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

Andrea Rosales Sanabria, Thomas W. Kelsey, Simon Dobson and Juan Ye *

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 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 proposal with a collection of state-of-the-art classification techniques on four real-world third-party datasets that involve different types of object sensors and are collected in different environments and on different subjects and six imbalanced datasets from the UCI-Irvine Machine Learning repository. Our results demonstrate that our hierarchical classifier approach performs better than state-of-the-art techniques including both structure- and feature-based learning techniques. The key novelty of our approach is that we reduce the bias of the ensemble classifier by training it on a subspace of data, which allows identification of activities with subtle differences, and thus provides well-discriminating features.

Keywords: Activity recognition, Representation learning, Sampling techniques, Ensemble learning, Smart home

1. Introduction

The European Commission has predicted that by 2025, the United Kingdom alone will see a rise of 44% in people over 60 years of age. This motivates the development of new solutions to improve and to guarantee an adequate quality of life and independence for elderly people. Sensor-based human activity recognition involves the abstraction low-level sensor data into high-level descriptions (i.e., activities) [32]. It has many exciting pervasive computing applications in our everyday life, one of which is ambient assisted living in the smart home.

Ambient sensors, including positioning or pressure sensors and RFID sensors, can be deployed to detect the whereabouts of older adults and their interactions with everyday objects. With the support of intelligent algorithms, we aim to infer their current activities (e.g., making a meal or performing personal hygiene) and also detect changes in their health conditions over time. This would enable timely intervention by sending alerts to users, family members, and/or caregivers [30]. There is evidence that some adverse health conditions can be prevented and controlled if detected in a timely manner [2]. This situation reinforces the motivation for developing preventive methods to guarantee a better quality of life.

Recent advance in data mining, machine learning, and deep learning have made it possible to learn complex correlations between low-level sensor data and high-level activities, but it remains challenging to distinguish activities that have both subtle differences and imbalanced distributions, since these can have significant implications in health-related applications. For example, life-threatening situations such as fall, strokes or heart attacks are infrequent and may have subtle differences when compared to the sensor data for other daily activities. Effective recognition of these incidents is central to the robustness of any activity recognition system.

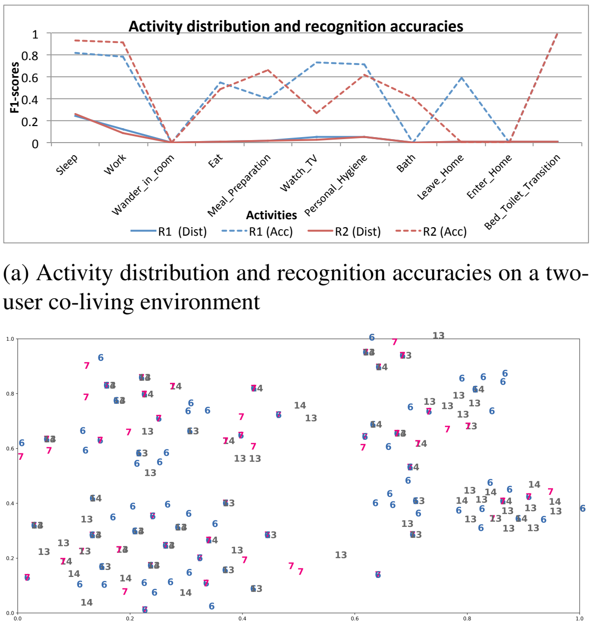
To illustrate the challenge of recognising minority and subtle activities, we use the following example. Figure 1a presents the distribution of a set of concurrent activities from two users recorded in a smart home

*Rosales et al. are in the School of Computer Science, University of St Andrews, UK.
jy31@st-andrews.ac.uk

setting [6] and activity recognition accuracies (measured in F1 scores) from a Support Vector Machine (SVM) with a RBF kernel. This example demonstrates that the SVM RBF can reliably recognise the majority activities such as “Sleep" and “Work" and the activities with distinct patterns such as “Bed_Toilet_Transition". However, it performs poorly on (i) distinguishing the activities from the same user occurring in the same area; for example, is a user wandering or working in the bedroom?, and (ii) differentiating between the users for the same type of activities performed in a public area; for example, is the user *R1* or *R2* leaving the house?. 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. Secondly, 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. Figure 1b visualises the sensor features for these four activities on a 2D plot using t-distributed Stochastic Neighbor Embedding (t-SNE). As we can see, these activities are quite mixed together and there is no clear boundary between them.

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?* To address these, 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 [8]. 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 third-party datasets 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 2 introduces the existing literature in activity recognition. Section 3 introduces dissimilarity representation and proposes DRHC. Section 4 describes the evaluation methodology and Section 5 presents the evaluation results and discusses the performance of DRHC over the other state-of-the-art classifiers. The paper concludes in Section 6.



(a) Activity distribution and recognition accuracies on a two-user co-living environment

(b) A 2D plot to visualise subtle activities. The labels stand for “R1 Leave Home" (6), “R1 Enter Home" (7), “R2 Leave Home" (13), and “R2 Enter Home" (14).

Fig. 1. An illustration of the problem and challenge of recognising minority and subtle activities

2. Related Work

Human activity recognition 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.

2.1. State of the Art of Sensor-based Human Activity Recognition

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) [5, 39].

Ye et al. [40] have applied ontologies to support automatic segmentation of sensor data for multi-user concurrent activities. They have employed the Pyramid Match Kernel to separate the activities with similar patterns to a certain degree. This is achieved by calculating the difference of sensor feature distributions in a hierarchical manner. However they still cannot distinguish the users for the same activities; for example, identifying which user is cooking.

van Kasteren et al. [35] apply HMM to model sequential relationships of sensor data and activities. The HMM is trained to obtain three probability parameters, where the prior probability of an activity represents the likelihood of the user starting from this activity; the state transition probabilities represent the likelihood of the user changing from one activity to another; and the observation emission probabilities represent the likelihood of the occurrence of a sensor observation when the user is conducting a certain activity. Even though the HMM has well built the temporal probabilist model between activities and sensor observations and thus successfully recognised activities, but it has achieved low accuracies on minority activities [34].

More recently, convolutional neural networks (CNNs) have become a popular approach to automatically extract features from low-level sensor data. Morales et al. [21] introduce transfer learning in wearable activity recognition using CNNs. Kernels in CNNs are supervised neural networks that can act as feature extractors by stacking several convolutional operators to create a hierarchy of progressively more abstract features. CNNs learn a hierarchical representation of the data and have the ability to identify salient patterns in the signal with deeper layers. They have applied transfer learning technique to learn model parameters for a classification task by incorporating data from a different but related classification task. They analyse kernel transfer between users in the same application domain (i.e., the same dataset), between applications domains, between sensor locations and between sensor modalities.

These techniques have demonstrated promising results in learning complex correlations between human activities and sensor features. However, few of these existing techniques have focused on learning good representations of sensor data so as to further distinguish activities that have subtle difference.

2.2. 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 [42]. 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 [42]. 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 [3].

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 [28]. 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 [20], 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.

2.3. Feature Selection

The goal of feature selection methods is to find a subset of optimal features to obtain high classification accuracy by analyzing low-dimensional data. Feature selection algorithms not only address the issue to find a good data representation to improve the recognition of minor activities with less distinctive patterns but also reduce the computational costs by removing redundant information.

David et al. [15] propose a novel hybrid feature selection algorithm based on the niche overlapping coefficient (FeSNOC). Their proposal combines *k*-Nearest Neighbour algorithm and the overlapping coefficient to find the best features. They use the niche overlapping coefficient to estimate the overlapping between classes and use this measure as a measure of similarity between two activity classes. FeSNOC consists of four steps. In the first step, FeSNOC uses *k*-Nearest Neighbour algorithm to calculate the classification accuracy using each feature. Then, it computes the overlapping coefficient for each feature and uses a linear relationship between the accuracy and overlapping coefficient. Based on the fit function obtained in this step, FeSNOC ranks and selects the features. David et al. [15] have shown that their proposal outperforms other filters and hybrid approaches such as Information Gain, One R, Gain Ratio, Chi-Square, and Relief-F.

Wang et al. [36] introduce a discrimination index based on neighborhood cardinality rather than neighborhood similarity classes to measure the uncertainty quantity of the distinguishing ability of a feature subset. This index has similar properties to Shannon entropy. They define joint discrimination index, conditional discrimination index, and mutual discrimination index. These measures are used to calculate the change of distinguishing information caused by the combination of multiple feature subsets. The conditional discrimination index is used to characterise the ability of a subset of features to distinguish samples with different decisions. The idea behind this index is that the smaller the conditional discrimination index, the greater the distinguishing ability of the feature subset. Guo et al. [11] propose to concatenate frequency-based sensor features with TF-IDF features, which demonstrates an improvement over frequency-based features alone.

Xiang et al. [38] use the least square regression (LSR) to enlarge the distance between different classes. To achieve this, they introduce a new technique called ϵ-dragging to force the regression targets of different classes moving along with opposite directions.

In the last years, many new techniques have been develop in machine learning that have proven successful in human activity recognition in smart home environments. Oukrich et at. [25] introduce a Multilayer Perceptron model made up of three layers that uses back propagation algorithm to recognise activities of daily living in smart home. Then they use minimal-redundancy-maximal-relevance criterion method for feature selection of observed motion and door sensors.

At this point, we have discussed about feature selection to remove redundant features while achieving good classification performance. Other way to reduce dimensionality is feature transformation. In contrast with feature selection, feature transformation methods transform the original features to a new feature subspace. Tao et al. propose a feature selection method that combines LDA and sparsity regularization. They proved that extending the $l_{2,1}$-norm-based formulation to the $l_{2,p}$-norm improves the ability to select the most discriminative features and remove the redundant ones. They tested their proposal on various real-word data sets to illustrate the advantages of the proposed method.

2.4. Imbalanced Class Distribution

Activity data collected in the real-world environments, as presented in Figure 1a, can often have imbalanced distributions. This issue occurs when the number of instances one activity is much lower than the ones of the other activities. This problem has yet attracted sufficient attention from researchers.

Due to the importance of this issue, a large amount of techniques have been developed to address the problem. Feuz et al. [9] 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 subclasses, 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.

The performance of classification algorithms can be greatly affected when the dataset is highly imbalance. A large number of different resampling techniques have been proposed in the literature to deal with the class imbalance problem. Galar et al. [10] have categorised resampling techniques into three groups: undersampling methods, which create a subset of the original data set by eliminating instances; oversampling methods, which create a superset of the original data-set by creating new instances from existing ones; and hybrids methods that combine both sampling methods. More et al. [22] have compared different techniques with respect to their effect on the recall on the minority class and the precision on the majority class. They have used a synthetic dataset with 1,000 instances and two classes. They conclude that combining SMOTE and Edited Nearest Neighbours (ENN) with as logistic regression classifier and BalanceCascade give the best performance. Guo et al. [12] have presented an improved SMOTE algorithm to address the imbalance problem, which uses the Euclidean distance of each minority class to adjust the distribution of all the classes, and generates new synthetic minority classes in the neighbourhood of remaining minority-class examples.

2.5. *Hierarchical Classifiers*

Ensembles and hierarchical classifiers are often used to recognise complex activities. Ensemble classifiers are known to increase the accuracy of single classifiers by combining several of them. Galar et al. [10] review the state of the art on ensemble techniques in the framework of imbalanced data sets with a particular focus on the binary classification problem. They proposed a new framework to classify ensemble-based methods in a new category depending on how they deal with cost-sensitive and data preprocessing level before training the classifier.

Nguyen et al. have designed a hierarchical HMM (HHMM) to recognise primitive and complex behaviours of multiple people [24]. They construct a unified graphical model composed of a set of HHMMs with data association. Banos et al. [2] 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.

3. Minority and Subtle Activity Recognition

3.1. *Problem Statement*

Recognising everyday routine activities can be challenging, as it involves understanding human behaviour from complex interactions between diverse sensor signals. Similar to [9], we list a formal definition of the terms in Table 1, based on which we define the problem of interest – recognising minority and subtle activities.

Table 1

Symbols and annotations

Symbol	**Annotation**
$X = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_n, y_n)\}$	the set of labelled training instances
$\vec{x}_i = [sf_1, sf_2, ..., sf_m]$	the *i*th training instance is a *m*-D vector consisting of features generated from a collection of sensors
$y_i \in Y$	the class label for the *i* training instance
$Y = \{y_1, y_2, ..., y_c\}$	the set of possible class labels
$X_{y_i} = \{(x, y) \in X \mid y = c_i\}$	the instances belonging to a class c_i
$P_{c_i} = g(X_{c_i})$	a pattern on a class label c_i is an abstract representation of its instances

Let X_{c_i} be a collection of instances belonging to a class c_i and P_{c_i} be a pattern of the class c_i, which is a generalised representation on its instances X_{c_i}. We define an activity class c_i is *minority* if its instances are significantly less than the averaged activity class size; that is, $\frac{|X_{c_i}|}{\frac{\sum |X_{c_j}|}{|C|}} \leqslant \theta$, $\forall c_j \in C$ and C is all the classes of interest; and *subtle* if its pattern representations are close to some other classes; that is, $dist(P_{c_i}, P_{c_j}) \leqslant \delta$, $\exists c_j \in C$.

For example in Figure 1a, 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. A subtle activity is an activity that has a similar pattern to some other activities. There are different ways of characterising pattern representations and assessing the similarity between them. For example, here 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 2

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

3.2. 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. The idea behind DR is that objects are given the same class label if their difference is sufficiently small. Hence, it should be easier for the classifiers to discriminate between them. A more formal definition is given as follow [8]:

Definition 1. 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 [16]. 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 $(1 \leqslant i \leqslant n)$ is the probability of the ith binary sensor being activated during a certain time interval (e.g., every 60 seconds) [40], 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.

3.3. Dissimilarity Representation based Hierarchical Classifier

As introduced above, dissimilarity representation can help learn discriminative feature representations, but the problem of imbalanced class distributions remains: the prototypes selected might represent varied patterns for majority classes, while the minority classes might either sit at the boundary of prototype clusters and be considered as noise, or be identified into a small number of prototypes. Either way, existing classifier techniques cannot learn them effectively. To address this problem, we introduce a hierarchical classifier that performs two-phase learning, where the first phase is to learn general features in order to recognise majority classes, and the second phase is to collect minority and subtle classes to identify any fine differences between them. Fig 2 illustrates the main workflow of our approach.

For the prototype selection, we apply a clustering algorithm to each activity separately, and select the centre of each cluster as a prototype. Once the prototypes are selected, we compute the pairwise dissimilarity matrix $D(x_i, R)$. Then we train a base classifier on the converted dissimilarity representations. We apply the trained classifier on the training data again, and collect all the misclassified instances. To further learn any differences, we apply a clustering technique on their dissimilarity representations to group them, where mem-

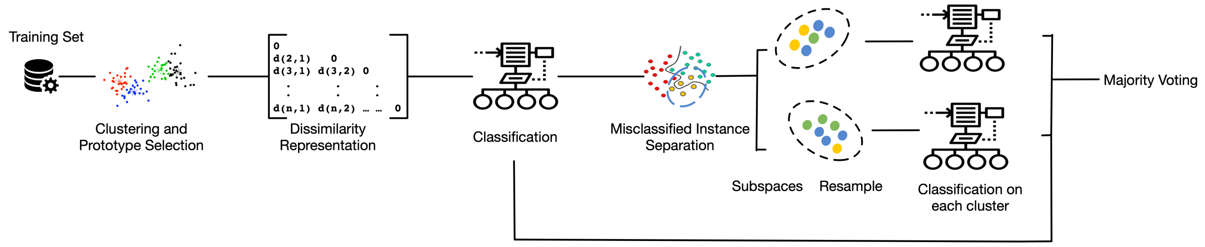

Training Set
Clustering and Prototype Selection
0
d(2,1) 0
d(3,1) d(3,2) 0
d(n,1) d(n,2) … … 0
Dissimilarity Representation
Classification
Misclassified Instance Separation
Subspaces
Resample
Classification on each cluster
Majority Voting

Fig. 2. DRHC Workflow

bership of each cluster means that instances have small differences. Clusters may suffer from data imbalance across classes, which can lead to poor performance in the classification process. Therefore, for each cluster, we use a resampling technique to collect instances for misclassified classes to enforce a balanced distribution. For example in our first-phase, the classifier might confuse the “Wander" activity with “Sleep" activity as they activate the same set of sensors, and there might be significantly more “Wander" examples than the “Sleep" examples in the misclassified set. In our second-phase, we will not only include these misclassified instances but also use the over-sampling techniques to sample instances from both activities in the training set in order to reach a balanced distribution.

Finally on each cluster g_l ($l = 1, ..., L$), a classifier f_l is built to further pinpoint features to separate them. We use majority voting to integrate inference results from f_l for the final decision. This gives rise to the Hierarchical Classifier Algorithm shown in Fig 1. DRHC novelly combines the following four aspects in an integrated and systematic approach: dissimilarity representation of sensor feature, clustering misclassified instances to enforce learning discriminative features, resampling to obtain balanced class distribution, and the application of an ensemble to integrate inference.

For classification, given an unlabelled sensor feature vector, DRHC will first calculate the probability class distribution using the classifier in the first phase. We choose the class with higher confidence score. Then, the unlabelled sensor feature vector will go through the second classification phase, were it is evaluated by each classifier f_l from each cluster. Each classifier f_l will output a confidence score. Finally, we use majority voting to obtain the final label.

4. 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:

Algorithm 1 Dissimilarity Hierarchical Classifier Algorithm

```
Require: a training set $T$
Compute a prototype set $\{r_1^{(i)}, r_2^{(i)}, ..., r_k^{(i)}\}$, where $r_j^{(i)}$ is the $j$ prototype for activity $i$, $i = 1, ..., C$
for $x \in T$ do
    compute dissimilarity representation $DR = \{d(x, r_j^{(i)})\}$, $i = 1, ..., C$ and $j = 1, ..., k$
train a baseline classifier $bc$ on $DR$ and test on $DR$
collect misclassified instances from $DR$ and cluster them into $L$ groups
for $l \in L$ do
    resample to obtain training set $g_l$
    train an ensemble classifier $f_l$ on $g_l$
```

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?
3. Does DRHC outperform the existing representation learning techniques in learning features?

4.1. Selection of Datasets

We test our algorithm on two types of data taken from collections available to the entire research community: smart home and general machine learning datasets. To evaluate performance as a classification method and demonstrate the generality, we use six imbalanced datasets from the UCI-Irvine Machine learning repository[1]. To evaluate performance on data with

[1] https://archive.ics.uci.edu/ml/datasets.html

binary event-driven sensors having imbalanced activity distributions we use three datasets collected and curated by the University of Amsterdam (named *HA*, *HB*, and *HC* respectively in the remainder of this paper) [35]. They were collected in three different residential settings, with three different users. In House A, the sensor network is composed of 14 state-change sensors on household objects like doors, cupboards, or toilet flush. In House B and C, each network node was equipped with heterogeneous sensors: passive infrared to detect motion in a specific area; pressure mats to measure whether someone is sitting on a couch or lying in bed; switches to monitor whether doors and cupboards are open or closed; mercury contacts to detect the movement of objects (e.g., drawers); and water flow sensors to detect the flush of toilet. All these sensors output binary readings (0 or 1), indicating whether or not a sensor fires. In these three datasets, the activities “Toilet", “Leave House", and “Sleep" dominate the datasets, while the activities “drink” is the least activity being recorded. The main goal of our algorithm is to be able to distinguish “drink” from the other similar activities “Breakfast” and “Dinner”.

We additionally examine the performance of our algorithm using the Interleaved ADL dataset from the CASAS smart home project from Washington State University [6], referred to as *WS*. This dataset was collected from a student apartment testbed during the 2009-2010 academic year. The apartment was 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 [6]. This dataset recorded 13 activities performed by 2 individuals. We use a semantic approach to separate sensor data for concurrent activities [40]. 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 user *R1* or *R2* is watching TV, and (2) distinguishing one user’s activities performed in the same area; for example, is the user sleeping or wandering in a room. These two types have been demonstrated as a challenging problem in multi-user concurrent activity recognition [40].

4.2. Metrics

Given that all the datasets have imbalanced distribution of activities, class-based precision, recall and F1 score are taken as an indicative of the performance of an algorithm [9]. For comparative purposes, F1 score represents the trade-off between Recall and Precision.

$$recall = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$precision = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

The **F1 score** is the harmonic mean of precision and recall:

$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

4.3. Technique and Parameter Setup of DRHC

DRHC can be configured with any appropriate distance metric, clustering technique, and ensemble classifier. There is no a generally agreed distance metric. Some dissimilarity metric are more discriminative than others. It depends on the complexity of the dataset. Finding a well-discriminating dissimilarity measure is difficult.

There might exist plausible dissimilarities which are defined on different representations and reflect different aspects of the data [37]. We are interested in finding out whether using multiple distance metrics can help capture subtle differences between activities. We hypothesis that training the classifiers in different dissimilarity representations and using an ensemble approach to take the final decision will improve activity recognition accuracy over the basic DR-based classifier. We consider the following commonly used distance measures from the literature, including Kullback-Leibler (KL) divergence, Mahalanobis, Cosine, Euclidean, and Bray-Curtis [27].

Let *u* and *v* be *n*-dimension vectors.

- **KL divergence.** Kullback-Leibler divergence is defined as

$$D_{KL}(u||v) = -\sum_i u(i) \log \frac{v(i)}{u(i)}.$$

This measure is not a distance metric but a relative entropy because $D_{KL}(u||v) \neq D_{KL}(v||u)$.

- **Mahalanobis.** The Mahalanobis distance between two points is

$$D_M(u,v) = \sqrt{(u-v)(1/V)(u-v)^T},$$

where $(1/V)$ is the inverse covariance.

- **Cosine**. The cosine measures the cosine angle between two data points. Cosine similarity is particularly used in the positive space, bounded in $[0,1]$. A value of 0 indicates "dissimilar" objects, while 1 means exactly the same. It is define as

$$D_C(u,v) = 1 - \frac{uv}{||u||_2||v||_2},$$

where $|| * ||_2$ is the 2-norm.

- **Bray-Curtis.** The Bray-Curtis distance computes the compositional dissimilarity between two different sites based on the counts of items in each site, which is commonly used in biology [4]. Here we use it to quantify the difference of activation frequencies of sensors on two activity classes. It is defined as

$$D_{BC}(u,v) = \frac{\sum_i(|u_i - v_i|)}{\sum_i(|u_i + v_i|)}.$$

We have experimented all the above metrics, 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 [39]. 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 (Synthetic Minority Over-sampling TEchnique) followed by Tomek link [23]. 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) [13], and Repeated Edited Nearest Neighbour (RENN) [23], in which the ENN algorithm is applied successively until it can remove no further points. This combined technique has achieved the best performance.

4.4. Comparison Process

We performed four stages of comparative evaluation. The first compared DRHC with four alternatives that are considered before proposing our final algorithm.

- *Without Subspace* – where instead of clustering misclassified instances, we apply a hierarchical ensemble on the whole set of misclassified instances to assess whether gathering similar instances together will help differentiate them. The assumption is that applying a classifier on each group of misclassified instances will allow for better discrimination of features to separate them.
- *Without Dissimilarity Representation* – where instead of generating the dissimilarity representation, we train the classifier on the sensor feature representation to assess whether the dissimilarity representations approach is necessary.
- *Without Resampling* – where instead of resampling the misclassified instances, we apply the classifier on each cluster to assess whether resampling is necessary to improve the performance of the classifier.

Stage two involves comparison of DRHC against baseline classifiers, with and without Dissimilarity Representation incorporated into the underlying data. To achieve this, we use the same DBSCAN algorithm to generate prototypes and then employ the above mentioned baseline techniques to train on the converted dissimilarity representations. In particular, we compare DRHC to SVM RBF with the "class weight" parameter set to "balanced". This automatically adjusts weights to be inversely proportional to class size, and has demonstrated promising results in both the handling of imbalanced classification problems and in activity recognition in previous studies [9].

In the third stage DRHC performance is compared with the techniques focus on imbalanced class distribution. These include resampling techniques (1) SMOTE, (2) Edited Nearest Neighbour (ENN) [7],

(3) Repeated Edited Nearest Neighbour (in which the ENN algorithm is applied successively until it can remove no further points) (RENN), and (4) a more recent technique, called intra-class clustering (ICC), where instances are clustered and candidate labels are generated to enforce a balanced class distribution within each cluster [9]. For each of these options we take training data with sensor dissimilarity representations and then train an ensemble SVM RBF balanced model. This controls for variability in performance due to the specific classifier used.

The first three stages are designed to assess the DRHC algorithm against (i) plausible variants of itself, (ii) commonly-used classifiers (with limited or no emphasis on the classification of minor and subtle activities), and (iii) more advanced and specialised classification techniques that might be considered suitable for our purposes. The fourth and final stage is to assess DRHC performance against current state-of-the-art representation learning techniques [3, 42]. As for stage 3, we control for classifier variability by using SVM RBF balanced as the underlying classifier. A key aspect of any activity recognition task is an appropriate feature representation of the sensor data, and the design of suitable classifiers [16]. We consider and assess three types of unsupervised feature learning techniques:

- **PCA**, Principal Component Analysis [17], one of the most popular techniques in learning correlations of features. It is a linear unsupervised algorithm that transforms feature variables into a smaller number of uncorrelated variables called principal components.
- **t-SNE**, t-Stochastic Neighbour Embedding, is a nonlinear technique for dimensionality reduction that minimises the divergence between the distributions [33]. It constructs a probability distribution over pairs of high-dimensional data points in such a way that objects from the same class have higher probability of being picked than those belonging to different classes. Then, it defines a similar probability distribution over the data points in the low-dimensional map, and minimises the Kullback-Leibler divergence between the two distributions with respect to the locations of the data points in the map.
- **Autoencoder**, an unsupervised representation learning technique based on neural networks, which has attracted increasing attention in the deep learning [19] community. This technique learns a function $h(x) \approx x$, i.e, an approximation to the identity function so that the output is similar to the input. Autoencoder consist of two stages: encoding and decoding. It was first used to reduce dimensionality by setting the number of encoders output units less than the input. With the activation function chosen to be nonlinear, an autoencoder has been shown to extract more useful features than some common dimensionality reductions techniques such as PCA [19].
- **NOC**, a recent feature selection technique based on Niche Overlapping Coefficient [26], which has achieved promising results in learning discriminative sensor features [15].

4.5. Statistical Analysis

At each stage, 100 iterations for 5-fold cross validation are performed on each dataset. For each iteration, we calculate the mean class- and instance-F1 scores for DRHC and the comparison techniques. We test the null hypothesis that DRHC will produce higher accuracies than the other alternatives using one-sided (greater) paired Welch's t-tests on the class-F1 scores across iterations, with significance levels for p-values set at 5%. All calculations were performed in R version 3.3.2 [14].

5. Results and Discussion

In this section we will discuss the evaluation results and validate that our proposed algorithm finds a suitable data representation that can improve activity recognition accuracies. In each results table, entries are averaged class- and instance-F1 score over 100 iterations plus or minus standard deviation, with best performance for a given dataset shown in bold. Starred entries denote statistically significantly inferior performance compared to DRHC. WS is the CASAS home sensor data; HA, HB and HC are the Amsterdam data; the remaining rows are unbalanced classification calibration datasets from the UCI machine learning repository.

We summarise the main findings in terms of the three questions proposed in Section 4:

1. Does DRHC outperform the state-of-the-art classifiers in recognising minority and subtle activities?
   Existing classifiers can learn common patterns from majority classes well. They, even less so-

phisticated classifiers like Naive Bayes, have achieved a good classification accuracies for well-represented activities, whereas they all tend to misclassify minority and less distinctive activities. DRHC leverages dissimilarity representation and highlights fine and discriminative features between these activities, leading to significantly improved accuracies.

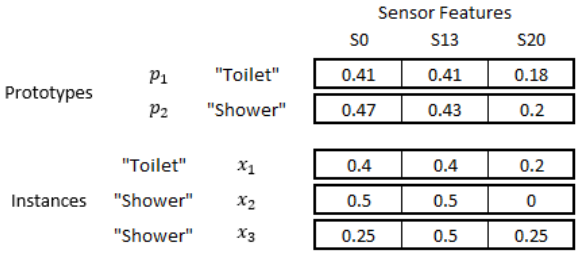
2. Does DRHC outperform the existing sampling techniques in targeting minority classes?
   Sampling techniques can help to recognise minority classes, as with them the base classifier's performance has been improved to a certain degree. However, still for minority and subtle activities, the sampling techniques alone cannot help. DRHC has leveraged the sampling techniques and combined with dissimilarity representations, which has demonstrated a better capability in learning discriminative features between them.
3. Does DRHC outperform the existing representation learning techniques in learning features?
   DRHC outperforms the state-of-the-art representation learning techniques. Even though some techniques like t-SNE clearly finds a good representation of the data, they are not able to recognise subtle differences and the classifier is biased by the majority classes.

5.1. Variations of DRHC

The results for the first stage of empirical evaluation are given in Table 3. DRHC is compared to three candidate methods that were considered during algorithm development, resulting in significantly improved F1 score performance in 23 of 30 comparisons ($p < 0.05$) and comparable performance (i.e. not significantly superior performance) in 3 of the remaining 7 comparisons. For the remaining 4 comparisons, replacement of clustering of misclassified instances with a hierarchical ensemble at the prototyping stage leads to superior average performance. From the results we can see that, in most cases, they all improve the baseline classifiers' results. We can see that only applying an hierarchical ensemble classifier on all instances (No Subspace) has the advantage of characterising diverse aspects of differences, but not necessarily leverage minority activities. We demonstrate that a hierarchical ensemble on all the misclassified instances into groups can help pinpoint dissimilarity features and recognise minor and subtle activities.

Difference between DRHC and 'No DR' is not significant, which 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.

A DR-based approach can work effectively if there exists well-defined and separated prototypes for each class, but this is not the case for the problem that we aim to address. We illustrate the challenge through an example in Figure 3. Here we list two prototypes p_1 and p_2 from the activities "Toilet" and "Shower" respectively, both of which not only fires the same set of sensors (e.g., $S0$ on the toilet, $S13$ in the bathroom, and $S20$ on the microwave)[2], but also the difference in sensor distribution is really small. Now let's look at three instances x_1, x_2, and x_3 from these two activities. The distance from the second instance x_2 to the above two prototypes is hardly observable, even though the other two instances are close but still considered as distinguishable. These subtle differences will not be able to capture in the basic DR approach or in the first-phase of DRHC. Only when we collect and compare them in a group, is the classifier able to characterise the differences.



			Sensor Features		
			S0	S13	S20
Prototypes	p_1	"Toilet"	0.41	0.41	0.18
	p_2	"Shower"	0.47	0.43	0.2
Instances	"Toilet"	x_1	0.4	0.4	0.2
	"Shower"	x_2	0.5	0.5	0
	"Shower"	x_3	0.25	0.5	0.25

Fig. 3. An example of highly similar prototypes in HB

We also compare the current design of DRHC with our previous work [31], and the current design has shown an improvement in 7 of 10 datasets, and comparable performance in 3 of the remaining experiments. We believe that applying a classifier on each cluster helps learn the small differences between activities that

[2] Sensor noise, like $S20$ fires during these two activities, is a common issue in most of the smart home datasets [41].

Table 3

DRHC compared to variants of our algorithm

	DRHC	No Subspace	No DR	No Resample	DRHC (v1) [31]
WS	.720 ± 0.043	**.729 ± 0.002**	.552* ± 0.023	.536* ± 0.018	.706 ± 0.016
HA	**.885 ± 0.018**	.823* ± 0.020	.855* ± 0.027	.765* ± 0.018	.806* ± 0.015
HB	.926 ± 0.007	.652* ± 0.019	**0.963 ± 0.004**	.671* ± 0.033	.545* ± 0.071
HC	**.631 ± 0.004**	.619* ± 0.005	.619* ± 0.009	.560* ± 0.042	.583* ± .016
Abalone	.164 ± 0.010	**.598 ± 0.002**	.179 ± 0.003	.160* ± 0.002	.229 ± 0.014
Ecoli	**.870 ± 0.008**	.279* ± 0.011	.866 ± 0.076	.786* ± 0.070	.792* ± .036
Glass	**.693 ± 0.005**	.279* ± 0.027	.603* ± 0.047	.594* ± 0.001	.606* ± .091
Letters	**.964 ± 0.002**	.360* ± 0.000	.956 ± 0.011	.955 ± 0.002	**.964 ± 0.021**
Letters Cons	**.990 ± 0.013**	.958* ± 0.002	.984 ± 0.005	.984 ± 0.008	**.990 ± 0.011**
Nursery	**.972 ± 0.004**	.827* ± 0.025	.986 ± 0.014	.986 ± 0.010	.738* ± .014

activate similar sensors that are more difficult to learn when trying to separate multiple and different activities.

As we can see, performing clustering on misclassified instances and assigning a classifier on each cluster achieves higher accuracy than only using a set of one-class classifier. The key reason is that clustering gathers instances that come from different classes but have similar patterns; *i.e.*, small distance between their feature vectors, and thus the classifier on a cluster is guided to look for difference in features to separate the instances. In comparison, the one-class classifiers will still focus on learning the boundary of each class, but for activities that have similar patterns, their boundaries can overlap, which does not help to distinguish them.

5.2. DRHC and State-of-the-art Classifiers

In the following, we will compare DRHC with the state-of-the-art classifiers, and in each result table we list both the averaged class- (top) and instance-F1 scores (bottom). Because we deal with imbalanced datasets, we focus our discussion on class-F1 scores.

Table 4 contains results for the initial phase of the second stage of empirical evaluation, in which DRHC is compared to the commonly-used classifier techniques including KNN, NB, SVM RBF, and SVM RBFB, **without** taking dissimilarity representation into account. DRHC exhibits significantly superior performance in 31 of 40 comparisons ($p < 0.05$), and comparable performance in 6 of the 9 remaining instances. DRHC has a lower average F1 score for 3 of the 30 comparisons. In Table 6 we again report the average F1 scores for DRHC compared to the same commonly-used classifier techniques, but for these instances the data are augmented with dissimilarity representations prior to classification. DRHC exhibits significantly superior performance in all the 40 comparisons ($p < 0.05$). It should be noted that in both Table 4 and Table 6 the DRHC results include dissimilarity representation calculations. The fourth column of Table 3 provides the results needed to compare baseline classifiers to DRHC without dissimilarity representation calculations.

Table 5 presents the comparison of averaged F1 scores between DRHC and SVM RBF B on each activity in the WS dataset. We can observe that even with the 'balanced' option, SVM RBF still performs worse on recognising the minority classes such as 'Watch TV', 'Eat', 'Bath', 'Leave Home' and 'Enter Home', or distinguishing subtle patterns between imbalanced activities like 'R1 Sleep', 'R1 Work' and 'R1 Wander in Room', all of which are occurring in the same room and will often activate the same set of sensors. Figure 4 shows the sensor feature distribution on these three activities. The imbalanced distribution makes learning the difference more challenging; that is, the activity 'R1 wander in Room' only takes 0.05% of the whole dataset while the other two activity classes dominate the dataset; *i.e.*, 24% and 12%. In comparison, DRHC outperforms most of the state-of-the-art techniques 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.

When comparing the results with and without dissimilarity representations on the state-of-the-art techniques between Table 4 and 6, the performance varies between datasets. On the smart home datasets, sensor and dissimilarity representations achieve compara-

Table 4

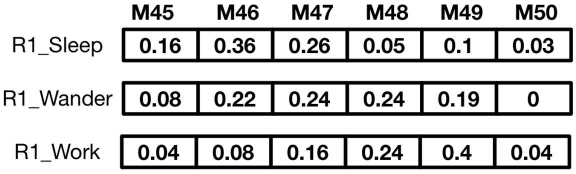
DRHC compared to benchmark classifiers on original features(class-F1 vs. instance-F1 scores)

	DRHC	kNN	NB	SVM RBF	SVM RBF (B)	RF
WS	.720 ± 0.043	.577* ± 0.006	.578* ± 0.022	.514* ± 0.018	.514* ± 0.018	**.799 ± 0.002**
	.830 ± 0.006	.807 ± 0.00	.711 ± 0.00	.730 ± 0.00	.730 ± 0.00	.807 ± 0.002
HA	**.885 ± 0.018**	.806* ± 0.026	.806* ± 0.057	.757* ± 0.052	.780* ± 0.023	.812* ± 0.010
	.886 ± 0.008	.806 ± 0.00	.804 ± 0.00	.794 ± 0.00	.801 ± 0.00	.803 ± 0.017
HB	**.926 ± 0.007**	.751* ± 0.019	.751* ± 0.037	.676* ± 0.003	.704* ± 0.036	.854* ± 0.009
	.970 ± 0.021	.847 ± 0.00	.874 ± 0.00	.771 ± 0.00	.714* ± 0.00	.854 ± 0.011
HC	**.631 ± 0.004**	.605* ± 0.041	.605* ± 0.101	.614* ± 0.088	.619* ± 0.011	.608* ± 0.007
	.797 ± 0.021	.811 ± 0.00	**.813 ± 0.00**	.811 ± 0.00	.809 ± 0.00	.812 ± 0.011
Abalone	.164 ± 0.010	.132* ± 0.004	.132* ± 0.004	.078* ± 0.002	.168 ± 0.002	**.211 ± 0.004**
	.164 ± 0.017	.241 ± 0.00	.229 ± 0.00	.143 ± 0.00	.138 ± 0.00	**.237 ± 0.048**
Ecoli	**.870 ± 0.008**	.795* ± 0.064	.795* ± 0.084	.723* ± 0.023	.807* ± 0.004	.753* ± 0.015
	.872 ± 0.018	.827 ± 0.00	.823 ± 0.00	.778 ± 0.00	.838 ± 0.00	.795 ± 0.019
Glass	**.693 ± 0.005**	.598* ± 0.006	.598* ± 0.010	.620* ± 0.015	.623* ± 0.035	.549* ± 0.026
	.705 ± 0.014	.657 ± 0.00	.655 ± 0.00	.637 ± 0.00	.667 ± 0.00	.663 ± 0.020
Letters	**.964 ± 0.002**	.939* ± 0.012	.939* ± 0.017	.961 ± 0.000	.961 ± 0.011	.701* ± 0.004
	.967 ± 0.011	.965 ± 0.00	.939 ± 0.00	.965 ± 0.00	.967 ± 0.00	.952 ± 0.007
Letters Cons	**.990 ± 0.013**	.988 ± 0.012	.988 ± 0.011	.983 ± 0.005	.984 ± 0.013	.702* ± 0.004
	.991 ± 0.011	.987 ± 0.00	.998 ± 0.00	.983 ± 0.00	.987 ± 0.00	**.994 ± 0.007**
Nursery	.972 ± 0.004	.941* ± 0.004	.941* ± 0.001	.981 ± 0.004	**.986 ± 0.012**	.822* ± 0.004
	.975 ± 0.002	.932 ± 0.00	.944 ± 0.00	.985 ± 0.00	**.986 ± 0.00**	.957 ± 0.005

Table 5

Predicting Human Activity of WS

	Class Dist.	DRHC	SVM RBF (B)
R1 Sleep	24%	0.81	0.5
R2 Work	9%	0.90	0.64
R2 Watch TV	3%	0.59	0.42
R2 Eat	1%	0.54	0.39
R1 Bed Toilet Transition	0.05%	1.00	0.71
R1 Eat	1%	0.57	0.41
R1 Leave Home	1%	0.68	0.49
R1 Enter Home	1%	0.68	0.48
R1 Meal Preparation	2%	0.77	0.55
R2 Personal Hygiene	5%	0.73	0.52
R1 Work	12%	0.69	0.49
R2 Meal Preparation	2%	0.79	0.56
R2 Leave Home	0.05%	0.36	0.25
R2 Enter Home	0.05%	0.00	0.00
R1 Wander in Room	0.05%	0.42	0.30
R1 Personal Hygiene	5%	0.77	0.5
R2 Sleep	26%	0.95	0.68
R1 Bath	0.05%	0.72	0.51
R2 Bath	0.05%	0.59	0.42
R2 Bed Toilet Transition	0.05%	1.00	0.71
R1 Watch TV	5%	0.73	0.52



	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’

ble F1 scores within 5% deviation. On the 6 machine learning datasets, the difference is slightly more observable, especially on the last 3 datasets where DR can enhance the F1 scores over 20%; *i.e.,* Nursery. Again this is still due to the nature of the datasets – whether it is possible to identify effective prototypes.

5.3. DRHC and Sampling Techniques

Table 7 summarises our findings when comparing the DRHC data to classifiers based on the resampling techniques SMOTE, RENN and ENN, and ICC in Section 4.4. The same ensemble classifier, SVM RBF balanced, is used in each comparison. DRHC exhibits significantly superior performance in 36 of 40 comparisons ($p < 0.05$), and has inferior average F1 score in 4 of the remaining 4 instances. It demonstrates that

Table 6

DRHC compared to benchmark classifiers on DR augmented data(class-F1 vs. instance-F1 scores)

	DRHC	kNN	NB	SVM RBF	SVM RBF (B)
WS	**.720 ± 0.043**	.566* ± 0.007	.571* ± 0.009	.514* ± 0.004	.514* ± 0.004
	.830 ± 0.006	.806* ± 0.00	.701 ± 0.001	.730 ± 0.000	.754 ± 0.000
HA	**.885 ± 0.018**	.796* ± 0.016	.767* ± 0.049	.791* ± 0.051	.779* ± 0.047
	.886 ± 0.008	**.896 ± 0.006**	.865 ± 0.004	.801 ± 0.001	.811 ± 0.017
HB	**.926 ± 0.007**	.670* ± 0.030	.682* ± 0.001	.588* ± 0.028	.716* ± 0.020
	.970 ± 0.021	.670 ± 0.030	.682* ± 0.001	.588* ± 0.028	.716* ± 0.020
HC	**.631 ± 0.004**	.617* ± 0.051	.544* ± 0.081	.531* ± 0.033	.536* ± 0.022
	.797 ± 0.021	.707 ± 0.011	.654 ± 0.000	.713 ± 0.003	.736 ± 0.011
Abalone	**.164 ± 0.010**	.147* ± 0.003	.141* ± 0.009	.116* ± 0.003	.116* ± 0.002
	.164 ± 0.017	.157 ± 0.013	.164 ± 0.000	.136 ± 0.001	.118 ± 0.000
Ecoli	**.870 ± 0.008**	.766* ± 0.037	.756* ± 0.044	.102* ± 0.048	.126* ± 0.033
	.872 ± 0.018	.866 ± 0.003	.870 ± 0.004	.342 ± 0.011	.376 ± 0.012
Glass	**.693 ± 0.005**	.594* ± 0.032	.317* ± 0.021	.089* ± 0.021	.085*± 0.015
	.705 ± 0.014	**.710 ± 0.012**	.507 ± 0.002	.209 ± 0.011	.280 ± 0.005
Letters	**.964 ± 0.002**	.797* ± 0.006	.335* ± 0.009	.565* ± 0.011	.569* ± 0.017
	.967 ± 0.011	.960 ± 0.000	.575 ± 0.001	.585 ± 0.001	.691 ± 0.007
Letters Cons	**.990 ± 0.013**	.870* ± 0.001	.317* ± 0.006	.448* ± 0.015	.160* ± 0.014
	.991 ± 0.011	.970 ± 0.001	.527 ± 0.002	.548 ± 0.005	.370 ± 0.011
Nursery	**.972 ± 0.004**	.704* ± 0.001	.380* ± 0.010	.581* ± 0.001	.472* ± 0.002
	.975 ± 0.002	.901 ± 0.002	.587 ± 0.011	.718 ± 0.002	.602 ± 0.001

Table 7

DRHC compared to resampling techniques(class-F1 vs. instance-F1 scores)

	DRHC	SMOTE	RENN	ENN	ICC
WS	**.720 ± 0.043**	.648* ± 0.010	.565* ± 0.006	.557* ± 0.008	.246* ± 0.007
	830 ± 0.006	.764 ± 0.011	.575 ± 0.003	.575 ± 0.008	.463* ± 0.011
HA	.885 ± 0.018	.806* ± 0.049	.764* ± 0.033	.782* ± 0.021	**.991 ± 0.013**
	.886 ± 0.008	.890 ± 0.014	.786 ± 0.011	.798 ± 0.012	**.993 ± 0.011**
HB	.926 ± 0.007	.809* ± 0.002	.605* ± 0.038	.605* ± 0.031	**.949 ± 0.008**
	.970 ± 0.021	.889 ± 0.011	.605* ± 0.038	.605* ± 0.031	.949 ± 0.008
HC	.631 ± 0.004	**.692 ± 0.065**	.555* ± 0.047	.552* ± 0.075	.488* ± 0.048
	.797 ± 0.021	.769 ± 0.005	.587 ± 0.014	.572 ± 0.007	.678 ± 0.014
Abalone	.164 ± 0.010	**.199 ± 0.001**	.138* ± 0.003	.136* ± 0.001	.116* ± 0.003
	.164 ± 0.017	.219 ± 0.002	.164 ± 0.003	.138 ± 0.002	.131 ± 0.003
Ecoli	**.870 ± 0.008**	.470* ± 0.073	.578* ± 0.075	.592* ± 0.047	.466* ± 0.070
	.872 ± 0.018	.574 ± 0.013	.579 ± 0.015	.612 ± 0.017	.646 ± 0.007
Glass	**.693 ± 0.005**	.440* ± 0.029	.105* ± 0.008	.086* ± 0.033	.152* ± 0.055
	.705 ± 0.014	.615 ± 0.004	.225 ± 0.006	.106 ± 0.011	.272 ± 0.011
Letters	**.964 ± 0.002**	.511* ± 0.008	.373* ± 0.004	.378* ± 0.017	.923* ± 0.014
	.967 ± 0.011	.661 ± 0.008	.473 ± 0.007	.398 ± 0.007	**.968 ± 0.001**
Letters Cons	**.990 ± 0.013**	.622* ± 0.003	.647* ± 0.017	.647* ± 0.012	.800* ± 0.002
	.991 ± 0.011	.661 ± 0.007	.689 ± 0.007	.674 ± 0.005	.902* ± 0.011
Nursery	**.972 ± 0.004**	.659* ± 0.002	.413* ± 0.005	.415* ± 0.017	.682* ± 0.017
	.975 ± 0.002	.769 ± 0.001	.537 ± 0.009	.505 ± 0.007	.705 ± 0.011

sampling techniques alone will help balance the class distribution.

Comparing Table 7 and the last column in Table 4, we can see that SMOTE, RENN and ENN improve the F1 scores on most of the smart home datasets and some of the machine learning datasets. 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, therefore throwing away potentially useful information.

Still DRHC produces higher recognition accuracies than SMOTE. One reason might be that the sampling technique could generate potentially misleading information through oversampling the minority class [9]. 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.

It is worthy of noticing that ICC outperforms DRHC on HA and HB datasets but produces much lower F1 scores on WS, HC, and the other machine learning datasets. As mentioned in 2.4, ICC decomposes majority classes into smaller sub-classes before training the classifier, this process creates more decision boundaries to separate the different classes which increases the classification performance but eventually may lead to over-fitting. Table 8 compares F1 scores on DRHC and the sampling techniques on the HA dataset where the sub-classes boundaries created by the clustering process improve the classification performance.

Table 8

Predicting Human Activity of House A

	Class Dist.	DRHC	SMOTE	ENN	RNN	ICC
Leave Home	14%	0.94	0.89	0.86	0.84	1
Toilet	33%	0.91	0.86	0.84	0.82	1
Shower	10%	0.85	0.80	0.78	0.76	0.99
Sleep	15%	0.98	0.92	0.89	0.87	1
Breakfast	8%	0.67	0.63	0.61	0.60	0.78
Dinner	15%	0.70	0.66	0.64	0.63	0.82
Drink	4%	0.56	0.52	0.51	0.50	0.64

5.4. DRHC and Representation Learning Techniques

Table 9 summarises our investigations into whether the representation learning techniques PCA, t-SNE, Autoencoder, and NOC can be used to improve activity recognition when compared to our DRHC algorithm. DRHC exhibits significantly improved predictive performance in 39 of 40 comparisons ($p < 0.05$) and comparable performance in 1 of 40.

PCA performs the worst and their poor performance might indicate 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. This means that 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 [18], which suggests that PCA misses important nonlinear structures of the data.

t-SNE transforms the input features 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. For example, Figure 1b plots the most common misclassified activities of the WS dataset in a two-dimensional scatter plot, including ‘R1 Leave Home’, ‘R1 Enter Home’, ‘R2 Enter Home’, and ‘R2 Leave Home’.

Autoencoders have been widely used in speech recognition, image classification, and face recognition [19], where it has achieved promising results in compressing data by learning linear and nonlinear relationships between features. However, it is not able to differentiate activities with less distinctive patterns. We have tried to configure the autoencoder with different parameters, such as different numbers of layers, different numbers of neurons, and various optimisation functions. No set of parameters significantly improves the classification accuracy indicating that autoencoder fails in representing noisy data with few spare feature vectors. However, we only experiment a standard sparse autoencoder and with more sophsiticated autoencoders such as variational autoencoder [29] the performance can be improved. But this attempt is out of the scope of this paper, and we will look into it in the future.

Table 9

DRHC compared to representation learning techniques(class-F1 vs. instance-F1 scores)

	DRHC	PCA	t-SNE	Autoencoder	NOC
WS	**.720 ± 0.043**	.247* ± 0.002	.677* ± 0.017	.505* ± 0.001	.475* ± 0.018
	.830 ± 0.006	.557 ± 0.000	.686 ± 0.000	.586 ± 0.000	.485 ± 0.011
HA	**.885 ± 0.018**	.654* ± 0.005	.722* ± 0.012	.252* ± 0.012	.735* ± 0.023
	.886 ± 0.008	.679 ± 0.000	.797 ± 0.000	.297 ± 0.000	.837 ± 0.012
HB	**.926 ± 0.007**	.247* ± 0.018	.359* ± 0.018	.115* ± 0.021	.540* ± 0.036
	.970 ± 0.021	.283 ± 0.000	.453 ± 0.000	.153 ± 0.000	.597 ± 0.016
HC	**.631 ± 0.004**	.417* ± 0.004	.468* ± 0.058	.110* ± 0.017	.451* ± 0.011
	.797 ± 0.021	.734 ± 0.000	.514 ± 0.000	.114 ± 0.000	.455 ± 0.002
Abalone	**.164 ± 0.010**	.113* ± 0.002	.101* ± 0.003	.115* ± 0.003	.156 ± 0.007
	.164 ± 0.017	.172 ± 0.000	.150 ± 0.000	.150 ± 0.000	**.164 ± 0.001**
Ecoli	**.870 ± 0.008**	.102* ± 0.061	.835* ± 0.047	.102* ± 0.037	.575* ± 0.004
	.872 ± 0.018	.274 ± 0.000	.839 ± 0.000	.391 ± 0.000	.597 ± 0.011
Glass	**.693 ± 0.005**	.089* ± 0.006	.424* ± 0.005	.089* ± 0.015	.588* ± 0.035
	.705 ± 0.014	.089 ± 0.000	.455 ± 0.000	.355 ± 0.000	.591 ± 0.015
Letters	**.964 ± 0.002**	.107* ± 0.012	.230* ± 0.007	.199* ± 0.004	.913* ± 0.011
	.967 ± 0.011	.565 ± 0.000	.252 ± 0.000	.252 ± 0.000	.924 ± 0.012
Letters Cons	**.990 ± 0.013**	.447* ± 0.013	.447* ± 0.008	.199* ± 0.006	.874* ± 0.013
	.991 ± 0.011	.565 ± 0.000	.565 ± 0.000	.251 ± 0.000	.924 ± 0.013
Nursery	**.972 ± 0.004**	.350* ± 0.013	.532* ± 0.007	.122* ± 0.001	.528* ± 0.012
	.975 ± 0.002	.380 ± 0.000	.587 ± 0.000	.387 ± 0.000	.552 ± 0.011

5.5. Discussion

The specific number and sequence of sub-techniques used in DRHC is justified by empirical evaluation (Table 3). Failure to implement resampling at the final stage of the algorithm leads to statistically significantly inferior performance in all cases. The situation is less clear-cut at the initial prototyping stage of the algorithm. The replacement of clustering by hierarchical sampling (both based on misclassification of instances for data that have not yet been augmented with dissimilarity representations) led to improved F1 score performance for two of our ten datasets, and omitting the dissimilarity representation stage also led to increase average F1 score for two of ten datasets. These results may be due to statistical fluctuation in the underlying data, but it may be the case that practitioners seeking to identify subtle and minority activities from sensor data should consider these modifications to the algorithm as described in Figure 1 in the event of unsatisfactory initial performance.

When compared to commonly-used classifiers, DRHC is a significant improvement, irrespective of the type of data (sensor-based or unbalanced benchmark) and whether or not dissimilarity representation is performed as a pre-process (Table 4 and Table 6). These results (71 of 80 comparisons a significant improvement in predictive ability) provide empirical validation for DRHC, demonstrating that the new algorithm achieves its intended purpose.

After selection of the balanced SVM with radial basis as the underlying ensemble classifier, we also report strong empirical evidence that DRHC outperforms implementations that employ alternative methods of resampling and relabelling.

In our algorithm dissimilarity representations are derived to use when clustering misclassified instances from a baseline classifier. The effectiveness of dissimilarity representations depends on the selection of prototypes; *i.e.*, whether the prototypes are representative of the datasets. We have employed clustering algorithms to select the centroids of each cluster as prototypes, which can be affected by the amount of training data being used. If there exists small size of training data, then the prototypes might not be representative, which can impact on the dissimilarity representation.

A plausible alternative approach would be to use standard unsupervised representation learning techniques to form the clusters. Our experiments suggest that this is always an inferior alternative, leading to statistically significantly inferior predictive performance in 30 of 30 tests (with the same ensemble classifier being used for each comparison). This (i) indicates that dissimilarity representations are crucial to the performance of predictors of subtle and infrequent activities, and (ii) provides strong evidence for the utility of DRHC as a proven advantage over existing alternatives.

However, difference between dissimilarity representations and normal feature representations 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.

6. Conclusion and Future Work

In this paper, we present a new technique based on dissimilarity representation, which leverages a multi-phase, hierarchical ensemble in recognising minority and subtle activities. A sequence of comprehensive empirical evaluation and comparison demonstrates that (i) this is a challenging task where existing structure- and feature-based learning techniques do not perform well in general, and (ii) our DRHC algorithm constitutes a 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 subspace of data with less noise so that the classifier could learn from minority activities and hence reliably identify well-discriminating features.

The problem remains of how to best to separate prototypes when we have activities that fire the same set of sensors, with the difference in their sensor distribution being almost undetectably small. Such subtle differences cannot be captured by the baseline classifiers we have employed to date. Future investigations include the design and evaluation of prototype selection and distance metrics to better represent and separate subtle difference in these distributions. So far, we have only considered static 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 accelerometer data. These data would also contain subtle and minority activities, and we speculate that dissimilarity representation based classification of combined static and mobile data will be useful in the future accurate detection of important events in the ageing population. Also to demonstrate the generality of DRHC, we will further evaluate DRHC with other feature extraction techniques [1].

References

[1] Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*, 10:138 – 154, 2014.

[2] 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.

[3] 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.

[4] J. R. Bray and J. T. Curtis. An ordination of the upland forest communities of southern wisconsin. *Ecological Monographs*, 27(4):326–349, 1957.

[5] 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.

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

[7] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1):21–27, Sept. 2006.

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

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

[10] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, July 2012.

[11] J. Guo, Y. Mu, M. Xiong, Y. Liu, and J. Gu. Activity feature solving based on tf-idf for activity recognition in smart homes. *Complexity*, 2019.

[12] S. Guo, Y. Liu, R. Chen, X. Sun, and X. Wang. Improved smote algorithm to deal with imbalanced activity classes in smart homes. *Neural Processing Letters*, Oct 2018.

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

[14] K. Hornik. R FAQ, 2017.

[15] I. HÃijbener and K. David. Fesnoc: A novel feature selection algorithm based on niche overlapping coefficient. In *PerCom 2018*, pages 265 – 278, 2018.

[16] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4ÃćÂĂÂŞ37, 2000.

[17] I. Joliffe and B. Morgan. Principal component analysis and exploratory factor analysis. *Statistical Methods in Medical Research*, 1(1):69–95, 1992. PMID: 1341653.

[18] 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.

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

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

[21] F. J. O. n. Morales and D. Roggen. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *ISWC '16*, pages 92–99, 2016.

[22] A. More. Survey of resampling techniques for improving classification performance in unbalanced datasets. 2016.

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

[24] 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.

[25] N. Oukrich, A. Maach, E. Sabri, E. Mabrouk, and K. Bouchard. Activity recognition using back-propagation algorithm and minimum redundancy feature selection method. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 818–823, Oct 2016.

[26] J. P. Pedroso. Niche search: An evolutionary algorithm for global optimisation. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature — PPSN IV*, pages 430–440, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[27] E. Pekalska and R. P. Duin. *The Dissimilarity Representation for Pattern Recognition*. World Scientific, 2005.

[28] 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.

[29] 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.

[30] D. C. Roschelle Fritz. Identifying varying health states in smart home sensor data: An expert-guided approach. *World Multi-conference on Systems, Cybernetics and Informatics*, 2017.

[31] A. R. Sanabria, T. W. Kelsey, and J. Ye. Representation learning for minority and subtle activities in a smart home environment. In *Proceedings of PerCom '19*, 2019.

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

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

[34] T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *UbiComp '08*, pages 1–9, New York, NY, USA, 2008. ACM.

[35] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse. *Human Activity Recognition from Wireless Sensor Network Data: Benchmark and Software*, pages 165–186. Atlantis Press, Paris, 2011.

[36] C. Wang, Q. Hu, X. Wang, D. Chen, Y. Qian, and Z. Dong. Feature selection based on neighborhood discrimination index. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):2986–2999, July 2018.

[37] A. Woznica, A. Kalousis, and M. Hilario. Learning to combine distances for complex representations. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 1031–1038, New York, NY, USA, 2007. ACM.

[38] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang. Discriminative least squares regression for multiclass classification and feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 23(11):1738–1754, Nov 2012.

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

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

[41] J. Ye, G. Stevenson, and S. Dobson. Detecting abnormal events on binary sensors in smart home environments. *Pervasive and Mobile Computing*, 33:32 – 49, 2016.

[42] G. Zhong, L.-N. Wang, X. Ling, and J. Dong. An overview on data representation learning: From traditional feature learning to recent deep learning. *The Journal of Finance and Data Science*, 2(4):265 – 278, 2016.