This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2855159, IEEE Transactions on Knowledge and Data Engineering

A Novel Feature Incremental Learning Method for Sensor-based Activity Recognition

Chunyu Hu, Yiqiang Chen, Xiaohui Peng, Han Yu, Chenlong Gao and Lisha Hu

Abstract—Recognizing activities of daily living is an important research topic for health monitoring and elderly care. However, most existing activity recognition models only work with static and pre-defined sensor configurations. Enabling an existing activity recognition model to adapt to the emergence of new sensors in a dynamic environment is a significant challenge. In this paper, we propose a novel feature incremental learning method, namely the Feature Incremental Random Forest (FIRF), to improve the performance of an existing model when a small amount of data on newly appeared features. It consists of two important components – 1) a mutual information based diversity generation strategy (MIDGS) and 2) a feature incremental tree growing mechanism (FITGM). MIDGS enhances the internal diversity of random forests, while FITGM improves the accuracy of individual decision trees. To evaluate the performance of FIRF, we conduct extensive experiments on three well-known public datasets for activity recognition. Experimental results demonstrate that FIRF is significantly more accurate and efficient compared with other state-of-the-art methods. It has the potential to allow the dynamic exploitation of new sensors in changing environments.

Index Terms—feature incremental learning, activity recognition, random forest.

1 Introduction

Studies have shown that the ability to perform activities of daily living (ADL) is an important indicator of a person's health [1]. In [2], it was found that cognition is closely associated with gait in complex ways. Changes in ADL are important indicators for small vessel disease and diseases in the white matter in the brain [3], [4]. According to these findings, activity recognition based health monitoring is of great significance for providing early warning of chronic diseases and improving people's quality of life. Real-time and accurate activity recognition is an important challenge for ADL monitoring. The emergence of wearable sensors provides an alternative way for ADL monitoring. With small sizes and low power consumption, wearable sensors show great promise in activity recognition.

In [5], experiments show that the emergence of a new dimension of sensory perception entails female mice a new capability of chromatic discrimination and enhanced long-wavelength sensitivity. Similar observations are drawn from applications of activity recognition. With the help of more diverse sensors, it is possible to achieve a better activity recognition performance [6], [7]. In [6], Tobias et al. enhanced accelerometer-based activity recognition with capacitive proximity sensing. Their work significantly improves the ADL recognition performance. In [7], different modality sensors, consisting of a depth camera and an inertial body sensor, are combined to recognize human actions. Experimental results show that the fusion approach led to a 2% to 23% improvement in recognition rate.

- Chunyu Hu, Yiqiang Chen, Xiaohui Peng, Chenlong Gao and Lisha Hu are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China; University of Chinese Academy of Sciences, Beijing, China; and Beijing Key Laboratory of Mobile Computing and Pervasive Device, Beijing, China (Corresponding author: Yiqiang Chen, e-mail: yqchen@ict.ac.cn).
- Han Yu (e-mail: han.yu@ntu.edu.sg) is with the School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore; Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY), Singapore; and Alibaba-NTU Singapore Joint Research Institute, Singapore.

Manuscript received January, 2018; revised March, 2018.

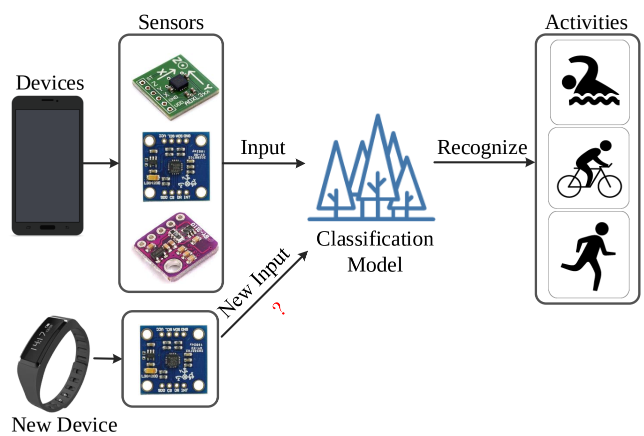


Fig. 1. Illustration of feature incremental learning scenario.

Traditional sensor-based activity recognition methods train fixed classification models with labeled data collected off-line. Such models are unable to adapt to dynamic changes in real applications. With the emerging of new wearable devices, more diverse sensors can be used to improve the performance of activity recognition. However, it is difficult to integrate a new sensor into a pre-trained activity recognition model. The emergence of new sensors will lead to a corresponding increase in the feature dimensionality of the input data, which may result in the failure of a pre-trained activity recognition model. Take Figure 1 as an example. A user normally uses his smart phone to record his ADL. When he acquires a new wristband, data from sensors embedded in the new wristband are available and provide a new dimension of data to help recognize his activities. However, the existing activity recognition model on his smart phone cannot recognize the data collected by his new wristband. The pre-trained activity recognition model is unable to take advantage of this new source of data. Similar problems can also be observed in smart home environments. In a smart home, sensors are often deployed in an incremental manner. A static or fixed activity recognition model fails to make full use of sensors deployed in this way, while retraining of the model is often time-consuming.

To take advantage of data generated by new sensors, traditional methods need to retrain a new model from scratch with newly acquired data. Knowledge stored in the existing model based on old data is abandoned. As we know, data annotation is a time-consuming and laborious task. Retraining a new model will result in a waste of both time and manpower. How to adapt an existing activity recognition model to the emergence of new sensors with minimal time and cost is a difficult challenge.

In this paper, we propose a novel supervised feature incremental activity recognition method, namely the Feature Incremental Random Forests (FIRF), which is able to improve the accuracy of a pre-trained model when a small amount of labeled data about new features arrive. This method first employs a novel Mutual Information based Diversity Generation Strategy (MIDGS) to select individual trees of the random forests which become ineffective with the emergence of new sensors, and then uses a new Feature Incremental Tree Growing Mechanism (FITGM) to update the selected individual trees. Under FIRF, FITGM improves the accuracy of individual learners in the ensemble, while MIDGS supports diversity among individual trees to improve the overall performance of an ensemble.

FIRF is designed to process signals collected from inertial sensors. Thus, we conduct extensive experiments on three inertial sensor based public activity datasets (i.e. Daily and Sports Activities Data Set (DSADS) [8], Physical Activity Monitoring Data Set (PAMAP2) [9]) and RealWorld(HAR) [10]) to evaluate the performance of FIRF. Experimental results show that FIRF is able to improve the recognition accuracy of an existing model using a small amount of data about new features. Compared with retraining a new recognition model from scratch, FIRF can exploit both the knowledge of the existing model and the newly arrived data with less training time. Besides, FIRF is able to gain better test accuracy than other state-of-the-art feature incremental learning methods.

The rest of this paper is organized as follows. In Section 2, we introduce the formulation of feature incremental activity recognition problem and the related existing work. In Section 3, we provide a brief introduction to random forest, which is the foundation of our proposed method. In Section 4, we describe the proposed FIRF method in detail. In Section 5, we conduct extensive experiments to evaluate the performance of FIRF and discuss the experimental results. Conclusions and future work are presented in Section 6.

2 Problem Formulation and Related Work

2.1 Problem Formulation

The feature incremental activity recognition problem can be formalized as: given n labeled instances $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iK}) \in \mathbb{R}^K$ is a K-dimensional feature vector and $y_i \in Y$ is the label of the ith sample, the activity recognition task in a static sensor configuration is to learn a function $f \colon \mathbb{R}^K \to Y$. With new sensors introduced, newly acquired data $D_1 = \{(\boldsymbol{x'}_{n+1}, y_{n+1}), (\boldsymbol{x'}_{n+2}, y_{n+2}), \ldots, (\boldsymbol{x'}_{n+m}, y_{n+m})\}$, $\boldsymbol{x'}_i = (x_{i1}, x_{i2}, \ldots, x_{iK}, x_{iK+1}, \ldots, x_{iK+K'}) \in \mathbb{R}^{K+K'}$ and $\mathbb{R}^K \in \mathbb{R}^{K+K'}$, where K' is the number of features corresponding to new sensors, feature incremental activity recognition is to learn the function $f_1 \colon \mathbb{R}^{K+K'} \to Y$.

2.2 Activity Recognition

With great significance in health monitoring, activity recognition has attracted much research attention [11], [12], [13]. Many machine learning methods, such as support vector machines (SVMs) [14], [15] and decision trees [16], [17], have been applied to recognize human activities through sensors [18], [19].

Compared with other machine learning methods, SVM has a distinct advantage in classification accuracy [14], [15], [20]. As a basic implementation of SVM is limited to making binary classifications, it cannot be used for multi-class activity recognition directly. Some variants of SVM [21], [22] have been introduced to tackle this issue. Decision trees are a simple but effective machine learning method. It can map sensor readings to corresponding activity labels with a hierarchical tree structure, which is human-understandable. In [23], Phan proposed a decision tree based method to mitigate misclassification in classifying previously-unseen real-world activities. In [17], the TransEMDT (Transfer learning EMbedded Decision Tree) algorithm was introduced to solve the cross-people activity recognition problem. Single layer feed-forward networks (SLFNs) [24] have also been used for activity recognition. The extreme learning machine (ELM), as a special variant of SLFN, is a unified solution for binary classification, multi-class classification, and regression problems. It performs well in terms of both accuracy and training time. In [25], Hu et al. proposed an activity recognition model, namely b-COELM, to meet the low-computational-complexity, lightweight and high accuracy requirements of wearable devices. In [26], Wang et al. proposed a fast and robust activity recognition model called TransM-RKELM (Transfer learning mixed and reduced kernel Extreme Learning Machine) which can adapt to new sensor location information efficiently and achieve good accuracy.

The aforementioned work is effective for activity recognition in a static or predefined sensor configuration. However, when new sensors emerge, these models cannot adapt. They are, thus, unable to make full use of the newly available sensors to improve the activity recognition performance. To exploit information from newly available sensors, model retraining is required, which consumes time and results in losing the knowledge stored in the previous models.

2.3 Transfer Learning

Transfer learning is an useful technique to allow activity recognition models to be reuse when new sensors are added into the system [27]. In general, the target domain of transfer learning is different from the source domain. Take [28] for example, knowledge is transferred between domains with different features, while [17] transfers knowledge between domains with different data distributions.

Transfer learning tries to transfer the knowledge embedded in a model learned based on data from the source domain such that it can work in a different target domain [29]. In [30], Zheng et al. designed a bridge between the activities in two domains by learning similarities from the label information. Their method is able to transfer the available labeled data from a set of existing activities in one domain to help recognize the activities in another different but related domain. In [31], Feuz et al. proposed a novel Feature-Space Remapping (FSR) technique, which is able to transfer knowledge between domains with different feature spaces. With an activity recognition model based on a particular home, FSR can enable it to perform well in a differently configured home. In [17], Zhao et al. proposed an algorithm known as TransEMDT. It can be used for personalized activity recognition model adaptation. In [28], a personalized activity ecosystem was introduced to transfer learned activity information between sensor platforms in real time.

Opportunistic learning can be viewed as a special case of heterogeneous domain adaptation [29], [32]. In feature incremental activity recognition, the source domain happens to be a subset of the target domain since $\mathbb{R}^K \subset \mathbb{R}^{K+K'}$. Heterogeneous online transfer learning (OTL) [33] attempts to handle a similar problem in which the feature space of the source domain is a subset of that of the target domain. Nevertheless, this technique can only handle binary classification problems.

2.4 Incremental Learning

Incremental learning is an effective solution to improve the performance of an existing model with new data. To handle changes in the distribution of input data, [34], [35], [36], [37] proposed four variants of online random forest algorithms. In [38], ORF-Saffari is combined with active learning to construct a cross-subject activity recognition model. Experimental results show that it is efficient in building personalized activity recognition classifiers. These methods focus on the dynamic changes of data distribution. They aim to refine the function $f : \mathbb{R}^K \to Y$ with new data from previously known features. Different from them, our problem involves dynamic changes not only in data distributions, but also in input dimensions.

To recognize new activities without retraining from scratch, [39] proposed the CIELM class incremental learning method. With CIELM, new activities can be recognized dynamically. In [40], Hu et al. designed a separating axis theorem based splitting strategy to insert internal nodes. Combined with splitting leaf nodes, their incremental learning method can match the performance of random forest. Ristin et al. proposed two variants of the random forest model to avoid retraining from scratch [41] when handling the class incremental learning problem. It can extend random forest initially trained with just 10 classes to 1,000 classes with an acceptable loss of accuracy. These work attempts to learn the function $f : \mathbb{R}^K \to Y'$ by focusing on handling changes in the output space.

To exploit all available sensor data in a dynamic scenario, [42] proposed a feature incremental and decremental learning method, namely FA-OSELM, for Wifi-based indoor localization. The performance of FA-OSELM depends on the parameter settings, which may result in drastic fluctuations of performance. Similar to FA-OSELM, Hou and Zhou proposed the One-Pass Incremental and Decremental learning approach (OPID) to adapt to evolving features and instances simultaneously [43]. Xing et al. [44] proposed a Perception Evolution Network (PEN) which can integrate the new sensor inputs into the learned model. Nevertheless, the influence of the order of addition of the sensors into the system is not taken into account. Both [45] and [46] proposed methods that can incorporate new sensors into an existing activity recognition system in an unsupervised manner. In [46], the clustering membership of the higher dimensional data and heuristics were utilized to label newly arriving data. In [45], similarity and transduction were used to label such new data. Similarly, Gregory et al. [47] proposed a self-supervised learning method - ARTMAP. It is able to learn features from unlabeled patterns without losing knowledge previously acquired from labeled patterns. However, the spreading of labels to newly arriving data in an unsupervised manner will further affect the classifier. Thus, [46], [45] and [47] are not able to handle the continuous emergence of new sensors.

3 Preliminaries

Random forest [48] is an extension of the decision tree ensemble. It has been demonstrated to be advantageous in machine learning tasks such as classification, semantic segmentation, and so on [34]. Feature randomization and bootstrap sampling are combined to achieve good generalization performance. Information Gain [49] and Gini index [50] are the most commonly used splitting criteria. The output of the random forest is a combination of individual decision trees using different combination rules, such as simple averaging [51], weighted averaging [52], majority voting [53] and weighted voting [54].

Assume a random forest R consists of M decision trees $h_1, h_2, \ldots, h_M$. The tree bagging mechanism samples n' instances from the training set D with replacement to form a new training set D_i for the construction of the ith decision tree. For each split in the decision tree, a random forest randomly selects k features to build a candidate feature set, and then chooses the optimal feature from this candidate feature set as the split feature. The output of the random forest R is computed as:

$$H(\boldsymbol{x}) = \arg\max_{y \in Y} p(y|\boldsymbol{x}) \tag{1}$$

where $H(\boldsymbol{x})$ is the prediction of the random forest.

4 Method

In this paper, we propose a novel feature incremental activity recognition method, which is named FIRF. It is able to adapt an existing activity recognition model to newly available sensors in a dynamic environment. Different from [45] and [46], FIRF is a supervised learning method. The data used for both initial training and incremental training are labeled. In a real application, when the sensor configuration changes, we can carry out the same activity for a period of time and then label these data later. This process can be repeated for each activity to collect training data for incremental learning. There are also other ways to label incremental data effectively. However, this is not the focus of this paper. We aim to provide a supervised model updating method for feature incremental scenarios. Figure 2 gives an overview of the method.

With an initial sensor configuration, we can construct a random forest-based activity recognition model. This step is commonly conducted with data collected offline. When new sensors emerge, FIRF is adopted to improve the performance of the existing model. In FIRF, we present two new strategies: 1) MIDGS which encourages diversity among individual decision trees in the incremental learning phase by identifying the individual learners that have high redundancy with the other individual learners and low recognition accuracy; and 2) FITGM which improve the performance of these identified individual decision trees with new data collected from both existing and newly emerging sensors.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2855159, IEEE Transactions on Knowledge and Data Engineering

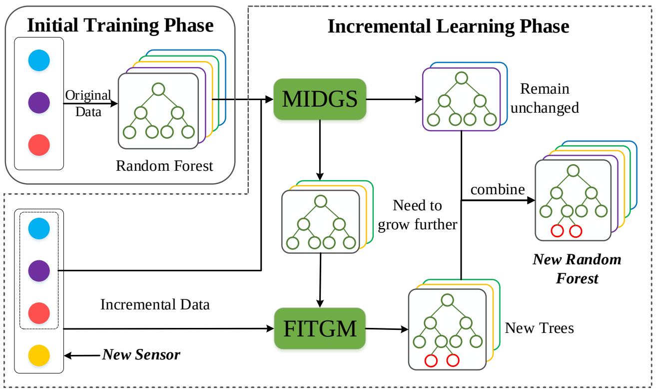


Fig. 2. Feature incremental random forest algorithm. With the emergence of new sensors, FIRF updates an existing random forest model with MIDGS and FITGM. MIDGS is used to select a proportion of decision trees to grow further with FITGM. (*Note:* frames with different colors in the random forest represent different individual trees.)

4.1 Motivation

In [55], Krogh and Vedelsby proposed the *error-ambiguity decomposition* rule. It is formulated as:

$$Err = \overline{err} - \overline{A} \tag{2}$$

where Err and $\overline{err}$ are the error of the ensemble classifier and the weighted average of the individual generalization errors, respectively. $\overline{A}$ is the weighted average of the individual ambiguities.

Error-ambiguity decomposition rule proves that high accuracy of individual learners and high diversity among the predictions of individual learners will contribute to good generalization performance of the ensemble learner.

Based on this conclusion, we aim to find a trade-off between the average individual generalization errors and the average individual ambiguities in the incremental learning phase. To reduce the value of $\overline{err}$ in equation (2), we leverage new data collected from both existing and newly emerging sensors to update existing individual trees in a given random forest model. We aim to improve the performance of individual trees by exploiting all available and useful information while avoiding the negative effect of non-useful sensors. To achieve this goal, we propose a novel tree growing mechanism, namely the FITGM. On the other hand, we also aim to increase the value of $\overline{A}$ in equation (2) in the incremental learning phase. To achieve this goal, we propose a novel diversity generation mechanism, namely the MIDGS, which updates only some of individual trees in the random forest with newly acquired data and keep the rest unchanged. This operation increases the diversity among the decision trees by training them with different training data.

4.2 MIDGS

According to the *error-ambiguity decomposition* rule [55], diversity is important for the performance of ensemble classifiers. A high diversity among individual learners will contribute towards better generalization performance. Thus, we take diversity into consideration in the feature incremental learning phase.

One of the popular diversity generation mechanisms is by manipulating the data samples [56]. The proposed MIDGS method belongs to this category. The basic idea of MIDGS is to build individual learners with different training data in the incremental learning phase. We select individual learners with lower diversity scores to update with batches of incremental data. Compared with updating the whole random forest, MIDGS improves diversity from the view of data samples which contributes towards improved accuracy. In addition, only updating part of the random forest also reduces latency time.

In the process of diversity generation, the most important challenge is to quantify the diversity of individual learners. In this section, we propose a new diversity scoring rule—the Mutual Information based Diversity Scoring (MIDS) rule. It jointly reflects both the accuracy and the redundancy of individual classifiers.

In information theory, the concept of mutual information, $I(X_1, X_2)$, [57] measures the inter-dependence between two variables.

$$I(X_1, X_2) = \sum_{x_1, x_2} p(x_1, x_2) \log \frac{p(x_1, x_2)}{p(x_1)p(x_2)}. \tag{3}$$

More specifically, $I(X_1, X_2)$ measures the amount of information obtained about X_1 through X_2. If X_1 is independent of X_2, the value of $I(X_1, X_2)$ is zero. A large value of $I(X_1, X_2)$ represents a strong dependency between X_1 and X_2. We can take advantage of mutual information to measure the diversity among individual learners. The MIDS rule is formulated as:

$$S(h_i) = I(h_i, y) - \frac{1}{M-1} \sum_{k \neq i} I(h_i, h_k). \tag{4}$$

The first term, $I(h_i, y)$, is the mutual information between the output of the ith individual learner and the target. It is used to measure the relevancy between the output of the ith decision tree and the target label. A larger value of $I(h_i, y)$ means that h_i achieves high accuracy. The second term, $\frac{1}{M-1} \sum_{k \neq i} I(h_i, h_k)$, is the average mutual information between the output of the ith decision tree and the other decision trees in the random forest. It is used to calculate the averaged redundancy between the ith decision tree and the other decision trees. A larger value of the second term indicates a stronger correlation between the ith individual learner and the other individual learners. Thus, a smaller value for the second term is desirable.

Equation (4) is an initial form of MIDS rule. However, in this form, the range of S varies with the dataset used, which limits its application. In order to address this limitation, we first investigate the range of each item in equation (4). Theorem 1 provides a bound of the value range of $I(X_1, X_2)$ as follows.

Theorem 1. For any two variables X_1, X_2,

$$0 \leq I(X_1, X_2) \leq \log |\chi| \tag{5}$$

with equality if and only if X_1 and X_2 are independent and identically distributed (i.i.d.) over χ.

The proof of Theorem 1 is presented in Appendix A. With theorem 1, we obtain:

$$0 \leq I(h_i, y) \leq \log |y|, \tag{6}$$

and

$$0 \leq I(h_i, h_k) \leq \log |y|, \tag{7}$$

where $|y|$ denotes the number of possible values of y.

Substituting (6) and (7) into (4), we obtain the ranges of $S(h_i)$:

$$-\log |y| \leq S(h_i) \leq \log |y|. \tag{8}$$

According to equation (8), the range of $S(h_i)$ depends to the number of possible values of the target. To eliminate the differences in the range of $S(h_i)$ as a result of the different datasets used, we normalize the range of $S(h_i)$ into $[-1, 1]$ as:

$$S(h_i) = \frac{1}{\log|y|}\left[I(h_i, y) - \frac{1}{M-1}\sum_{k\neq i} I(h_i, h_k)\right]. \quad (9)$$

As we can see from equation (9), a larger value of $S(h_i)$ indicates that the ith decision tree has high prediction accuracy and low redundancy with respect to other individual classifiers. Based on the MIDS rule, we further propose the MIDGS method, which encourages diversity while minimizing the sacrifice in accuracy. The main process of MIDGS is as follow:

(1) Compute the predictions based on the feature incremental data $\hat{Y} = \{\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, \ldots, \hat{\boldsymbol{y}}_M\}$ with the initial random forest model $R = \{h_1, h_2, \ldots, h_M\}$, where $\hat{\boldsymbol{y}}_i$ denotes the prediction obtained by the ith decision tree.
(2) Compute the diversity score $S(h_i)$ of each decision tree h_i using equation (9).
(3) Sort the values of $S(h_i)$ in descending order.
(4) Select the decision trees with small values of $S(h_i)$ and update them with the feature incremental data.
(5) The updated decision trees and the unchanged decision trees together form a new random forest $R^{'}$.

To maximize the ensemble diversity, it is important to determine an appropriate fraction of the decision trees selected to grow further in Step (4). There are two methods to determine the fraction of decision trees to update. One way is to define a specific number of decision trees to update. This method can be affected by the size of the random forest. Another way is by sorting the value of $S(h_i)$ and selecting a certain proportion p of decision trees with low $S(h_i)$ values to grow further. In Section 5.4, we conduct a group of experiments to evaluate the effect of the parameter p.

4.3 FITGM

According to equation (2), more accurate individual classifiers will contribute to better overall performance of the ensemble. In this section, we introduce a novel FITGM to improve individual classifier accuracy. It updates individual decision trees with new data obtained from both existing sensors and newly emerging sensors. Old data points are not required when updating individual decision trees. With new data collected from existing sensors, FITGM grows the decision trees further by splitting the leaf nodes, which is similar to [34]. With new data acquired from newly emerging sensors, FITGM modifies the subtree to take advantage of useful information contributed by these new sensors. Figure 3 illustrates the two modes of operation of FITGM.

4.3.1 Modification of a subtree

With newly emerging sensors, new features need to be extracted. When a new feature can achieve better score (i.e. information gain or Gini index) on a decision node, we discard the current split and its child nodes. A new subtree of this node is retrained with this new feature as the split attribute (Figure 3(b)). Modifications of a subtree will take advantage of new features to improve the performance of an existing decision tree. However, it may result in loss of information stored in an existing tree structure and additional training time. In order to address these problems, we propose two constraints to control the modification of subtrees: 1)

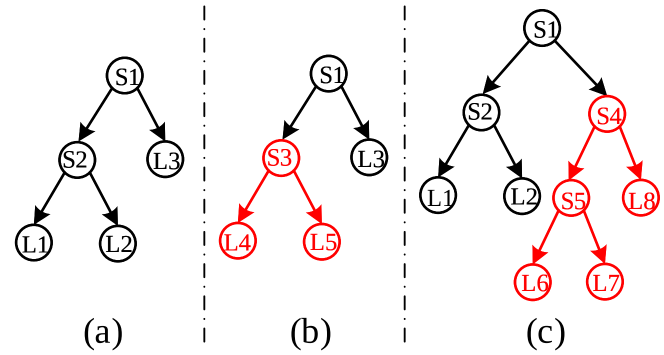


Fig. 3. The feature incremental tree growing mechanism: (a) Random forest learned in the off-line phase; (b) Modification of a subtree; (c) Incremental growing on a leaf node.

the number of samples, n_1, with new features must satisfy either equation (10) or equation (11):

$$n_1 \geq \frac{n_{INC}}{2^{d-1}}, \quad (10)$$

where n_{INC} is the number of samples in an incremental data block, and d is the node depth (assuming the node depth of the root is 1 and its child is 2).

$$n_1 \geq n_{node} \quad (11)$$

where n_{node} is the number of samples belonging to this node previously; and 2) the score of a new feature is larger than the score of the current split (i.e. $score(f_{new}) > score_{node}$). Both constraints are used to ensure that the benefit of modification of subtrees outweighs the loss of knowledge stored in existing subtrees.

4.3.2 Split on a leaf node

In the feature incremental learning scenario, data distribution affects when new data will arrive. To improve the performance when the tree grows, we adopt a similar tree growing strategy as that proposed in [34]. A leaf node will split once when the following conditions are both met. Firstly, the number of samples accumulated in a leaf exceeds a threshold. Secondly, the samples in a leaf belong to different classes. When these conditions occur, we choose an attribute with the best splitting score to split the leaf (Figure 3(c)).

FITGM is decribed in Algorithm 1. The *UpdateStatics()* function is used to record the number of instances belonging to this nodes.

4.4 Feature Incremental Random Forest

With MIDGS and FITGM, we are able to improve both the diversity and the accuracy of individual decision trees through feature incremental learning. As shown in Figure 2, we adopt a two-stage incremental learning strategy. In the first stage, samples collected offline are used to construct an initial random forest model. In this stage, the initial random forest model is trained with data from sensors installed beforehand. When new sensors emerge, their readings cannot be utilized by the initial random forest model. Thus, we propose the FIRF to exploit the newly emerging sensors. To handle the challenge of feature incremental data, FIRF chooses some individual trees using MIDGS to grow further with new sensor data, while the rest of decision trees

Algorithm 1 FITGM

Input: incremental training set with new features $D_1 = \{(\boldsymbol{x}_i^1, y_i^1)|i = 1, 2, , n_1\}$; $\boldsymbol{x}_i^1 = (\boldsymbol{x}_{iold}^1, \boldsymbol{x}_{inew}^1)$; $\boldsymbol{x}_{iold}^1 = (x_{i1}^1, x_{i2}^1, \ldots, x_{iK}^1)$, $\boldsymbol{x}_{inew}^1 = (x_{iK+1}^1, \ldots, x_{iK+K'}^1)$; decision tree T.

Output: decision tree $T^{'}$;

if T is an internal node **then**
  if $n_1 \geq \frac{n_{INC}}{2^{d-1}}$ or $n_1 \geq n_{node}$ **then**
    maxScore=computeScore($\boldsymbol{x}_{new}^1$)
    if $maxScore > T.score$ **then**
      Reconstruct subtree(T,D_1);
    else
      UpdateStatics(T,D_1);
    end if
  else
    UpdateStatics(T,D_1);
  end if
else if isNotPure(T) and $n_1 \geq threshold$ **then**
  splitLeaf(T,D_1)
end if

remain unchanged. This operation increases the diversities of individual learners. The selected decision trees are grown using FITGM to improve the classification accuracy. The details of FIRF are described in Algorithm 2:

Algorithm 2 FIRF

Input: incremental training set with new features $D_1 = \{(\boldsymbol{x}_i^1, y_i^1)|i = 1, 2, , n_1\}$; $\boldsymbol{x}_i^1 = (\boldsymbol{x}_{iold}^1, \boldsymbol{x}_{inew}^1)$; $\boldsymbol{x}_{iold}^1 = (x_{i1}^1, x_{i2}^1, \ldots, x_{iK}^1)$, $\boldsymbol{x}_{inew}^1 = (x_{iK+1}^1, \ldots, x_{iK+K'}^1)$; random forest $R = \{T_i, i = 1, 2, \ldots, M\}$; update proportion rate δ.

Output: updated random forest $R^{'} = \{T_i^{'}, i = 1, 2, \ldots, M\}$.

for $i = 1 : M$ **do**
  $\hat{\boldsymbol{y}}_i$=predict(T_i,D_1);
end for
for $i = 1 : M$ **do**
  $S(\hat{\boldsymbol{y}}_i) = \frac{1}{\log|\boldsymbol{y}|}(I(\hat{\boldsymbol{y}}_i, \boldsymbol{y}) - \frac{1}{M-1}\sum_{k \neq i} I(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_k))$
end for
$[S, index]$ =sort(S);
for $i = 1 : \delta * M$ **do**
  T'_{index_i} =FITGM(D_1, T_{index_i});
end for
for $i = \delta * M + 1 : M$ **do**
  $T'_{index_i} = T_{index_i}$
end for

5 EXPERIMENTAL EVALUATION

In this section, we conduct extensive experiments to validate the performance of the proposed method. All experiments are conducted on a Dell OptiPlex 9020 (Intel Core i7-4790/24GB DDR3) desktop computer with Matlab R2014a.

5.1 Comparison Methods

Random forest (RF) [48], Extra-Trees (ET) [58] and online random forest (ORF-Saffari) [34] are adopted to evaluate the performance of FIRF in recognizing continuous newly emerging features. Another four feature incremental learning methods, including Feature Adaptive Online Sequential Extreme Learning machine (FA-OSELM) [42], One-Pass Incremental and Decremental learning approach (OPID) [43], heterogeneous Online Transfer Learning (heterogeneous OTL) [33], and self-supervised ARTMAP (SS ARTMAP) [47], are used to evaluate the performance of FIRF in feature incremental activity recognition. Brief introductions of these methods are listed as follows.

1) *Random forest (RF)* [48]: It was first proposed by in 2001. We gave a brief introduction to RF in Section 3.
2) *Extra-Trees (ET)* [58]: It is also a classical tree-based ensemble method. Different form RF, ET splits a tree node with both random attributes and random cut-point choices. It has high computational efficiency.
3) *Online random forest (ORF-Saffari)* [34]: It is a variant of ET which can handle online data. It combines online bagging with online random decision trees.
4) *Feature Adaptive Online Sequential Extreme Learning Machine (FA-OSELM)* [42]: It can transfer the original model to a new one with a small number of data with new features.
5) *One-Pass Incremental and Decremental learning approach (OPID)* [43]: It learns with incremental and decremental features in one shot.
6) *Heterogeneous Online Transfer Learning (heterogeneous OTL)* [33]: It tackles data with different feature representations. In this method, the feature space of the target domain is different from that of the source domain.
7) *self-supervised ARTMAP* (SS ARTMAP) [47]: Self-supervised ARTMAP is able to integrate knowledge from labeled patterns with some features, knowledge from unlabeled patterns with more features, and knowledge form self-labeled patterns.

5.2 Evaluation Datasets

We adopt three public activity datasets to evaluate the performance of FIRF. They are DSADS [8], PAMAP2 [9] and RealWorld(HAR) [10]. The details of these three datasets are listed in Table 1.

1) *DSADS* consists of 19 daily activities and sports activities performed by 8 subjects (4 males and 4 females). Three types of sensors (i.e. accelerometers, gyroscopes and magnetometers) were placed on five different body parts. Each subject performed the activities according to his/her own style.
2) *PAMAP2* comprises 18 different physical activities performed by 9 subjects wearing 3 inertial measurement units (IMU) and a heart rate monitor each. As there are many missing values in the data collected by the heart rate monitor, we only use the data from IMU in the experiments. Each IMU consists of two accelerometers, one magnetometer and one gyroscope. Thus, readings collected from 12 sensors are used in the experiments.
3) *RealWorld(HAR)* includes 8 physical activities performed 15 subjects. For each activity, the acceleration of seven body positions were recorded simultaneously. We adopt the acceleration of chest, forearm, shin, thigh, upper arm and waist in the experiments.

5.2.1 Feature Extraction

For RealWorld(HAR), we extracted 43 features following the feature extraction method presented in [38]. For DSADS and PAMAP2, we firstly synthesize the readings of the 3 axes from one

TABLE 1
Details of the three public activity datasets

Dataset	Samples	Sensors	Classes	Users
DSADS	9,120	15	19	8
PAMAP2	7,352	12	12	9
RealWorld(HAR)	115,947	6	8	15

sensor into a synthetic value using formula $a = \sqrt{x^2 + y^2 + z^2}$, which eliminates the impact caused by the difference in orientation among the sensors. Then, the sliding window technique was used to segment the sensor readings. In our experiments, we set the window size to 5 seconds [59] and 5.12 seconds [9] for DSADS and PAMAP2, respectively with 50% overlap between consecutive windows. For both DSADS and PAMAP2, we followed [60] to extract 27 features (Table 2) from each sensor.

TABLE 2
Features extracted per sensor from DSADS and PAMAP2

ID	Feature	Description
1	Mean	Average value of samples in window
2	STD	Standard deviation
3	Minimum	Minimum
4	Maximum	Maximum
5	Mode	The value with the largest frequency
6	Range	Maximum minus minimum
7	Mean crossing rate	Rate of times signal crossing the mean value
8	dc	Direct component
9-13	Spectrum peak position	First 5 peaks after FFT
14-18	Frequency	Frequencies corresponding to the 5 peaks
19	Energy	Square of norm
20-23	Four shape features	Mean, STD, Skewness, Kurtosis
24-27	Four amplitude features	Mean, STD, Skewness, Kurtosis

5.3 Experimental Settings

In our experiments, we divide the whole dataset into two equal sized subsets: a training set and a testing set. To mitigate the impact of sensor sequence, we randomly sample five sensor arriving orders and take the mean value as the final result.

5.3.1 Recognizing continuous newly emerging features

To evaluate the performance of FIRF in recognizing continuous newly emerging features, we add the features extracted from new sensor at different time steps. Thus, we further divide the training set into equal sized subsets according to the number of sensors involved. The initial model is trained with features extracted from only one sensor. Then, we add new sensors one by one in subsequent time steps. During each update, data with old features and features from a previously unseen sensor are used for training. Table 3 contains the detailed experimental settings for recognizing continuous newly emerging features.

For RF and ET, only new data blocks are used to retrain a new model when a new sensor emerges, while ORF-Saffari updates its model with the new data that corresponds to the initial sensors. FIRF updates with new data from both old sensors and new sensors. We set the number of trees $M = 100$, which is the same as most of the related work in random forest [48]. The number of features randomly selected at each node is set as $\lfloor \sqrt{K-1} \rfloor$, where K is the number of features. For ORF-Saffari, we implemented it in Matlab and set $num_{epochs} = 10$ (number of passes through the training data). The other parameters of ORF-Saffari are set according to [38].

5.3.2 Comparisons with other feature incremental learning methods

For comparisons with other feature incremental learning methods, we divided the training set into the initial training set and the incremental learning training set with equal sizes. For the initial training set, the number of sensors is set to $\lceil \frac{num_{sensor}}{2} \rceil$. While all sensors are used for training in the incremental learning phase. The detailed experimental settings for comparisons with other feature incremental learning methods are listed in Table 4.

For OPID, we set $\lambda = 0.5$, $\rho = 1e-3$ and $\gamma = 0.8$. For FA-OSELM, C is chosen from $\{2^{-20}, 2^{-19}, \ldots, 2^{19}, 2^{20}\}$ and L is chosen from $\{10, 20, \ldots, 990, 1,000\}$. Grid research is adopted to find the best value for C and L. For heterogenous OTL, we adopt one-versus-one strategy to tackle multiclass classification and set $\gamma_1 = \gamma_2 = 1$, $\sigma_1 = \sigma_2 = 4$ and $C = 5$ according to [33]. For self-supervised ARTMAP, we set $\overline{\beta} = 0.01$ and $\overline{\overline{\beta}} = 0.01$. The other parameters are set as default.

5.4 Parameter Sensitivity

The proportion of decision trees selected to grow further, p, is the most important parameter for FIRF. In this section, we evaluate the effect of adapting p. Figure 4 shows the accuracy achieved by FIRF under different values of p.

From Figure 4, it can be observed that the accuracy of FIRF increases and then decreases as value of p increases. This result is consistent with the *error-ambiguity decomposition* rule. Diverse individual learners will increase high accuracy. When the value of p is very large or very small, the diversity among individual decision trees is small. This is because that, in these cases, most individual trees are updated with the same incremental data. From the view of data samples, the diversity among individual trees is small. In contrast, a medium value of p results in high accuracy. From Figure 4, FIRF with $p = 0.4$ and $p = 0.6$ achieved the best performance. Besides, the effect of parameter p is more pronounced when the number of available sensors is small. The difference in accuracy can be as high as close to 10% under DSADS, 12% under PAMAP2 and 3% under RealWorld(HAR). Therefore, selecting an appropriate value of p is important for the performance of FIRF. According to the experimental results presented in Figure 4, a medium value of p (between 0.4 and 0.6) is preferred. Thus, in the following experiments, we set the value of parameter p to 0.4.

5.5 Performance in Recognizing Continuous Newly Emerging Features

In this section, we evaluate the performance of FIRF in recognizing continuous newly emerging features. RF, ET and ORF-Saffari are adopted as comparison methods.

5.5.1 Classification Accuracy

Figure 5 compares the testing accuracy achieved FIRF, RF, ET and ORF-Saffari under the three public activity recognition datasets.

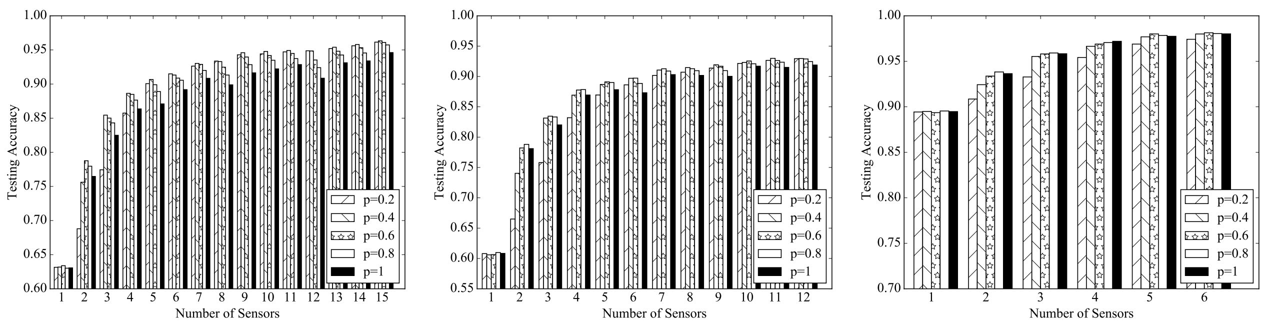
TABLE 3
Experimental settings for recognizing continuous newly emerging features

	Initial block			Incremental block				Test block
	Instance	Feature	Sensor	Instance	Feature	Sensor	Number of block	Instance
DSADS	304	27	1	304	27+27*i	1+i	14	4,560
PAMAP2	306	27	1	306	37+27*i	1+i	11	3,676
RealWorld(HAR)	9,662	43	1	9,662	43+43*i	1+i	5	57,974

[1] i is the number of incremental learning times.

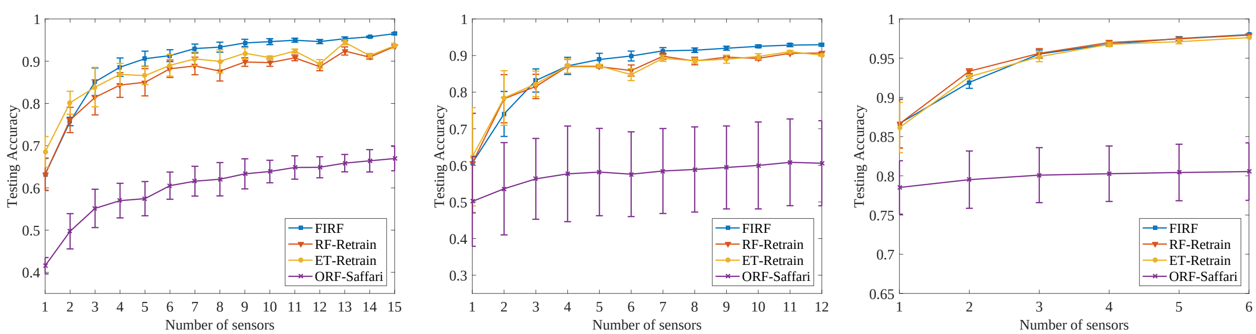
TABLE 4
Experimental settings for comparisons with other feature incremental learning methods

	Initial block			Incremental block			Test block
	Instance	Feature	Sensor	Instance	Feature	Sensor	Instance
DSADS	2,280	216	8	2,280	405	15	4,560
PAMAP2	1,838	162	6	1,838	324	12	3,676
RealWorld(HAR)	28,986	129	3	28,987	258	6	57,974



(a) DSADS (b) PAMAP2 (c) RealWorld(HAR)

Fig. 4. The effect of parameter p on the accuracy of FIRF under the three public activity datasets.



(a) DSADS (b) PAMAP2 (c) RealWorld(HAR)

Fig. 5. The mean and standard deviation of the accuracy of the four comparison methods under the three public activity recognition datasets.

As more sensors became available, the activity recognition models all showed increases in testing accuracy. The increases in testing accuracy eventually slowed down, which indicates that data from some sensors may be redundant. The testing accuracy of FIRF consistently increased with additional sensors over all three datasets. In contrast, the testing accuracy of RF and ET fluctuated significantly as the number of sensors increased.

FIRF achieved the highest testing accuracy on all three datasets in most cases. Under DSADS and when the number of sensors is larger than two, the testing accuracy of FIRF was 2.89%-5.96% and 0.84%-5.28% higher than RF and ET, respectively. Under PAMAP2, the testing accuracy of FIRF rose from 60.64% to 92.97%, while that of RF rose from 60.62% to 90.67%. ET performed the best when the number of sensors was no more than two. However, when the number of sensors increased beyond two, it under-performed FIRF by 0.03%-5.09%. Under RealWorld(HAR), there was little difference between FIRF, RF and ET in classification accuracy. For the retraining model, only a small size of data was used in the training of classification model, which resulted in a lower classification accuracy. In contrast, FIRF was able to take advantage of both the knowledge in existing model and newly arriving data. Thus, it achieved better performance in most cases. FIRF and batch learning methods significantly outperform the ORF-Saffari by more than 20%. This indicates that more sensors can contribute to higher testing accuracy.

In terms of standard deviation, FIRF, RF and ET showed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2855159, IEEE Transactions on Knowledge and Data Engineering

similar performance, while ORF-Saffari performed the worst on all three datasets. This means that the performance of ORF-Saffari is more susceptible to different sensor sequences. FIRF, RF and ET are more robust and less susceptible to the different sensor sequences. FIRF achieved the best performance in most cases.

5.5.2 Training Time

As shown in Figure 6, with increasing number of sensors, the training time for RF and ET both increased linearly under all three datasets. Under DSADS, the training time for RF increased from $108.93s$ to $346.79s$, while that for ET increased from $25.48s$ to 38.41. Under PAMAP2, the training time for RF increased from $96.26s$ to $261.15s$, while that for ET increased from $19.64s$ to $26.27s$. Under RealWorld(HAR), the training time for RF rose from $1.68 \times 10^4 s$ to $4.17 \times 10^4 s$, while that for ET rose from $420.77s$ to $482.31s$. As the number of available sensors increased, the number of features (K) grew linearly. Thus, the number of features selected to split each node ($\lfloor \sqrt{K-1} \rfloor$) grew accordingly, leading to the increase in retraining time. For the two batch learning algorithms, the construction of RF required more training time. This can be attributed to the splitting mechanism used by RF. The naive strategy of enumerating all partitions and then choosing the optimal split position makes the number of partitions grow exponentially, which is very time-consuming. In contrast, ET splits nodes with fully randomized split-criteria. Thus, compared with RF, ET requires less training time.

Different from batch learning methods, the training time of ORF-Saffari is more affected by the number of training samples than by the number of features. Thus, it showed some fluctuations when the number of features increased. The training time for ORF-Saffari weas $485.72s$-$605.47s$ under DSADS, $457.31s$-$581.80s$ under PAMAP2 and $1.87 \times 10^4 s$-$2.26 \times 10^4 s$ under RealWorld(HAR). As multiple passes through the training data was used by ORF-Saffari, longer training time was required under both DSADS and PAMAP2.

Compared with RF and ORF-Saffari, FIRF requires consistently less training time in most cases, ranging from $49.36s$ to $109.47s$ under DSADS, $45.15s$ to $97.39s$ under PAMAP2 and $8.68 \times 10^3 s$ to $1.68 \times 10^4 s$ under RealWorld(HAR). The advantage of FIRF in training time can be attributed to two factors. Firstly, the diversity scoring strategy of FIRF enables it to only select a few decision trees to retrain. The rest of the decision trees remain unchanged. Secondly, only part of a decision tree is reconstructed by the FITGM algorithm under FIRF, which contributes to lower time consumption.

5.5.3 Testing Time

From Figure 7, as the number of features and instances increased, the testing time increased for all methods under the three public activity recognition datasets. ET, which is trained with randomized split-criteria, consumed more time in the prediction phase than both FIRF and RF. With the least time spent in training, the classification model constructed by ET is more complex, which resulted in longer testing time. RF showed a small advantage over FIRF in testing time under the three datasets. The testing time for ORF-Saffari was significantly more than RF-Retrain and FIRF. As stated in [61], the testing time for RF is directly related to the average depth of the decision trees in the random forest. From the results presented in Figure 7, the random forest constructed by FIRF was comparable with RF and much better than both ET and ORF-Saffari, which means that the model constructed by FIRF is similar to the model learned by RF and much better than ET and ORF-Saffari.

5.6 Comparison with Other Feature Incremental Learning Methods

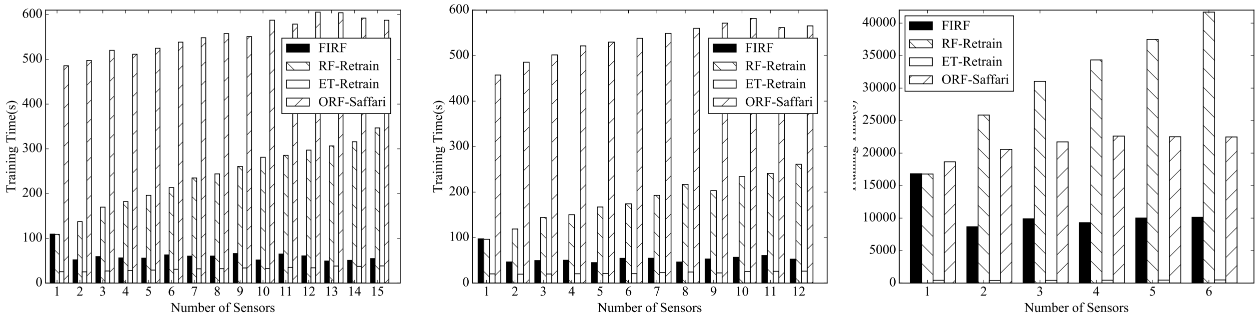
In this section, we compare the performance of FIRF with four feature incremental learning methods, including OPID [43], FA-OSELM [42], heterogeneous OTL [33] and self-supervised ARTMAP [47].

As shown in Figure 8, FIRF achieved the best test accuracy under all three datasets. Under DSADS, FIRF achieved a $7.03\% - 21.59\%$ advantage in terms of testing accuracy over the other three state-of-the-art methods. Under PAMAP2 dataset, the testing accuracy of FIRF was higher than FA-OSELM, OPID, heterogeneous OTL and self-supervised ARTMAP by 2.19%, 5.17%, 12.34% and 15.53%, respectively. Under RealWorld(HAR), OPID achieved 92.83% testing accuracy, while heterogeneous OTL achieved 95.91% testing accuracy. FA-OSELM performed better than both of them. FIRF achieved a 2.62%-5.7% advantage over the three comparison methods. By contrast, self-supervised ARTMAP performs the worst under all three datasets. As can be observed from Figure 8, the testing accuracy after incremental learning by self-supervised ARTMAP is worse than the initial phase. This is due to the spreading of the labels to newly arrived data in an unsupervised manner which affects the updating of the classifier. FIRF achieves the best performance in both the initial training phase and the incremental learning phase. The good performance in initial training phase can be attributed to the good performance of RF. Many studies have shown that RF is better or at least comparable to other state-of-the-art methods [34]. This group of experiments does not show the strength of FIRF in improving the accuracy incrementally well, which can be attributed to the good performance of the initial model. The room for improvement for FIRF in this area is small. In conclusion, FIRF demonstrated the best performance in testing accuracy compared with the four state-of-the-art feature incremental learning methods.

6 CONCLUSIONS

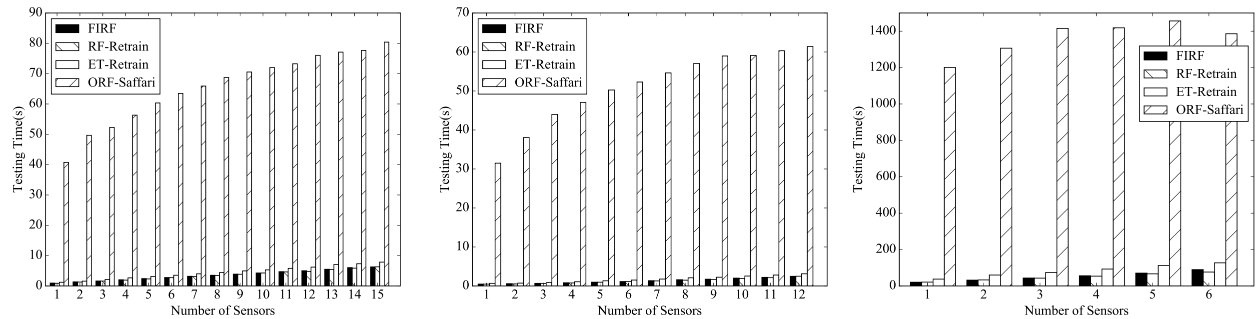
Incremental learning with the data from a fixed set of sensors can achieve good performance in activity recognition. With the advance of ubiquitous technology, data from more sensors can be dynamically employed to improve the performances of activity recognition models. Instead of retraining a new model from scratch, updating an existing model incrementally is more advantageous for reducing training time and the need for data annotation. This paper presents a novel feature incremental activity recognition model, namely FIRF. It is designed to adapt an existing activity model to take advantage of newly available sensors in a dynamic environment. The proposed method increases the diversities of individual learners in the random forest, while improving their classification performance.

Experimental results show that FIRF performed better than state-of-the-art feature incremental learning methods. It is a promising choice for handling the emergence of new sensors in dynamic environments. To achieve good performance, an appropriate choice of parameter p is very important. Through experimentation, we found that values between 0.4 and 0.6 are appropriate for p. Compared to existing approaches, FIRF shows a great improvement in time consumption. As the number of features increases,



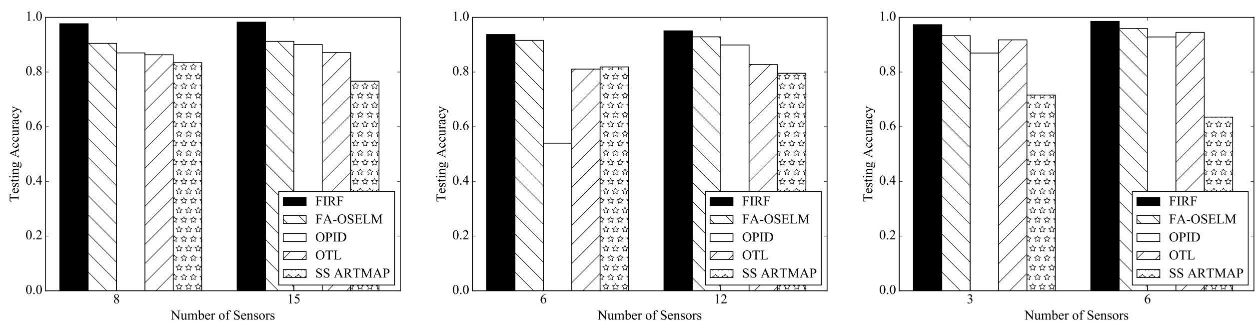
(a) DSADS (b) PAMAP2 (c) RealWorld(HAR)

Fig. 6. Training time required by the four comparison methods under the three public activity recognition datasets.



(a) DSADS (b) PAMAP2 (c) RealWorld(HAR)

Fig. 7. Testing time required by the four comparison methods under the three public activity recognition datasets.



(a) DSADS (b) PAMAP2 (c) RealWorld(HAR)

Fig. 8. Comparison with other feature incremental learning methods in terms of testing accuracy.

the training time of FIRF remains consistently low. In addition, FIRF also requires less testing time, which indicates that it is able to build models with optimal structures. In summary, compared with other methods, FIRF is a good choice for feature incremental activity recognition.

FIRF is a supervised feature incremental learning method, which requires data annotation for both initial training and incremental training. In our future research, we plan to investigate how to incorporate semi-supervised strategies into FIRF to reduce the need for data annotation. Ethical and privacy issues [62] will also be studied to engender trust with users.

Acknowledgments

This work is supported by the National Key Research and Development Plan of China (No. 2017YFB1002801); Natural Science Foundation of China under Grant No. 61572471, No.61502456, and No.61472399; Beijing Municipal Science & Technology Commission No.Z161100000216140; and Nanyang Technological University, Nanyang Assistant Professorship (NAP).

References

[1] H. Lin, J. Hou, H. Yu, Z. Shen, and C. Miao, "An agent-based game platform for exercising people's prospective memory," in *Proc. WI-IAT'15*. IEEE, 2015, pp. 235–236.

[2] V. J. Verlinden, J. N. van der Geest, A. Hofman, and M. A. Ikram, "Cognition and gait show a distinct pattern of association in the general population," *Alzheimer's & Dementia*, vol. 10, no. 3, pp. 328 – 335, 2014.

[3] Y. Chen, M. Huang, C. Hu, Y. Zhu, F. Han, and C. Miao, "A coarse-to-fine feature selection method for accurate detection of cerebral small vessel disease," in *Proc. IJCNN'16*. IEEE, 2016, pp. 2609–2616.

[4] Y. Chen, C. Hu, B. Hu, L. Hu, H. Yu, and C. Miao, "Inferring cognitive wellness from motor patterns," *IEEE Trans. Knowl. Data Eng.*, 2018, to be published. doi:10.1109/TKDE.2018.2820024.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2855159, IEEE Transactions on Knowledge and Data Engineering

[5] G. H. Jacobs, G. A. Williams, H. Cahill, and J. Nathans, "Emergence of novel color vision in mice engineered to express a human cone photopigment," *Science*, vol. 315, no. 5819, pp. 1723–1725, 2007.

[6] T. Grosse-Puppendahl, E. Berlin, and M. Borazio, "Enhancing accelerometer-based activity recognition with capacitive proximity sensing," in *Proc. AmI'12*. Springer, 2012, pp. 17–32.

[7] C. Chen, R. Jafari, and N. Kehtarnavaz, "Improving human action recognition using fusion of depth camera and inertial sensors," *IEEE Trans. Hum. Mach. Syst.*, vol. 45, no. 1, pp. 51–61, 2015.

[8] K. Altun, B. Barshan, and O. Tunel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognit.*, vol. 43, no. 10, pp. 3605–3620, 2010.

[9] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. ISWC'12*. IEEE, 2012, pp. 108–109.

[10] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *Proc. PerCom'16*, 2016, pp. 1–9.

[11] W.-X. Li and N. Vasconcelos, "Complexactivityrecognition via attribute-dynamics," *Int. J. Comput. Vis.*, vol. 122, no. 2, pp. 334–370, 2017.

[12] H. Guo, L. Chen, L. Peng, and G. Chen, "Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble," in *Proc. UbiComp'16*. ACM Press, 2016, pp. 1112–1123.

[13] L. Yao, F. Nie, Q. Z. Sheng, T. Gu, X. Li, and S. Wang, "Learning from less for better, semi-supervised activity recognition via shared structure discovery," in *Proc. Ubicomp'16*. ACM Press, 2016, pp. 13–24.

[14] H. Zhao and Z. Liu, "Human action recognition based on non-linear SVM decision tree," *J. Computat. Inform. Syst.*, vol. 7, no. 7, pp. 2461–2468, 2011.

[15] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. IWAAL'12*. Springer, 2012, pp. 216–223.

[16] L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, and W. Stork, "Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity," in *Proc. EMBS'08*. IEEE, 2008, pp. 5250–5253.

[17] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *Proc. IJCAI'11*, vol. 11, 2011, pp. 2545–2550.

[18] Y. Iwasawa, K. Nakayama, I. E. Yairi, and Y. Matsuo, "Privacy issues regarding the application of dnns to activity-recognition using wearables and its countermeasures by use of adversarial training," in *Proc. IJCAI'17*, 2017, pp. 1930–1936.

[19] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.

[20] H. Qian, Y. Mao, W. Xiang, and Z. Wang, "Recognition of human activities using svm multi-class classifier," *Pattern Recognit. Lett.*, vol. 31, no. 2, pp. 100–111, 2010.

[21] V. Franc and V. Hlavác, "Multi-class support vector machine," in *Proc. ICPR'02*. IEEE, 2002, pp. 236–239.

[22] C. Hsu and C. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, 2002.

[23] T. Phan, "Improving activity recognition via automatic decision tree pruning," in *Proc. Ubicomp'14*, 2014, pp. 827–832.

[24] C. Randell and H. Muller, "Context awareness by analysing accelerometer data," in *Proc. ISWC'00*. IEEE, 2000, pp. 175–176.

[25] L. Hu, Y. Chen, S. Wang, and Z. Chen, "b-coelm: A fast, lightweight and accurate activity recognition model for mini-wearable devices," *Pervasive Mob. Comput.*, vol. 15, pp. 200–214, 2014.

[26] Z. Wang, D. Wu, R. Gravina, G. Fortino, Y. Jiang, and K. Tang, "Kernel fusion based extreme learning machine for cross-location activity recognition," *Inform. Fusion*, vol. 37, pp. 1–9, 2017.

[27] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: a survey," *Knowl. and Inf. Syst.*, vol. 36, no. 3, pp. 537–556, 2013.

[28] K. D. Feuz and D. J. Cook, "Collegial activity learning between heterogeneous sensors," *Knowl. and Inf. Syst.*, vol. 53, no. 2, pp. 337–364, 2017.

[29] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[30] V. W. Zheng, D. H. Hu, and Q. Yang, "Cross-domain activity recognition," in *Proc. Ubicomp'09*, 2009, pp. 61–70.

[31] K. D. Feuz and D. J. Cook, "Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (fsr)," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 1, pp. 1–27, 2015.

[32] C. A. K. Yuan Shi, "Learning with previously unseen features," in *Proc. IJCAI'17*, 2017, pp. 2722–2729.

[33] P. Zhao and S. C. H. Hoi, "OTL: a framework of online transfer learning," in *Proc. ICML'10*, 2010, pp. 1231–1238.

[34] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *ICCVW'09*. IEEE, 2009, pp. 1393–1400.

[35] O. H. Elgawi, "Online random forests based on corrfs and corrbe," in *CVPRW'08*. IEEE, 2008, pp. 1–7.

[36] M. Denil, D. Matheson, and D. Nando, "Consistency of online random forests," in *Proc. ICML'13*, 2013, pp. 1256–1264.

[37] B. Lakshminarayanan, D. M. Roy, and Y. W. Teh, "Mondrian forests: Efficient online random forests," in *Adv. Neural. Inf. Process. Syst.*, 2014, pp. 3140–3148.

[38] T. Sztyler and H. Stuckenschmidt, "Online personalization of cross-subjects based activity recognition models on wearable devices," in *Proc. PerCom'17*. IEEE, 2017, pp. 180–189.

[39] Z. Zhao, Z. Chen, Y. Chen, S. Wang, and H. Wang, "A class incremental extreme learning machine for activity recognition," *Cogn. Comput.*, vol. 6, no. 3, pp. 423–431, 2014.

[40] C. Hu, Y. Chen, L. Hu, and X. Peng, "A novel random forests based class incremental learning method for activity recognition," *Pattern Recognit.*, vol. 78, pp. 277–290, 2018.

[41] M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool, "Incremental learning of random forests for large-scale image classification," *IEEE Trans. Pattern Ana. Mach. Intell.*, vol. 38, no. 3, pp. 490–503, 2016.

[42] X. Jiang, J. Liu, Y. Chen, D. Liu, Y. Gu, and Z. Chen, "Feature adaptive online sequential extreme learning machine for lifelong indoor localization," *Neural Comput. Appl.*, vol. 27, no. 1, pp. 215–225, 2016.

[43] C. Hou and Z. H. Zhou, "One-pass learning with incremental and decremental features," *IEEE Trans. Pattern Ana. Mach. Intell.*, 2017, to be published. doi:10.1109/TPAMI.2017.2769047.

[44] Y. Xing, F. Shen, and J. Zhao, "Perception evolution network adapting to the emergence of new sensory receptor." in *Proc. IJCAI'15*, 2015, pp. 3967–3973.

[45] V. F. Rey and P. Lukowicz, "Label propagation: An unsupervised similarity based method for integrating new sensors in activity recognition systems," *Proc. Ubicomp'17*, vol. 1, no. 3, pp. 1–24, 2017.

[46] D. Bannach, B. Sick, and P. Lukowicz, "Automatic adaptation of mobile activity recognition systems to new sensors," in *Proc. Ubicomp'11*, 2011.

[47] G. P. Amis and G. A. Carpenter, "Self-supervised artmap," *Neural Netw.*, vol. 23, no. 2, pp. 265–282, 2010.

[48] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[49] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[50] L. Breiman, J. H. Friedman, R. Olshen, and C. J. Stone, "Classification and regression trees," *Biometrics*, vol. 40, no. 3, p. 358, 1984.

[51] L. Xu, A. Krzyzak, and C. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Syst. Man Cybern.*, vol. 22, no. 3, pp. 418–435, 1992.

[52] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," *Neural Networks for Speech and Image Processing*, pp. 126–142, 1993.

[53] N. Jamali and C. Sammut, "Majority voting: Material classification by tactile sensing using surface texture," *IEEE Trans. on Robot.*, vol. 27, no. 3, pp. 508–521, 2011.

[54] N. Littlestone and M. K. Warmnuth, "The weighted majority algorithm," *Inform. Comput.*, vol. 108, pp. 212–261, 1994.

[55] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Proc. NIPS'95*, vol. 7, 1995, pp. 231–238.

[56] Z. H. Zhou, *Ensemble Methods: Foundations and Algorithms*. CRC press, 2012.

[57] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[58] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

[59] B. Barshan and M. C. Yuksek, "Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units," *The Comput. J.*, vol. 57, no. 11, pp. 1649–1667, 2014.

[60] L. Hu, Y. Chen, J. Wang, C. Hu, and X. Jiang, "OKRELM: online kernelized and regularized extreme learning machine for wearable-based activity recognition," *Int. J. Mach. Learn. Cyb.*, pp. 1–14, 2017.

[61] G. Louppe, "Understanding random forests: From theory to practice," Ph.D. dissertation, University of Liege, Belgium, 2014, arXiv:1407.7502.

[62] H. Yu, Z. Shen, C. Miao, C. Leung, V. R. Lesser, and Q. Yang, "Building ethics into artificial intelligence," in *Proc. IJCAI'18*, 2018, pp. 1–7.

Chunyu Hu received the B.S. and M.S. degrees in computer science and technology from Shandong Normal University, Jinan, China, in 2012 and 2015, respectively. She is now a Ph.D. candidate in the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Her current research interests include machine learning, pervasive computing, and activity recognition.

Yiqiang Chen (S'01M'03) received the B.S. and M.S. degrees in computer science from Xiangtan University, Xiangtan, China, in 1996 and 1999, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003.

In 2004, he was a Visiting Scholar Researcher with the Department of Computer Science, Hong Kong University of Science and Technology (HKUST), Hong Kong. He is currently a professor and the director of the Pervasive Computing Research Center at the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). His research interests include artificial intelligence, pervasive computing, and human computer interaction.

Xiaohui Peng received the B.S. degree from Northwestern Polytechnical University, Xi'an, China, in 2006, the M.S. degree from Peking University, Beijing, China, in 2009, and the Ph.D. degree from the University of Tokyo, Tokyo, Japan, in 2016. Now he is an assistant professor in the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS). His research interests include edge computing and the Internet of Things.

Han Yu received his B.Eng. (Hons) degree and Ph.D. degree from the School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore in 2007 and 2014, respectively. He is currently a Nanyang Assistant Professor (NAP) at SCSE, NTU. He was a Visiting Scholar with the Department of Computer Science, Hong Kong University of Science and Technology (HKUST), Hong Kong from 2017 to 2018. From 2015 to 2018, he held the prestigious Lee Kuan Yew Post-Doctoral Fellowship (LKY PDF) at the Joint NTU-UBC Research Centre of Excellence in Active Living for the Elderly (LILY). His research focus is on artificial intelligence (AI)-powered crowd work. Specifically, he designs stochastic optimization techniques to enhance crowd workers' experience through data-driven algorithmic management.

Chenlong Gao received the B.S. degree in software engineering from Northeast Normal University, Changchun, China, in 2016. Now he is a Ph.D. candidate in the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His current research interests include artificial intelligence, pervasive computing, and activity recognition.

Lisha Hu received the B.S. and M.S. degrees in applied mathematics from Hebei University, Baoding, China, in 2009 and 2012, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2017. She is currently a lecturer in Hebei University of Economics and Business. Her research interests include machine learning, activity recognition and wearable computing.