



A Novel Method for Wearable Activity Recognition with Feature Evolvable Streams

Yixiao Wang^{1,2}, Chunyu Hu^{1,2(✉)}, Hong Liu³, Lei Lyu³, and Lin Yuan^{1,2}

¹ Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

10431210389@stu.qlu.edu.cn, {hcy,yuanl}@qlu.edu.cn

² Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan, China

³ School of Information Science and Engineering, Shandong Normal University, Jinan, China

lvlei@sdsnu.edu.cn

Abstract. Wearable activity recognition plays an important role in human health monitoring. Traditional wearable activity recognition models are trained in an offline mode with static and pre-defined sensor configurations. However, in real scenarios, data arrives in streams and wearable sensors dynamically appear or disappear, resulting in corresponding changes in the feature space, which is referred to as feature evolution. Addressing the issue of feature evolution is a significant challenge in wearable activity recognition. In this paper, we propose a new method, namely Online Learning method for Feature Evolvable Streams (OLFES). OLFES learns the optimal model depth online according to the complexity of the data stream, recovers the old features through the feature space generation strategy, and finally integrates the prediction results according to a weighted combination strategy. Extensive experimental results on data science datasets and activity recognition datasets demonstrate the feasibility and effectiveness of the proposed method.

Keywords: Activity recognition · Feature evolution · Online learning

1 Introduction

Human activity recognition (HAR) has become a hot topic for many areas, including healthcare, interactive games and sports [1, 2]. It is able to detect and identify human activities by analyzing the data collected from various types of sensors. Wearable sensors, such as heart rate monitoring sensors, accelerometers, gyroscopes, and magnetometers, exhibit robustness to occlusion, point-of-view variations, illumination changes, and background fluctuations. This characteristic enables individuals to transcend the limitations imposed by their immediate

environment and offers significant advantages in activity recognition. Therefore, wearable activity recognition is gaining increasing popularity among a growing number of individuals.

Existing activity recognition methods typically assume that data is collected with fixed sensors. The feature space will not undergo any changes. However, this assumption does not hold in realistic scenarios. Specifically, in an open and dynamic environment, the feature space evolves over time as sensors appear or disappear. Traditional activity recognition models face two challenges. Firstly, they are unable to utilize the data from newly emerged sensors to recognize user behavior. Secondly, they fail to function properly when existing sensors are removed. A straightforward idea is to learn a new activity recognition model on the new feature space and use the new model to recognize user's activities. However, there are some weaknesses in this method. On one hand, the new feature space has just emerged and the collected samples are insufficient to train a well-performing model. On the other hand, the knowledge learned by the old model on the vanished old feature space cannot be reused, resulting in a waste overhead of previously collected data. To solve these problems, a new learning paradigm for online learning from evolvable feature spaces, named feature evolution learning [3]. There have been many related researches [4–6] and have been applied in many fields [7,8]. It not only rely on the current features, but also try to leverage the knowledge stored in the existing model by recovering the disappeared features.

In the pioneering work [3], an important observation is that the feature space does not change in an arbitrary way, and there is an evolving phase in the process of change in which the old feature space and the current feature space exist simultaneously. In the activity recognition task, we usually put on a new wearable device in advance before the old one runs out of power. Therefore, Hou [3] et al. used Online Gradient Descent (OGD) to update the model in the old feature space and the model in the new feature space. At the same time, the mapping matrix from the current feature to the old feature is learned by taking advantage of the simultaneous existence of the old and new feature spaces in the evolution stage, so as to establish the mapping relationship between the two feature spaces. After the old feature space disappears, the new model learned from the current feature space can be aided by the historical data in the old feature space by mapping from the current feature space to the old feature space. Liu [9] et al. introduce a second-order online learning method into the evolvable feature space to improve the convergence speed of the model. Zheng [10] proposed a feature evolution learning method for image streams by replacing the fully connected layer in Online Deep Learning (ODL) with ResNet and approximating the connection between two feature spaces by minimizing the potential representation mapping. Lian [11] et al. use variational self-encoders to map two feature spaces into a shared latent subspace that can encode arbitrary expressible mapping functions for feature reconstruction. However, for sensor stream data, shallow learners have limited learning capabilities and cannot face high-dimensional features. Meanwhile, linear mapping as well as self-coding reconstruction are not sufficient to recover the vanished feature space well.

To address these challenges, we propose a novel Online Learning method for Feature Evolvable Streams, namely OLFES. OLFES is designed to support the appearance and disappearance of sensors in real scenes. Specifically, OLFES updates the classifier online by Hedge Backpropagation (HBP), which not only learns the predictive model sequentially from the data stream, but also enables the transition from shallow to deep models. In the evolving stage, a mapping from new features to old features is learned through a feature space generation strategy, then the learned mapping relation is used to recover the old features when only the new feature space is available, and finally the prediction results of the old and new models are integrated in a weighted combination. Experiments were conducted on datasets from both data science and activity recognition domains, and OLFES was able to obtain better classification accuracy and minimal regret.

The rest of the paper is organized as follows. In Sect. 2, we present the related work on activity recognition and feature evolvable stream learning. In Sect. 3, we present the problem description of OLFES. In Sect. 4, we describe the proposed OLFES method in detail. In Sect. 5, extensive experiments are conducted to evaluate the performance of OLFES and specific experimental results are given. Conclusions and future work are presented in Sect. 6.

2 Relate Work

2.1 Activity Recognition

Traditional human activity recognition is mainly based on pattern recognition and feature engineering, which consists of two main parts: feature extraction and classification algorithms. Feature extraction requires the researcher to manually design the extracted features. For example, Bao and Intille [12] collected a large amount of accelerometer data using an accelerometer sensor and then manually extracted four features of the data samples: mean, energy, frequency domain entropy, and correlation coefficient, and then fed these extracted features to a classifier for training. In terms of classification algorithms, various machine learning methods have been applied in the field of wearable activity recognition, such as SVM [13], decision trees [14].

Compared to shallow machine learning models, deep learning models using neural networks have significant advantages in terms of classification accuracy [15]. In [16], a deep learning algorithm based on deep self-encoders is used to reduce the feature dimensionality of the data, minimize unwanted activities and extract hierarchical features of the data to learn the most appropriate feature vector for activity recognition. In literature [17], Vollmer C et al. used sparse coding method to learn human activity features using feature dimension reduction strategy to reduce computational complexity and time. In the literature [18], Zeng et al. proposed a convolutional neural network-based human activity recognition method, which can automatically extract features for activity recognition by capturing the local dependence and scale invariance of the signal, and changes in the same activity can be easily captured. Wang [19] et al. proposed a neural network framework consisting of CNN and bidirectional LSTM to extract spatial

and temporal data from the cell phone the raw acceleration data from mobile phones to extract spatial and temporal features. Wen [20] developed their own convolutional and recurrent network architectures, but did not illustrate their performance in real-world heterogeneous environments.

However, the existing work requires that the dataset is fully available before training and that the existing models are not applicable to scenarios with changing feature spaces. Real-world scenarios where data often arrives in streams and the number of sensors changes dynamically due to various factors make static models inapplicable.

2.2 Feature Evolvable Stream Learning

In an open environment [21], the feature dimension of the data is likely to change. The earliest study of feature changes was incremental learning of features in incremental learning. Guan [22] et al. proposed the ITID algorithm to solve the problem of new features appearing in the training of neural networks. When new features appear, they do not train from scratch using new data, but train the old network with the old features, train the new network on the new features, and finally combine them together. The paper [23] combined the PCA method based on ITID algorithm to reduce the interference between input features and improve the accuracy of model classification. Hu [24] et al. proposed FIRF, a random forest-based feature incremental learning method, to solve the problem of incremental features in wearable activity recognition due to the addition of new sensors. However, the above method does not consider the scenario of feature disappearance in an online environment.

In [3], Hou et al. proposed a feature evolution scenario that includes feature addition and disappearance, which assumes that the change of features is not arbitrary but there is an evolving stage, and learns the mapping relationship between the two feature spaces during the evolving so as to recover the data using the newly emerged features after the initial feature space disappears, and trains the classifier on the two feature spaces separately, and finally integrates the two prediction results are integrated. Zhang [25] et al. considered a more realistic scenario where the disappearance of features is temporary and arbitrary, and proposed an adaptive integration learning method FIDE for the problem of feature increment and decrement in gesture recognition. Based on [3], Zhang [26] et al. further considered the scenario where features and distributions evolve simultaneously, and proposed evolutionary differences to measure differences between successive batches of data and implement a difference minimization algorithm using deep neural networks. Similarly, Hou [27] et al. proposed the one-time incremental decremental learning approach (OPID), which assumes the scenario that there will be an overlap of old and new features when the features evolve, i.e., the old features will not disappear completely when they disappear, but only partially, and the model that learns the disappearing features and the surviving features well will make predictions for the surviving features, and then the multiclassification vectors obtained from the predictions and the new features will be put together to learn the new. Zheng [10] proposed a feature evolution

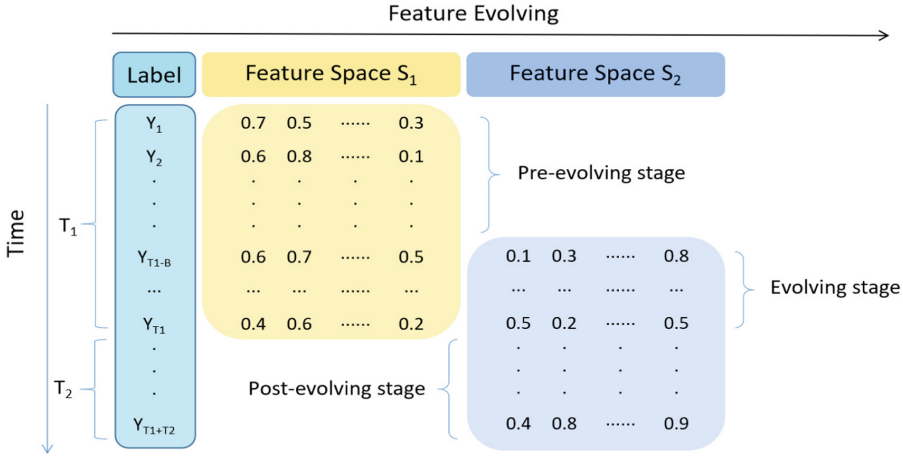


Fig. 1. Illustration of feature evolution

learning method for image streams by replacing the fully connected layer in a neural network with ResNet and approximating the connection between two feature spaces by minimizing the potential representation mapping. The current state-of-the-art method is proposed by Lian [11] et al. who combine online deep learning with a variational self-coding mapping approach to feature evolving streams. However, the above methods are not suitable for complex and variable sensor data streams.

3 Problem Statement

We focus on the data stream classification task in the feature space evolution scenario. Assume $\{(x_t, y_t) | t = 1, 2, \dots, T_1 + T_2\}$ is the input data stream, where x_t is the instance observed at moment t . The true label $y_t \in \{1, 2, \dots, c\}$, where c is the number of classes. In the wearable activity recognition task, we assume S_1 and S_2 denote the previous feature space and the current feature space, respectively. Consistent with the scenario presented in [3], OLFES can also be divided into three stages: the pre-evolving stage, the evolving stage and the post-evolving stage. As shown in Fig. 1, in the pre-evolving stage, the data stream comes from the old feature space S_1 . Then, in the evolving stage, a small number of instances come from both the old feature space S_1 and the new feature space S_2 . Subsequently, in the post-evolving stage, instances come only from the new feature space S_2 . A detailed description of the three time periods is given below.

- Pre-evolving stage: For each round in $t = 1, \dots, T_1 - B$, the classifier receives an instance $x_t^{S_1} \in \mathbb{R}^{d_1}$ collected from the feature space S_1 , where d_1 is the dimension of the feature space S_1 and T_1 is the number of rounds in S_1 .
- Evolving stage: For each round in $t = T_1 - B + 1, \dots, T_1$, the classifier receives an instance consisting of two vectors $x_t^{S_1}$ and $x_t^{S_2}$. $x_t^{S_1} \in \mathbb{R}^{d_1}$ and $x_t^{S_2} \in \mathbb{R}^{d_2}$

are sampled from feature spaces S_1 and S_2 , where d_2 is the dimensionality of feature space S_2 .

- Post-evolving stage: For each round in $t = T_1 + 1, \dots, T_1 + T_2$, the classifier receives an instance $x_t^{S_2} \in \mathbb{R}^{d_2}$ sampled from the feature space S_2 , where T_2 is the number of rounds of S_2 .

Note that in Fig. 1 we only show the feature evolution learning process in one cycle which contains three different stages. This evolution can be extended to the case with multiple cycles easily, where the post-evolving stage can be seen as the pre-evolving stage for the next cycle.

At any moment t in the cycle, the classifier f_t receives the instance x_t . Assume the prediction is denoted as \hat{y} . Our goal is to learn a series of classifiers $\{f_1, f_2, \dots, f_{T_1+T_2}\}$, that minimizes the empirical risk:

$$\min_{f_1, \dots, f_{T_1+T_2}} \frac{1}{T_1 + T_2} \sum_{t=1}^{T_1+T_2} \mathcal{L}(y_t, f_t(x_t)), \quad (1)$$

where $\mathcal{L}(\cdot, \cdot)$ denotes a loss function, such as cross-entropy loss or square loss.

4 Our Method

In this section, we present the proposed OLFES method. This section is organized as follows: First, in Sect. 4.1 we introduce the online deep learning method with HBP, when S_1 and S_2 are newly available we train deep adaptive models in an online manner. Then, Sect. 4.2 presents the feature space generation strategy used to recover old features by reconstructing S_1 from S_2 . Finally, Sect. 4.3 describes how to combine the models learned in the old and new feature spaces in a weighted manner to obtain the final prediction. The architecture of OLFES is described in Fig. 2.

4.1 Online Deep Learning with HBP

Traditional neural network algorithms usually require the entire training data to be available before training. However, there are many realistic scenarios where data arrives in the form of streams, limiting the use of neural networks. Therefore, a Hedge Backpropagation method is proposed to enable the use of deep neural networks in an online environment [28].

Assuming a deep neural network with L hidden layers, the final prediction result can be expressed as:

$$F(x) = \sum_{l=0}^L \alpha^{(l)} f^{(l)}. \quad (2)$$

where $f^{(l)} = \text{softmax}(h^{(l)}\theta^{(l)})$, $\forall l = 0, \dots, L$, $h^{(l)} = \sigma(W^{(l)}h^{(l-1)})$, $\forall l = 1, \dots, L$, and $h^{(0)} = x$.

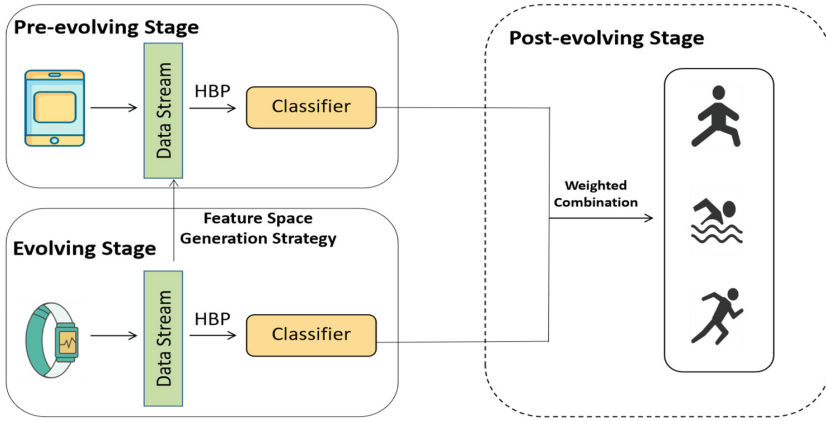


Fig. 2. Architectural description of our OLFES

The final prediction results in the use of $h^{(0)}, \dots, h^{(L)}$ feature representations of the weighted combinations of the learned classifiers, thus allowing the learner to acquire network knowledge from “shallow to deep”. The weight of each classifier is denoted as $\alpha^{(l)}$. Our goal is to learn and update the parameters $\alpha^{(l)}, \theta^{(l)}$ and $W^{(l)}$ in the model and train the model by minimizing the loss function: $\mathcal{L} = \sum_{l=0}^L \alpha^{(l)} \mathcal{L}(f^{(l)}(x), y)$.

For the update of α , in the initial iteration, all the weights α are uniformly distributed, i.e., $\alpha^{(l)} = \frac{1}{L+1}$. In each subsequent iteration, the classifier $f^{(l)}$ makes a prediction $\hat{y}_t^{(l)}$. Subsequently, the basic fact is revealed that the weights of the classifier are updated according to the losses in this round:

$$\alpha_{t+1}^{(l)} \leftarrow \alpha_t^{(l)} \beta^{\mathcal{L}(f^{(l)}(x), y)}, \tag{3}$$

where $\beta \in (0, 1)$ is the discount rate parameter and $\mathcal{L}(f^{(l)}(x), y) \in (0, 1)$. In each iteration, the weights of the classifier are discounted according to the performance of the classifier by a factor of $\beta^{\mathcal{L}(f^{(l)}(x), y)}$. At the end of each round, the value of weight α will be renormalized such that $\sum_l \alpha_t^{(l)} = 1$. Since the shallow model tends to converge faster than the deep model, we set the smoothing parameter $S \in (0, 1)$ to balance exploration and exploitation [29] by setting a minimum weight for each classifier. The weights of α are smoothed after updating them in each iteration: $\alpha^{(l)} \leftarrow \max(\alpha^{(l)}, \frac{S}{L})$.

For the learning of the parameters $\theta^{(l)}$, consider the online gradient descent approach, where the input to the l^{th} classifier is $h^{(l)}$, which is similar to the update of the output layer weights in the original feedforward network and can be expressed as:

$$\begin{aligned} \theta_{t+1}^{(l)} &\leftarrow \theta_t^{(l)} - \eta \nabla_{\theta_t^{(l)}} \mathcal{L}(F(x_t, y_t)) \\ &= \theta_t^{(l)} - \eta \alpha^{(l)} \nabla_{\theta_t^{(l)}} \mathcal{L}(f^{(l)}, y_t). \end{aligned} \tag{4}$$

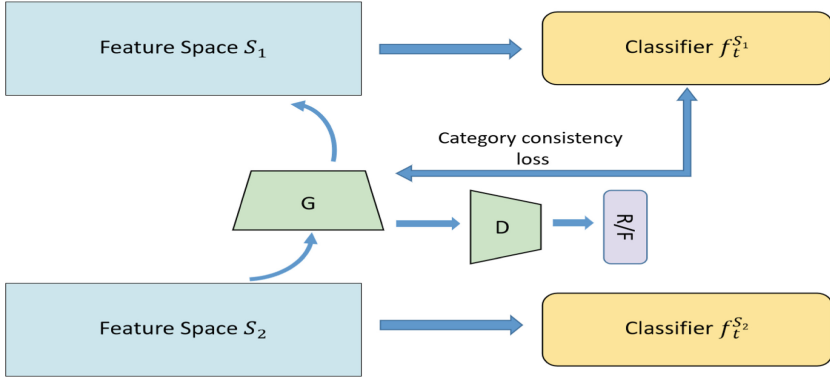


Fig. 3. In the evolving stage, the generator is used to generate S_1 from S_2 while imposing an additional loss of category consistency.

For the learning of the parameters $W^{(l)}$, where the error derivatives are back-propagated from the output layer, which is different from the previous back-propagation. Here, the error derivatives are back-propagated from each classifier $f^{(l)}$. Therefore, using the adaptive loss function \mathcal{L} and applying it to the OGD, the update of $W^{(l)}$ can be expressed as:

$$W_{t+1}^{(l)} \leftarrow W_t^{(l)} - \eta \sum_{j=l}^L \alpha^{(j)} \nabla_{W^{(l)}} \mathcal{L}(f^{(j)}, y_t), \quad (5)$$

where η is the learning rate. Here, the summation (gradient term) starts from $j = l$ because only the deeper classifiers rely on $W^{(l)}$ for prediction. Hedge enjoys the regret of $R_T \leq \sqrt{T \ln N}$, where N represents the depth of the network in our method.

4.2 Feature Space Generation Strategy

In the post-evolving stage, the old feature space disappears. To be able to use the knowledge learned in the old model, we used a generative model to recover the vanished features. The generation model GAN [30] contains two networks: a generator and a discriminator. Where the generator is trained to generate samples and the discriminator is trained to discriminate between the samples generated by the generator. The competing strategy makes both networks continuously improve. Inspired by the idea of conditional GAN [30,31], we propose a new generative network to establish the connection between two feature spaces in the evolving stage. The framework of our model is shown in Fig. 3.

An important observation is that in the evolving stage, the two feature spaces coexist and the corresponding instance labels are consistent. In wearable activity recognition, the activity observed by the sensor at the same time is always the same. Our feature space generation strategy consists of a generator G and

a discriminator D . The generator is used to generate instances $\tilde{x}_t^{S_1}$ in the corresponding S_1 feature space from instances arriving in the S_2 feature space: $\tilde{x}_t^{S_1} = G(x_t^{S_2})$. The discriminator is used to discriminate the instances in the generated S_1 feature space. In addition, to ensure the generated S_1 instances get the same predicted labels as the original instances, we additionally impose a category consistency loss for the generator to ensure the consistency of the classification results.

At this point, our optimization goal is:

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_C(G) \quad (6)$$

$$\mathcal{L}_C(G) = -\frac{1}{N} \sum_j \sum_{c=1}^M y_{ic} \log(P_{ic}), \quad (7)$$

where $\mathcal{L}_C(G)$ is the category consistency loss, M represents the number of categories. y_{ic} is the true category of the symbolic function sample i equal to c takes 1, otherwise takes 0. P_{ic} is the probability that the observed sample belongs to category c .

4.3 Prediction with Weighted Combination

During the T_2 time span, the old feature space S_1 disappears. Since the evolving stage B is very short, it is unlikely that the model $f_t^{S_2}$ learned on the new features will achieve a good convergence. Relying on $f_t^{S_2}$ alone to predict instances in S_2 would lead to great regret.

In order to be able to benefit from the recovered feature space, we utilize the model $f_t^{S_1}$ trained on the old feature space. Based on the generative model learned in Sect. 4.2, we reconstruct the old features from the new features. When $f_t^{S_2}$ training instances are insufficient, the predictions of the two models are weighted and combined to produce accurate predictions. Specifically, we train the model $f_t^{S_2}$ in time period T_2 using instances received in the new feature space; at the same time, the mapping relations learned in the evolving stage are utilized to recover the data in the old feature space, following which we continue to update the learned model $f_t^{S_1}$. Finally, the predictions of the two models are ensembled and represented as follows:

$$\hat{y}_t = p \cdot f_t^{S_1}(\tilde{x}_t^{S_1}) + (1 - p) f_t^{S_2}(x_t^{S_2}), \quad (8)$$

where p is the weight of the classifier $f_t^{S_1}$ learned in the feature space S_1 .

5 Experiment

In this section, we compare the proposed OLFES algorithm with four other state-of-the-art methods on six datasets involving two domains. We describe the experimental setup in detail, including the datasets, the comparison methods, and the evaluation metrics, in Sect. 5.1. Section 5.2 shows the results of the comparison between OLFES and the other methods, along with the corresponding conclusions.

Table 1. The statistical information of the dataset. T_1 represents the number of instances in the entire feature space S_1 , B represents the number of instances in the evolving stage, and T_2 represents the number of instances in the post-evolving stage where only S_2 exists. d_1 and d_2 represent the dimensions of the two feature spaces, respectively, and c represents the number of categories.

Datasets	T_1	B	T_2	d_1	d_2	c
Magic	19,019	1,919	17,100	10	30	2
Adult	32,559	3,559	29,000	14	30	2
Dry Bean	13,611	1,400	12,211	16	32	7
DSADS	130,000	15,000	115,000	18	27	19
PAMAP2	200,000	20,000	180,000	12	24	12
HHAR	200,000	15,000	185,000	6	6	6

5.1 Experimental Setup

Datasets. To evaluate the performance of OLFES, we conducted comparative experiments on three UCI data science dataset as well as three UCI activity recognition datasets. For the UCI activity recognition dataset, we use the collected raw sensor data for training. Table 1 shows the details of these public datasets.

- Magic [32]: UCI data science dataset. Magic is MC generated to simulate registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique. This dataset initially has only one feature space S_1 , and we artificially created a new feature space S_2 by means of a random Gaussian matrix. These two feature spaces were concatenated in the manner showed in Fig. 1 to simulate a feature evolution scenario.
- Adult [33]: UCI data science dataset. Adult was extracted from the 1994 Census database. Prediction task is to determine whether a person makes over 50K a year. The data set is processed in the same way as above and cascaded as in Fig. 1.
- Dry Bean [34]: UCI data science dataset. Dry Bean contains 13,611 images of seven different registered dry beans with 16-dimensional features such as area and perimeter extracted by a computer vision system. Process in the same manner and connect in series as shown in Fig. 1.
- DSADS [35]: UCI activity recognition dataset. DSADS consists of 19 daily and physical activities performed by eight subjects. Three types of sensors (accelerometer, gyroscope, and magnetometer) were placed at five different positions of the body. To simulate the feature evolution scenario, we divided the sensors of five body positions into two parts. The first part contains sensor data from two positions, represents the feature space S_1 ; the second part contains sensor data from the remained three positions, represents the feature space S_2 .

- PAMAP2 [36]: UCI activity recognition dataset. PAMAP2 consists of 18 different physical activities performed by 9 subjects. Each subject wore 3 inertial measurement units (IMU) and a heart rate monitor. Some subjects were missing 6 optional activities, we selected only activities from the 12 protocols. As there are many missing values in the data collected by the heart rate monitor, we only use the data collected by the IMU in our experiments. Each IMU consists of two accelerometers, one magnetometer and one gyroscope. We divided the three IMUs into two parts. The first part contains the data of one IMU, which represents the feature space S_1 ; the second part contains the data of the remained two parts IMU, which represents the feature space S_2 .
- HHAR [37]: UCI activity recognition datasets. The HHAR consisted of 6 different daily activities performed by 9 subjects. Each subject wore two different devices (smartphone and smartwatch), each of which contains two embedded sensors, an accelerometer and a gyroscope. We divided the two devices into two parts. The first part contains data from smartphones, represents the feature space S_1 ; the second part contains data from smartwatches, represents the feature space S_2 .

Comparison Methods. We compare OLFES with four state-of-the-art methods. The details of the comparison methods are listed below.

- FESL-C [3]: Pioneer of feature evolvable streams. Use a shallow learner and employ a linear mapping function to learn the mapping relationship. The final prediction is a weighted combination of the two models learned on the old and new feature spaces.
- FESL-S [3]: Instead of combining old and new models weighted for prediction as in FESL-C, the best single model is utilized for prediction.
- OLD³S [11]: The most advanced algorithm available. For the first time, online deep learning is combined with variational self-encoder mapping. The final prediction is a weighted combination of the old and new models.
- OLFES-N: A variant of OLFES. For the vanished feature space, we do not do anything with it and only use the new model trained on the new feature space for classification.

Evaluation Metrics. Online Classification Accuracy (OCA) as well as Average Cumulative Regret (ACR) [11] are frequently used measures in online learning methods. OCA dynamically evaluates the accuracy of the classifier f_t at round t and validates it on the most recent K instances:

$$OCA(f_t) = \frac{1}{K} \sum_{i=t-K}^t [y_i = f_t(x_i)]. \quad (9)$$

Table 2. Comparison results of the average cumulative regret (ACR) over the six datasets, with the symbol \bullet indicating the optimal result.

Datasets	FESL-C	FESL-S	OLD ³ S	OLFES-N	OLFES
Magic	.0473±.014	.0448±.013	.0414±.020 \bullet	.0504±.024	.0456±.015
Adult	.0500±.013	.0420±.017	.0514±.013	.0493±.021	.0346±.020 \bullet
Dry Bean	.1313±.018	.1132±.016	.0576±.019	.0657±.018	.0495±.014 \bullet
DSADS	.1593±.039	.1590±.036	.0942±.026	.0826±.035	.0589±.014 \bullet
PAMAP2	.1182±.012	.1235±.011	.1075±.009	.0902±.008	.0877±.006 \bullet
HHAR	.1137±.012	.1085±.012	.0658±.010	.0618±.009	.0569±.008 \bullet

ACR evaluates the degree of regret of the online classifier compared to the optimal f^* by accumulating the OCA difference between f_t and f^* in the T round. In general, a lower ACR indicates better performance of an online algorithm. ACR can be expressed as:

$$ACR = \frac{1}{T} \sum_{t=1}^T [\max_{f^*} OCA(f^*) - OCA(f_t)]. \quad (10)$$

5.2 Experimental Results

In this section, we show the results of the proposed OLFES algorithm compared with the other four methods. Table 2 gives the ACR of all the compared methods on the datasets of Table 1. The OCA of all the compared methods on the datasets of Table 1 is given in Fig. 4. Results for all methods are averaged over 10 independent runs.

From Table 2, we can observe that our OLFES method can achieve the smallest ACR on the Adult, Dry Bean, DSADS, PAMAP2, and HHAR datasets. And on the Magic dataset, OLD³S achieved the smallest ACR. The reason why both variants of FESL will achieve the highest ACR in most cases compared to the nonlinear mapping methods is that the linear mapping methods reconstruct the feature space with a large error compared to the original feature space, leading to a decrease in accuracy.

OLFES has a higher OCA in the non-evolving stages than FESL-C, FESL-S and OLD³S methods in Fig. 4. The reason for this is that the two variants of FESL learn linear shallow models and cannot learn nonlinear functions in complex application scenarios. Although OLD³S uses the HBP method to update the parameters of the network online, it employs a variational self-encoder mapping into a latent space before the data stream is fed to the classifier. Since the data stream is complex and variable, the self-encoder produces a certain error in the mapping, thus leading to a gap in the OCA.

At the end of the evolving stage, OLFES does not lead to a sudden drop in OCA performance, indicating that our feature space generation strategy is

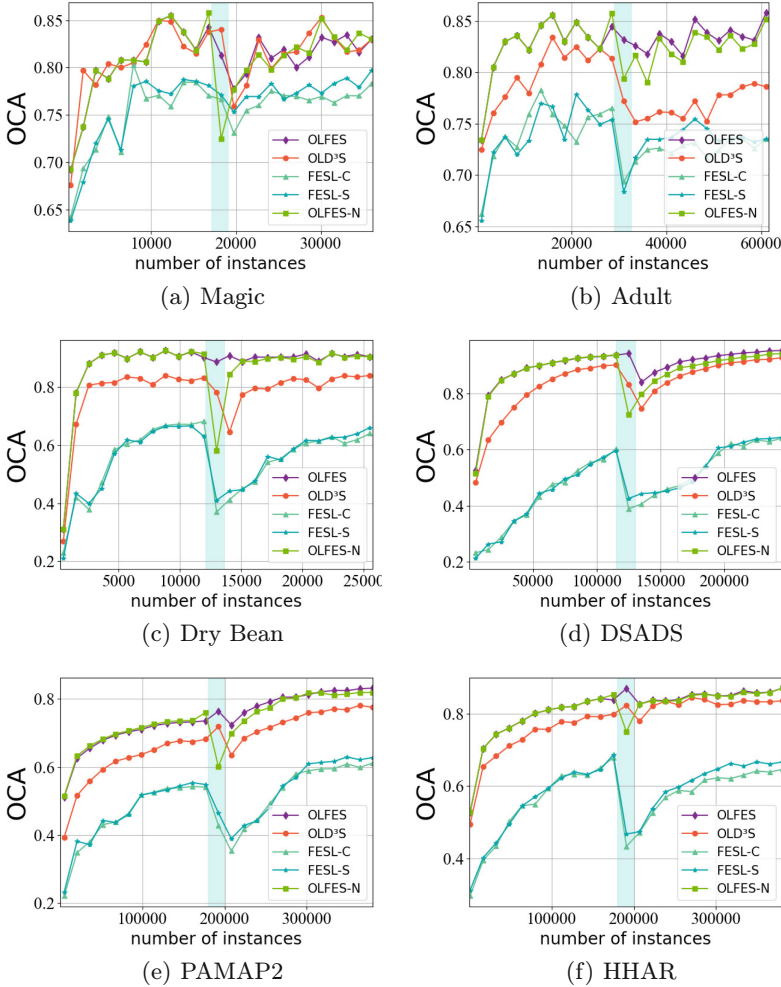


Fig. 4. Compare the online accuracy of five methods on six datasets. For all six subplots, the horizontal axis indicates the number of instances and the vertical axis indicates the online classification accuracy.

superior to the linear mapping and the variational self-encoder mapping methods. In terms of the degree of decline, the two variants of FESL have the largest gap between before and after evolution, and the OLD³S method is significantly better than the method using linearity. The reason for this is that the complex relationship between two feature spaces is often nonlinear and cannot be approximated by a simple linear mapping.

In the comparison results with OLFES-N, our method does not lead to a steep drop in OCA in the case of evolution of the old feature space, which has a clear advantage. The reason is that the old features generated through the

proposed mapping method positively help the learning of the current features, indicating that our feature space generation strategy is reasonable and effective.

6 Conclusion

In this paper, a novel Online Learning method for Feature Evolvable Streams (OLFES) is proposed for the feature evolution problem due to the appearance and disappearance of sensors in wearable activity recognition. The method can not only learn the optimal depth of the model from the data stream, but also reconstructs the old features from the new ones after the old have disappeared, allowing the model to use the information of the old and new features and make accurate online predictions through a weighted combination. Comparative experiments with four other state-of-the-art methods on three data science datasets and three activity recognition datasets demonstrate the effectiveness of OLFES.

However, only the appearance and disappearance of sensors of the same type and frequency are considered in our method. In real life, there are many sensors of different types and frequencies. Compared with a single type of sensor, multiple types of sensors often bring complementary information that can help to further improve the performance of the model. In our future research, we will plan how to incorporate multimodal sensors into the paradigm of feature evolvable learning to further enhance the generalizability of the model.

Acknowledgement. This study is supported by the National Natural Science Foundation of China (Grant No. 62002187, No. 62276156 and No. 61976127), Qilu University of Technology (Shandong Academy of Sciences) Talent Scientific Research Project (No. 2023RCKY127 and No. 2023RCKY128), Technology Small and Medium Enterprises Innovation Capability Improvement Project of Shandong Province (No. 2023TSGC0279).

References

1. Antunes, R.S., et al.: A survey of sensors in healthcare workflow monitoring. *ACM Comput. Surv. (CSUR)* **51**(2), 1–37 (2018)
2. Wang, Y., Cang, S., Yu, H.: A survey on wearable sensor modality centred human activity recognition in health care. *Expert Syst. Appl.* **137**, 167–190 (2019)
3. Hou, B.J., Zhang, L., Zhou, Z.H.: Learning with feature evolvable streams. *IEEE Trans. Knowl. Data Eng.* **33**(6), 2602–2615 (2019)
4. He, Y., Dong, J., Hou, B.J., Wang, Y., Wang, F.: Online learning in variable feature spaces with mixed data. In: 2021 IEEE International Conference on Data Mining (ICDM), pp. 181–190. IEEE (2021)
5. Schreckenberger, C., He, Y., Lüdtkke, S., Bartelt, C., Stuckenschmidt, H.: Online random feature forests for learning in varying feature spaces. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 4587–4595 (2023)
6. Gu, S., Qian, Y., Hou, C.: Learning with incremental instances and features. *IEEE Trans. Neural Netw. Learn. Syst.* (2023)

7. Nie, X., Gao, R., Wang, R., Xiang, D.: Online multiview deep forest for remote sensing image classification via data fusion. *IEEE Geosci. Remote Sens. Lett.* **18**(8), 1456–1460 (2020)
8. He, Y., Wu, B., Wu, D., Beyazit, E., Chen, S., Wu, X.: Toward mining capricious data streams: a generative approach. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(3), 1228–1240 (2020)
9. Liu, Y., Li, W., Gao, Y.: Confidence-weighted learning for feature evolution. *Int. J. Softw. Inf.* **12**(2) (2022)
10. Zheng, T., Wang, X., Chen, Y., Yu, F., Li, J.: Feature evolvable learning with image streams. In: *Intelligent Data Analysis (Preprint)*, pp. 1–17 (2022)
11. Lian, H., Atwood, J.S., Hou, B.J., Wu, J., He, Y.: Online deep learning from doubly-streaming data. In: *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 3185–3194 (2022)
12. Bao, L., Intille, S.S.: Activity recognition from user-annotated acceleration data. In: Ferscha, A., Mattern, F. (eds.) *Pervasive 2004*. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24646-6_1
13. Zhao, H., Liu, Z.: Human action recognition based on non-linear svm decision tree. *J. Comput. Inf. Syst.* **7**(7), 2461–2468 (2011)
14. Zhao, Z., Chen, Y., Liu, J., Shen, Z., Liu, M.: Cross-people mobile-phone based activity recognition. In: *Twenty-Second International Joint Conference on Artificial Intelligence*. Citeseer (2011)
15. Ronao, C.A., Cho, S.-B.: Deep convolutional neural networks for human activity recognition with smartphone sensors. In: Arik, S., Huang, T., Lai, W.K., Liu, Q. (eds.) *ICONIP 2015*. LNCS, vol. 9492, pp. 46–53. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26561-2_6
16. Zhou, X., Guo, J., Wang, S.: Motion recognition by using a stacked autoencoder-based deep learning algorithm with smart phones. In: Xu, K., Zhu, H. (eds.) *WASA 2015*. LNCS, vol. 9204, pp. 778–787. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21837-3_76
17. Vollmer, C., Gross, H.-M., Eggert, J.P.: Learning features for activity recognition with shift-invariant sparse coding. In: Mladenov, V., Koprinkova-Hristova, P., Palm, G., Villa, A.E.P., Appollini, B., Kasabov, N. (eds.) *ICANN 2013*. LNCS, vol. 8131, pp. 367–374. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40728-4_46
18. Zeng, M., Nguyen, L.T., Yu, B., Mengshoel, O.J., Zhu, J., Wu, P., Zhang, J.: Convolutional neural networks for human activity recognition using mobile sensors. In: *6th International Conference on Mobile Computing, Applications and Services*, pp. 197–205. IEEE (2014)
19. Wang, J., Long, Q., Liu, K., Xie, Y., et al.: Human action recognition on cellphone using compositional BIDIR-LSTM-CNN networks. In: *2019 International Conference on Computer, Network, Communication and Information Systems (CNCI 2019)*, pp. 687–692. Atlantis Press (2019)
20. Hammerla, N.Y., Halloran, S., Plötz, T.: Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880* (2016)
21. Zhou, Z.H.: Open-environment machine learning. *Natl. Sci. Rev.* **9**(8), nwac123 (2022)
22. Guan, S.U., Liu, J.: Incremental neural network training with an increasing input dimension. *J. Intell. Syst.* **13**(1), 71–94 (2004)

23. Wang, T., Liu, F., Zhou, W., Zhu, X., Guan, S.U.: Neural incremental attribute learning based on principal component analysis. In: 2016 IEEE International Conference on Big Data Analysis (ICBDA), pp. 1–5. IEEE (2016)
24. Hu, C., Chen, Y., Peng, X., Yu, H., Gao, C., Hu, L.: A novel feature incremental learning method for sensor-based activity recognition. *IEEE Trans. Knowl. Data Eng.* **31**(6), 1038–1050 (2018)
25. Zhang, Y., Chen, Y., Yu, H., Yang, X., Sun, R., Zeng, B.: A feature adaptive learning method for high-density semg-based gesture recognition. *Proc. ACM Interact. Mobile Wearable Ubiqu. Technol.* **5**(1), 1–26 (2021)
26. Zhang, Z.Y., Zhao, P., Jiang, Y., Zhou, Z.H.: Learning with feature and distribution evolvable streams. In: International Conference on Machine Learning, pp. 11317–11327. PMLR (2020)
27. Hou, C., Zhou, Z.H.: One-pass learning with incremental and decremental features. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(11), 2776–2792 (2017)
28. Sahoo, D., Pham, Q., Lu, J., Hoi, S.C.: Online deep learning: learning deep neural networks on the fly. arXiv preprint [arXiv:1711.03705](https://arxiv.org/abs/1711.03705) (2017)
29. Hoi, S.C., Jin, R., Zhao, P., Yang, T.: Online multiple kernel classification. *Mach. Learn.* **90**, 289–316 (2013)
30. Goodfellow, I., et al.: Generative adversarial networks. *Commun. ACM* **63**(11), 139–144 (2020)
31. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
32. Bock, R.: MAGIC Gamma Telescope. UCI Machine Learning Repository (2007)
33. Becker, B., Kohavi, R.: Adult. UCI Machine Learning Repository (1996)
34. Dry Bean Dataset: UCI Machine Learning Repository (2020)
35. Barshan, B., Altun, K.: Daily and Sports Activities. UCI Machine Learning Repository (2013)
36. Reiss, A.: PAMAP2 Physical Activity Monitoring. UCI Machine Learning Repository (2012)
37. Blunck, Henrik, B.S.P.T.K.M., Dey, A.: Heterogeneity Activity Recognition. UCI Machine Learning Repository (2015). <https://doi.org/10.24432/C5689X>