IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 17, 2021, accepted May 1, 2021, date of publication May 5, 2021, date of current version May 18, 2021.
Digital Object Identifier 10.1109/ACCESS.2021.3077764

Baseline Model Training in Sensor-Based Human Activity Recognition: An Incremental Learning Approach

JIANYU XIAO, LINLIN CHEN, (Member, IEEE), HAIPENG CHEN, AND XUEMIN HONG, (Member, IEEE)
National-Local Joint Engineering Research Center of Navigation and Location Services, Xiamen University, Xiamen 361005, China
Corresponding author: Xuemin Hong (xuemin.hong@xmu.edu.cn)

This work was supported in part by the Science and Technology Key Project of Fujian Province, China, under Grant 2019HZ020009 and Grant 2018HZ0002-1, in part by the Natural Science Foundation of China under Grant 62077040, and in part by the Key Laboratory of Wireless Sensor Network and Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences.

ABSTRACT Human activity recognition (HAR) based on wearable sensors has attracted significant research attention in recent years due to its advantages in availability, accuracy, and privacy-friendliness. HAR baseline model is essentially a general-purpose classifier trained to recognized multiple activity patterns of most user types. It provides the input for subsequent steps of model personalization. Training a good baseline model is of fundamental importance because it has significant impacts on the ultimate HAR accuracy. In practice, baseline model training in HAR is a non-trivial problem that faces two challenges: insufficient training data and biased training data. This paper proposes a novel baseline model training scheme to tackle the two challenges using Deep InfoMax (DIM)-based unsupervised feature extraction and Broad Learning System (BLS)-based incremental learning, respectively. Experimental results demonstrate that the proposed scheme outperform conventional methods in terms of overall accuracy, computational efficiency, and the ability to adapt to dynamic scenarios with changing data characteristics.

INDEX TERMS Human activity recognition, baseline model, broad learning system, DIM, incremental learning.

I. INTRODUCTION

Human Activity Recognition (HAR) based on sensor data is a thriving research topic that has a wide range of applications in fields of health-care [1], security [2] and surveillance [3]. A diverse types of sensors have been used in HAR, including kinetic sensors, visual sensors, sound sensors, and radio signal sensors, etc [4]. Among the various types of sensor modality for HAR, HAR based on wearable kinetic sensors has attracted significant research attention due to several advantages in data availability, high accuracy and privacy friendliness. The first advantage is data availability: the sensing data is widely available because low-cost kinetic sensors can be easily embedded into smartphones, smartwatches or helmets [5]. The second advantage is high accuracy as the sensors can directly measure the acceleration and angular velocity related to human body movements. The third advantage is privacy friendliness, which means the sensing data include little information about users apart from what is crucial to the HAR task. In this paper, we restrict our focus on HAR systems based on wearable kinetic sensors.

The HAR task is essentially a pattern recognition problem, to which various types of machine learning methods can be applied [6]. A particular challenge in HAR is to address the issue of user diversity. For instance, a model trained on a data set of teenagers does not perform well on the elderly because the behavior characteristics of teenagers and the elderly are very different. In recent years, deep learning (DL) based methods [4], [7]–[10] have emerged as a promising solution that have been shown to achieve higher accuracy than conventional machine learning methods [6]. The advantage of the DL-based method comes firstly from its capability to learn deep representations of sensor data with little human

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

experience and domain knowledge, and secondly from its capability to be fine-tuned and personalized for different users.

A DL-based HAR system typically involves two phases, as shown in Fig.1. The first phase (the left part in Fig.1) is to train a baseline neural network (NN) model using an initial data set, which include labeled data from different groups of people. In the literature, the baseline model is also called "user-independent model" [11], which means it is a general model applicable to all users. However, the generality of the baseline model comes at a cost of reduced accuracy. Consequently, a second phase (the right part in Fig.1) is often used to further personalize the baseline model based on data from a particular user. It has been shown that personalized HAR models can significantly improve the accuracy of HAR tasks [12].

The performance of DL-based HAR systems rely heavily on the availability and quality of labeled data. It is commonly assumed that labeled data can be collected in a online fashion once the HAR system is deployed. However, in practice, although unlabeled data from wearable kinetic sensors is widely available, labeled data is difficult to obtain because it requires a high degree of user participation. Furthermore, labeled data collected online tends to have a highly unbalanced demographic property, which means that much of the data is concentrated to a small number of active users or certain types of users. For example, because teenagers are more willing to interact with the system, labeled data collected online tends to be biased towards teenagers. In summary, the quality of labeled data can have significant impacts on the generality of the baseline model and the accuracy of the ultimate personalized model.

In the literature, a majority of published work have focused on the task of training personalized models (i.e., phase two) [4]. To this end, transfer learning based approaches have been proposed to deal with the problem of imbalanced data set, such that labeled data of a particular type of users can also help to optimize the models for other types of users [13], [14]. Moreover, incremental learning has been proposed not only to speed up the online training process, but also to provide the flexibility of learning new classes of user activities [12], [15]–[21]. It has been shown that incremental learning approaches can adapt well when the user behavioral pattern changes over time or a new kind of user activity needs to be recognized [22].

In contrast to the issue of personalized model training (i.e., phase two), the issue of baseline model training (i.e., phase one) has received inadequate research attention so far. According to a recent study [12], if the baseline model is not accurate enough in the first place, the second phase of personalization can hardly improve the accuracy of HAR. It should be noted that training a good baseline model is a non-trivial problem. There are two major challenges.

- Insufficient training data: The baseline model is a highly complex model that needs to accommodate the diversity of user types (e.g., gender, age, body height), sensor embedding methods (e.g., hand-held, limbs, helmet), and activity types. Training such a model requires a large volume of labeled data. An initial training data set generated in experimental settings is insufficient to train a good industrial-class DL model.
- Biased training data: A common way to enlarge the training data set is to include labeled data from user feedback. However, in practice, the user generated data is often biased towards certain types of users. Biased training data could result in a biased baseline model that does not generalize well to all users.

To tackle the common problems of insufficient training data and biased training data, a wide range of methods have been proposed. These methods can be broadly categorized into two types: data-oriented and model-oriented. Data-oriented methods aim to improve the quality of the training data set. For example, resampling techniques can be used to perform up-sampling or down-sampling on a particular class of data, thereby changing the size or composition of the training dataset as needed [23], [24]. On the other hand, model-oriented methods aim to design learning algorithms that can tolerate certain impairments of the data set. For example, unsupervised learning algorithms can be used for unlabeled data, transfer learning can be used to complement an insufficient data set with large data sets in other domains, and incremental learning can be used for continuous online learning when the incoming training data is biased. In this paper, we focus on the model-oriented method only and propose a algorithm that effectively integrates unsupervised learning and incremental learning.

This paper aims to propose a method for training the baseline model of HAR. To our best knowledge, this is the first paper which directly addresses the problem of baseline model training in HAR. Our paper tackles the above-mentioned challenges directly and makes the following contributions.

- To address the first challenge, a novel unsupervised DL scheme is proposed to learn good representations from unlabeled sensor data. The DL scheme is based on enhanced Deep InfoMax (DIM) [25], which aims to maximize the mutual information between extracted high-level features and the original data.
- To address the second challenge, a broad-learning (BL) [26] based incremental learning scheme is proposed to fine-tune the baseline model with streaming labeled data. A novel technique called random factor scaling is further introduced to the BL system to enhance its performance.
- The DIM and BL network is integrated as a holistic solution for baseline model training and online updating. Experimental results show that the proposed DIM-BLS scheme outperforms other approaches in HAR tasks.

The remainder of this paper is organized as follows. Section II briefly reviews the related work. In Section III, details of the proposed DIM-BLS model for HAR are presented. Experimental evaluations are shown in Section IV. Finally, conclusions and future work are presented in Section V.

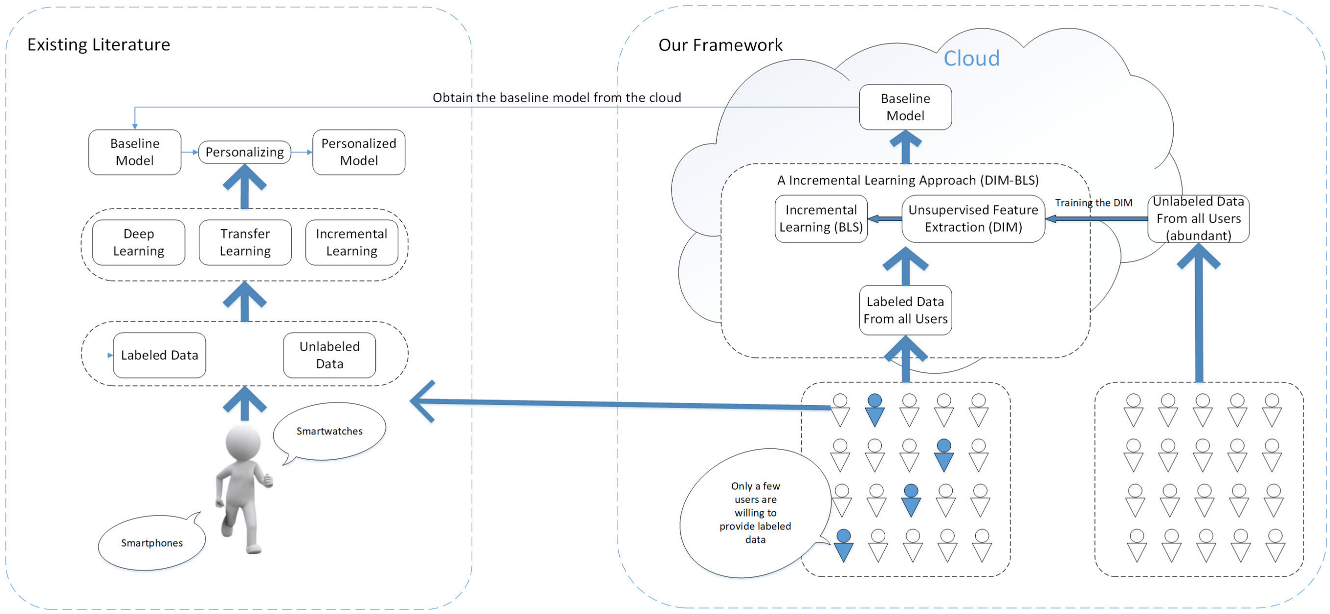

FIGURE 1. **Illustration of HAR systems including online baseline model training and personal model training.**

II. RELATED WORK

HAR is essentially a pattern recognition problem, to which many traditional machine learning based approaches have been applied [7]. These traditional approaches rely on either expert-crafted features [8], [27], [28] or some self-learned shallow features to train the classifier. However, such features cannot sufficiently represent the latent characteristics of complex human activities [7] and usually become the performance bottleneck.

In recent years, DL-based methods have drawn increasing attention in HAR as they can offer higher accuracy than traditional approaches [4], [29]. The accuracy improvement comes from their capability to extract non-linear high-level features from sensor data using deep neural networks, e.g., Convolutional Neural Network (CNN) or Long Short-Term Memory (LSTM) [30], [31]. However, DL-based methods have several drawbacks. First, DL-based methods require a complex process of parameter initialization and tuning, which might significantly increase the computational complexity [4], [7], [9], [10]. Second, the method is data-driven and relies heavily on the quality of training data. The third problem is known as the *Classifier Flexibility* problem, which means that the method cannot adapt well in dynamic scenarios with changing data distributions. For example, a DL model trained on data from children cannot accurately identify the activities of the elderly.

The classifier flexibility problem is an essential challenge in HAR. To tackle this challenge, transfer learning has been proposed as a primary means to exploit the common knowledge embedded in different HAR tasks. In [32], the authors combined naive Bayes and SVM to transfer the knowledge between activities. Morales and Roggen [13] proposed a model that consists of multiple CNN layers and LSTM layers, fine-tuned the model with different freezing layers to achieve knowledge transfer between different data sets with different distributions. Paper [14] studied distribution of sensor data of HAR and proposed an unsupervised deep transfer learning approach based on Maximum Mean Discrepancy (MMD) [33].

Incremental learning is another popular paradigm proposed to address the classifier flexibility challenge in HAR. In [15], the authors proposed a personalization algorithm that integrates SVM classification and K-means clustering method to adjust the model of person A with selected confident samples from another person B. An approach named hybrid user-assisted incremental model adaption was proposed in [34] to address the challenge in a dynamic smart-home environment. Wang *et al.* [16] proposed an incremental activity recognition classifier based on probabilistic neural networks (PNN) and adjustable fuzzy clustering (AFC). In [17], incremental learning and active learning were adopted to learn from evolving data streams. Several categories of incremental learning cases were considered in [18] using the RF-based methods.

In summary, although there is a wealth of HAR literature, a large body of published work is dedicated to the problem of personalized model training. In contrast, the problem of baseline model training is still an under-investigated subject.

III. PROPOSED FRAMEWORK FOR HAR

A. GENERAL FRAMEWORK AND TRAINING PROCEDURE

In this section, we propose a novel online learning scheme for baseline model training in HAR. The general framework of the proposed scheme is depicted in Fig.1. We consider a cloud-based HAR system that collects sensor data from users and updates the baseline model in a timely fashion.

There are two types of user generated data: unlabeled and labeled. The former can be collected passively without user participation, thus it is abundant and covers all user types. The latter is data with activity labels, which require users to actively participate in a data labeling procedure. In practice, only a small portion of users are willing to provide labeled data with high quality. Therefore, the labeled data set has a much smaller size compared with the unlabeled data set. Moreover, the labeled data set is likely to be biased towards certain user types.

The proposed baseline HAR model has two concatenated functional blocks: an unsupervised feature extraction block that applies Deep InfoMax (DIM), followed by an incremental learning block that applies Broad Leaning System (BLS). DIM is proposed in 2018 [25] as a novel approach to estimate and maximize the mutual information between high dimensional data sets. Exact calculation of the mutual information between high dimensional random variables is numerically intensive and typically intractable in practice [35]. The main idea of DIM is to use NNs to estimate mutual information and train an encoder by maximizing the mutual information between the input and output. A typical application of DIM is to use it as an unsupervised feature extraction technique. It has been shown that DIM outperformed a number of popular unsupervised learning methods [25] and can be successfully applied to many feature extraction tasks [36]–[38].

Compared with other widely applied methods such as auto-encoder (AE) [39] and reverse generative adversarial network (GAN) [40], which should train the encoder and decoder as pairs, DIM does not need to train a decoder. As a result, DIM is a computationally-efficient solution to solve the problem of insufficient training data. Besides, kinetic sensors in HAR could be deployed in multiple body parts such as hand, head helmet, foot, and body. As a result, the sensor data could have different inherent features. AE and GAN use reconstruction error and distribution divergence as the optimization criteria, respectively. In practice, these two criteria do not work well for the HAR data. Instead, DIM uses mutual information as the criterion for feature extraction. This is a general-purpose criterion that better suits the HAR scenario.

The second function block of our proposed scheme uses BLS. BLS was proposed by Chen and Liu [26] in 2018 as a computationally efficient incremental learning technique. BLS establishes a flat network that maps the original inputs to feature nodes. The feature nodes are further transformed to enhancement nodes, which can better represent nonlinear features of the input data. More importantly, through matrix calculations, BLS is able to perform effective and efficient incremental learning from new data [26], [41]. As a result, BLS is a computationally-efficient solution among other possible choices because its incremental mechanism is based on linear calculations.

We note that the proposed DIM-BLS framework is deliberately designed and tailored to the scenario of baseline HAR model training. The original BLS suffers from poor feature extraction performance due to its shallow structure.

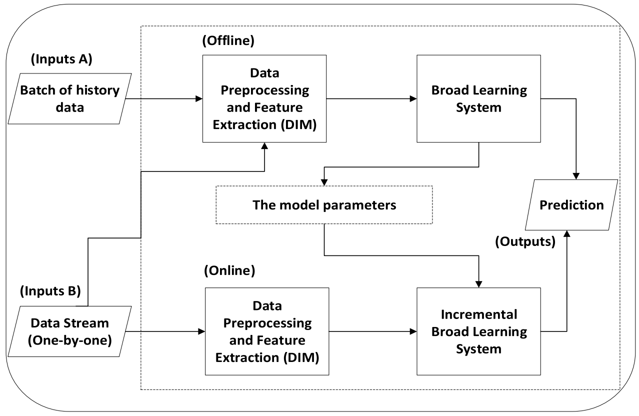

FIGURE 2. The proposed HAR framework with integrated DIM and BLS functional blocks.

In our framework, this weakness is overcome by DIM, a deep structure feature extraction module. The proposed DIM-BLS combination exploits the complementary characteristics of DIM and BLS. The deep-structure DIM ensures the effectiveness of feature extraction, while the shallow-structure BLS ensures the efficiency of incremental learning. Furthermore, BLS-based incremental learning is introduced into HAR due to the following merits. First, it enables efficient update of the baseline model using only the newly incoming labeled data. In other words, there is no need for time-consuming model retraining using the entire data set. Second, when the incoming labeled data is biased toward certain user types, BLS-based incremental learning ensures that the baseline model can learn from new data while not losing its capability with respect to other user types. In other word, the baseline model does not "forget" what it has already learned from historical data.

The training procedure of the DIM-BLS network include two stages, as illustrated in Fig.2. The first stage (the upper half in Fig.2) is *offline training*, in which an initial data set is used to train the DIM and the baseline model. The second stage (the lower half in Fig.2) is *online incremental training*. This stage initializes the network parameters from the first stage and refines the baseline model by training it with new data samples coming one-by-one or small-batch by small-batch. In what follows, we will explain the proposed functional blocks and training procedure in detail.

B. UNSUPERVISED FEATURE EXTRACTION BASED ON DIM

In both the online or offline learning stages, the input sensor data is processed by DIM to extract high-level features before being fed into BLS. In a DIM model, a NN is used to encode high dimensional data into features with lower dimension. The DIM model trains the NN to maximize the mutual information between the input data and its feature. In [25], it was shown that apart from considering global features, it may be useful to maximize the average mutual information between the encoded features and some local regions of the original data.

The rationale of DIM is as follows. Given $\mathcal{X}$ and $\mathcal{Y}$, the domain and range of a continuous and (almost everywhere) differentiable parametric function, we define function $E_\psi : \mathcal{X} \to \mathcal{Y}$ as a neural network encoder with parameter ψ, $\mathbf{X} := \left\{x^{(i)} \in \mathcal{X}\right\}_{i=1}^{N}$ as the training data set, $\mathbb{P}$ as the empirical data distribution, and $\mathbb{U}_{\psi,\mathbb{P}}$ as the marginal distribution induced by pushing samples via the NN. The encoder is trained respect to the following two criteria.

- Maximize mutual information between data $\mathbf{X}$ and it's encoded features, including local and global features;
- Match the encoded features with some prior distribution.

Based upon the above objectives of MI maximization and prior matching, DIM borrow ideas from literature [42] and [35] to construct an objective function as

$$\arg\max_{\omega_1,\omega_2,\psi}(\alpha\widehat{\mathcal{I}}_{\omega_1,\psi}(X;E_\psi(X)) + \frac{\beta}{M^2}\sum_{i=1}^{M^2}\widehat{\mathcal{I}}_{\omega_2,\psi}(X^{(i)};E_\psi(X))) + \arg\min_{\psi}\arg\max_{\phi}\gamma\widehat{\mathcal{D}}_\phi(\mathbb{V}||\mathbb{U}_{\psi,\mathbb{P}}), \quad (1)$$

where $\widehat{\mathcal{I}}_{\omega,\psi}(X;E_\psi(X))$ is a mutual information estimator. $\mathbb{V}$ is a certain prior distribution and $\widehat{\mathcal{D}}_\phi\left(\mathbb{V}\|\mathbb{U}_{\psi,\mathbb{P}}\right)$ is a divergence estimator. Parameters ω_1 and ω_2 are the discriminator parameters for the global and local objectives, respectively; M is the number of local feature windows. Symbols α, β, and γ are hyperparameters. Detailed proof and derivation of DIM can be found in [25].

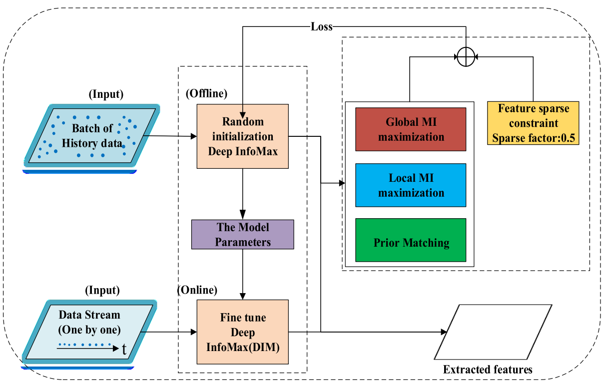

FIGURE 3. **Structure and training process of the DIM functional block.**

Fig.3 shows the training process of DIM. First, an encoding network and two discriminator networks are initialized for evaluating the global and local mutual information. Next, the loss function is computed according to (1). Here, we further impose a sparse constraint on the output features (with a sparse factor of, e.g., 0.5) to enhance feature extraction. Based on the above loss function, we construct positive/negative samples using the method of random scrambling of dimensions [25]. The stochastic gradient descent (SGD) method is then used to train the network in the offline stage. In the online training stage, only the mutual information network is fine-tuned with the incoming new data stream. We note that if the new incoming data is insufficient, an oversampling technique as introduced in [43] could be used to further enhance the training performance.

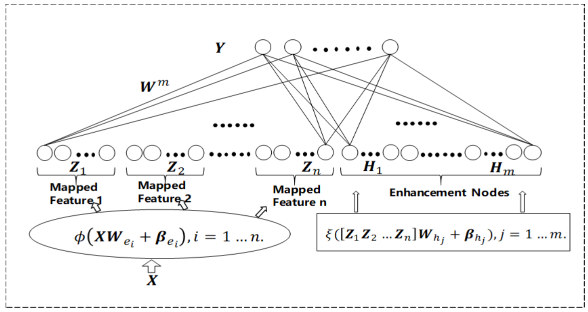

FIGURE 4. **Structure of the BLS functional block.**

C. BASIC BLS MODEL

As illustrated in Fig.4, the BLS network is constructed based on the random vector functional link neural network (RVFLNN) [44], which is a random vector single-layer neural network with flat functional-link structure. However, different from the original RVFLNN, the input data in BLS are transformed by randomly generated feature mappings before fed into the network. In addition, the BLS network could be broaden by adding enhancement nodes into the network. The general procedure of BLS is as follows. First, a function is used to convert the input into random features. The features are further connected to the enhancement nodes by nonlinear or linear activation functions. Finally, output results are determined by the direct feed of the mapping nodes along with the enhancement nodes. The weights of the output layer are determined by the pseudoinverse of matrices (using efficient approximation algorithms such as Ridge Regression [45]) or the back propagation (BP) algorithm.

We assume that the input training set $\mathbf{X}$ has N samples and each sample has M dimensions. The output matrix Y belongs to $\mathbb{R}^{N\times C}$, where C is the number of labels. The network is built with n feature mapping nodes and m enhancement nodes. The *ith* group of feature mapping is given by

$$\mathbf{Z}_i = \phi_i(\mathbf{X}\mathbf{W}_{e_i} + \boldsymbol{\beta}_{e_i}), \quad i = 1, \ldots, n, \quad (2)$$

where $\mathbf{W}_{e_i}$ is a N by M matrix and $\boldsymbol{\beta}_{e_i}$ is a N by 1 vector. The entries of $\mathbf{W}_{e_i}$ and $\boldsymbol{\beta}_{e_i}$ are randomly generated. Function ϕ_i can be different for each feature mapping node. Let $\mathbf{Z}^n \equiv [\mathbf{Z}_1, \ldots, \mathbf{Z}_n]$ denote all the feature nodes, and denote the mth group of enhancement nodes as

$$\mathbf{H}_m \equiv \xi_m(\mathbf{Z}^n\mathbf{W}_{h_m} + \boldsymbol{\beta}_{h_m}), \quad (3)$$

where $\mathbf{W}_{h_m}$ and $\boldsymbol{\beta}_{h_m}$ are randomly generated. Similarly, functions ξ_j and ξ_k can be different for $j \neq k$. The constructed BLS network is formulated as

$$\begin{aligned}\mathbf{Y} &= [\mathbf{Z}_1, \ldots, \mathbf{Z}_n | \xi(\mathbf{Z}^1\mathbf{W}_{h_1} + \boldsymbol{\beta}_{h_1}),\\ &\quad \ldots, \xi(\mathbf{Z}^n\mathbf{W}_{h_m} + \boldsymbol{\beta}_{h_m})]\mathbf{W}^m\\ &= [\mathbf{Z}_1, \ldots, \mathbf{Z}_n | \mathbf{H}_1, \ldots, \mathbf{H}_m]\mathbf{W}^m\\ &= [\mathbf{Z}^n | \mathbf{H}^m]\mathbf{W}^m\\ &= \mathbf{A}_n^m\mathbf{W}^m, \end{aligned} \tag{4}$$

where the $\mathbf{W}^m$ is calculated by the pseudoinverse of $\mathbf{A}_n^m$, i.e. $\mathbf{W}^m = [\mathbf{Z}^n|\mathbf{H}^m]^+\mathbf{Y}$.

The pseudoinverse of the matrix can be approximately calculated by the Ridge regression [45].

$$[\mathbf{Z}^n|\mathbf{H}^m]^+ = \lim_{\lambda\to 0}(\lambda\mathbf{I} + [\mathbf{Z}^n|\mathbf{H}^m][\mathbf{Z}^n|\mathbf{H}^m]^T)^{-1}[\mathbf{Z}^n|\mathbf{H}^m]^T. \tag{5}$$

D. INCREMENTAL LEARNING BASED ON BLS

As introduced in [26], incremental learning is achieved in BLS by changing the network structure to insert new enhancement nodes or new feature nodes, or simply update the weights of network by a pseudoinverse method. An important characteristic of incremental BLS is that only the weights of new links need to be computed, while the existing links are left unchanged. In other words, the network is only partially updated with low computational demands. In contrast, conventional DL methods typically requires a computationally intensive update of the entire network parameters. When new data samples come in, a BLS incremental algorithm is applied to train the network. The training process, as originally proposed in [26], is adopted and explained here for the convenience of readers.

Denote $\mathbf{X}_a$ as the new input dataset. The incremental features and the corresponding incremental matrix can be updated as

$$\begin{aligned}\mathbf{A}_x &= [\phi(\mathbf{X}_a\mathbf{W}_{e_1} + \boldsymbol{\beta}_{e_1}), \ldots, \phi(\mathbf{X}_a\mathbf{W}_{e_n} + \boldsymbol{\beta}_{e_n})|\\ &\quad \xi(\mathbf{Z}_x^n\mathbf{W}_{h_1} + \boldsymbol{\beta}_{h_1}), \ldots, \xi(\mathbf{Z}_x^n\mathbf{W}_{h_m} + \boldsymbol{\beta}_{h_m})].\end{aligned} \tag{6}$$

Consequently, the entire BLS matrix, which includes both the historical and incremental information, is given by

$$\mathbf{U}_n^m = \begin{bmatrix}\mathbf{A}_n^m\\ \mathbf{A}_x^T\end{bmatrix}. \tag{7}$$

Matrix $\mathbf{U}_n^m$ is also updated by the pseduoinverse method, i.e.,

$$\mathbf{U}_n^m = [(\mathbf{A}_n^m)^+ - \mathbf{B}\mathbf{D}^T|\mathbf{B}], \tag{8}$$

where $\mathbf{D}^T = \mathbf{A}_x^T\mathbf{A}_n^{m+}$,

$$\mathbf{B}^T = \begin{cases}(\mathbf{C})^+, & if\ \ \mathbf{C} \neq 0\\ (\mathbf{1} + \mathbf{D}^T\mathbf{D})^{-1}(\mathbf{A}_n^m)^+\mathbf{D}, & if\ \ \mathbf{C} = 0,\end{cases} \tag{9}$$

and $\mathbf{C} = \mathbf{A}_x^T - \mathbf{D}^T\mathbf{A}_n^m$. The new weights of the updated model can be calculated as

$$\mathbf{V}_n^m = \mathbf{W}_n^m + (\mathbf{Y}_a^T - \mathbf{A}_x^T\mathbf{W}_n^m)\mathbf{B}, \tag{10}$$

where $\mathbf{Y}_a$ is the matrix of data labels corresponding to original data $\mathbf{X}_a$.

E. REFINEMENT OF BLS

The theoretical foundation of BLS can be traced back to the random vector functional-link neural network (RVFLNN) [44], which is a linear feature extraction method. In the original structure of RVFLNN and BLS, feature nodes are linear combinations of inputs with different weights. However, the initial weights of these feature nodes are randomly generated. The linear correlation between the weights of two feature nodes could become damaging to the performance [46]. Consequently, in this paper we propose to refine the BLS model by introducing random scaling factors and nonlinear mapping function to each feature window, thereby reducing the correlation between each feature window output. Heuristically, we multiply the outputs of feature nodes by different factors. If two feature nodes have higher correlation, the difference between their scaling allocated factors will be greater, such that greater effects of decorrelation can be achieved after passing through a subsequent non-linear activation function.

More specifically, a random scaling factor r_i is allocated to each $\mathbf{W}_{e_i}$, where $i = 1, \ldots, n$. The basic principle is that the greater correlation between two feature windows, the larger distance of the scaling factors is allocated. We let ρ_{ij} denote the linear correlation coefficient between $\mathbf{W}_{e_i}$ and $\mathbf{W}_{e_j}$, $i \neq j$. To avoid the undesirable situation where the distance of some scaling factor is too small, we further enforce the mean value of the distance to be greater than $\frac{(a-b)}{(n-1)}$, where $a \leq r_i \leq b$, $\forall i$. Base on the above principles, we introduce a linear optimization algorithm to generate the random scaling factor r_i. The optimization problem is formulated as

$$\max_{r_i}\ \sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1,j\neq i}^{n}\rho_{ij}\left|r_i - r_j\right| + \delta\sum\nolimits_{i=1}^{M}d_i$$
$$\text{st.}\ \begin{cases}a < r_i < b\\ \sum\nolimits_{i=1}^{n-1}\sum\nolimits_{j=i+1}^{n}\left|r_i - r_j\right| \geq \dfrac{1}{2}n(b-a),\\ M \leq \dfrac{1}{2}n\end{cases} \tag{11}$$

where there are M clusters of high correlation feature windows and δ is a penalty factor. Symbol $d_i, i \in [1, M]$ is the greatest correlation in corresponding cluster. After random scaling factors allocation, a nonlinear mapping function (i.e., activation function) is used to further reduce the correlation between feature nodes. The detailed procedure of correlation reduction is presented in Algorithm 1.

Finally, the above-introduced DIM and BLS functional blocks are combined to yield the proposed DIM-BLS model. Moreover, the DIM-BLS algorithm is further optimized for computational efficiency. The original BLS introduces an "enhancement layer" to perform single-layer non-linear transform on input data. The purpose of this layer is to let BLS acquire certain capability of extracting nonlinear features. In the proposed DIM-BLS algorithm, the non-linear features of data are sufficiently captured by DIM. Compared with DNN-based DIM, the enhancement layer in BLS is simplistic and redundant. Therefore, in the integrated DIM-BLS

Algorithm 1 Correlation Elimination of Feature Nodes in BLS

Initialization:

Random scaling factor $r_i, i \in [1, n]$;
Current number of clusters $M = 0$;
Penalty factor δ;
The set of classified feature window $S = \{s_i\}$;
correlation threshold of feature window e $= 0.8$;
Correlation coefficient matrix ρ;
Upper and lower bounds of random scaling factor [a,b];
The sum of the interval distances between the scaling factors $sum_{dis} = 0$.

Output:

Random scaling factors allocation result $r_i, \forall i$.

for $sum_{dis} < 0.5n(b-a)$ **do**
  Generate random factors r_i in interval [a, b];
  Calculate $sum_{dis} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} |r_i - r_j|$.
end for
for $i, j = 1 : n$ **do**
  if $\rho_{i,j} > e$ **then**
    for $K = 1 : M$ **do**
      if $iorj \in s_k$ **then**
        Add i, j in s_k.
      end if
    end for
    if $i \notin S$ and $M < 0.5n$ **then**
      $M++$
      Add i, j in s_M.
    else
      Add i in $s_{random(1,M)}$
    end if
  end if
end for
for $i = 1 : n$ **do**
  if $i \notin S$ and $M < 0.5n$ **then**
    $M++$
    Add i in s_M.
  else
    Add i in $s_{random(1,M)}$
  end if
end for
for $i = 1 : M$ **do**
  $d_i = max(s_i)$
end for
Optimize Eq.(11)

algorithm, the enhancement layer in BLS is removed to improve the overall computational efficiency. As shown in 5, two non-trivial refinements are introduced in the proposed DIM-BLS algorithm to exploit the synergy between DIM and BLS. The output of DIM will be combined with the result of feature nodes to form the feature dictionary of BLS. These feature nodes will be further implemented with random scaling factors to reduce the linear correlation.

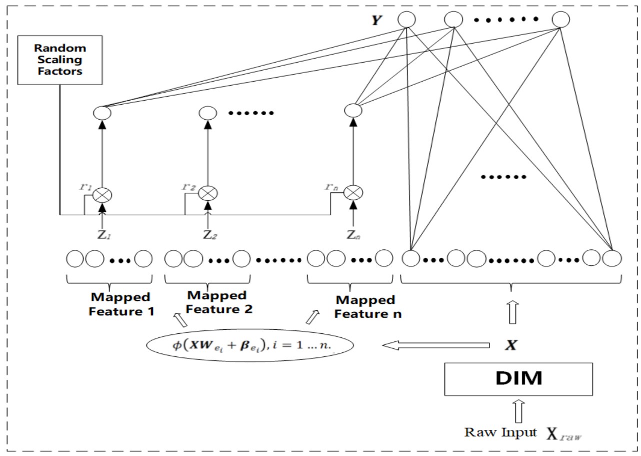

FIGURE 5. Illustration of the DIM-BLS structure.

In the offline stage, the DIM model is pre-trained by a large amount of unlabeled data. After that, labeled data samples are used to fine-tune DIM. The BLS then uses the output of DIM as extracted features to train its model. In the online stage, when new data stream arrives, the incremental BLS algorithm is used to update the entire model.

IV. EXPERIMENTS

In this section, the performance of the proposed HAR scheme is evaluated for both the offline and online training stages. First, we will show that the BLS-based methods outperform other machine learning methods in the offline training stage. In addition, we will show that the DIM-BLS model yields superior performance in the online training stage.

A. EXPERIMENTAL SETTINGS

Two public datasets are used for our performance evaluation: *WISDM* [47] and *HAPT* [48]. The *WISDM* dataset is collected from 36 users who use single Android phone to conduct simple activities, i.e., walking, sitting, jogging, standing, walking upstairs, and walking downstairs. The *HAPT*[1] data set is collected from 60 volunteers who perform daily activities that include six types of movements: walking, walking upstairs, walking downstairs, sitting, standing, and laying. Both data sets have been widely used in the HAR research field.

A total of six representative HAR schemes are evaluated and compared. Among them, two are BLS-based methods, two are DL-based methods, and two are conventional machine learning methods. The two BLS-based methods differ in feature engineering: one uses DIM for unsupervised feature extraction, while the other uses classic expert-crafted feature extraction methods proposed in [48]. The two DL-based methods used convolution neural network (CNN) and long-short term memory (LSTM) NNs as introduced in [49]. The two traditional machine learning methods also

[1] Here the *HAPT* dataset is the combination of two datasets (*HAR* and *HAPT*) collected by the same research team. We denote the collective data set as *HAPT*.

rely on expert-crafted features [48] and use support vector machine (SVM) [32] and random forest (RF) [18] as classifiers, respectively. We believe that these four methods are among the most representative methods in the HAR literature and can well-serve the purpose of performance comparison.

In our experiments, the BLS network is constructed by a total of 14×95 feature nodes and 1×11000 enhancement nodes. The input mapping functions $\phi_i(\cdot)$ are linear functions, while the activation function uses the *tansig* function [50] to reduce the correlation of feature nodes. Detailed parameter settings are presented in Table 1. All experiments are conducted on the same computing platform with a 3.20-GHz Intel i5-3470 CPU processor.

TABLE 1. Hyperparameters.

Models	Parameter	Value
DIM	learning rate λ	3e-4
	sparse factor	0.5
	feature dimension	512
	Optimization	Adam(Hu et al.,2009)
BLS	shrinkage scale s	0.8
	penalty factor δ	0,3
	regularization parameter	2^{-30}
SVM	penalty factor C	0.073
	gamma	30
	Kernel	RBF
Random Forest	number of estimators	500
	splitting criterion	gini
	max depth	10
Neural Network	learning rate λ	0.005
	number of epoch	300
	Loss Function	MSE

B. PERFORMANCE EVALUATION OF OFFLINE TRAINING

In the offline training stage, the labeled dataset is used to train all the six HAR schemes. Results on classification accuracy and training time are presented in Table 2 and Table 3. We observe that the two BLS-based models generally outperform other models in classification accuracy and training speed. Taking the *HAPT* dataset for example, the BLS-based model achieved over 98% overall classification accuracy, showing 1 to 4 percent improvements compared with CNN (94.12%), LSTM (93.46%), SVM (96.75%) and RF (92.67%). For each activity type, the highest accuracy is given by either the DIM-BLS model or the BLS model. Similar results are also observed for the *WISDM* data set, but with exceptions that CNN and LSTM yield the best performance in categories of jogging and walking down-stair, respectively. Nevertheless, the two BLS-based models still give the best overall performance.

In terms of training time, the BLS-based models are comparable to the traditional machine learning methods, but far superior to the DL-based methods. For fair comparison, the operation time of expert-crafted feature extraction is also taken into account. Take Table 2 for example, the best performance is given by SVM, which costs 389.66 seconds. This is closely followed by BLS and D-BLS, which costs 401.53 and 437.11 seconds, respectively. The training time increases almost two-times with the RF model and four-times with the two DL-based models. Experiments over the *WISDM* dataset give similar performances as shown in Table 3.

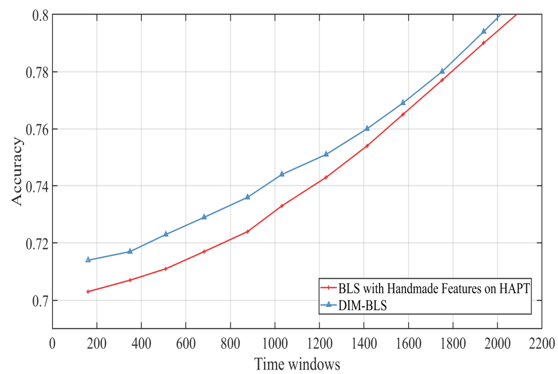

FIGURE 6. Comparison between DIM and expert-crafted features in incremental BLS on the *HAPT* data set.

From the above discussions, we can see that the two BLS-based models outperform other types of models in general, validating the feasibility and benefits of using BLS for HAR. However, at the first glance, it may seem that the DIM-BLS scheme does not perform better than the classic BLS scheme. This naturally raises some doubts on the value of DIM-BLS. To clarify such doubts, we argue that integrating DIM with BLS brings three key benefits. First, using DIM for unsupervised feature extraction eliminates the need of expert involvements in feature design. This means that the DIM-BLS model is a more general framework that is applicable to other tasks apart from HAR. Second, the potential of DIM is fully realized when a large unlabeled dataset is available for model training. Unfortunately, we do not have access to a large unlabeled HAR dataset at this moment. Still, we can see that even with a small data set, DIM is able to give good enough performance comparable to expert-crafted features. To further validate this point, we compare the learning curve of the two BLS-based schemes with DIM-generated features and expert-crafted using the *HAPT* data set. The results are shown in Fig.6, from which we can see that DIM-BLS works slightly better along the learning curve. Third, using mutual information as the criteria for feature extraction, DIM can better cope with the scenario of dynamic online training. In particular, DIM-BLS can better address the challenging task of incremental learning with biased data. Our next experiment is designed to demonstrate such a benefit.

TABLE 2. Performance comparisons in *HAPT*.

Activities \ Methods	% of Record Correctly Predicted					
	DIM-BLS	**BLS**	**SVM**	**RF**	**LSTM**	**CNN**
Walking	**99.34**	99.18	97.77	95.89	98.85	98.22
Upstairs	95.62	**95.86**	95.16	89.75	87.26	87.61
Downstairs	97.91	**97.98**	97.03	90.61	97.11	96.51
Sitting	**97.71**	97.45	96.28	92.93	90.97	89.64
Standing	**98.53**	98.52	96.35	94.28	88.62	88.43
Laying	100	100	97.12	92.47	100	100
Overall	**98.15**	98.06	96.75	92.67	94.12	93.46
Training time(s)	437.11	401.53	**389.66**	668.01	1344.32	1409.77

TABLE 3. Performance comparisons in *WISDM*.

Activities \ Methods	% of Record Correctly Predicted					
	DIM-BLS	**BLS**	**SVM**	**RF**	**LSTM**	**CNN**
Walking	96.45	**96.54**	95.26	92.88	93.38	93.74
Upstairs	93.95	**94.23**	92.98	89.90	93.86	93.77
Downstairs	90.97	91.99	89.78	88.76	**92.16**	92.03
Jogging	89.82	88.64	87.54	85.79	90.47	**90.55**
Siting	94.06	**95.19**	94.82	91.07	87.04	87.15
Standing	94.23	**95.47**	95.24	90.10	86.45	86.72
Overall	94.59	**95.14**	93.97	90.11	91.33	91.51
Training time(s)	401.03	**378.33**	380.61	642.74	1758.94	1686.16

C. PERFORMANCE EVALUATION OF ONLINE TRAINING

The baseline model obtained in the offline training stage is limited by the initial dataset used to train the offline model. The purpose of online training is to continuously improve the baseline model once user-generated labeled data (i.e., new data) can be obtained from an operating HAR system. In practice, the user generated data is often biased towards certain types of users. This means that the incoming data could have a different distribution with the initial data. The challenge of online training is to build a baseline model that performs well on both the initial data and new data.

By carefully rearranging the public dataset (e.g., *HAPT*), we are able to create datasets with desirable properties to simulate an online training scenario. Let $\mathbf{P}_A$ and $\mathbf{P}_B$ denote the initial dataset and new dataset, respectively. To simulate the case of data distribution bias, we use the *average unit step size* as a metric to divide the entire public dataset into two sub-datasets: the *long-step* sub-dataset and the *short-step* sub-dataset. Intuitively, these two sub-datasets could be interpreted as datasets for adults and children, respectively. Subsequently, we let $\mathbf{P}_A$ contain all data from the *long-step* sub-dataset and a few data from the *short-step* sub-dataset. On the other hand, we let $\mathbf{P}_B$ contain data mostly from the *short-step* sub-dataset. This simulates a scenario in which the initial dataset is biased towards adults, while new data mostly comes from children. Taking the *HAPT* data set for example, after such rearrangement, 35 users are included into $\mathbf{P}_A$ and 25 users are included into the $\mathbf{P}_B$.

Data sets $\mathbf{P}_A$ and $\mathbf{P}_B$ are used to evaluate the online learning performance of all the six HAR schemes. We first use dataset $\mathbf{P}_A$ to train a baseline model with respect to different schemes. It is then assumed that new (labeled) data from $\mathbf{P}_B$ arrives

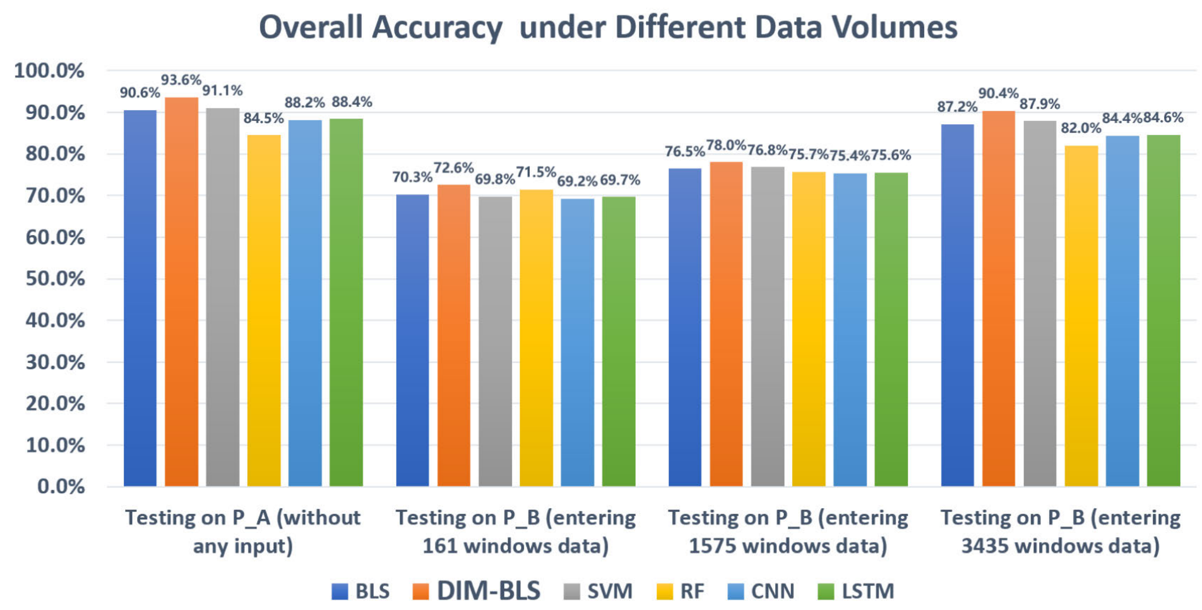

FIGURE 7. Overall accuracy under different data volumes.

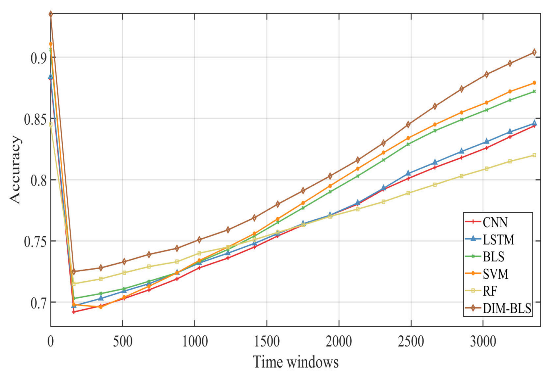

FIGURE 8. Accuracy curves of different incremental learning methods based on the *HAPT* data set.

one-by-one as streaming data. The baseline model is then updated using the new data and tested against all test samples from both $\mathbf{P}_A$ and $\mathbf{P}_B$ to calculate its accuracy.

Using the *HAPT* dataset, the overall performances of six algorithms are shown in Fig.7 and Fig.8. The X-axis in Fig.8 represents the length of time window, which starts from the moment just before new data $\mathbf{P}_B$ streams in. At the beginning (i.e., Time 0), the baseline model is trained on $\mathbf{P}_A$ and tested by $\mathbf{P}_A$. We can see that all six schemes have relatively high accuracy around 89.4%. Starting from Time 1, new data $\mathbf{P}_B$ comes in. A sharp fall of accuracy to around 70% is observed in Time 1 for all six schemes. This is an expected result, which indicates that models trained on $\mathbf{P}_A$ does not generalize well when the test set is expanded to also include $\mathbf{P}_B$. However, with the labeled data from $\mathbf{P}_B$ keep entering, all six schemes can gradually learn from the new data and improve their accuracy.

In Fig.7 and Fig.8, it is observed that the DIM-BLS scheme consistently outperforms other schemes. A closer look reveals that the DIM-BLS scheme establishes its wining margin at the early beginning of Time 1, where DIM-BLS has the best recognition accuracy rate of 72.6%, followed by 71.5% of RF and 70.3% of BLS. This means that DIM-BLS can better adapt to new data distribution and user types. Because the BLS scheme performs no better than DL-based schemes, we infer that the good model generalization capability of DIM-BLS comes from DIM.

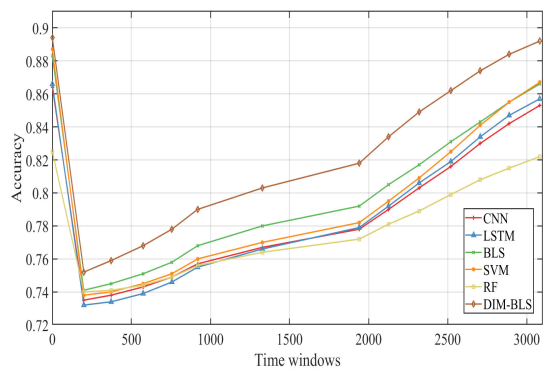

FIGURE 9. Accuracy curves of different incremental learning methods based on the *WISDM* data set.

The online learning performance are also evaluated for dataset *WISDM* and shown in Fig.9. Base on the *WISDM* dataset, there are 23 users in $\mathbf{P}_A$ and 13 users in $\mathbf{P}_B$. Results show that DIM-BLS consistently outperforms other schemes by almost 2% in terms of accuracy. This further confirms the advantage of the DIM-BLS scheme in the online learning scenario.

V. CONCLUSION

In this paper, a framework named DIM-BLS has been proposed for baseline model training in HAR. The framework has been designed to tackle two particular challenges in HAR baseline model training: insufficient training data and biased training data. It has been shown that DIM can overcome the first challenge by effectively learning useful representations from unlabeled data, while BLS can overcome the second challenge by means of fast incremental learning. A special technique called random factor scaling has been introduced to enhance the performance of DIM-BLS. Experiments on *HAPT* and *WISDM* datasets have shown that the proposed BLS-based scheme outperformed other types of machine-learning methods in the offline training stage of HAR. Moreover, the enhanced DIM-BLS scheme can better adapt to online learning scenarios with varying data distributions. We conclude that the proposed DIM-BLS framework is a promising scheme for online HAR baseline model training.

REFERENCES

[1] A. Prati, C. Shan, and K. I.-K. Wang, "Sensors, vision and networks: From video surveillance to activity recognition and health monitoring," *J. Ambient Intell. Smart Environ.*, vol. 11, no. 1, pp. 5–22, 2019.

[2] X. Ji, J. Cheng, W. Feng, and D. Tao, "Skeleton embedded motion body partition for human action recognition using depth sequences," *Signal Process.*, vol. 143, pp. 56–68, Feb. 2018.

[3] G. Batchuluun, J. H. Kim, H. G. Hong, J. K. Kang, and K. R. Park, "Fuzzy system based human behavior recognition by combining behavior prediction and recognition," *Expert Syst. Appl.*, vol. 81, pp. 108–133, Sep. 2017.

[4] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.

[5] R. Yao, G. Lin, Q. Shi, and D. C. Ranasinghe, "Efficient dense labelling of human activity sequences from wearables using fully convolutional networks," *Pattern Recognit.*, vol. 78, pp. 252–266, Jun. 2018.

[6] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1192–1209, 3rd Quart., 2013.

[7] Y. Bengio, "Deep learning of representations: Looking forward," in *Proc. Int. Conf. Statstical Lang. Speech Process. (SLSP)*, vol. 7978. Berlin, Germany: Springer-Verlag, 2013, pp. 1–37.

[8] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognit.*, vol. 108, Dec. 2020, Art. no. 107561.

[9] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Syst. Appl.*, vol. 105, pp. 233–261, Sep. 2018.

[10] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Activity recognition with evolving data streams: A review," *ACM Comput. Surv.*, vol. 51, no. 4, p. 71, Jul. 2018.

[11] A. Ignatov, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Appl. Soft Comput.*, vol. 62, pp. 915–922, Jan. 2018.

[12] P. Siirtola, H. Koskimäki, and J. Röning, "Personalizing human activity recognition models using incremental learning," 2019, *arXiv:1905.12628*. [Online]. Available: http://arxiv.org/abs/1905.12628

[13] F. J. O. Morales and D. Roggen, "Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations," in *Proc. ACM. Int. Symp. Wearable Comput.*, 2016, pp. 92–99.

[14] R. Ding, X. Li, L. Nie, J. Li, X. Si, D. Chu, G. Liu, and D. Zhan, "Empirical study and improvement on deep transfer learning for human activity recognition," *Sensors*, vol. 19, no. 1, p. 57, Dec. 2018.

[15] Q. V. Vo, M. T. Hoang, and D. Choi, "Personalization in mobile activity recognition system using K-medoids clustering algorithm," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 7, Jul. 2013, Art. no. 315841.

[16] Z. Wang, M. Jiang, Y. Hu, and H. Li, "An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 4, pp. 691–699, Jul. 2012.

[17] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Adaptive mobile activity recognition system with evolving data streams," *Neurocomputing*, vol. 150, pp. 304–317, Feb. 2015.

[18] 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, Jun. 2018.

[19] L. Mo, Z. Feng, and J. Qian, "Human daily activity recognition with wearable sensors based on incremental learning," in *Proc. 10th Int. Conf. Sens. Technol. (ICST)*, Nov. 2016, pp. 1–5.

[20] S. Ntalampiras and M. Roveri, "An incremental learning mechanism for human activity recognition," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–6.

[21] C. Hu, Y. Chen, X. Peng, H. Yu, C. Gao, and L. Hu, "A novel feature incremental learning method for sensor-based activity recognition," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 6, pp. 1038–1050, Jun. 2019.

[22] Q. Yang, Y. Gu, and D. Wu, "Survey of incremental learning," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 399–404.

[23] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019.

[24] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, "SMOTE-RSB*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory," *Knowl. Inf. Syst.*, vol. 33, no. 2, pp. 245–265, Nov. 2012.

[25] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Sep. 2019, pp. 2–25. [Online]. Available: https://openreview.net/forum?id=Bklr3j0cKX

[26] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[27] Z. Qin, Y. Zhang, S. Meng, Z. Qin, and K.-K.-R. Choo, "Imaging and fusing time series for wearable sensor-based human activity recognition," *Inf. Fusion*, vol. 53, pp. 80–87, Jan. 2020.

[28] H. He, Y. Tan, and W. Zhang, "A wavelet tensor fuzzy clustering scheme for multi-sensor human activity recognition," *Eng. Appl. Artif. Intell.*, vol. 70, pp. 109–122, Apr. 2018.

[29] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. Int. Cont. Artif. Intell. (IJCAI)*, vol. 15, 2015, pp. 3995–4001.

[30] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.

[31] S. Chung, J. Lim, K. J. Noh, G. Kim, and H. Jeong, "Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning," *Sensors*, vol. 19, no. 7, p. 1716, Apr. 2019.

[32] J.-H. Hong, J. Ramos, and A. K. Dey, "Toward personalized activity recognition systems with a semipopulation approach," *IEEE Trans. Human-Machine Syst.*, vol. 46, no. 1, pp. 101–112, Feb. 2016.

[33] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Scholkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, Jul. 2006.

[34] C.-H. Lu, Y.-C. Ho, Y.-H. Chen, and L.-C. Fu, "Hybrid user-assisted incremental model adaptation for activity recognition in a dynamic smart-home environment," *IEEE Trans. Human-Machine Syst.*, vol. 43, no. 5, pp. 421–436, Sep. 2013.

[35] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "MINE: Mutual information neural estimation," 2018, *arXiv:1801.04062*. [Online]. Available: http://arxiv.org/abs/1801.04062

[36] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15535–15545.

[37] A. Anand, E. Racah, S. Ozair, Y. Bengio, M.-A. Côté, and R. D. Hjelm, "Unsupervised state representation learning in Atari," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8769–8782.

[38] X. Liu, F. Zhang, Z. Hou, Z. Wang, L. Mian, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," vol. 1, no. 2, 2020, *arXiv:2006.08218*. [Online]. Available: http://arxiv.org/abs/2006.08218

[39] B. Yan and G. Han, "Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system," *IEEE Access*, vol. 6, pp. 41238–41248, 2018.
[40] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process.*, vol. 35, no. 1, pp. 53–65, Jan. 2017.
[41] C. L. P. Chen and Z. Liu, "Broad learning system: A new learning paradigm and system without going deep," in *Proc. 32nd Youth Academic Annu. Conf. Chin. Assoc. Automat. (YAC)*, May 2017, pp. 1271–1276.
[42] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*. [Online]. Available: http://arxiv.org/abs/1511.05644
[43] Z. Lin, H. Chen, Q. Yang, and X. Hong, "A flexible approach for human activity recognition based on broad learning system," in *Proc. 11th Int. Conf. Mach. Learn. Comput. (ICMLC)*, 2019, pp. 368–373.
[44] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, Apr. 1994.
[45] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, pp. 80–86, Feb. 2000.
[46] X. Gong, T. Zhang, C. L. P. Chen, and Z. Liu, "Research review for broad learning system: Algorithms, theory, and applications," *IEEE Trans. Cybern.*, early access, Mar. 17, 2021, doi: 10.1109/TCYB.2021.3061094.
[47] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Mar. 2011.
[48] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. Eur. Symp. Artif. Neural Netw. (ESANN)*, Apr. 2013, p. 3.
[49] H. Chen, F. Xiong, D. Wu, L. Zheng, A. Peng, X. Hong, B. Tang, H. Lu, H. Shi, and H. Zheng, "Assessing impacts of data volume and data set balance in using deep learning approach to human activity recognition," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2017, pp. 1160–1165.
[50] I. Sahin and I. Koyuncu, "Design and implementation of neural networks neurons with RadBas, LogSig, and TanSig activation functions on FPGA," *Electron. Electr. Eng.*, vol. 120, no. 4, pp. 51–54, Apr. 2012.

JIANYU XIAO received the B.E. degree in communication engineering from Xiamen University, Xiamen, China, in 2018, where he is currently pursuing the master's degree in communication and information system. His research interests include cloud computing, edge computing, and virtual resource scheduling.

LINLIN CHEN (Member, IEEE) received the B.E. degree in communication engineering from Xiamen University, Xiamen, China, in 2018, where she is currently pursuing the master's degree in communication and information system. Her research interests include cloud computing, artificial intelligence, and resource scheduling systems.

HAIPENG CHEN received the B.S. degree in communication engineering and the M.S. degree in information communication engineering from Xiamen University, Xiamen, China, in 2017 and 2020, respectively. His research interests include incremental learning and broad learning.

XUEMIN HONG (Member, IEEE) received the Ph.D. degree from Heriot-Watt University, Edinburgh, U.K., in 2008.

He is currently a Professor with the School of Informatics, Xiamen University, China. He has published over 60 articles in refereed journals and conference proceedings. His current research interests include cognitive communication networks and wireless localization systems.