly detection and dimension reduction steps are considered as clustering pre-processing. Clustering is performed on the remaining samples after completing anomaly detection and using the features converted by dimension reduction (Fig. 1 F).

The clusters need to be labeled for retraining, in which several clusters may be merged into a single class, and, finally, m ($\leq k$) new classes are identified (Fig. 1 G). This process requires knowledge about particular samples; therefore, the user involvement is required. However, during this task, the user's mental workload should be minimized. We assume that the most suitable timing for a request to complete labeling is estimated in advance and that a recallable query message is generated using the user's contextual information, as proposed in [20]. Finally, the position recognizer is retrained on the data corresponding to the discovered m classes and to the existing n known classes, reaching an ability to recognize $n + m$ classes.

The framework proposed by Guo et al. can be also used to generate new class candidates via dimension reduction and clustering; however, it does not imply performing the anomaly detection. Moreover, the importance of dimension reduction has not been examined. Therefore, an experiment aimed to formulate the design principles of the new class candidate generation component (Fig. 1 D, E, and F) is performed, as described in the following section 4.

In this study, we consider that the labeled class (Fig. 1 G) is added to the training dataset corresponding to storing position recognition as a known class and becomes one of the recognition targets after retraining the classifier (Fig. 1 B). Generally, the higher classification accuracy can be achieved when the user's own data are used to train the recognition model for a particular user [21]. We also leverage the retraining process to personalize the classifier even on the existing known classes. We assume that the initial classifier is trained on the data obtained for other persons corresponding to n position, as presented in Fig. 1. Let us assume that the user stores the smartphone at a known position denoted as Position-1. At an initial stage, even this known position is subject to labeling, and the existing labeled data are discarded once being re-labeled by the user. Thus, the training data corresponding to an existing class are replaced with the user's own data, which is expected to improve the recognition accuracy of the class. After a sufficient period of time, the whole training dataset is replaced with the user's own data, which we consider as a steady state. In this study, we evaluate the accuracy of classification in the steady state, implying the training dataset corresponds to the user.

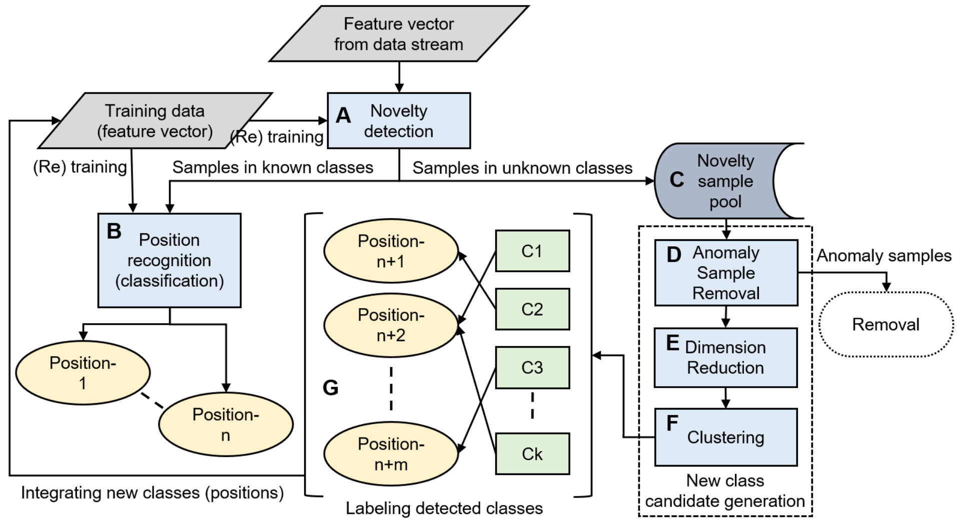

Fig. 1 Framework for incremental recognizable addition of smartphone carrying positions

4 Applicability of New Class Candidate Generation

The new class candidate generation includes the following three functions: anomaly detection, dimension reduction, and clustering. Concerning each function, the details about the methods used in the conducted experiments are provided below.

4.1 Anomaly Sample Removal

Local outlier factor (LOF) and isolation forest (IForest) are tested with regard to anomaly detection [22, 23]. LOF uses the density of samples to detect missing samples. As shown on the left of Fig. 2, samples in dense areas are more likely to be judged known, while samples in less dense areas are more likely to be judged unknown. IForest is an ensemble-based method to detect missing samples during the process of dividing the feature space selected randomly. As shown on the right of Fig. 2, the area will be divided, and samples in small divided areas with a small number of samples are more likely to be judged as unknown. Thus, isolated samples with no other surrounding samples are judged as anomaly and are removed.

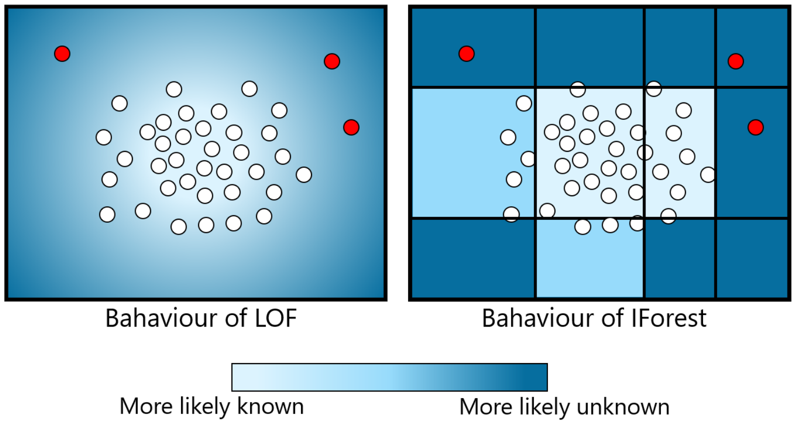

Fig. 2 Bahaviour of anomaly detection: The left and right parts are LOF and IForest, respectively.

4.2 Dimension reduction

Principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE) are used to evaluate dimension reduction. PCA is a method used to reduce dimensions by calculating a regression equation that converts a given sample into an arbitrary number of smaller dimensions. In its turn, t-SNE is a method to transform all samples into two or three dimensions, so that similar samples can be gathered and dissimilar samples can be separated.

Since t-SNE does not perform regression, new samples cannot be mapped to the model after dimension reduction. However, the features reduced by the dimension reduction are used only for clustering. Clustered unknown samples are used as training samples for the position recognition component, in which the original non-reduced features are used. Therefore, there is no problem if we use t-SNE not conceived to use new samples.

4.3 Clustering

The clustering methods can be divided into the ones requiring to specify the number of clusters and the others with no need for such information. As a method corresponding to the former, k-means clustering is considered in this study. The k-means method is used to calculate the distance to each sample from the representative sample of each cluster and to update the cluster accordingly.

DBSCAN is considered as a method that does not require specifying the number of clusters. It is used to perform clustering by grouping the nearby samples into the same cluster. The samples that do not belong to any cluster are denoted as outliers.

This means that DBSCAN can be used to detect anomalies simultaneously with clustering. In addition, X-means is considered as a method to estimate the number of clusters. This method applies k-means given the minimum and maximum number of clusters as parameters. Then, it outputs the number of clusters considered to be the best one for clustering.

5 Experiment

5.1 Overview

Comparative experiments were conducted to evaluate the applicability of the three main components: the anomaly detection, dimension reduction, and clustering methods. Python 3.7.4 was used as the programming language, and `scikit-learn` and `pyclustering` libraries were used. The main parameters of each method are summarized in Table 2, which are the default values except for k-means.

Table 2 Main parameters used in the Python libraries

Method	Main parameters
LOF	The number of neighbors=20, contamination='auto'
IForest	The number of estimators=100, contamination='auto'
PCA	The number of dimensions=2
t-SNE	The number of dimensions=2, perplexity=30.0
k-means	The number of clusters=-1 to +6 from the number of positions(Change according to the experiment)
DBSCAN	eps=0.5, the number of minimum samples=5
X-means	The number of minimum clusters=2, The number of maximum clusters=20

Experiments were conducted for each of the three clustering methods. Preprocessing was applied before clustering. In anomaly detection, in addition to the two considered methods, there was a case in which no anomaly detection was performed, so that there were three experimental conditions investigated with regard to anomaly detection. Similarly, in dimension reduction, in addition to the two considered methods, there was a case in which no dimension reduction was performed, so that there were three experimental conditions corresponding to dimension reduction. In this study, the dimension reduction was set to two, which was the minimum dimension that could be reduced. Then, the pre-processing step was executed considering the nine conditions: combination consisting of three conditions corresponding to anomaly detection and three conditions of dimension reduction. However, considering that DBSCAN could perform anomaly detection simultaneously with clustering, the anomaly detection methods were not applied before DBSCAN.

5.2 Datasets

The three datasets including three-axis acceleration signals corresponding to a wide variety of possible carrying positions were used in the experiments (Table 3). All data were provided by the volunteer subjects during walking. Feature vectors were calculated based on the raw data signal, which consisted of 30, 63, and 16 types of features originated from both time and frequency domains for datasets A, B, and C, respectively.

In dataset A, for six positions, namely, jacket pocket, trouser front/back pockets, and a hand (calling, watching the screen in the portrait direction, and swinging during walking), the data were collected from both left and right sides. Similarly, in dataset C, the data were collected from both left and right sides of trousers pocket. Considering that the left and right sides could be regarded as same positions, we obtained the number of positions equal to 11, 9, and 4 in dataset A, B, and C, respectively. Therefore, for each of the three datasets A, B, and C, the number of possible combinations of unknown positions was 2047 (= $\sum_{k=1}^{11} {}_{1}1C_k$), 511 (= $\sum_{k=1}^{9} {}_{9}C_k$), and 15 (= $\sum_{k=1}^{4} {}_{4}C_k$), respectively.

Concerning the actual usage, the clustered candidate classes were generated based on the data obtained from the users of the mobile devices. Therefore, this experiment was performed on a subject-by-subject basis. However, it was not practical for all subjects to consider all combinations of unknown positions. Therefore, in each combination of unknown positions, the data corresponding to the one randomly selected subject were used.

Table 3 Datasets used in the study

Dataset	Person	Position
A [6]	70	neck (hanging), chest pocket, jacket pocket, trouser front/back pockets, bag (backpack, handbag, shoulder bag), hand (calling, watching the screen in the portrait direction, swinging during walking)
B [2]	20	neck (hanging), chest pocket, jacket pocket, trouser front/back pockets, bag (backpack, handbag, shoulder bag, messenger bag)
C [24]	10	trousers left/right pockets, arm, wrist, belt

5.3 K-Means as a clustering method with the specified arbitrary number of clusters

The purpose of this evaluation was to verify the change in the result corresponding to changes in the pre-processing condition or the number of clusters in k-means that required specifying the number of clusters.

5.3.1 Method

Clustering was often evaluated in terms of its own accuracy such as normalized mutual information (NMI) [18, 19]. However, in the case of the proposed framework, even if the accuracy of clustering itself is good, it is meaningless if the test sample cannot be correctly recognized during the subsequent position recognition. Hence, it was more important to verify that the new classes could correctly recognize the test sample during subsequent position recognition and not to focus on the accuracy of clustering itself. Therefore, the accuracy of position recognition after the moment when clustering was performed and the labeled class was changed to a known class, was used as an evaluation metric of new class candidate generation in this experiment. The detailed evaluation method will be described later in this section.

After completing the pre-processing step corresponding to the nine conditions, clustering was performed by using k-means clustering with the number of clusters specified as the actual number of positions. However, we consider that the left and right positions, such as jacket pocket, should be treated as different positions because the features of the sample tend to be different between left and right. Therefore, in datasets A and C, if the samples included the left and right positions, k varied from +1 to +6 with respect to the actual number of positions. This number was considered as an ideal number. In other words, the total number of positions when the left and right positions are considered separately is the ideal number. For example, if the unknown positions were the three following positions: jacket pocket, backpack, and handbag in dataset A, clustering was performed using the ideal number $k = 4$ and considering that the jacket pocket represents two positions. Moreover, clustering was performed by specifying the number of clusters as 1, +1, and +2 with respect to the ideal number.

For the practical viewpoint, labeling was performed by users; however, in this study, labeling was conducted by a simulated user, meaning that each generated cluster was marked with the label assigned to the largest number of the constituent samples. Fig. 3 presents an example in which the three samples with the true labels being circle, triangle, and square are clustered at $k = 4$. At this time, as shown on the left side of the figure, it is assumed that each of the four generated clusters contains a mixture of the three types of labels. Therefore, as shown on the right side of the figure, each cluster is labeled by the most common label.

Using the data corresponding to the labeled classes as the training dataset for position recognition, we calculated the accuracy according to Eq. (1), given that the number of correctly classified test samples is $N_{correct}$ and the number of all test samples is N_{test}.

$$Accuracy = \frac{N_{correct}}{N_{test}} \tag{1}$$

To evaluate the accuracy of position recognition, it is necessary to leave the position recognition test sample. That is, part of the sample of positions used for new class candidate generation are used for evaluation. Therefore, 10-fold cross validation is used. That is, the samples of the dataset are divided into 10 groups, and the samples

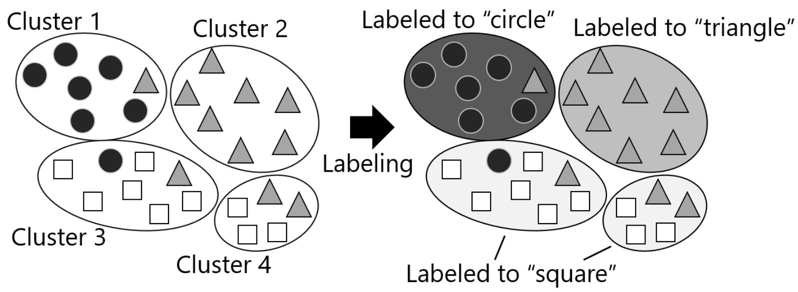

Fig. 3 Labeling: The left and right parts are the results of clustering and simulated labeling, respectively.

in 9 groups are used in new class candidate generation, and the generated clusters are labeled accordingly. After that, the classification is performed using the remaining one group of samples as a test sample for position recognition. This process is iterated 10 times while varying the test group, and the average accuracy across 10 iterations is calculated. In this study, we use RandomForest as one of the most common classification methods. Here, the number of trees denoting the number of estimators was set to 10.

In practice, test samples are first judged as unknown or known by using novelty detection (Fig. 1 A), and only those that are judged as known are then classified. As mentioned in Section 3, once the smartphone user performs labeling, the training and test datasets will be used as the user data thereafter. Moreover, novelty detection can be used to ensure the high accuracy when the training and test datasets correspond to the same person [10]. Therefore, in this study, we evaluate the classification results under the assumption that the novelty detection performs perfectly. The position recognition accuracy of the nine pre-processing conditions is compared between each other. In addition, the position recognition accuracy in the case when each cluster contains the samples from a single class, is calculated as an ideal condition. For each dataset, the average of the accuracy values across all combinations of unknown positions is calculated.

5.3.2 Results and Analysis

The position recognition accuracy values for each dataset are summarized for the case when the number of clusters is specified as the ideal number, as provided in Table 4. Under ideal conditions, the accuracy increases to nearly 1.0. This is reasonable because the classification component is trained on the user's own data. For all datasets, the accuracy increases to near-ideal conditions when t-SNE is applied with the purpose of dimension reduction. Conversely, if PCA is considered for dimension reduction, the accuracy is lower than that corresponding to the other methods. In addition, there is no large difference in the accuracy among the three conditions for anomaly detection compared with the changes associated with conditions of dimension reduction. Therefore, if the ideal number of clusters (k) is specified, k-means clustering with t-SNE can be efficiently applied for the purpose of new class candidate generation.

Table 4 Position recognition accuracy in the case of combining anomaly detection and dimension reduction, as well as considering the ideal condition. It should be noted that the bold face represents the maximum in each dataset except for the ideal condition.

	Non Non	Non PCA	Non t-SNE	LOF Non	LOF PCA	LOF t-SNE	IForest Non	IForest PCA	IForest t-SNE	ideal condition
Dataset A	0.962	0.887	**0.987**	0.974	0.896	**0.987**	0.974	0.898	0.969	0.997
Dataset B	0.856	0.820	0.969	0.862	0.826	**0.970**	0.878	0.836	0.963	0.999
Dataset C	0.965	0.963	**0.997**	0.972	0.963	**0.997**	0.978	0.965	**0.997**	0.999

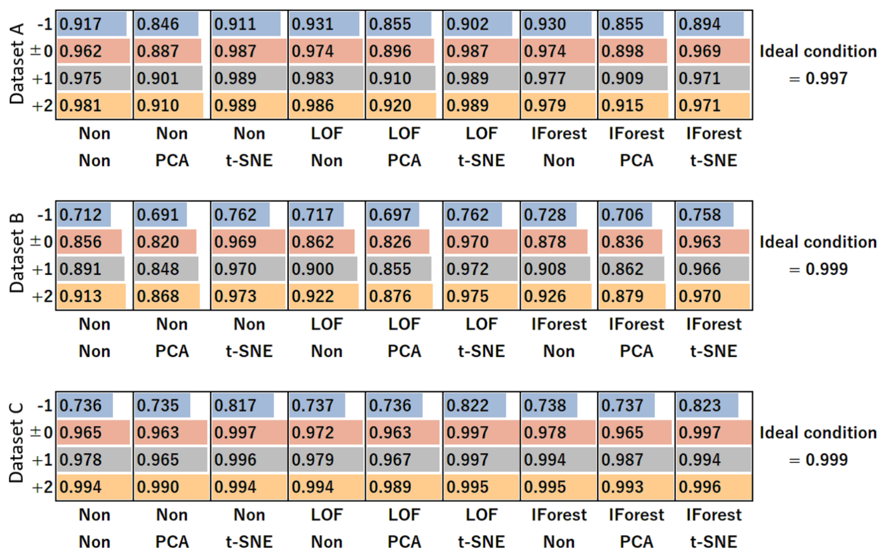

Fig. 4 The position recognition accuracy result for the case when the number of clusters is specified from 1 to +2 with respect to the ideal number

Figs. 4 present the variations of position recognition accuracy values obtained by changing the number of clusters k from 1 to +2 relatively to the ideal number. When the number of clusters is 1 from the ideal number, the accuracy deteriorates considerably. This is due to the fact that if the number of clusters is less than the ideal number, unlabeled positions will always appear. In other words, as the training dataset for particular position does not exist in the known class, the test datasets of these positions cannot be considered in the position recognition.

As the number of clusters increase, the accuracy increases gradually. This is because with an increase in the number of clusters, each cluster is less likely to contain samples of various positions. As the ideal condition corresponds to the accuracy estimated in the case when the number of clusters is equivalent to the number of samples, the accuracy converges to the ideal condition, when the number of clusters is increased. However, with an increase in the number of clusters, it becomes more difficult for the user to label them in line with the actual usage. Therefore, the number of clusters should be determined considering the burden of labeling on the user.

5.4 DBSCAN as a clustering method without the need for specifying arbitrary number of clusters

The purpose of this evaluation was to verify the applicability of DBSCAN, a method without the need for specifying the number of clusters, in a practical situation.

5.4.1 Method

DBSCAN could be applied to detect anomalies simultaneously with clustering. We focused on the rate of samples judged to be anomaly ($R_{outlier}$). The number of samples determined as outliers was denoted as $N_{outlier}$. Then, $R_{outlier}$ was calculated according to Eq. (2).

$$R_{outlier} = \frac{N_{outlier}}{N_{test}} \tag{2}$$

Likewise, $R_{outlier}$ in LOF and IForest were calculated. Therefore, we compared $R_{outlier}$ calculated by DBSCAN, LOF, and IForest. For each dataset, the average of $R_{outlier}$ for all combinations of unknown positions was calculated. The three pre-processing conditions, no dimension reduction, PCA, and t-SNE, under which anomaly detection was not performed, were considered before applying DBSCAN.

5.4.2 Results and Analysis

Table 5 presents the result corresponding to $R_{outlier}$. For datasets A and C, almost all samples were determined to be abnormal by DBSACN in the cases without dimension reduction (Non+DBSCAN) and with t-SNE (t-SNE+DBSCAN). Moreover, for dataset B, almost all samples were determined to be abnormal according to all three conditions of DBSACN. Therefore, we conclude that DBSCAN is not suitable for the proposed framework.

As mentioned in Section 5.3.2, anomaly detection has only minor effect on the accuracy, which implies that $R_{outlier}$ of LOF and IForest is a reasonable value.

However, if the number of training samples needs to be small, we consider that IForest is more applicable as more samples are removed compared with LOF.

Table 5 The rate of samples determined to be abnormal ($R_{outlier}$)

	Non+DBSCAN	PCA+DBSCAN	t-SNE+DBSCAN	LOF	IForest
Dataset A	1.000	0.354	0.969	0.058	0.130
Dataset B	1.000	0.999	0.998	0.030	0.104
Dataset C	1.000	0.270	0.999	0.021	0.146

5.5 X-means: the clustering method that estimates the number of clusters internally

The purpose of this evaluation was to verify the applicability of X-means as a method to estimate an appropriate number of clusters. The difference between the estimated number of clusters and the ideal number, was calculated and discussed.

5.5.1 Method

The number of clusters estimated by X-means was evaluated. The three processes, including anomaly detection, dimension reduction, and estimation of the number of clusters by X-means, were executed.

The difference between the estimated number of clusters and the ideal number of clusters (D_K) was presented by Eq. (3). Here, the estimated and ideal number of clusters were denoted as $K_{estimated}$ and K_{ideal}, respectively. For each dataset, the average of D_K for all combinations of unknown positions was calculated. D_ks obtained according to the nine conditions of pre-processing were compared.

$$D_K = K_{estimated} - K_{ideal} \tag{3}$$

In addition, the rates of the number of times, when D_K was negative, ($R_{negative}$) was defined according to Eq. (4), in which the number of times D_K was negative, and the number of possible combinations of unknown positions was represented as $C_{negative}$ and C_{test}, respectively.

$$R_{negative} = \frac{C_{negative}}{C_{test}} \tag{4}$$

Table 6 The difference between the calculated number of clusters and the ideal number (D_K).

	Non Non	Non PCA	Non t-SNE	LOF Non	LOF PCA	LOF t-SNE	IForest Non	IForest PCA	IForest t-SNE
Dataset A	+11.1	+3.9	+5.6	+10.4	+3.2	+5.6	+10.2	**+2.1**	+4.2
Dataset B	+15.3	+4.3	+6.5	+15.2	+3.9	+6.5	+15.2	**+3.4**	+5.9
Dataset C	+12.5	+3.7	**+3.3**	+12.2	+3.8	+4.4	+11.0	**+3.3**	+3.7

Table 7 The rate between the number of times when the calculated number of clusters is less than the ideal number ($R_{negative}$).

	Non Non	Non PCA	Non t-SNE	LOF Non	LOF PCA	LOF t-SNE	IForest Non	IForest PCA	IForest t-SNE
Dataset A	0.001	0.129	0.024	0.000	0.124	0.024	0.000	0.192	0.036
Dataset B	0.000	0.105	0.017	0.000	0.095	0.014	0.000	0.114	0.016
Dataset C	0.000	0.000	0.020	0.000	0.000	0.000	0.000	0.000	0.000

5.5.2 Results and Analysis

Table 6 presents the result of D_K. D_K is smaller when PCA is applied as a dimension reduction for all datasets. By contrast, if dimension reduction is not applied, D_K is larger than the others.

As shown in graphs presented in Fig. 4, increasing the number of clusters does not significantly affect the accuracy. In addition, considering the burden of labeling, it is preferable that D_K is small. Moreover, there is no considerable difference in D_K among the three conditions of anomaly detection. For example, in dataset B, the difference between the maximum and minimum values among Non+t-SNE, LOF+t-SNE, and IForest+t-SNE is 0.6, which is much smaller than the difference between those among LOF+Non, LOF+PCA, and LOF+t-SNE, namely, 11.3. Therefore, dimension reduction has to be applied.

Moreover, as shown in Table 7 that presents the result of $R_{negative}$, D_K is more likely to be negative if PCA is applied. As mentioned in Section 5.3.2, the accuracy may be greatly deteriorated when the number of clusters is negative with respect to the ideal number, which needs to be avoided at the highest priority. Therefore, performing PCA as dimension reduction is also deemed inappropriate. On the basis of the above discussion, we can conclude that applying t-SNE as dimension reduction in X-means is efficient. Furthermore, there is no large difference in $R_{negative}$ among the three conditions for anomaly detection. For example, in dataset B, the difference between the maximum and minimum values among Non+t-SNE, LOF+t-SNE, and IForest+t-SNE is 0.003, which is much smaller than the difference between those among LOF+Non, LOF+PCA, and LOF+t-SNE, which is 0.095.

6 Conclusions and Future Work

In this study, we focused on the task of detecting new class candidates in the framework for an extensible on-body smartphone localization system. The considered framework was intended to discover new device storing positions that were not initially supported by the system and accordingly, to retrain the classifier using the data on new positions labeled by users. The representative sample was labeled to reduce the burden on the user, rather than labeling each sample one by one, in which clustering was important.

The three datasets were considered in the task of new class candidate cluster generation using anomaly detection, dimension reduction, and clustering. We found that applying t-SNE as dimension reduction before clustering through the k-means method was efficient if the ideal number of clusters could be specified. Using anomaly detection did not significantly affect the results; however, if the number of samples to be kept as training ones had to be reduced, IForest could be applied efficiently so as to remove more samples.

DBSCAN was consideded as a method that did not require specifying the number of clusters, but was found to be unsuitable. The X-means method was used as an approach to estimate the appropriate number of clusters. Thus, without dimension reduction, the obtained number of clusters was much larger than the ideal number. We note that, generally, when PCA is applied with purpose of dimension reduction, the number of clusters often becomes smaller than the ideal number. Therefore, it was found that applying t-SNE as dimension reduction before executing X-means was an efficient scheme. Moreover, anomaly detection did not significantly affect the results. We therefore conclude the processing pipeline for new class candidate generation as follows:

- Use k-means clustering on the number of clusters estimated by X-means after performing dimension reduction by t-SNE, without anomaly sample removal in advance.

As shown in Section 5.5.2, the number of clusters determined in X-means tends to be larger than the ideal number even when using t-SNE. Moreover, as shown in Section 5.3.2, if the number of clusters is smaller than the ideal one, the result deteriorates significantly, and this outcome needs to be avoided. On the contrary, with an increase in the number of clusters, the user's burden caused by labeling also augments. Therefore, we will seek to identify a precise estimation method aiming to obtain the ideal number of clusters. Furthermore, the effects of parameter variation will be investigated to find more effective processing pipeline in the future because the main parameters of each method are the default values except for k-means (Table 2).

Acknowledgment

This work is funded by the Japan Society for the Promotion of Science (JSPS) (Grant No. 18H03228).

References

1. Y. Cui, J. Chipchase, and F. Ichikawa, "A Cross Culture Study on Phone Carrying and Physical Personalization," *Proc. of the 12th International Conference on Human-Computer Interaction (HCI International 2007)*, pp. 483–492, 2007.
2. K. Fujinami, "On-Body Smartphone Localization with an Accelerometer," *Information*, vol. 7, no. 2, p. 21, 2016.
3. K. Fujinami, "Smartphone-based Environmental Sensing Using Device Location as Metadata," *International Journal on Smart Sensing and Intelligent Systems*, vol. 9, no. 4, pp. 2257–2275, 2016.
4. A. Exler, C. Dinse, Z. Günes, N. Hammoud, S. Mattes, and M. Beigl, "Investigating the Perceptibility Different Notification Types on Smartphones Depending on the Smartphone Position," *Proc. of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'17)*, pp. 970–976, 2017.
5. A. Mannini, A. M. Sabatini, and S. S. Intille, "Accelerometry-based recognition of the placement sites of a wearable sensor," *Pervasive and Mobile Computing*, vol. 21, pp. 62–74, 2015.
6. K. Fujinami, T. Saeki, Y. Li, T. Ishikawa, T. Jimbo, D. Nagase, and K. Sato, "Fine-grained Accelerometer-based Smartphone Carrying States Recognition during Walking," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 8, 2017.
7. K. Alanezi and S. Mishra, "Design, implementation and evaluation of a smartphone position discovery service for accurate context sensing," *Computers & Electrical Engineering*, vol. 44, pp. 307–323, 2015.
8. O. D. Incel, "Analysis of Movement, Orientation and Rotation-Based Sensing for Phone Placement Recognition," *Sensors*, vol. 15, no. 10, pp. 25474–25506, 2015.
9. J. Guo, X. Zhou, Y. Sun, G. Ping, G. Zhao, and Z. Li, "Smartphone-Based Patients' Activity Recognition by Using a Self-Learning Scheme for Medical Monitoring," *Journal of Medical Systems*, vol. 40, no. 6, p. 140, 2016.
10. M. Saito and K. Fujinami, "Evaluation of Novelty Detection Methods in On-Body Smartphone Localization Problem," *Proc. of the 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE 2019)*, pp. 470–474, 2019.
11. M. Saito and K. Fujinami, "Unknown On-Body Device Position Detection Based on Ensemble Novelty Detection," *Sensors and Materials*, vol. 32, no. 1, pp. 27–40, 2020.
12. Y. Shi, Y. Shi, and J. Liu, "A Rotation Based Method for Detecting On-Body Positions of Mobile Devices," *Proc. of the 13th international conference on Ubiquitous computing (UbiComp'11)*, pp. 559–560, 2011.
13. T. Sztyler and S. Heiner, "On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition," *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2016.
14. J. Wiese, T. Saponas, and A.J. Brush, "Phoneprioception: Enabling Mobile Phones to Infer Where They Are Kept," *Proc. of CHI*, pp. 2157–2166, 2013.
15. R. Yang and B. Wang, "PACP: A Position-Independent Activity Recognition Method Using Smartphone Sensors," *Information*, vol. 7, no. 4, 2016.
16. A. AlSuwaidi, B. Grieve, and H. Yin, "Feature-Ensemble-Based Novelty Detection for Analyzing Plant Hyperspectral Datasets," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 4, pp. 1041–1055, 2018.

17. J. Yin, Q. Yang, and J. J. Pan, "Sensor-Based Abnormal Human-Activity Detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1082–1090, 2008.
18. D. Trabelsi, S. Mohammed, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "An Unsupervised Approach for Automatic Activity Recognition Based on Hidden Markov Model Regression," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 829–835, 2013.
19. Y. Kwon, K. Kang, and C. Bae, "Unsupervised Learning for Human Activity Recognition Using Smartphone Sensors," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6067–6074, 2014.
20. K. Fujinami, T.T. Vu, K. Sato, "A Framework for Human-Centric Personalization of Context Recognition Models on Mobile Devices,", *Proc. of 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 885-888, 2019.
21. J.W. Lockhart and G.M. Weiss, "The Benefits of Personalized Smartphone-Based Activity Recognition Models," *Proc. of the 2014 SIAM International Conference on Data Mining*, pp.614–622, 2014.
22. D. Pokrajac, A. Lazarevic, and L. Latecki, "Incremental Local Outlier Detection for Data Streams", *Proc. of the 2007 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2007*, pp. 504-515, 2007.
23. Z. Ding and M. Fei, "An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window", *IFAC Proceedings Volumes*, vol. 46, no. 20, pp. 12-17, 2013.
24. M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Fusion of Smartphone Motion Sensors for Physical Activity Recognition," *Sensors*, vol. 14, no. 6, pp. 10146–10176, 2014.