Online Learning of Wearable Sensing for Human Activity Recognition

Yiwei Zhang, Bin Gao, *Senior Member, IEEE*, Daili Yang, Wai Lok Woo, *Senior Member, IEEE*, and Houlai Wen

***Abstract*—This article presents a novel semisupervised learning method for wearable sensors to recognize human activities. The proposed method is termed a tri-very fast decision tree (VFDT). The proposed method is a more efficient version of the Hoeffding tree and three VFDTs are generated from the original labeled example set and refined using unlabeled examples. Based on the heuristic growth characteristics of VFDT, a tri-training framework is proposed which uses unlabeled data to update the model without labeled data. This significantly reduces the computational time and storage of the data processing. In addition, the proposed method is embedded into wearable devices for online learning, while the test data flow is regarded as the unlabeled data to update the model. The experiment collects data stream of 16 min with motion state switching frequently while the wearable devices recognize motions in real time. An experimental comparison has also been undertaken for performance evaluation between the wearable and computation using a desktop computer. The obtained results show that only minor difference in terms of the *f*1-score rendered by the proposed method online or offline. This is a prominent characteristic for wearable computing within the Internet of Things (IoT). Data set can be linked as https://faculty.uestc.edu.cn/gaobin/zh_CN/lwcg/153392/list/index.htm.**

***Index Terms*—Online learning, real-time activity recognition, semisupervised learning, wearable device.**

I. Introduction

HUMAN wellbeing and mental evaluation play an important healthcare element in daily lives. Human activity recognition (HAR) is an emerging field in the healthcare context as it involves wellbeing and personal fitness tracking, monitoring of elderly and frail people, and assessment of rehabilitation. There are several approaches to capture human's behaviors and these include cameras, smartphones, and wearable devices. Wearable devices have recently played an increasing important role in HAR. The reason for such interest lies in many applications, which has been made possible by the progressive reduction of their physical size and costs. There are wide examples of wearable devices which include sport and physical activities [1]–[3], surveillance [4], human computer interaction [5], [6], rehabilitation [7], [8], and monitoring elderly people for ambient assisted living (AAL) purposes with the Internet of Things (IoT) [9], [10].

The applications of machine learning and deep learning algorithms for HAR are crucial. Machine learning methods, such as the kernel neural network (KNN) and support vector machine (SVM) have been used to recognize the daily activities [13]. In addition, the deep neural network (DNN) [14], convolutional neural network (CNN) [15]–[17], restricted Boltzmann machine (RBM) [18], [19], and recurrent neural network (RNN) [20] models are widely used [13]. As HAR is a time-series classification problem with local dependency, the LSTM network model is known to be well suited [14] as it utilizes the temporal correlations between neurons. Jiang and Yin [15] proposed an active learning approach based on deep CNN learning. Carneiro and Nascimento [21] constructed a multimodal model deep convolution neural network (DCNN) that it processes each sensor separately and extracts information from multiple temporal scales. Yver [22] proposed an algorithm that uses a temporal series self-characterization mechanism based on the use of a deep recurrent network (DRNN) to predict upcoming segments of temporal data.

HAR systems have achieved satisfactory results in terms of accuracy using machine learning and deep learning algorithms. Most methods, however, are designed for offline processing. This is especially the case in the field of healthcare services but it requires real-time decision. Online learning presents an alternative approach to solve the problems as it has capability to keep training the model on the wearable devices. This is possible by continuously reading the sensors' real-time data stream and using it for classification of the user's activity and further training. For current relevant online works, the lack of data is a problem [23], [24]. Semisupervised learning using unlabeled data is an effective direction. In [25], a relational *k*-means-based transfer semisupervised SVM learning framework (known as RK-TS3VM) is used to build a prediction model with labeled and unlabeled samples. Co-training is one of the major semisupervised learning paradigms that it iteratively trains two classifiers on

Manuscript received 29 January 2022; revised 17 April 2022; accepted 1 July 2022. Date of publication 6 July 2022; date of current version 21 November 2022. This work was supported in part by the China Postdoctoral Science Foundation under Grant 2021T140094, and in part by the National Natural Science Foundation of China under Grant 31800961. *(Corresponding author: Bin Gao.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the University of Electronic Science and Technology of China.

Yiwei Zhang, Bin Gao, Daili Yang, and Houlai Wen are with the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 610056, China (e-mail: bin_gao@uestc.edu.cn).

Wai Lok Woo is with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 7RU, U.K.

Digital Object Identifier 10.1109/JIOT.2022.3188785

different views and uses the predictions of either classifier on the unlabeled examples for data augmentation. It is a representative paradigm of disagreement-based methods [27]. Laine and Aila [28] proposed a tri-training algorithm that does not require sufficient and redundant views. In order to achieve better generalization performance, a co-training algorithm named COTRADE [29] is implemented. Miyato *et al.* [30] combined self-paced learning and co-training to adopt two different networks as classifiers and the final model behaved better performance. Noury *et al.* [31] proposed a temporary ensemble which uses the semisupervised method to train the neural network, introduced the self-ensemble mechanism, and calculated the mean square deviation by using the average of the current model prediction results and the historical prediction results. Chen *et al.* [32] proposed the mean teachers which takes into account the accumulation of labeled and unlabeled loss functions as the joint loss function so as to solve the problem for large-scale data sets. Bianchi *et al.* [33] used a virtual confrontation training (VAT) in semisupervised learning which is equivalent to regularizing the model and alleviating the overfitting problem.

When the size of data sets is large, online computation is a problem. The use of complicated machine-learning-based methods such as DNNs need to transfer the data to the server with the help of bluetooth or WiFi [34], [35]. Shi *et al.* [36] designed a wearable sensor which embeds an inertial measurement unit (IMU) and a Wi-Fi section to send data on a cloud service while coupling the sensor to a CNN. With IoT connectivity, many commercial products, such as Apple Watch, Fitbit, Microsoft Band, and smartphone apps are already available for continuous collection of physiological data. Jahanjoo *et al.* [37] embedded the Torch framework on smart phones to implement the deep learning model and used a powerful platform Intel Edison to demonstrate on-node activity classification. Nevertheless, since these experiments were tested in ideal environments and the hardware resources of most wearable devices are limited, designing embeddable algorithms with feasible computational cost is the target. The works aim to achieve the balance between accuracy and numerical complexity tradeoff. Reducing dimensionality of the input features vector is one way to reduce the computational time. Principal component analysis (PCA) and/or independent component analysis (ICA) [38], [39] are two well-known methods for dimensionality reduction. Chen and Guestrin [40] proposed a two-segment feature extraction method that calculates the features with low computational cost. Decision tree (DT) is a simple classification model since the tree-like model is easy to implement with low storage requirements and less computational complexity [41]. The DT model was embedded on the low-power microcontroller STM32 and it offered activity information in an acceptable accuracy [42]. The very fast DT (VFDT) uses a type of Hoeffding tree for streaming data mining. With increasing levels of noise added to the training examples, it embodies better accuracy and robustness than the C4.5 tree [43].

In this article, an online learning prediction model is proposed to recognize human daily activities and it is embedded on the wearable devices. The contributions can be drawn as follows.

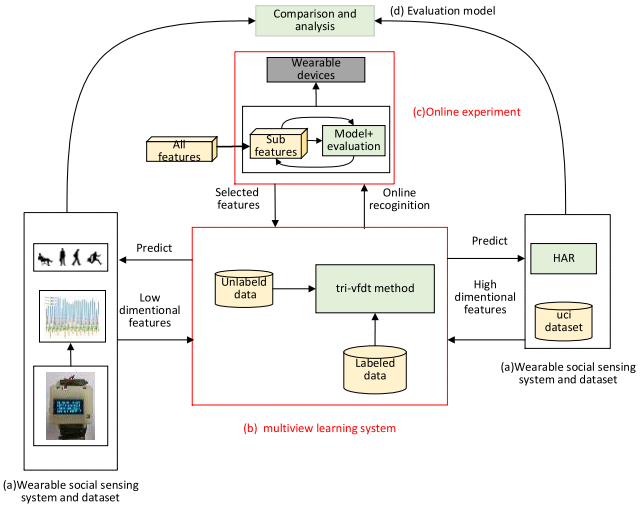

Fig. 1. Block diagram of activity recognition system.

1) Design of wearable devices with multiple sensors to collect human behavioral data. A semisupervised classification model tri-VFDT is proposed so that unlabeled examples could be used for classification and further training.
2) VFDT is selected as the base classifier to process streaming data. According to the characteristics of VFDT, the tri-training semisupervised framework is modified so that computing time and space are saved.
3) After filtering the high-dimensional behavior features with the simulated annealing [46] algorithm and initialization offline, the simplified model is embedded on the devices to learn and recognize the human's behaviors online.

This article is organized as follows: Section II presents the online learning system and the details of the proposed method. Experimental results and discussions are elaborated in Section III. Finally, we summarize the proposed work and identify the future work in Section IV.

II. Methodology

A. Description of Wearable Online Learning and Analysis Framework

In this section, the proposed framework for wearable social-sensing and online learning system will be presented. Fig. 1 illustrates the proposed system in four parts: 1) a wearable social perception system for behavior signal collection; 2) a multiview learning system; 3) online testing of wearable devices; and 4) comparison and evaluation.

In overall, the first part consists of a wearable social perception system. The second part proposes the tri-VFDT algorithm to extract the information of unlabeled data so as to optimize the original classifier. Moreover, in order to embed the algorithm in the wearable devices, the computational complexity has to be carefully reduced. The third part is online testing. In the experiment, the proposed model is applied to online learning and integrated into the wearable device to process the streaming data in real scene directly. The fourth part is evaluation.

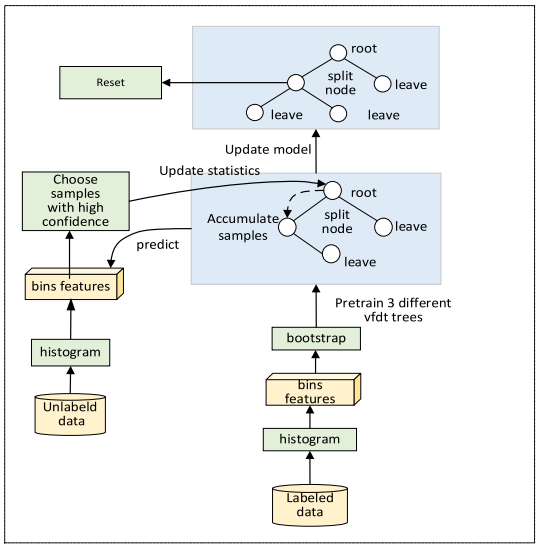

Fig. 2. Block diagram of tri-VFDT.

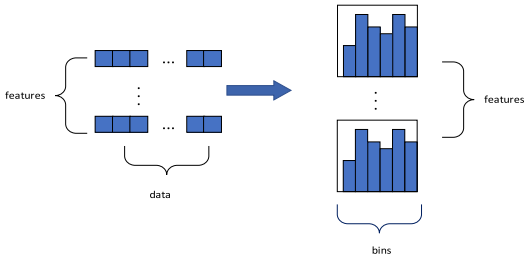

Fig. 3. Schematic diagram of the histogram algorithm.

B. Framework of the Proposed Online Learning Method

In this section, first, the proposed algorithm tri-VFDT is introduced, then the process of method used for online learning system is described.

In Fig. 2, we present our design of the semisupervised learning system. The proposed method focuses on optimizing computational complexity and storage when extracting the information of unlabeled data. A suitable classifier VFDT is applied as the base learner. According to the characteristics of VFDT, a new tri-training framework is proposed, and the histogram algorithm is applied for further feature simplification.

1) Histogram Algorithm: The histogram algorithm can be used to reduce computational complexity. Selecting the best segmentation points is time consuming for tree models, especially for continuous features. The histogram algorithm improves the efficiency of the greedy algorithm when the amount of data is too large to load the memory at one time.

Fig. 3 shows the schematic diagram of the histogram algorithm. The histogram algorithm discretizes continuous float numbers into k integers and transforms continuous features into discrete features. The discretized value is used as the index to accumulate statistics in the histogram. When looking for the best splitting point, k histograms are traversed. After feature discretization, the segmentation points can no longer be precisely predicted and this will affect the results. However, the performance on different data sets show that the discretization of the segmentation points does not have a great impact on the final accuracy whereas, occasionally, this even makes it more robust. Coarse segmentation points have regularization effects which effectively prevent model overfit and reduce the variance of prediction. Presorted is used in XGBOOST [40] to speed up the search for the optimal splitting point and the histogram algorithm is used in Lightgbm to speed up the computation [47]. Table I shows the performance comparison between the presorted and histogram algorithms. Through feature extraction, the data with dimension [30883 × 69] is input into the model, where the first number represents the size of the data, and the second number represents the number of features. Table IV shows the categories of input features, the acceleration value of x-, y-, and z-axes extracted from the features from each category, respectively, After the "bins" processing, the dimension of the sample changes from to [2987 × 69], which is 1/10 of the original data. For data storage, the original storage is float and the storage space of each float is 4 bytes. After the bins processing, continuous values are mapped to discrete values and into different bins. The number of each bin is stored with an integer, and the storage space is 1 byte. The data storage s is now changed from (30883 × 69 × 4) to (2987 × 69 × 1).

TABLE I
COMPARISON BETWEEN PRESORTED AND HISTOGRAM

Performance comparison	Pre-sorted	histogram
Memory consumption	$2 * \#features * \#data * 4Bytes$	$\#feature * k * 1Bytes$
Statistics accumulation	$O(\#data * feature)$	$O(\#data * feature)$
Information gain calculation	$O(\#data * feature)$	$O(k * \#feature)$
Cache status	occasionally miss cache	none

First, a column of features is sorted. The total number of samples is 30 883, which is called *total_cnt*. We set the minimum number of samples contained in the bucket, *min_data_in_bin* = 10. We make *distinct_values* as an array that holds feature values, which increase monotonically. We set the array count as the number of samples corresponding to the value of the feature in *distinct_values*. We make *num_distinct_values* the number of feature values, and its value is the length of *distinct_values* . We set max_*bin* = 2987, which is the maximum number of buckets.

When the feature values have zero, all zeros are divided into one bucket. Let the number of positive numbers, negative numbers, and zero be right, respectively, *right_cnt_data*, *left_cnt_data*,*cnt_zero*. Here is the numbers of positive and negative buckets, respectively

$$left\_\max\_bin = int\left(\frac{left\_cnt\_data}{total\_cnt - cnt\_zero} * (\max\_bin - 1)\right)$$
$$right\_\max\_bin = \max\_bin - 1 - left\_\max\_bin)).$$

When this happens, the two boundaries of each bucket only need to be found out. Through the relationship of the value of each element in the counts array and *max bin*, the two boundaries are set. In this way, the boundaries of all buckets are calculated and stored in an array *bin upper bound* as the dividing point group. According to the dividing point group of each column of features, the value of the feature can be converted to the number of the bucket, and input into the model for calculation. In this way, the continuous value can be converted into a discrete value.

2) Base Classifier VFDT: To solve the problem of limited data storage, VFDT learns by seeing each example only once, instead of reading the entire data set repeatedly and sequentially. Therefore, examples from an online stream are not stored while parameters are stored instead. Given a stream of examples, once the root attribute is chosen, the succeeding examples will be passed down to the corresponding leaves to choose the appropriate attributes iteratively. The Hoeffding bound (HB) [44] is applied to determine the necessary number of examples at each node which acts as a threshold for tree pruning based on certain conditions. For n independent observations of a real-valued random variable r whose range is denoted as R, the HB illustrates that with confidence $1-\delta$ the true mean of r is at least $\bar{r}-\varepsilon$ where

$$\epsilon = \sqrt{\frac{R^2 ln(1/\delta)}{2n}}. \tag{1}$$

HB has the attractive property that it is independent of the probability distribution generating the observations. With this property, when VFDT is used to process streaming data, it is applicable to HB regardless of the data distribution of real-time data. Similar as a batch learner, it trains and updates the initial model once a period of time. With the increase of epochs, the tree becomes deeper.

3) Improved Tri-Training Framework: Tri-training is considered as a multiview method. Compared with co-training, the tri-training neither requires different supervised learning algorithms that partition the instance space into a set of equivalence classes nor does it use tenfold cross validation on the original labeled example set. Three identical initial classifiers are refined during the tri-training process, the final hypothesis is produced via majority voting. Unlabeled samples with high confidence of prediction results are selected to expand the training set while the model is refined until the prediction error guaranteed. Labeled and unlabeled data are input to the system, it labels the unlabeled data through three classifiers.

Labeled data are separated to *train_X* and *val_X* as the training set and verification set, respectively, and their labels are *train_y* and *val_y*, that is, $labeled\_X = train\_X \cup val\_X$, $labeled\_y = train\_y \cup val\_y$, and the training set and test set are segmented 8:2.

Assuming that three classifiers h_i, h_j, and h_k are initially trained from $train\_X \cup train\_y$, the function MeasureError(h_j, h_k) attempts to estimate the classification error rate, $e[i]$, of the hypothesis derived from h_j and h_k

$$\text{MeasureError}(h_j, h_k) = \frac{\sum h_j(x) \neq val\_y}{\sum h_j(x) == h_k(x)}. \tag{2}$$

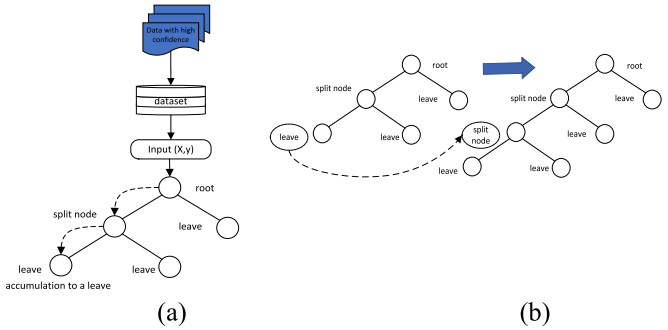

Fig. 4. Tri-VFDT updating process.

In the basic form of tri-training, *labeled_X* is used to pretrain the model and calculate $e[i]$ for each iteration but this can lead to overfitting. Therefore, in our approach, a new verification data set *valX* is used to evaluate whether the accuracy of the base classifier is improved. If $e[i]$ is less than the classification error rate of the previous round $e\_prime[i]$, the classification accuracy of $h_j \cup h_k$ is improved (since the initial value of $e\_prime[i]$ is 0.5, $e[i]$ is derived from the first iteration and compared with the initial value 0.5). In the tth iteration, the samples with the same prediction results of h_j and h_k are selected as high confidence samples to expand the sample set L_i^t for h_i (L_i^t is an empty set originally). The update sample sets L_j^t and L_k^t for h_j and h_k are expanded, respectively. After expanding all sample sets, the three classifiers are updated with the updated sample sets L_i^t, L_j^t, and L_k^t, respectively.

After updating h_i, let $e\_prime[i] = e[i]$. In a new round of iteration,$e[i]$ is calculated by the updated h_j and h_k. Comparing $e[i]$ with $e\_prime[i]$, if $e[i]$ is less than $e\_prime[i]$, h_i continues to improve and the iteration continues. If $e[i]$ is greater than $e\_prime[i]$, the update process of h_i is terminated. When h_i, h_j, and h_k terminate the updating step, the training process ends. After batch data input, the base classifier is updated.

Our approach takes on the integration of VFDT with the tri-training framework. Compared with the traditional machine learning algorithm, our approach uses a smaller amount of effective data to train the model. Fig. 4 shows the training process of the model.

Fig. 4(a) illustrates the process of data input and accumulation. Once the update samples are selected, the samples are input into VFDT in the form of one entry per time. Fig. 4(b) shows the process of node splitting. When the splitting condition satisfies Hoeffding's inequality, the node splits. After splitting, the data accumulated in the node will be deleted. In the tth round of iteration, the original tri-training updates with $labeled\_X \cup L_i^t$, the proposed method updates part of the classifier with new batch input data L_i^t. The newly generated samples are put into the root of the tree as it directly transferred to the leaf and accumulated in the leaf. The new samples directly determine whether the leaf is split, rather than all data being used to determine the nodes split. The theoretical proof as to why the accuracy of a single classifier VFDT has improved is illustrated as follows.

Three different sample sets are generated by bootstrap to initialize three base classifiers h_i, h_j, and h_k. L is the original

labeled example set, S_i is sampled from L, and the sampling rate is 0.8.

Assuming the size of *labeled_X* is L, which is divided into training set *train_X* and verification set *val_X*, h_i, h_j, and h_k are the three VFDT trees that have been pretrained

$$L = train\_X \cup val\_X. \tag{3}$$

Inspired by the approach taken from Goldman and Zhou [42] with the finding of Angluin and Laird [43], we can show the following analysis: if a sequence σ of m samples is drawn, where the sample size m satisfies the following:

$$m \geq \frac{2}{\epsilon^2(1-2\eta)^2}\ln\left(\frac{2N}{\delta}\right) \tag{4}$$

where ϵ is the hypothesis worst case classification error rate, η (< 0.5) is an upper bound on the classification noise rate, N is the number of hypotheses, and δ is the confidence. Let $c = 2\mu\ln(2N/\delta)$, where μ makes (1) hold equality, then (4) becomes

$$m = \frac{c}{\epsilon^2(1-2\eta)^2}. \tag{5}$$

To simplify the computation, it is helpful to compute the quotient of the constant c divided by the square of the error

$$u = \frac{c}{\epsilon^2} = m(1-2\eta)^2. \tag{6}$$

Let η_L denote the classification noise rate of L, that is, the number of examples in L be mislabeled in η_L. Assuming that h_j and h_k make the same classification on z examples, among these examples, h_j and h_k make correct classification on z' examples, then e_i^t can be estimated as

$$e_i^t = \frac{(z-z')}{z}. \tag{7}$$

Thus, the number of examples in L_i^t that are mislabeled is $e_i^t L_i^t$. Therefore, the classification noise rate in the tth round can be denoted as

$$\eta_t = \frac{\eta_L|L| + e_i^t|L_i^t|}{\left|L\bigcup L^t\right|}. \tag{8}$$

Since the base classifier is VFDT, each node is trained independently by a part of the input data, the adjacent nodes of the tree will not interact with each other while the overall accuracy is determined by the whole tree. In the tth iteration, according to (6)

$$u^t = \left|L\bigcup L_i^t\right|\left(1-2\frac{\eta_L|L| + e_i^t|L_i^t|}{\left|L\bigcup L_i^t\right|}\right)^2. \tag{9}$$

Similarly, u^{t-1}can be computed as

$$u^{t-1} = \left|L\bigcup L_i^{t-1}\right|\left(1-2\frac{\eta_L|L| + e_i^t\left|L_i^{t-1}\right|}{\left|L\bigcup L_i^{t-1}\right|}\right)^2. \tag{10}$$

As shown in (6), since u is in proportion to $(1/\epsilon^2)$, it can be derived t if $u_t > u_{t-1}$,then $\epsilon_t < \epsilon_{t-1}$,which implies that h_i can be improved through utilizing L_i^t. This condition can be expressed in (11) by comparing (9) and (10)

$$\left|L\bigcup L_i^t\right|\left(1-2\frac{\eta_L|L| + e_i^t|L_i^t|}{\left|L\bigcup L_i^t\right|}\right)^2 > \left|L\bigcup L_i^{t-1}\right|\left(1-2\frac{\eta_L|L| + e_i^t\left|L_i^{t-1}\right|}{\left|L\bigcup L_i^{t-1}\right|}\right)^2. \tag{11}$$

Since $|L^t| > |L^{t-1}|$, if $e_1^t < e_1^{t-1}$, then $\epsilon^t < \epsilon^{t-1}$. Therefore, in the process of training, h_i improves the accuracy. Since the final result is produced by voting, the accuracy of the tri-VFDT model is improved by t iteration is compared to $(t-1)$ iteration. The pseudocode of the proposed tri-VFDT algorithm is presented in Table II.

TABLE II
PSEUDOCODE OF TRI-VFDT ALGORITHM

Algorithm: $tri-VFDT$
Input: L: Original labeled example set
U: Unlabeled example set
$learner$: vfdt model
for $i\epsilon\{1,2,3\}$ do
$S_i \leftarrow BoostrapSample(L)$
$h_i \leftarrow learn(S_i)$
$e_i' = 0.5, l_i' = 0$
end of for
repeat until none of $h_i(i=\{1,2,3\})$ changes
for $i\epsilon\{1,2,3\}$ do
$L_i \leftarrow \phi; update_i \leftarrow False$
$e_i \leftarrow MeasureError(h_j \& h_k)(j,k \neq i)$
if $e_i < e_i^{'}$
then for every $x \in U$ do
If $h_i(x) = h_k(x)(j,k \neq i)$
Then $L_i \leftarrow L_i \cup \{(x, h_j(x))\}$
then $update_i \leftarrow True$
end of for
for $i \in \{1,2,3\}$ do
If $update_i = True$
then $h_i \leftarrow update(L_i); e_i' = e_i;\ l_i' = |L_i|$
end of for
end of repeat
Output: $h(x) = argmax_y \sum_{h_i(x)=y} 1$

4) Feature Selection: Feature selection is required before integrating the model into a wearable device. Selecting the important features alleviates the dimension disaster problem and reduces the difficulty of the learning task.

The wrapper method is a popular feature selection method and regards the selection of subsets as a search optimization problem. As a kind of wrapper method, the simulated annealing algorithm is a stochastic optimization algorithm based on the Monte Carlo iterative solution. It starts from a higher initial temperature with the continuous decline of temperature parameters since it combines with the probability jump characteristics to randomly find the global optimal solution of the objective function in the solution space.

The initial temperature T_0 is set to 100, and the final temperature is T_f is set to 1, the superparameter α is 0.95, and the training set and test set are segmented 8:2. The VFDT is

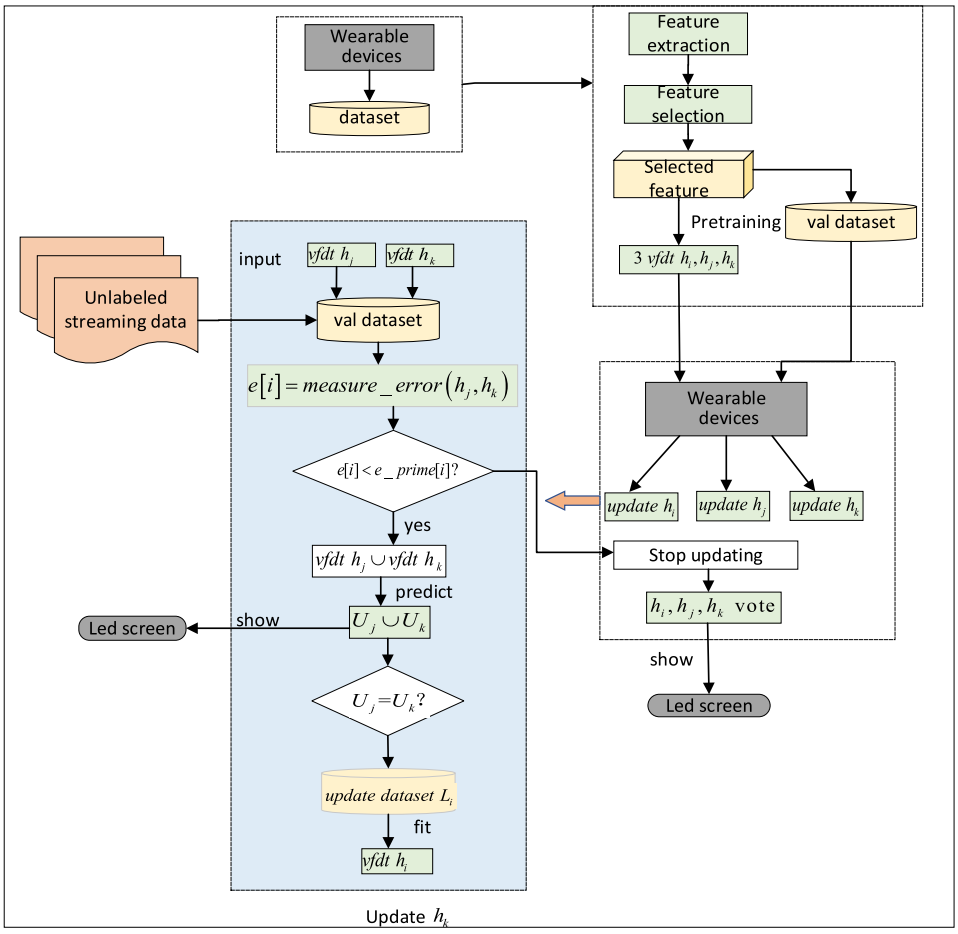

Fig. 5. Proposed online learning system.

used as the learner, and the solution space M maps all features. In each iteration, n features randomly selected from all features (n is artificially defined) that are mapped to a feasible solution m. The $f1$-score of the learner is used as the value of the objective function $E(m)$. Different from the fitting residuals, the $f1$-score is not a penalty attribute, it needs to be increased. In the next iteration, several features are selected from the unselected features and exchanged with the selected features to generate a new feasible solution m', and the energy change is calculated as $\Delta E = E(m') - E(m)$. Different from the classical simulated annealing algorithm, the purpose of feature selection is to increase $E(m)$. Therefore, when $\Delta E > 0$, m' is accepted as the new solution, otherwise, the new solution m' are accepted with the probability of $p = e^{-[\Delta E/(kT)]}$. Let $T = \alpha T$, if the termination condition is satisfied, that the temperature reaches the minimum temperature ($T \leq T_f$), the currently selected feature is output as the optimal solution and the program is terminated. On the contrary, iteration is continued until the temperature is cooled to the final temperature. Table II shows the selected features. The accuracy of a single VFDT tree is changed from 90% to 86%. The accuracy is reduced by 4%, and the dimension of features is reduced from 69 to 5. The 69-D features applied are described in Table V and Section III. The selected features are shown in Table III.

TABLE III
SELECTED FEATURES

Class	Feature	Description
Time domain	min_{az}	Max value of samples for z axis
	min_{ax}	Min value of samples for x axis
Frequency domain	std_{af}	Standard deviation of samples for combined acceleration
	$mean_{af}$	Mean value of samples for combined acceleration
	$max_{\|az\|}$	Max value of samples for z axis

5) Online Learning Based on Wearable Devices: Fig. 5 shows the scenario of applying the above tri-VFDT method to the online learning system. Three VFDT trees are trained offline with the selected features, which are used as the initialization of the base learner. The parameters of the model are transferred onto wearable device. The streaming data is put into the model as unlabeled data. Suppose h_i is the learner to be updated while the other two base learners are h_j and h_k, respectively. A batch of data, after collected, is predicted by h_j and h_k. The prediction result is recorded as U_j, U_k, one of

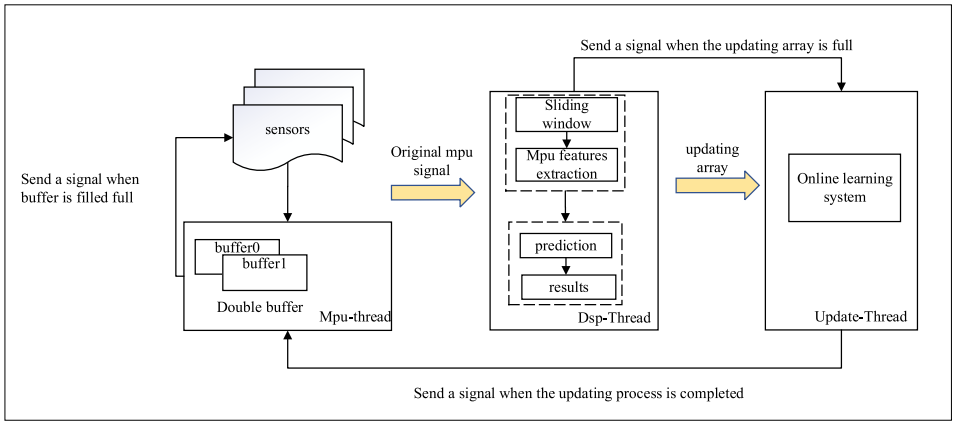

Fig. 6. Proposed implementation of the firmware.

which is displayed on the LED screen. After comparison, the samples with the same value are selected as the samples with high confidence probability to fill the sample set.

The three base classifiers continue to be updated until convergence to steady state. When the three classifiers stop updating, the current iteration ends. When the updating process terminates, a new batch of data is collected.

Once the convergence condition (accuracy is 95% for a period of time or the depth of VFDT tree is 6) is satisfied, the updating process stops and the classifier is used for pattern recognition. Through voting, the final result is stored and displayed.

The implementation diagram of the firmware is shown in Fig. 6. The double buffer caching mechanism is developed to enhance the recording stage and reduce power consumption. Mpu data are collected through Mpu-thread. The original Mpu signal is sent to Dsp-Thread for preprocessing and features extraction. Updating_array is set to transfer the expanded sample set. After the behavior recognition in the Dsp-Thread, the result is filled into the array. When the updating_array is full, it is passed to the online learning system for online processing. After Update-thread is finished, a signal is sent to Mpu-thread to collect data.

III. Evaluation and Analysis

A. Experiment Setup and Data Set Preparation

In this experiment, the $f1$-score is used to measure the prediction results. It can be regarded as a weighted average of the Precision and Recall. Thus, the $f1$-score helps objectively analyze the performance of the classifier. Precision and Recall are defined as

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \tag{12}$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{13}$$

where tp is true positive, fp is false positive, and fn is false negative. Precision is the fraction of correctly predicted positive samples to the total predicted positive samples. Recall can be represented as the fraction of correctly predicted positive samples to the total positive samples in actual label. The

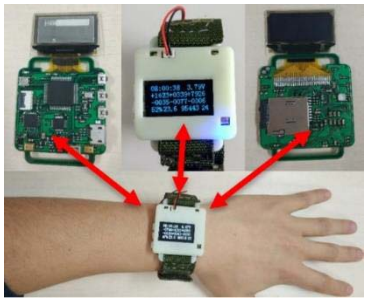

Fig. 7. Hardware platform of the self-designed wearable device.

$f1$-score is calculated by using the Precision and Recall with the same weight

$$f1 - \text{score} = 2\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{14}$$

The experiments will be carried out on a wearable social sensing platform. The self-designed wearable device and its related hardware platform is shown in Fig. 7. The microprocessor is an ARM-Cortex4 microcontroller with a DSP function and the model is STM32F405. Besides, the sensors system consists of a 6-axis behavior sensor (MPU6050) and the temperature and humidity sensor (SI7021), which collect physiological signals and behavioral activity. The display module is an OLED screen. In order to record large amounts of data and save it, the wearable device contains a power management unit with a 2200-mAh lithium battery and a micro SD card.

We designed three experiments, which are offline experiment, continuous data acquisition experiment, and online recognition experiment based on wearable devices. The offline experiment will be carried out on DATA SET I and DATA SET II.

DATA SET I is collected with 60 volunteers within an age bracket of 19–48 years (30 males and 30 females) from the University of Electronic Science and Technology. The wearable devices is worn at volunteer's preferred wrist to collect behavioral signals of six kinds of activities, including sitting, standing, walking, cycling, lying, and running, each for 5 min. Five 5-min recorded CSV files are stored in the SD card,

TABLE IV
DESCRIPTION OF DATA SET I

Data type	Sampling rate	Data fragment length	Data size
3-axis acceleration	125Hz	5 minutes	$5 \times 60 \times 3$
3-axis angular acceleration	125Hz	5 minutes	$5 \times 60 \times 3$

TABLE V
DESCRIPTION OF ACTIVITY FEATURES

Category	Id	Feature	Description
Time domain	1	Mean	Average value of samples in a window
	2	Std	Standard deviation of samples in a window
	3	Min	Minimum of samples in a

TABLE VI
DESCRIPTION OF DATA SET II

Data type	Sampling rate	Data fragment length	Data size
3 axis acceleration	50Hz	20 minutes	$5 \times 60 \times 3$
3 axis angular acceleration	50Hz	20 minutes	$5 \times 60 \times 3$

representing the behavior data of each movement state. The specification of the DATA SET I is shown on Table IV.

In the experiment, the behavioral data consist of three-axes acceleration data and three-axes angular velocity data. This 6-axis data are used to extract behavioral features using the sliding window. The time interval between the sliding windows is 3 s. In addition, the features are divided into time-domain features and frequency-domain features as listed in Table V. The x-, y-, and z-axes and angular velocity are calculated, respectively. The accelerations of x-, y-, and z-axes and combined angular acceleration are calculated, respectively. For the characteristics of x-, y-, and z-axes linear acceleration and angular acceleration, 18 kinds of characteristics in Table V are calculated, respectively, and a total of 72-D features are generated. According to the experimental analysis, the features that have a negative impact on the classification accuracy by several dimensions are removed, and finally changed from 72 dimensions to 69 dimensions.

The public data set HAR Using Smartphones Data set from UCI [45] is used as DATA SET II. Table VI shows the summary of the original data information of each volunteer.

Online experiment is carried out with self-designed wearable devices. The behavioral data are collected with 22 volunteers (14 males and 8 females) within an age bracket of 24–30 years. In the process of online experiment, the function includes data collection, online calculation, real-time recognition, and result display. In order to prove the effectiveness and robustness of the proposed method, the 16-min continuous experiment involves multiple transitions of different states of motion. The specific process is as follows: standing for 2→min, sitting for 2→min, walking for 3→min, running for 3→min, walking for 2→min, sitting for 3→min, and standing for 1→min.

TABLE VII
DESCRIPTION OF DATA SET III

Data type	Sampling rate	Data fragment length	Data size
3 axis acceleration	125Hz	16 minutes	$4 \times 16 \times 60 \times 3$
3-axis angular acceleration	125Hz	16 minutes	$4 \times 16 \times 60 \times 3$

TABLE VIII
TRAINING TIME WITH LABELED AND UNLABELED DATA

Dataset	Tri-DT	Tri-VFDT(no histogram)	Proposed method
Dataset I	17.80s	13.13s	8.47s
Dataset II	38.49s	29.81s	20.01s

DATA SET III is used as a comparison with online experiment. In order to form a control experiment, same volunteers completed the above process. The classification accuracy is compared with online experiment. The specification of DATA SET III is shown on Table VII.

B. Results and Analysis

In this section, in order to evaluate the proposed method, several similar methods are selected for comparison. The impact of unlabeled data will also be discussed. Labeled data are input to DT, random forest, and VFDT. We use DT as the base learner, tri-training as the semisupervised framework (tri-DT), and unlabeled data are input as data augmentation. The classification effect, time complexity, and storage of the model will be analyzed.

1) Time Complexity and Storage: Table XI shows the storage costs and time complexity of several related algorithms where M is the number of trees, m is the data dimension, n is the feature dimension, and d is the depth of a tree, v is the maximum number of values per feature, c is the number of classes, l is the number of leaves in the tree, n_leave is the total number of data at leave, $n_{\min}$ is the minimum sample number at a leave, and s is the number of splits. Since VFDT stores data in dictionary form, the storage is independent of the number of examples seen. Through the histogram algorithm, the amount of data is further reduced. As in Table XI, since $lnkc \ll n$ and $mn_vkc \ll nd$, the storage and time complexity are reduced in the proposed method.

Table VIII demonstrates the proposed tri-VFDT is more advantageous than tri-DT methods in terms of speed with unlabeled data. In DATA SET I and DATA SET II, the proposed method uses almost half the time of tri_DT to complete the training process.

2) Model Results and Analysis: The results of behavior recognition on DATA SET I and DATA SET II are shown in Tables IX and XI, respectively.

In Table IX, the $f1$-score of VFDT is higher than DT on each movement state except for walking. The result of tri-dt is higher than that of DT, which shows the effect of unlabeled

TABLE IX
$f1$-SCORE OF DATA SET I(%)

motion	Decision Tree	VFDT	Random Forest	Tri-DT	Proposed
walking	85.37	91.36	93.31	88.15	92.92
running	93.62	95.85	95.86	93.39	97.05
biking	91.33	90.65	92.35	91.24	92.45
standing	79.64	85.96	90.39	88.72	90.32
sitting	84.93	76.62	87.75	85.82	83.54
Lying	85.49	89.40	92.34	88.54	92.21
macro-average	86.01	88.97	92.37	88.98	92.32
weighted-average	85.37	89.36	93.31	88.15	**93.74**

TABLE X
$f1$-SCORE OF DATA SET II(%)

motion	Decision Tree	VFDT	Random Forest	Tri-DT	Proposed
walking	86.45	79.46	92.23	89.74	82.53
walking upstairs	66.49	72.18	89.89	83.72	75.1
walking downstairs	78.04	84.07	93.67	87.42	86.98
sitting	69.76	67.25	75.97	67.16	71.54
standing	78.43	76.97	81.00	75.78	77.76
lying	90	93	97	96	100
macro-average	79.86	80.15	88.79	83.97	83.19
weighted-average	80.04	79.99	88.2	82.41	82.95

TABLE XI
COMPARISON OF COMPUTATIONAL COMPLEXITIES OF RELATED METHODS

Method	Random Forest (CART)	Decision Tree (CART)	VFDT	Proposed
Time complexity	$O\left(M\sqrt{m}nd\right)$	$O(mnd)$	$O(n_v mvc)$	$O(mn_v kc)$
storage	$O(n+Ms)$	$O(n+s)$	$O(lmvc+s)$	$O(lnkc+s)$

data. For the proposed tri-vfdt model, compared with tri-DT and VFDT, the $f1$-score is significantly improved. It shows that when a large number of data update the VFDT tree, the $f1$-score will continue to improve. The random forest composed of ten DTs is selected as the reference model. The algorithm proposed in this article consists of three Hoeffding trees, and the $f1$-score is close to the random forest in the case of low cost and time.

In Table X, after updating the unlabeled data, the average $f1$-score of the model is improved. The average $f1$-score of the model is improved after updating the unlabeled data. For the motions that are difficult to recognize such as sitting, when compared with DT, the $f1$-score of tri-DT decreases. This shows that the tri-training framework is greatly affected by the base learner when selecting samples with high confidence to update the model. If the original learning ability of the classifier is poor, the "wrong selection rate" in the selection process is high. During the updating process, the errors accumulate resulting in worse results after iteration. The proposed algorithm limits the sample error rate. If the classification accuracy is not improved, the updating process stops, which maintains the original accuracy.

Compared with Tables VIII and XI, the influence of the model changes with the amount of data. The input size of DATA SET I is 7352×561, and the input size of DATA SET II after feature extraction is 30883×49. When the amount of data increases and the feature dimension decreases, the accuracy of the VFDT model is higher than that of DT. The reason is that VFDT grows heuristically, which means the number of layers cannot be set artificially and is affected by the amount of data. When the amount of data is small, the number of tree layers is low, resulting in poor classification results. When the evaluation index of the base learner decreases, the updating process stops immediately. Therefore, when the $f1$-score of VFDT is low, the $f1$-score of tri-VFDT is close to it; when the classification effect of VFDT is better, the improvement of tri-VFDT is more obvious.

The proposed method has significant better performance than DT and VFDT in terms of accuracy computational time and storage. In addition, the time complexity and storage of the proposed method are much smaller than the random forest. With less parameters, the proposed model is convenient to be embedded on wearable devices and achieves online computation.

3) Discussion on Parameter Configuration: In this section, the impact of hyperparameters on accuracy and computation time is discussed. The experiment will be set on DATA SET I. First, the analysis will focus on the impacts of the ratio of unlabeled data to all the data; second, the effects of important parameters of VFDT are studied.

For the first part, by changing the ratio of labeled data to unlabeled data, the model effects of tri-DT and the proposed

TABLE XII
DIFFERENT PROPORTIONS OF UNLABELED DATA (%)

proportion	20%		40%		60%		80%	
motion	tri-DT	proposed	tri-DT	proposed	tri-DT	proposed	tri-DT	proposed
walking	87.15	91.92	88.70	90.23	89.38	84.05	89.47	84.14
running	93.39	96.05	93.99	95.02	94.51	91.70	93.33	91.67
biking	90.24	92.45	91.26	91.65	91.60	88.63	91.64	83.32
standing	87.72	89.32	80.63	82.45	80.57	81.74	79.10	82.88
sitting	85.82	82.54	88.96	90.13	89.45	86.76	88.83	83.90
lying	85.04	91.21	88.64	89.84	89.01	86.53	88.38	89.68
macro-average	87.08	91.32	89.00	90.04	89.38	84.05	88.90	89.84
weighted-average	86.15	**91.92**	88.70	**90.23**	87.51	**89.70**	89.47	**85.84**

TABLE XIII
$f1$-SCORE ON DIFFERENT δ(%)

motion	0.01	0.05	0.1	0.5
walking	91.92	90.42	81.28	70.25
running	96.05	95.29	85.37	74.55
biking	92.45	91.57	82.34	72.97
standing	90.32	84.84	74.13	73.81
sitting	83.54	90.55	80.97	70.77
lying	91.21	90.53	80.78	70.40
macro-average	**92.32**	**90.57**	**80.95**	**70.78**
weighted-average	91.92	91.03	81.27	70.83

TABLE XIV
$f1$-SCORE ON DIFFERENT τ(%)

motion	0.01	0.05	0.1	0.2	0.5
walking		91.92	88.44	89.66	81.16
running		96.05	92.48	94.98	85.52
biking		92.45	91.44	91.62	82.05
standing	calculation time too long	90.32	82.29	83.18	74.65
sitting		83.54	89.16	90.06	80.90
lying		91.21	88.66	89.86	80.85
macro-average		92.32	89.1	90.04	80.93
weighted-average		91.92	88.44	89.66	81.16

tri-VFDT are compared, the parameter of VFDT is set to $n_{\min} = 100$, $\delta = 0.05$, and $\tau = 0.01$. Table XII shows the results of different proportions of unlabeled data.

When unlabeled data account for 20% of all data, the $f1$-score of tri-VFDT is significantly higher than tri-DT. With the highest proportion of labeled data, tri-VFDT performs the best. After the improvement of the semisupervised framework, the improvement effect is obvious. When the ratio of unlabeled data is 40% and 60%, the effect of tri-VFDT is similar to that of tri-DT. When the unlabeled ratio increases to 80%, the classification accuracy of tri-VFDT decreases significantly, even lower than that of tri-DT. Since the amount of data used for model initialization is too small and the base classifier is under fitted, this results in low accuracy in the prediction results when they are used to expand the sample set. As a result, the classification results decline after the semisupervised framework. This shows that the proposed semisupervised framework is suitable for data with a large initial sample set.

For the second part, the percentage of unlabeled data is set at 20%. Tables XIII and XIV show the effects of different superparameters. There are three important hyperparameters, τ limits the information gain, $n_{\min}$ limits the splitting condition of nodes, according to Hoeffding's inequality, finite data can replace infinite data with probability of $1-\delta$. The experiment shows that $n_{\min}$ has a little effect on the results.

When τ is set to 0.01 and $n_{\min}$ is set to 100, Table XIII shows the results on different δ. With the increase of δ, the training time decreases, but the accuracy of classification decreases. When delta increases from 0.01 to 0.5, the classification accuracy decreases to 21.54%.

When δ is set to 0.01 and $n_{\min}$ is set to 100, Table XIV shows the results on different τ, where it is observed that τ has a great influence on the calculation time. When τ decreases, the number of calculation rounds increases and the accuracy improves, but the calculation time also increases. When τ is set to 0.05, the model performs best and the $f1$-score is 91.21%. When τ is set to 0.01, the training time consumes a much longer time.

4) Online Experiment Testing on Wearable Devices: The proposed method is an end-to-end algorithm for online learning and real-time recognition, which is specially designed for wearable devices. The model is embedded into the wearable device to classify the events using the real data directly. Three difficulties have been encountered with online experiments. First, in the static data sets, different motion states are balanced; when verification on wearable devices, in a short period of time, the samples may be extremely unbalanced, and the data of a certain motion may be greatly enhanced at a certain time, which will affect the results. Second, in real scenes, especially in strenuous sports such as running, the tester's body shaking may have an impact on the collection and calculation of wearable devices. Finally, due to the limitation of hardware devices, multithreads are not parallel. Data acquisition thread, data preprocessing thread, recognition thread, and updating thread are executed alternately. The collected data are predicted intermittently, not all of them are calculated. Some information of the streaming data may be lost.

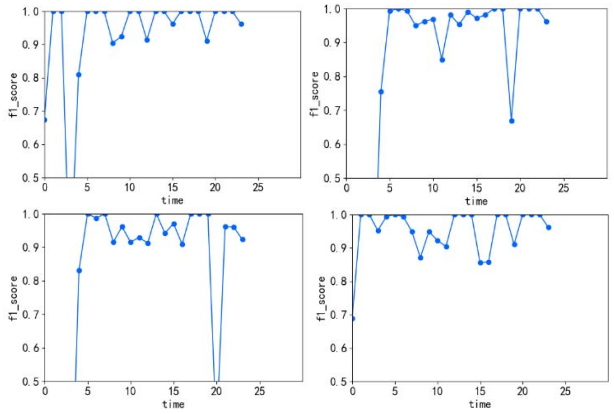

Fig. 8. $f1$-score for activity recognition on wearable devices.

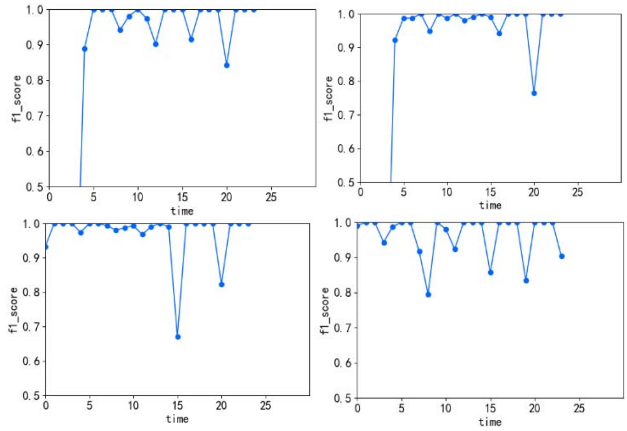

Fig. 9. $f1$-score for activity recognition on desktop computer.

When the motion state is switched, the delay problem occurs, the prediction results cannot match the label. In the process of updating the model, the termination condition of iteration is the accuracy of the classifier is no longer improved and the running time cannot be limited. When updating thread is executed, the collection thread cannot be executed. Therefore, some data will be lost. The prediction results are written into a fixed length array. In order to align the prediction result with the label, the missing part is filled with 0, resulting in a low accuracy (less than or equal to 50%) in a certain segment. Fig. 8 shows part of the online recognition results. The results of the control experiment on desktop computer of 2.3-GHz quad-core Intel i7 processor, 8 GB of RAM are shown in Fig. 9.

Comparing the results on wearable devices and desktop computer, the $f1$-score of the model embedded into the wearable devices decreases slightly but basically still keeps above 82.5%. In contrast, the results on wearable devices are more volatile and this is due to the environmental noise caused by the shaking of the equipment and the limitation of the hardware. After a period of study, it is observed that the $f1$-score gradually improved and tended to be stable.

IV. Conclusion and Future Work

In this article, a semisupervised learning algorithm tri-VFDT has been proposed for online learning and was embedded into wearable devices. The dictionary storage mode of base learner VFDT is suitable for streaming data. In addition, with the improved tri-training semisupervised framework, unlabeled data have been used to improve the accuracy and robustness. Compared with other related methods, the proposed method has higher performance in terms of the $f1$-score with lower training time and storage on two different data sets. After feature selection with the simulated annealing algorithm, the calculation method of features and the pretrained model parameters are embedded into wearable devices. Compared with the verification experiment on the desktop computer, the obtained $f1$-score using the wearable device has minimal difference. Future research will focus on a longer duration experiment while reducing the impact of noise on the model.

References

[1] H. Qiu, X. Wang, and F. Xie, "A survey on smart wearables in the application of fitness," in *Proc. IEEE 15th Int. Conf. Depend. Auton. Secure Comput. IEEE 15th Int. Conf. Pervasive Intell. Comput. IEEE 3rd Int. Conf. Big Data Intell. Comput.*, Orlando, FL, USA, Jan. 2018, pp. 303–307.

[2] M. N. S. Zainudin, M. N. Sulaiman, N. Mustapha, and T. Perumal, "Monitoring daily fitness activity using accelerometer sensor fusion," in *Proc. IEEE Int. Symp. Consum. Electron. (ISCE)*, Kuala Lumpur, Malaysia, 2017, pp. 35–36.

[3] A. Ahmadi *et al.*, "Toward automatic activity classification and movement assessment during a sports training session," *IEEE Internet Things J.*, vol. 2, no. 1, pp. 23–32, Feb. 2015.

[4] V. Bianchi, P. Ciampolini, and I. De Munari, "RSSI-based indoor localization and identification for ZigBee wireless sensor networks in smart homes," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 2, pp. 566–575, Feb. 2019.

[5] A. Ancans, A. Rozentals, K. Nesenbergs, and M. Greitans, "Inertial sensors and muscle electrical signals in human–computer interaction," in *Proc. 6th Int. Conf. Inf. Commun. Technol. Accessibility (ICTA)*, 2017, pp. 1–6.

[6] N. Mora, V. Bianchi, I. De Munari, and P. Ciampolini, "A BCI platform supporting AAL applications," in *Universal Access in Human–Computer Interaction. Design and Development Methods for Universal Access* (Lecture Notes in Computer Science 8513), C. Stephanidis and M. Antona, Eds. Cham, Switzerland: Springer, 2014.

[7] S. Qiu, Z. Wang, H. Zhao, L. Liu, and Y. Jiang, "Using body-worn sensors for preliminary rehabilitation assessment in stroke victims with gait impairment," *IEEE Access* vol. 6, pp. 31249–31258, 2018.

[8] I. Bisio, A. Delfino, F. Lavagetto, and A. Sciarrone, "Enabling IoT for in-home rehabilitation: Accelerometer signals classification methods for activity and movement recognition," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 135–146, Feb. 2017.

[9] F. Montalto, C. Guerra, V. Bianchi, I. De Munari, and P. Ciampolini, "MuSA: wearable multi sensor assistant for human activity recognition and indoor localization," in *Ambient Assisted Living* (Biosystems and Biorobotics), vol. 11. Cham, Switzerland: Springer, 2015.

[10] V. Bianchi, C. Guerra, M. Bassoli, I. De Munari, and P. Ciampolini, "The HELICOPTER project: Wireless sensor network for multi-user behavioral monitoring," in *Proc. Int. Conf. Eng. Technol. Innov. Eng. Technol. Innov. Manag. Beyond New Challenges New Approaches (ICE/ITMC)*, Jan. 2018, pp. 1445–1454.

[11] G. De Leonardis *et al.*, "Human activity recognition by wearable sensors" in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, 2018, pp. 1–6.

[12] P. Vepakomma, D. De, S. K. Das, and S. Bhansali, "A-wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities," in *Proc. IEEE 12th Int. Conf. Wearable Implantable Body Sens. Netw. (BSN)*, Jun. 2015.

[13] Y. Chen and Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 1488–1492.

[14] S. Ha, J. M. Yun, and S. Choi, "Multi-modal convolutional neural networks for activity recognition," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 3017–3022.

[15] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 1307–1310.
[16] V. Radu, N. D. Lane, S. Bhattacharya, C. Mascolo, M. K. Marina, and F. Kawsar, "Towards multimodal deep learning for activity recognition on mobile devices," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. Adjunct*, 2016, pp. 185–188. [Online]. Available: http://doi.acm.org/10.1145/2968219.2971461
[17] C. Liu, L. Zhang, Z. Liu, K. Liu, X. Li, and Y. Liu, "Lasagna: Towards deep hierarchical understanding and searching over mobile sensing data," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 334–347. [Online]. Available http://doi.acm.org/10.1145/2973750.2973752
[18] M. Edel and E. Kppe, "Binarized-blstm-rnn based human activity recognition," in *Proc. Int. Conf. Indoor Positioning Indoor Navig. (IPIN)*, Oct. 2016, pp. 1–7.
[19] 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. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016786551830045X
[20] F. Ordez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, Jan. 2016. [Online]. Available: http://dx.doi.org/10.3390/s16010115
[21] G. Carneiro and J. Nascimento, "Incremental on-line semi-supervised learning for segmenting the left ventricle of the heart from ultrasound data," in *Proc. IEEE Int. Conf. Proc. Comput. Vis.*, Nov. 2011, pp. 1700–1707.
[22] B. Yver, "Online semi-supervised learning: Application to dynamic learning from RADAR data," in *Proc. Int. Radar Conf. Surveillance Safer World*, Oct. 2009, pp. 1–6.
[23] P. Zhang, X. Zhu, and L. Guo, "Mining data streams with labeled and unlabeled training examples," in *Proc. IEEE 9th Int. Conf. Data Min.*, 2009, pp. 627–636.
[24] Z.-H. Zhou and M. Li, "Semi-supervised learning by disagreement," *Knowl. Inf. Syst.*, vol. 24, pp. 415–439, Sep. 2010.
[25] Z.-H. Zhou and M. Li, "Tri-Training: Exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, Nov. 2005.
[26] M.-L. Zhang and Z.-H. Zhou, "COTrade: Confident co-training with data editing," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 6, pp. 1612–1626, Dec. 2011.
[27] F. Ma, D. Meng, Q. Xie, Z. Li, and X. Dong, "Self-paced co-training," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2275–2284.
[28] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *Proc. ICLR*, 2017.
[29] A. Tarvainen and H. Valpola, 'Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Assoc., 2017.
[30] T. Miyato, S. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.
[31] N. Noury *et al.*, "Fall detection—Principles and methods," in *Proc. IEEE Eng. Med. Biol. Soc. (EMBS)*, Lyon, France, Aug. 2007, pp. 1663–1666.
[32] K. H. Chen, J. J. Yang, and F. S. Jaw, "Accelerometer-based fall detection using feature extraction and support vector machine algorithms," *Instrum. Sci. Technol.*, vol. 44, no. 4, pp. 333–342, 2016.
[33] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I. De Munari, "IoT wearable sensor and deep learning," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8553–8562, Oct. 2019.
[34] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 56–64, Jan. 2017.
[35] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *Proc. IEEE 13th Int. Conf. Wearable Implantable Body Sens. Netw. (BSN)*, 2016, pp. 71–76.
[36] G. Shi, C. S. Chan, W. J. Li, K.-S. Leung, Y. Zou, and Y. Jin, "Mobile human airbag system for fall protection using MEMS sensors and embedded SVM classifier," *IEEE Sensors J.*, vol. 9, no. 5, pp. 495–503, May 2009.
[37] A. Jahanjoo, M. N. Tahan, and M. J. Rashti, "Accurate fall detection using 3-axis accelerometer sensor and MLF algorithm," in *Proc. Int. Conf. Pattern Recognit. Image Anal. (IPRIA)*, Shahrekord, Iran, Apr. 2017, pp. 90–95.
[38] M. Saleh and R. L. B. Jeannès "Elderly fall detection using wearable sensors: A low cost highly accurate algorithm," *IEEE Sensors J.*, vol. 19, no. 8, pp. 3156–3164, Apr. 2019.
[39] G. De Leonardis *et al.*, "Human activity recognition by wearable sensors: Comparison of different classifiers for real-time applications," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, 2018, pp. 1–6.
[40] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22Nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2016, pp. 785–794.
[41] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, 1963.
[42] S. Goldman and Y. Zhou, "Enhancing supervised learning with unlabeled data," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 327–334.
[43] D. Angluin and P. Laird, "Learning from noisy examples," *Mach. Learn.*, vol. 2, no. 4, pp. 343–370, 1988.
[44] W. Hoeffding, "Probability inequalities for sums ofbounded random variables," *J. Amer. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, 1963.
[45] D. Anguita, A. Ghio, L. Oneto, X. P. Perez, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. 21th Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn.*, Bruges, Belgium, Apr. 2013, pp. 437–442.
[46] K. Bouleimen and H. Lecocq, "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version," *Eur. J. Oper. Res.*, vol. 149, pp. 268–281, Sep. 2003.
[47] G. Ke, Q. Meng, T. Finley, and T. Wang, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Assoc., 2017.

Yiwei Zhang received the B.Sc. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2018, where she is currently pursuing the M.Sc. degree in human activity recognition and machine translation.

Her research interests include wearable sensing, recommendation system, and machine learning.

Bin Gao (Senior Member, IEEE) received the B.Sc. degree in communications and signal processing from Southwest Jiao Tong University, China, in 2005, the M.Sc. degree in communications and signal processing with Distinction and the Ph.D. degree from Newcastle University, Newcastle upon Tyne, U.K., in 2007 and 2011, respectively.

He worked as a Research Associate of Wearable Acoustic Sensor Technology with Newcastle University, from 2011 to 2013. He is currently a Professor with the School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include sensor signal processing, machine learning, social signal processing, nondestructive testing, and evaluation, where he actively publishes in these areas.

Prof. Gao is also a very active reviewer for many international journals and long standing conferences. He has coordinated several research projects from the National Natural Science Foundation of China.

Daili Yang received the B.Sc. degree from the School of Physics and Electronic Engineering, Hainan Normal University, Haikou, China, in 2009, and the M.Sc. degree from the School of Electrical Engineering, Xinjiang University, Urumqi, China, in 2014. He is currently pursuing the Ph.D. degree in control science and engineering with the University of Electronic Science and Technology of China, Chengdu, China.

His research interests include wearable sensors and physical and mental health.

Wai Lok Woo (Senior Member, IEEE) received the B.Eng. degree in electrical and electronics engineering and the M.Sc. and Ph.D. degrees in statistical machine learning from Newcastle University, Newcastle upon Tyne, U.K., in 1993, 1995, and 1998, respectively.

He was the Director of Research with the Newcastle Research and Innovation Institute, and the Director of Operations with Newcastle University. He is currently a Professor of Machine Learning with Northumbria University, Newcastle upon Tyne. His research interests include the mathematical theory and algorithms for data science and analytics, artificial intelligence, machine learning, data mining, latent component analysis, multidimensional signal, and image processing. He has published more than 400 papers on these topics on various journals and international conference proceedings.

Dr. Woo was a recipient of the IEE Prize and the British Commonwealth Scholarship. He serves as an Associate Editor for several international signal processing journals, including the *IET Signal Processing*, *Journal of Computers*, and *Journal of Electrical and Computer Engineering*. He is a member of the Institution Engineering Technology.

Houlai Wen received the B.Sc. degree from the School of Information Engineering, Southwest University of Science and Technology, Mianyang, China, in 2018. He is currently pursuing the M.Sc. degree in control engineering with the University of Electronic Science and Technology of China, Chengdu, China.

His research interests include smart sensor and physical and mental health monitoring.