Representation Learning for Minority and Subtle Activities in a Smart Home Environment

Andrea Rosales Sanabria, Thomas W. Kelsey, and Juan Ye
School of Computer Science, University of St Andrews
St Andrews, UK
{ar296, juan.ye}@st-andrews.ac.uk

***Abstract*—Daily human activity recognition using sensor data can be a fundamental task for many real-world applications, such as home monitoring and assisted living. One of the challenges in human activity recognition is to distinguish activities that have infrequent occurrence and less distinctive patterns. We propose a dissimilarity representation-based hierarchical classifier to perform two-phase learning. In the first phase, the classifier learns general features to recognise majority classes, and the second phase is to collect minority and subtle classes to identify fine difference between them. We compare our approach with a collection of state-of-the-art classification techniques on a real-world third-party dataset that is collected in a two-user home setting. Our results demonstrate that our hierarchical classifier approach outperforms the existing techniques in distinguishing users in performing the same type of activities. The key novelty of our approach is the exploration of dissimilarity representations and hierarchical classifiers, which allows us to highlight the difference between activities with subtle difference, and thus allows the identification of well-discriminating features.**

***Index Terms*—Smart home, activity recognition, dissimilarity representation, representation learning**

I. Introduction

Sensor-based human activity recognition is to extract high-level descriptions (i.e., activities) from low-level sensor data [20]. One of the key challenges is to recognise activities that have infrequent occurrence and less distinctive patterns, which can have a significant implication in health-related applications. For example, life-threatening situations like fall or heart attack are often not frequent and may have subtle difference from other daily activities. Being able to recognise them effectively will enhance the robustness of an activity recognition system.

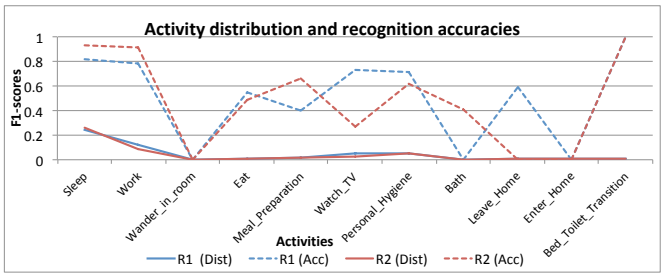

Fig. 1. Activity distribution and recognition accuracies on a two-user co-living environment.

To illustrate the challenge of recognising minority and subtle activities, we use the following example. Figure 1 presents the distribution of a set of concurrent activities from two users recorded in a smart home setting [4] and activity recognition accuracies (measured in F1-scores) from a Support Vector Machine (SVM) with a RBF kernel.

As we can see, SVM can fairly well recognise the majority activities like 'Sleep' and 'Work' and the activities with distinct patterns like 'Bed_Toilet_Transition'. It performs poorly when (1) distinguishing different activities from the same user occurring in the same area; for example, whether a user leaves or enters the house, and (2) differentiating the users for the same type of activities performed in a public area; for example, whether the one who leaves the house is the user R1 or R2. Some activities do not occur often, especially the leaving and entering home activities only occur 1% on *R1* and 0.05% on *R2*. Hence there are too few samples to train a reliable classifier, and also learning their discriminative features can be challenged by the majority classes. In addition, these activities can have less discriminative patterns from their majority counterpart; that is, they might activate the same set of sensors but with little difference in distributions.

In this paper, we hypothesise that learning good feature representations can help recognise minority and subtle activities. In particular, we address two research questions: *what constitute good feature representations?*, and *how can they be learnt?*. We explore the recent representation learning techniques and focus on Dissimilarity Representation (DR) that has achieved promising results in structural pattern recognition in computer vision [5]. We propose a Dissimilarity Representation based Hierarchical Classifier (DRHC), with the aim to learn discriminative features in order to better distinguish minority activities with less distinctive patterns. We have evaluated our technique on a third-party dataset and have demonstrated its effectiveness by comparison with (i) state-of-the-art classification techniques, (ii) resampling techniques that target at imbalanced datasets, and (iii) other representation learning techniques.

The rest of the paper is organised as follows. Section II introduces the existing literature in activity recognition. Section III introduces dissimilarity representation and proposes DRHC. Section IV describes the evaluation methodology and Section V presents the evaluation results and discusses the performance of DRHC over the other state-of-the-art classifiers. The paper concludes in Section VI.

II. Related Work

Human activity recognition has been an active field for more than a decade. It aims to develop methods to understand human behaviour from a series of observations derived from motion, location, physiological signals and environmental information. A general process in human activity recognition is to collect and integrate data from various sensors, extract features, and apply a learning technique to infer activities from the features.

Various data- and knowledge-driven techniques have been applied to human activity recognition, including ontological reasoning, Naive Bayes, Decision Trees, Hidden Markov Models (HMM), Conditional Random Fields (CRF), Neural Networks, and Support Vector Machines (SVM) [3], [22]. These techniques have demonstrated promising results in learning complex correlations between human activities and sensor features. However, few has focused on learning good representations of sensor data so as to further distinguish activities that have subtle difference.

A. Representation Learning

Representation learning has become a crucial task in machine learning. It can be either linear or nonlinear, either supervised (i.e., features are learned using labelled input data), or unsupervised (i.e., features are learned with unlabelled input data). Traditional feature learning aims to learn transformations of the data that make it easier to extract useful information when building a classifier [2]. Within this group, the most popular feature learning is Principal Component Analysis (PCA). This linear unsupervised algorithm transforms feature variables into a smaller number of uncorrelated variables called principal components. Another well-known linear supervised algorithm is linear discriminant analysis (LDA), which finds a linear combination of features that separates two or more classes of objects. It has been successfully used in face recognition [2]. Unlike these approaches, manifold learning is a nonlinear method that learns the high-dimensional structure of the data from the data itself, without the use of predetermined classifications [2].

Although, little research has addressed the problem of representation learning for human activity recognition. Plotz et al. highlight the idea of feature learning, which focuses on two learning techniques: Principal Component Analysis (PCA) and Autoencoder [18]. In the context of activity recognition, PCA can perform poorly because it can miss important nonlinear structures of the data. To tackle this problem, they propose an alternative raw data representation based on the empirical cumulative distribution function of the sample data. Furthermore, Mannini et al. propose the Pudil algorithm based on a sequential forward-backward floating search [13], which is a feature selection method to detect and discard the features that are demonstrated to make minimal contribution to a correct response from the classifier. Nguyen et al. have applied Bayesian nonparametric to learn representations of user context and community profiles [16].

B. Imbalanced Class Distribution

Activity data collected in the real-world environments, as presented in Figure 1, can often have imbalanced distributions. This problem has yet attracted sufficient attention. Feuz et al. [6] propose intra-class clustering (ICC) technique to learn from imbalanced classes without changing data distribution. ICC decomposes a large majority class into smaller sub-classes by clustering, which leads to a more balanced distribution. This technique is applied before training the classifier. Each class or classes are individually decomposed into sub-classes, each instance of which will be assigned a new class label. This new set of training data is then used to build a classification model. They have designed different strategies of selecting the number of clusters and determining labels for decomposed classes. Their evaluation have demonstrated that creating a more balanced class distribution leads to improved classifier performance. Adding new classes creates new decision boundaries, which improves the performance of classifiers of high bias classifiers like Naive Bayes. This work is most similar to ours in terms of dealing with skewed class distribution. The main difference is that we focus on minority and subtle classes and also instead of separating the classes into more balanced sub-classes, we apply a hierarchical approach to deal with majority and minority classes at different levels.

C. Hierarchical Classifiers

Ensembles and hierarchical classifiers are often used to recognise complex activities. Nguyen et al. have applied a hierarchical HMM (HHMM) to recognise primitive and complex behaviours of multiple people [15]. They construct a unified graphical model composed of a set of HHMMs with data association. Banos et al. [1] present a fusion classification approach called Hierarchical-weighted classification (HWC). This model combines hierarchical decision (HD) technique and majority voting (MV). HD the classifiers' decision are made in strict order of classification capabilities. It gives more importance to those classifiers which generally perform better. The MV is a democracy-based model where all the classifiers have the same opportunity to take a decision. The HWC is composed by three classifications levels. Each classifier has the same opportunity of collaborating on the final decision, but ranking the relative importance of each one through the use of weights based on the individual performance of each classifier. Their model outperforms other multiclass approaches and improves the scalability and robustness with respect to other traditional fusion techniques.

Our proposed approach also is built on a hierarchical classifier but the difference from the above work is that the hierarchy comes from a collection of sub-groups, within each of which data have high similarity. The employed classifiers employed are dedicated to learn specific difference to differentiate them.

III. Minority and Subtle Activity Recognition

A. Problem Statement

Recognising everyday routine activities can be challenging, as it involves understanding human behaviour from complex interactions between diverse sensor signals.

Let X_c be a collection of instances belonging to a class c and P_c be a pattern of the class c, which is a generalised representation on its instances X_c. We define an activity class

c is *minority* if its instances are significantly less than the averaged activity class size; that is, $\frac{|X_c|}{\frac{\sum |X_{c_j}|}{|C|}} \leq \theta$, $\forall c_j \in C$, where C is a collection of classes of interest; and *subtle* if its pattern representations are close to some other classes; that is, $dist(P_c, P_{c_j}) \leq \delta$, $\exists c_j \in C$.

For example in Figure 1, if we consider the threshold θ as 0.5, then the activity 'R1 leave home' is considered as a minor class as the ratio of its instances to the averaged instances of all the classes is 0.2, while the activity 'R1 sleep' is considered not as a minor class as its ratio is 5.04. There are different ways of characterising pattern representations and evaluating the distance between them. For example, if we take an intuitive way – calculating the Euclidean distance between the centre points of two activity classes, and set the threshold δ as 0.1, then we can consider the four activities of leaving and entering home of both users as subtle, as their distances are only about 0.001. The thresholds can be configured differently to suit the characteristics of datasets and the requirements of the applications.

TABLE I
DISTANCE MATRIX BETWEEN SUBTLE ACTIVITIES

	R1 Leave Home	R1 Enter Home	R2 Leave Home	R2 Enter Home
R1 Leave Home	0	0.0016	0.0004	0.0008
R1 Enter Home	0.0016	0	0.0029	0.0012
R2 Leave Home	0.0004	0.0029	0	0.0011
R2 Enter Home	0.0008	0.0012	0.0011	0

B. Dissimilarity Representation

Dissimilarity representation (DR) represents data as the difference between two objects. It is proposed as a more flexible representation than feature representation, with the purpose of having more information about the structure of the objects [17]. A more formal definition is given as follow [5]:

Given a representation or prototype set $R := \{r_1, r_2, ..., r_n\}$, a training set $T := \{x_1, x_2, ..., x_n\}$, and a dissimilarity measure d. A **Dissimilarity Representation (DR)** of an object x is a set of dissimilarities between x and the objects in R expressed as a vector $D(x, R) = [d(x, r_1), d(x, r_2), ..., d(x, r_n)]$.

The prototype set R is generally a subset of the training set T. The key idea of prototype selection is to find representative instances from training set. The most common approaches are clustering techniques and learning vector quantisation (LVQ) algorithm [10]. After prototype selection, the original feature space will be mapped to a dissimilarity space where each object is represented as a dissimilarity vector $d(x_i, r_j)$ between an original object x_i and a prototype r_j. For binary sensors, an object x_i in the feature space can be represented as $[s_1, s_2, ..., s_n]$, where s_i $(1 \leq i \leq n)$ is the probability of the ith binary sensor being activated during a certain time interval (e.g., every 60 seconds) [23], and n is the number of sensors being deployed. A prototype r_j represents a particular pattern for a subset of objects and a dissimilarity vector $d(x_i, r_j)$ indicates the distance from an object to a pattern. Thus, the dissimilarity representation $D(X, R)$ converts an original object that expresses the activation probability of each sensor into a distance object that suggests the closeness of an original object to each representative pattern in the original feature space.

We can train a classifier on the converted dissimilarity representations, which is dedicated to learn differences to separate objects in different classes. It is different from feature representation based classification that aims to learn the correlations between features and classes. We hypothesise that learning the difference between classes can better characterise distinctive patterns of activities and thus achieve higher recognition accuracies.

C. Dissimilarity Representation Generation

For the prototype selection, we apply a clustering algorithm to each activity separately, and select the centre of each cluster as a prototype. A good prototype that is well separated from the others is crucial to generate dissimilarity vectors.

To guarantee good prototype selection, we use the niche overlapping index [7], which has demonstrated promising results in a recent feature selection study [9]. The niche overlap occurs when two organismic units use the same resources or other environmental variables. Following a similar principle, we will use the overlapping coefficient between prototypes to select the best discriminative prototypes. Let S_i and S_j be two species and let $\mathbf{X} = (X_1, X_2, ..., X_N)$ be a random vector of resources variables. A variable X_t is assumed to be described by the probability density function $f_{it}(x)$ and $f_{jt}(x)$ for species S_i and S_j respectively. The **Niche Overlapping Coefficient (NOC)** β for the variable X_t is defined by:

$$\beta_{ij}(X_t) = \int \min(f_{it}(x), f_{jt}(x))dx, \tag{1}$$

with $0 \leq \beta \leq 1$ and $\beta_{ij} = \beta_{ji}$.

For a group of species S we obtain $S(S-1)/2$ independent pair of species. The averaged NOC for the group can be estimated as

$$\beta_g(X_t) = \frac{\sum_{j=1}^{S-1}(\sum_{j=i+1}^{S} \beta_{ij})}{S(S-1)/2}, \tag{2}$$

where β_{ij} is given by equation 1.

In the following, we introduce how to use NOC in prototype selection. We first cluster X_c – the training instances on each class label c, and take the mode and the mean of each cluster as the prototypes for c. We denote the set of prototypes for a class label c as $\mathcal{R}_c$. In the end, we collect a set of prototypes for all activities in C: $R = \mathcal{R}_{c_1} \cup \mathcal{R}_{c_2} \cup \cdots \mathcal{R}_{c_{|C|}}$. Let $r^i_{c_q}$ ($r^j_{c_s}$) be the ith (jth) prototype on the class c_q (c_s); that is, $r^i_{c_q} \in \mathcal{R}_{c_q}$ and $r^j_{c_s} \in \mathcal{R}_{c_s}$, the niche overlapping index between these two prototypes is calculated as

$$\mathcal{O}_{ij} = \frac{\sum r^i_{c_q} r^j_{c_s}}{\sqrt{\sum {r^i}^2_{c_q} {r^j}^2_{c_s}}}. \tag{3}$$

The overlapping measure O_i of a prototype $r^i_{c_q}$ is computed by averaging all the $\mathcal{O}_{ij}$ from Equation (3) between $r^i_{c_q}$ and the prototypes in the other classes; *i.e.*,

$$O_i = \frac{\sum\limits_{\substack{\forall r^j_{c_s} \in \mathcal{R}_{c_s} \\ \forall c_s \in C, c_s \neq c_q}} \mathcal{O}_{ij}}{|R - \mathcal{R}_{c_q}|}.$$

We rank the prototypes according to the averaged overlapping measures and select the prototypes with a lower overlapping index. That is, we set the mean of the overlapping measures on all the prototypes as a threshold ϕ, and select the prototypes if their overlapping measure O is no greater than ϕ. The smaller the overlapping measure, the better the prototype, suggesting a good separation from the others. The representation set $R := \{r_1, r_2, ..., r_n\}$ is built with the prototypes selected.

After prototype selection, the original feature space will be mapped to a dissimilarity space where each object is represented as a dissimilarity vector $d(x_i, r_j)$ between a original object x_i and a prototype r_j. These converted dissimilarity representations will be fed into a classifier to learn differences to separate objects between classes.

D. Hierarchical Classifier

We design a hierarchical classifier to perform two-phase learning, which is illustrated in the workflow in Figure 2. The first phase of learning performs classification based on dissimilarity representations to distinguish the majority of classes, while the second phase is to focus on subtle classes that cannot be correctly separated from other classes. To do so, we test the classifier in the first phase with the training data again and collect all the misclassified instances. The second phase learning is designed as a stacked ensemble that is built on one-class classifiers (OCCs) for classes that have been misclassified in the first phase. OCCs characterise each class and the ensemble will focus on learning the correlations between probability distributions of OCCs and class labels. Such combination can lead to a more effective way to distinguish classes with similar patterns. The process of the second phase learning is described below.

Let $\mathcal{M}$ be the set of misclassified instances.

1. Employ a resampling technique on the misclassified instances as they can be imbalanced; that is, $\mathcal{M}_B$ = resample($\mathcal{M}$).

2. Train an OCC on each misclassified class in $\mathcal{M}_B$.

3. Build a stacked ensemble on the conditional probabilities of OCCs; that is, train another classifier on the union of all conditional probabilities from each OCC, $P_M = [P_{c_1}\ P_{c_2}\ \cdots\ P_{c_m}]$, where c_i is a class label in $\mathcal{M}_B$ and $1 \leq i \leq m$.

IV. Experiment and Evaluation

We hypothesise that DRHC algorithm can significantly improve the accuracies of recognising minority activities with less distinctive patterns by learning good representations of sensor data. More specifically, we are mainly interested in the following three questions: (1) Does DRHC outperform the state-of-the-art classifiers in recognising minority and subtle activities?; (2) Does DRHC outperform the existing sampling techniques at targeting minority activities?; and (3) Does DRHC outperform the existing representation learning techniques in learning features?.

To address the above questions, we will compare the accuracies of DRHC with a collection of state-of-the-art classification techniques on a real-world third-party dataset that is collected in a two-user home setting.

A. Selection of Datasets

We mainly test our algorithm on smart home datasets that involve binary event-driven sensors with imbalanced activity distributions. For this purpose, we identify the dataset, the Interleaved ADL dataset from the CASAS smart home project at the Washington State University [4], referred to as *WS*. This dataset is collected in a student apartment testbed during the 2009-2010 academic year. The apartment is instrumented with various types of sensors to detect user movements, interaction with selected items, the states of doors and lights, consumption of water and electrical energy, and temperature, resulting in 2, 804, 812 sensor events. This dataset recorded 13 activities performed by 2 individuals, as shown in Figure 1. We use a semantic approach to separate sensor data for concurrent activities [23]. There are two main goals of our algorithm on this dataset: (1) distinguishing two users for the same type of activities performed in a public area; for example, whether the one who watches TV is the user *R1* or *R2*, and (2) distinguishing one users' activities performed in the same area; for example, whether a user sleeps or wanders in a room. These two types have demonstrated as a challenging problem in multi-user concurrent activity recognition [23].

B. Metrics

Given that all the datasets have an imbalanced distribution of activities, we use the class-based F1-score to indicate the performance of an algorithm [6].

C. Technique and Parameter Setup of DRHC

DRHC can be configured with any appropriate distance metric, clustering technique, and classifier. For dissimilarity prototype generation, we have experimented different distance metrics, including cosine, Euclidean, Kullback-Leibler divergence, Mahalanobis, and Bray-Curtis, and prototype generation algorithms including the traditional LVQ, KMeans and DBSCAN clustering algorithms. Among them, the best results are achieved with cosine and DBSCAN, which are reported in the following section.

We have also experimented with different techniques as the base classifier, including SVMs with the linear and RBF kernels, Naive Bayes (NB), K Nearest Neighbour (KNN), Decision Tree (DT), and Random Forest (RF). Each of these techniques has demonstrated promising results in activity recognition [22]. In our experiments, the SVM with the RBF kernel and Random Forest have performed the best. For the sake of computation performance, we select SVM RBF as the base classifier for DRHC and for all the others.

We choose a combined resampling technique – SMOTE

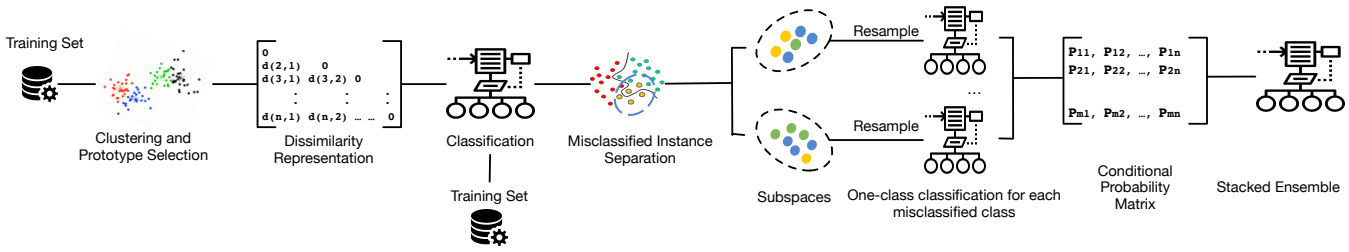

Fig. 2. Workflow of dissimilarity representation-based ensemble.

(Synthetic Minority Over-sampling TEchnique) followed by Tomek link [14]. That is, we first over-sample minority class instances by creating synthetic examples as close as to their nearest neighbors, and then remove the majority class instances that are part of a Tomek link. A pair of instances is called a Tomek link if they are each other's nearest neighbours but belong to different classes. We have compared the performance of this combined sampling technique with the other techniques including SMOTE, Edited Nearest Neighbour (ENN) [8], and Repeated Edited Nearest Neighbour (RENN), in which the ENN algorithm is applied successively until it can remove no further points [14]. This combined technique has achieved the best performance.

D. Process

We run 100 iterations for 5-fold cross validation on each dataset. In each iteration, we test all the algorithms including both DRHC and the state-of-the-art classifiers. We report the mean and standard deviation of F1-scores in each table in Section V. To compare the results, we run Welch's t-test on the accuracies of all the iterations and calculate the p-values. We hypothesise that the DRHC outperforms all the other classifiers. We select a standard significance level of 95% for the test, meaning that if the p-value in the test result is smaller than 0.05, then we accept that there is a statistically significant improvement.

V. Results and Discussion

In this section we will present and discuss the results.

A. DRHC and State-of-the-Art Classifiers

To demonstrate the effectiveness of the DRHC algorithm, we will compare with three types of classifiers. First of all, we collect a wide range of classifiers in different types, including KNN, SVM, NB, and RF. To note that we focus on learning sensor data representation, but not sequential relationships, so we exclude any sequence-based learning, like Hidden Markov Model and Sequential Mining.

TABLE II
DRHC Compared to State-of-the-Art Classifiers

DRHC	SRHC	SVM RBF (B)	SVM RBF	RF (B)	RF	KNN	NB
0.71 ± 0.04	0.69 ± 0.06	0.54* ± 0.01	0.52* ± 0.01	0.56* ± 0.01	0.53* ± 0.08	0.53* ± 0.05	0.49* ± 0.02

Figure 3 presents the F1-scores of recognising activities on the *WS* dataset from DRHC, the state-of-the-art techniques, and SRHC (the variant of DRHC, replacing the dissimilarity representations with the usual sensor features in Section III-B). Table II reports the mean and standard deviation of averaged F1-scores over 100 iterations, and the star * indicates that there is a statistically significant improvement of DRHC over the state-of-the-art techniques.

State-of-the-art techniques perform poorly on recognising the minority and subtle activities, especially distinguishing users for activities in common areas – whether it is R1 or R2 of preparing a meal, watching TV, or bathing, or separating one user's activities in the same room – whether R1 is wandering in room, or working or sleeping. Figure 4 shows the sensor feature distribution on these three activities, because their occurrence activates the same set of sensors. Also the distributions on these activities are imbalanced; that is, the activity 'R1 wandering in room' only takes 0.05% of the whole dataset while the other two activity classes dominate the dataset; *i.e.*, 24% and 12%. The small difference is more difficult to be learned with the dominance of the majority classes. SVM and RF have improved the recognition accuracies when configured with the balanced class distribution option, by boosting some minority classes. In comparison, DRHC outperforms them on most of activity classes and leads to significantly improved overall F1-scores. With the two-phase learning, especially the second-phase of learning in DRHC, we can look into discriminative features that well separates 'R1 wander in room' from the other two.

Difference between DRHC and SRHC is not significant, and we reject our hypothesis that the dissimilarity representation is more effective in distinguishing subtle differences between classes. This is mainly due to the effectiveness of identified prototypes. The problem remains of how to best to separate prototypes when we have activities that activate the same set of sensors, with the difference in their sensor distribution being almost undetectably small. Such subtle differences can challenge prototype selection. Future investigations include the design and evaluation of prototype selection and distance metrics to better represent and separate subtle difference in these distributions.

B. DRHC and Sampling Techniques

The next experiment compares DRHC with the techniques that focus on imbalanced class distribution. These include resampling techniques mentioned in Section IV, plus a more

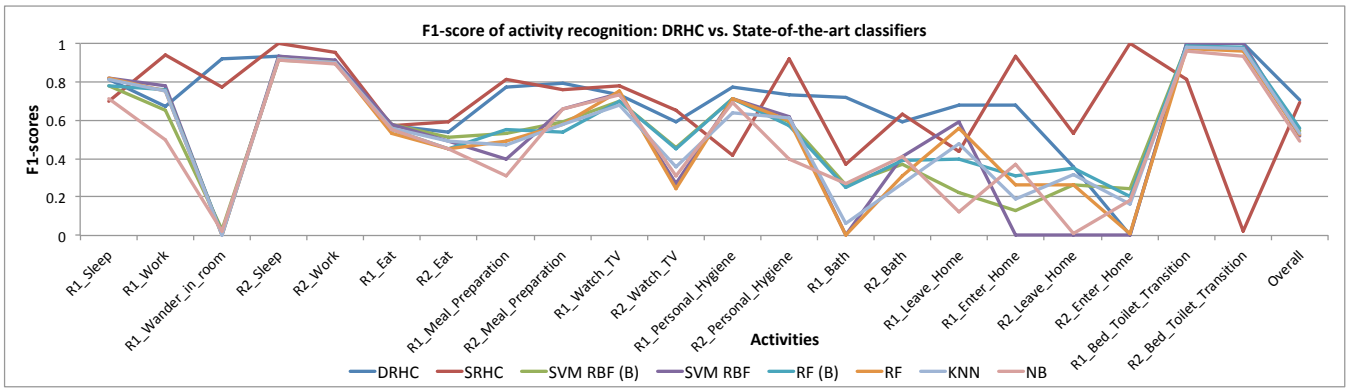

Fig. 3. Comparison of DRHC with the state-of-the-art classifiers.

	M45	M46	M47	M48	M49	M50
R1_Sleep	0.16	0.36	0.26	0.05	0.1	0.03
R1_Wander	0.08	0.22	0.24	0.24	0.19	0
R1_Work	0.04	0.08	0.16	0.24	0.4	0.04

Fig. 4. Sensor feature distribution on the activities 'R1 sleep', 'R1 work', and 'R1 wander in room'.

recent technique, called intra-class clustering (ICC), where instances are clustered and candidate labels are generated to enforce a balanced class distribution between clusters [6]. For each of these options we take training data with sensor dissimilarity representations and train a SVM classifier.

Figure 5 presents the F1-scores across all the activity classes on the *WS* dataset. DRHC consistently outperforms all these techniques in Table III. It demonstrates that sampling techniques alone will help balance the class distribution. For example, all these sampling techniques enables more accurate recognition on the activities of 'R1 wander in room', and bathing, leaving and entering home of both, compared to the results on SVM RBF classifier in Figure 3. Especially, SMOTE consistently outperforms the other sampling techniques. The reason is that ENN and RENN undersample the majority classes by removing data points. The more imbalanced the data set is, the more samples will be discarded when using these techniques.

DRHC outperforms SMOTE and one reason might be that the sampling technique could generate potentially misleading information through oversampling the minority class [6]. SMOTE might introduce instances that do not add any information about the minority classes which can be consider as noisy instances rather than true representation of them.

TABLE III
DRHC COMPARED TO RESAMPLING TECHNIQUES

DRHC	SMOTE	RENN	ENN	ICC
0.71 ±0.04	0.65* ±0.01	0.57* ±0.01	0.56* ±0.01	0.25* ±0.01

C. DRHC and Representation Learning Techniques

The third and final stage is to assess DRHC performance against the current representation learning techniques [2]: PCA, t-Stochastic Neighbour Embedding (t-SNE) [21], and Autoencoder [12]. We feed the generated dissimilarity representations to the above representation learning techniques, and input the learned representations to the SVM RBF balanced classifier for classification. Table IV reports the mean and standard deviation of F1-scores. The results show a statistically significant improvement of DRHC over the compared representation learning techniques.

TABLE IV
DRHC COMPARED TO REPRESENTATION LEARNING TECHNIQUES

DRHC	PCA	t-SNE	Autoencoder
0.71 ±0.04	0.25* ±0.00	0.68* ±0.02	0.51* ±0.00

Detailed results are shown in Figure 5. PCA performs the worst possibly indicating that compressing the data loses meaningful information of the classes leading to a very low F1-score. In addition, we need to retain 99% of the variability in order to have a good representation. That is, we need to preserve almost the same number of feature vectors so that the classifier could distinguish between activities. The results using PCA are not very outstanding and its poor performance is consistent with the literature [11], which suggests that PCA misses important nonlinear structures of the data.

t-SNE transforms the input feature vectors into 2 or 3 dimensions, which has been widely used in visualising high-dimensional data. The features learnt from t-SNE can well separate some classes, but not for classes with little difference. t-SNE technique is able to learn good features to separate classes, nevertheless the classifier is biased by the majority classes in each cluster, which still results in the poor recognition accuracies on the minority classes.

Autoencoders have been widely used in speech recognition, image classification, and face recognition [12], achieving promising results in compressing data by learning linear and nonlinear relationships between features. However, they are less able to differentiate activities with less distinctive patterns. We have configured the autoencoder with different parameters, such as different numbers of layers, different numbers of neurons, and various optimisation functions. No set of parameters

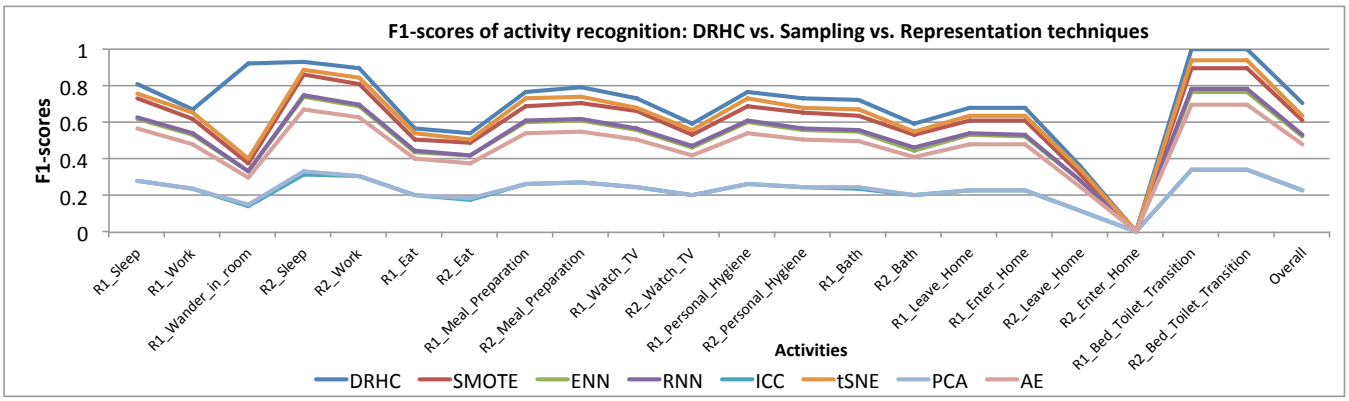

Fig. 5. Comparison of DRHC with the sampling and representation learning techniques.

significantly improves the classification accuracy, indicating that an autoencoder fails in representing noisy data with few spare feature vectors. However, we implement a standard sparse autoencoder, and with more sophisticated autoencoders (such as the variational autoencoder [19]) discriminatory performance may be improved. This is out of scope for this paper, however, and will be the focus of future investigations.

VI. Conclusion and Future Work

In this paper, we present a new technique, DRHC, based on dissimilarity representation, which leverages a dissimilarity representation-based multi-level ensemble in recognising minority and subtle activities. A sequence of empirical evaluation and comparison demonstrates that this is a challenging task where existing structure- and feature-based learning techniques do not perform well in general, and our DRHC algorithm constitutes a statistically significant improvement on existing methods. The key novelty of our approach is that we reduce the bias of the ensemble classifier by training it on a subset of classes so that the classifier could focus on minority activities and hence reliably identify well-discriminating features. So far, we have only considered ambient sensor data (e.g. doors opening and closing, motion sensors firing, lights turning on and off, etc). Recent developments in wearable technologies such as smart watches also allow collection of more dynamic data such as accelerometer and heart rate. These data would contribute to detecting subtle and minority activities, and dissimilarity representation based classification of combined ambient and mobile data will be useful in the future accurate detection of important events in the ageing population.

References

[1] O. Banos, M. Damas, H. Pomares, F. Rojas, B. Delgado-Marquez, and O. Valenzuela. Human activity recognition based on a sensor weighting hierarchical classifier. *Soft Computing*, 17:333–343, 2012.

[2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, Aug. 2013.

[3] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(6):790–808, Nov 2012.

[4] D. Cook and M. Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. *Methods of Information in Medicine*, 48:480–485, 2009.

[5] R. P. Duin and E. Pekalska. The dissimilarity representation for structural pattern recognition. *Pattern Recognition*, 7042:1–24, 2011.

[6] K. D. Feuz and D. J. Cook. Modeling skewed class distributions by reshaping the concept space. In *AAAI '17*, pages 1891–1897, 2017.

[7] E. Harner and R. C. Whitmore. Multivariate measures of niche overlap using discriminant analysis. *Theoretical Population Biology*, 12(1):21 – 36, 1977.

[8] P. Hart. The condensed nearest neighbor rule (corresp.). *IEEE Trans. Inf. Theor.*, 14(3):515–516, Sept. 2006.

[9] I. Hübener and K. David. Fesnoc: A novel feature selection algorithm based on niche overlapping coefficient. In *PerCom '18*, pages 1–7, March 2018.

[10] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:437, 2000.

[11] U. Kruger, J. Zhang, and L. Xie. Developments and applications of nonlinear principal component analysis – a review. In *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 1–43, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[12] K. Liang and H. Chang. Representation learning with smooth autoencoder. In *12th Asian Conference on Computer Vision, Singapore*, number Part II, 2014.

[13] A. Mannini and A. M. Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2):1154–1175, 2010.

[14] A. More. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv:1608.06048*, 2016.

[15] N. T. Nguyen, S. Venkatesh, and H. Bui. Recognising behaviours of multiple people with hierarchical probabilistic model and statistical data association. In *BMVC '06*, pages 126.1–126.10, 2006.

[16] T. Nguyen, V. Nguyen, F. D. Salim, and D. Phung. SECC: Simultaneous extraction of context and community from pervasive signals. In *PerCom 2016*, pages 1–9, March 2016.

[17] E. Pekalska. *Dissimilarity Representations in pattern recognition*. PhD thesis, Delft University of Technology, 2005.

[18] T. Plotz, N. Y. Hammerla, and P. Olivier. Feature learning for activity recognition in ubiquitous computing. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[19] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin. Variational autoencoder for deep learning of images, labels and captions. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2352–2360. Curran Associates, Inc., 2016.

[20] K. Shirahama, M. Grzegorzek, and L. Koping. Codebook approach for sensor-based human activity recognition. In *UBICOMP/ISWC '16 ADJUNCT*, 2016.

[21] L. van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15:1–21, 2014.

[22] J. Ye, S. Dobson, and S. McKeever. Situation identification techniques in pervasive computing: a review. *Pervasive and mobile computing*, 8:36–66, 2012.

[23] J. Ye, G. Stevenson, and S. Dobson. Kcar: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing*, 19:47 – 70, 2015.