BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 07/15 pp87–106
DOI: 10.26599/BDMA.2023.9020006
Volume 7, Number 1, March 2024

Incremental Data Stream Classification with Adaptive Multi-Task Multi-View Learning

Jun Wang, Maiwang Shi, Xiao Zhang*, Yan Li, Yunsheng Yuan, Chenglei Yang, and Dongxiao Yu*

Abstract: With the enhancement of data collection capabilities, massive streaming data have been accumulated in numerous application scenarios. Specifically, the issue of classifying data streams based on mobile sensors can be formalized as a multi-task multi-view learning problem with a specific task comprising multiple views with shared features collected from multiple sensors. Existing incremental learning methods are often single-task single-view, which cannot learn shared representations between relevant tasks and views. An adaptive multi-task multi-view incremental learning framework for data stream classification called MTMVIS is proposed to address the above challenges, utilizing the idea of multi-task multi-view learning. Specifically, the attention mechanism is first used to align different sensor data of different views. In addition, MTMVIS uses adaptive Fisher regularization from the perspective of multi-task multi-view learning to overcome catastrophic forgetting in incremental learning. Results reveal that the proposed framework outperforms state-of-the-art methods based on the experiments on two different datasets with other baselines.

Key words: data stream classification; mobile sensors; multi-task multi-view learning; incremental learning

1 Introduction

Data stream classification is the process of analyzing big data in real time to enable intelligent systems, such as online system monitoring, intelligent transportation systems[1], and financial risk management. With the rapid development of mobile networks and social media, data stream classification has gained increasing interest in academia and industry[2]. In addition, data stream analysis is a crucial component of the big data and analytics technology that enables Industry 4.0[3].

The demand for real-time data stream processing is increasing. Therefore, developing incremental learning methods is urgently needed. Incremental learning is a learning paradigm that allows continuous model updating based on new incoming data rather than training it once on the entire dataset. The purpose of incremental learning is to process the continuous flow of information in the real world, and retain, integrate, and optimize previous knowledge based on absorbed new knowledge. In the field of machine learning, incremental learning is dedicated to solving a common flaw in model training: catastrophic forgetting[4–6]. That is to say, when a general machine learning model is trained on an entirely new task, its performance on the previous task will typically degrade prominently. Multi-task multi-view learning provides a useful paradigm for various types of data collected from heterogeneous devices with multiple modalities in real-world scenarios, such as sensor-based Human Activity Recognition (HAR)[7] and traffic data analysis in smart

• Jun Wang, Maiwang Shi, Xiao Zhang, Yan Li, Yunsheng Yuan, and Dongxiao Yu are with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China. E-mail: wj_521@sdu.edu.cn; 975255673@qq.com; xiaozhang@sdu.edu.cn; yhyanli@163.com; 995716009@qq.com; dxyu@sdu.edu.cn.

• Chenglei Yang is with the School of Software, Shandong University, Jinan 250101, China. E-mail: chl_yang@sdu.edu.cn.

* To whom correspondence should be addressed.

Manuscript received: 2022-11-27; revised: 2023-04-19; accepted: 2023-04-25

cities. The goal of multi-task multi-view learning is to improve efficiency and accuracy by jointly learning objectives of multiple tasks with multiple views collected from different sources[8]. Compared to models with single-task multi-view learning, models with multi-task multi-view learning paradigms have high efficiency and low inference cost during training. The models are implemented in a way that promotes generalization capabilities using simultaneous learning of shared representations between related tasks and views[9−11]. However, research on the application of multi-task multi-view learning to incremental learning is limited. In the multi-task multi-view scenarios, learning a task can first cause interference with previously learned tasks due to updates in shared parameters. Second, different sensors can provide various views, leading to differences in input data that may be ignored in the learning process. Finally, data imbalance, in which certain tasks may have more samples than others, can lead to the neglect of less represented tasks in favor of overrepresented tasks during the learning process, resulting in catastrophic forgetting.

Some challenges arise when solving multi-task multi-view incremental data stream classification on mobile sensors. (1) Sensor data are continuously generated at a fast pace, occupying a large storage space. Therefore, incremental learning is an urgent need. (2) Different tasks and views interfere with each other in the training process, and data distribution differs between various tasks or views. Thus, the heterogeneity of tasks and views often differs among participants and sensors when performing the same activity. Using the traditional single-task single-view incremental learning may not effectively handle the specific catastrophic forgetting. (3) Existing methods addressing catastrophic forgetting, such as the traditional Elastic Weight Consolidation (EWC) approach[12], often ignore the significance of different components in the Fisher information matrix and fail to adapt to the evolution of the model structure. However, current data stream classification research only partially addresses these challenges, and few attempts have been made to address all these challenges within a single framework.

For example, the method in Ref. [13] was designed to run on mobile devices, where convolutional neural networks are used to learn local interactions in each perceptual view as well as global interactions between different sensor inputs and perform regression and classification tasks based on data from moving sensors. Reference [14] used incremental learning to build a personalized recognition model of human activities. The algorithm Learn++ is an integrated method that can use any classifier as a basic classifier. Reference [15] proposed a criterion of the “Kappa” architecture (namely MOLESTRA) to ensure the utilization of cognitive and learning relationships among the streaming data. Reference [16] transfered multi-view data stream from multiple views to a shared potential subspace and integrated distinguishing information by maximizing and minimizing the interclass and intraclass separability of streaming data, respectively. However, the existing work has not addressed all the aforementioned challenges of streaming data analysis within the overall framework.

Therefore, an adaptive Multi-Task Multi-View Incremental learning framework for data Stream classification (MTMVIS) is proposed to solve the above challenges. The general overview of MTMVIS is depicted in Fig. 1, where the notations are shown in Table 1. First, a hierarchical attention mechanism is used to weigh each measurement value within a certain time interval to align data collected from different sensors at the same body location. Task-view relationships with Multi-gate Mixture-of-Expert (MMoE)[17] are then comprehensively modeled A gating network for each task-view, which can determine a customized feature representation by combining the output of multiple experts, is constructed. In addition, another attention layer is used to construct a view fusion layer for each task to measure the importance of different views to the same task. Inspired by a single-task single-view method, i.e., IADM[18], a special adaptive output layer for each task is introduced, and each layer of the network in the model is trained to obtain a specialized feature representation, which serves as the input to the final output layer. Each task obtains a specific output representation by weighting the output combination of the multilayer network. In addition, homoscedastic uncertainty[19] is used to tune the loss weights of tasks adaptively during training. Finally, adaptive Fisher regularization is employed to overcome the catastrophic forgetting problem. Results of extensive experiments on two different datasets demonstrate that MTMVIS is superior to other state-of-the-art methods. The main contributions can be summarized as follows.

- A novel framework for adaptive multi-task multi-view incremental learning is proposed to address the

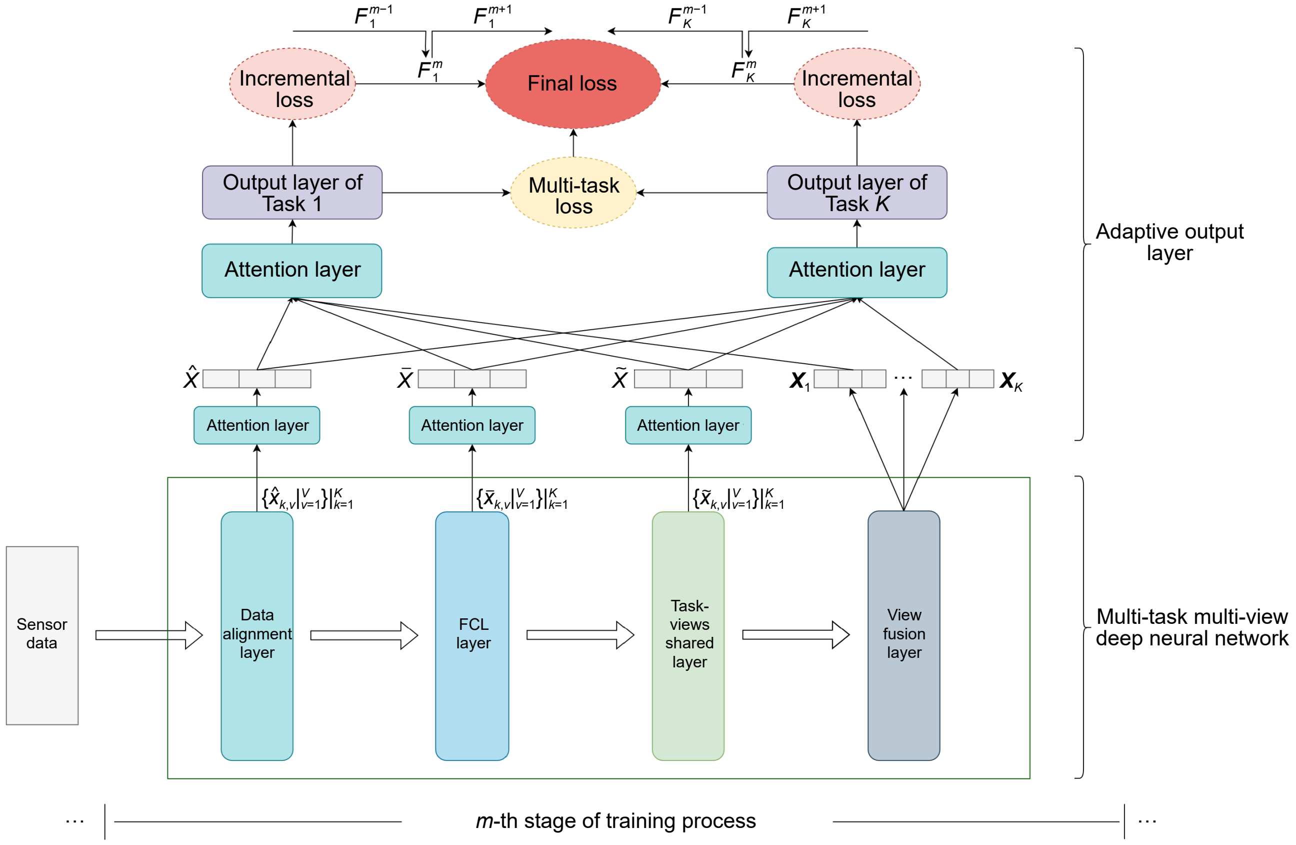


Fig. 1 Overview of the proposed MTMVIS framework. Specifically, each layer of the multi-task multi-view neural network is used to extract features separately, and the attention layer is employed to weight the output of all layers of the multi-task multi-view deep neural network as the final output layer for each task. Adaptive Fisher regularization is utilized to alleviate catastrophic forgetting problems and enhance model scalability. The details of multi-task multi-view deep neural network are shown in Fig. 3 in the following Section 3.1. The notations appearing in this figure are shown in Table 1.

catastrophic forgetting problem. Different from traditional incremental learning methods, the proposed model utilizes the relevant knowledge of multi-task multi-view learning.

- Compared with traditional EWC methods for solving the catastrophic forgetting problem, adaptive Fisher regularization is introduced to enhance the scalability of the model.

- The proposed approach is evaluated on the basis of two real datasets, namely RealWorld-HAR[20] and GLass Eating And Motion (GLEAM)[21], to demonstrate the effectiveness and efficiency of MTMVIS.

2 Related Work

2.1 Multi-task learning for data stream classification

The volume of data stream displaying heterogeneity has been increasing with the rise of mobile networks. Moreover, different participants exhibit various features for the same activity due to variations in sensor worn, usage environment, or personal habits, which cannot be resolved using traditional single-task models. Multi-task learning approaches generally provide superior prediction results because the noise of a single model is biger than the combined noise of the corresponding multiple models. Multi-task learning is a machine learning paradigm that leverages inter-task correlations. This approach learns all relevant tasks and knowledge among these tasks simultaneously to enhance generalization performance[22]. The two most common approaches for multi-task learning are hard parameter sharing and soft parameter sharing according to Ref. [10]. Previous work has focused on joint feature selection and feature learning, where task correlations are represented as mutual blocks of tasks[23–25].

Most work assumes that multiple tasks utilize a

Table 1 Notations used in this paper.

Variable	Description
K	Number of tasks
V	Number of views
S	Number of sensors
N	Amount of data in one window
$x_{k,v}^{n,s}$	The n-th data generated by the s-th senson from the v-th view of the k-th task in one window
X_{k_l}	Data generated by the l-th attention layer for the k-th task in one window
$\hat{x}_{k,v}$	Sum of the representation of all measurements weighted by the attention values for the k-th task
$\bar{x}_{k,v}$	Feature representation of the v-th view of the k-th task
$\tilde{x}_{k,v}$	Gate's output representation of the v-th view of the k-th task
$\widehat{X}$	Sum of the representation of all input feature vectors $\hat{x}_{k,v}$, weighted by the attention values
$\bar{X}$	Sum of the representation of all input feature vectors $\bar{x}_{k,v}$, weighted by the attention values
$\widetilde{X}$	Sum of the representation of all input feature vectors $\tilde{x}_{k,v}$, weighted by the attention values
$\vec{X}_R$	Fusion represention of feature $\tilde{x}_{k,v}$, weighted by the attention values
t_m	Total number of time windows within the m-th stage
λ_{MV}	Hyperparameter on multi-view loss L_{MV} in training process
λ_{MT}	Hyperparameter on multi-task loss L_{MT} in training process
$\theta_k^{m_p}$	p-th parameter in the mode after the training of the m-th stage for the k-th task
$F_k^{m_p}$	p-th diagonal value of the diagonal Fisher matrix of the m-th stage for the k-th task

shared representation. However, in practical applications, sharing a set of common features for all tasks often ignores task-specific features due to task heterogeneity. Some recent methods have proposed different viewpoints. Inspired by Mixture-of-Expert (MoE)[26], Ref. [17] proposed MMoE, which introduces multiple expert networks and then introduces a gating network for each task. The gating network learns different combinations of expert networks for each task; that is, adaptive weighting is performed on the output of the expert network. In addition, the conventional method for multi-task loss simply adds the loss of each task or sets a uniform weight loss. Furthermore, manually adjusting the weights is an inefficient and difficult task. Reference [19] proposed the methed of setting the weight of the loss function of different tasks by considering the homoscedastic uncertainty between different tasks.

Furthermore, some works consider data stream classification based on multi-task learning. For example, Ref. [27] proposed a multi-task deep learning architecture for maritime surveillance using Automatic Identification System (AIS) data streams. Multi-task learning is used to learn multiple tasks simultaneously via shared layers and task-specific layers. The fusion layer is utilized to combine the learned representations from different tasks and produces a final prediction. Additionally, Ref. [28] proposed a novel approach for multi-task data stream classification using a double-coupling learning method. The proposed model mainly focuses on coupling the mutual influences between different tasks and the temporal relationship within data streams to achieve multi-task data stream classification and avoid catastrophic forgetting. Furthermore, MOLESTRA[15] combines multi-task grouping and overlapping learning techniques with sliding learning windows for data stream classification. Moreover, Ref. [29] developed two complex models based on a fully convolutional network to predict node and edge flows, which are connected and trained together by coupling their respective potential representatives to the middle layer. However, most existing studies only consider multiple tasks and fail to analyze additional fine-grained information. Some models cannot even handle incremental data or ignore the existing catastrophic forgetting problem in multi-task incremental learning. Developing approaches for predicting data stream classification incrementally using multi-task learning is necessary to solve task heterogeneity to address the aforementioned issue.

2.2 Multi-view learning for data stream classification

The sensors located in different parts of the body acquire a considerable amount of information due to the increase in the number of various sensors. For instance, sensors located in distinct areas of the body are often considered distinct views. Moreover, co-learnable features exist across different views. Therefore, a multi-view learning approach can be employed to tackle this issue.

Multi-view learning emulates the consistency and diversity among multiple data views, resulting in superior learning performance compared to single-view approaches. In traditional multi-view learning, functions are employed to model specific views, and all functions are jointly optimized to leverage redundant views effectively with identical inputs, thereby enhancing learning performance[30]. Multi-view learning is also frequently employed to enhance single-view clustering techniques[31]. Reference [32] discovered that Expectation-Maximization (EM) based multi-view algorithms demonstrate considerably superior performance to their corresponding single-view counterparts. Furthermore, recent studies have demonstrated that multi-view spectral clustering holds substantial research implications. References [33–35] provided insights into the challenges and approaches encountered by Low-Rank Representations (LRR) in multi-view data clustering. Reference [33] proposed structured LRR via decomposing potential low-dimensional data clustering representations that characterize the data clustering structure of each view. Reference [34] demonstrated that LRR essentially comprises an optimized local structure for spectral clustering, which is a potential clustering orthogonal projection-based representation, leading to a highly desirable multi-view consensus. Reference [35] developed an effective and efficient approach to learning low-rank kernelized hash functions that are shared across views.

Moreover, the co-training algorithm[36] utilizes a view predictor to forecast other view labels, which, in turn, extends the training dataset. Reference [37] constructed a "co-regularization" specification associates with data, where a synthetic eproduction kernel associated with a single Reproducing Kernel Hilbert Space (RKHS), simulated the theoretical analysis, and extended the algorithm for co-regularization. References [22, 38] explored the basis vectors of two sets of variables by maximizing the correlation between their basis vectors. Therefore, these vectors were directly applied to dual-view data for selecting shared potential subspaces. Further development of these vectors was realized in multi-view regression[39].

Some existing works consider data stream classification based on multi-view learning. For example, Ref. [40] built an integrated global clustering model on each data block, integrating data from multiple views in a streaming manner through a split-merge clustering algorithm applied at each time window. Reference [16] transfered multi-view streaming data from different views to a shared potential subspace, and integrated distinguishing information by minimizing and maximizing the intraclass and interclass separability of streaming data, respectively. However, all these methods are only used to address single-task learning. The proposed MTMVIS focuses on the multi-task learning paradigm based on multi-view learning.

2.3 Multi-task multi-view learning

The research on multi-task multi-view learning has recently generated a wide range of interest. Many real-world scenarios can be formalized as multi-task multi-view learning problems, such as human activity recognition and news classification. Reference [8] is the earliest work on multi-task multi-view learning, which presented a graph-based framework where an effective algorithm was proposed to manipulate the framework (namely GraM2). However, GraM2 can only handle data with nonnegative eigenvalues. Reference [41] assumes that the views of different prediction models should be congruent, considering the idea of common normalization. Therefore, Ref. [42] proposed a highly general algorithm (namely CSL-MTMV), which assumes that multiple relevant tasks with the same view should be shared in a low-dimensional common subspace. Inspired by linear discriminant analysis, Ref. [43] focused on double heterogeneity through the shared latent space between multiple views. Reference [44] introduced algorithm MFM based on a multi-linear decomposition machine through learning the multi-linear structure shared by task-specific feature maps and task-views. A unified deep learning framework (namely DMTMV)[11] was proposed to solve multi-task multi-view problems by establishing three types of networks: feature-sharing network, specific feature network, and task network, and using layer-by-layer regularization to update parameters. However, these methods have generally assumed that views and tasks are independent and not studied the fine-grained task-view interaction relationship.

2.4 Incremental learning for data stream classification

The change in data distribution over time is usually called concept drift, which is a challenging problem in data stream classification. A considerable amount of recent work has been studying incremental learning to address the aforementioned problem[45–47]. Specifically, incremental learning can continuously process the continuous flow of information in the real world and absorb new knowledge. Simultaneously, incremental learning can retain or even integrate and optimize previous knowledge. Learning from emerging data from nonstationary data distribution is inevitably prone to inducing catastrophic forgetting[4,5]. Incremental learning uses three different paradigms, namely, regularization[48], replay[49], and parameter isolation, to address the catastrophic forgetting problem. Among these paradigms, Ref. [12] is one of the representatives of regularization. At the end of each training iteration, the EWC method computes a diagonal Fisher matrix from the final loss, where each value is a parameter reflecting the importance of the model to which it belongs. In the next iteration of the training process, the model uses a regularization term to limit the variation of the parameters.

Incremental learning for data stream classification has always been a long-standing research field[12, 45]. An ensemble-based incremental learning method for unbalanced streaming data is proposed by Ref. [50], the algorithm for concept drift, which reduces class imbalance by using bagging-based subsets and employs different methods to focus on class-specific performance. Inspired by the Hoeffding Decision Tree (HDT)[51] for efficient data stream classification, Ref. [52] used the proposed learning procedure in HDT to adapt to the recently introduced fuzzy decision tree and utilized a uniform fuzzy partition for each input feature to address the data stream classification problem. Reference [53] proposed a new incremental semi-supervised learning framework based on streaming data to address the problem of insufficient labeled samples in data stream classification. This framework uses autoencoders for learning features generated from streaming data to normalize the network construction by establishing pairwise similarity and dissimilarity constraints. However, the existing approaches only focus on solving the catastrophic forgetting of models in single-task scenarios. Data stream classification can generally be formalized as a multi-task multi-view problem. Therefore, using the idea of multi-task multi-view learning may be necessary to solve catastrophic forgetting.

3 MTMVIS: Adaptive Multi-Task Multi-View Incremental Learning Framework for Data Stream Classification

Owing to the nonstationary distribution characteristics of streaming data, old knowledge is disturbed by new knowledge when the knowledge is continuously acquired from nonstationary data distribution, resulting in a rapid decline in overall performance. This phenomenon is called catastrophic forgetting. Incremental learning is introduced to address the issue of catastrophic forgetting in data streams. Furthermore, the following two scenarios involving sensor data are examined: the first scenario’s data from the same type of sensors are gathered from different participants, and the second scenario’s data are collected from different parts of the same body using various sensors. Task heterogeneity or view heterogeneity arise because each participant or sensor behaves differently for the same activity. Each participant is assumed to represent a specific task, and each sensor corresponds to a view, as shown in Fig. 2. A multi-task multi-view incremental learning approach is adopted to tackle these scenarios. Therefore, an adaptive multi-task multi-view incremental learning framework for data stream classification, called MTMVIS, is proposed as depicted in Fig. 1. Specifically, the proposed MTMVIS first aligns the data to extract features effectively. Then, MTMVIS uses MMoE[17] to select appropriate expert networks for different task-views and construct a special view fusion layer for each task. Finally, MTMVIS combines adaptive Fisher regularization and task homoscedasticity uncertainty[19] to solve the catastrophic forgetting problem. The details of the MTMVIS are presented as follows.

3.1 Problem statement

Using incremental learning to predict the data stream classification on the mobile sensors and improve the overall performance of the model via multi-task multi-view learning is considered in this paper. K tasks, where each task comprises V different views, are presented considering a real-world scenario of different individuals wearing multiple sensors of the same number on different body parts. Specifically, the streaming data generated by sensors are manually divided into M stages, and each stage has t_m time windows with T seconds, where $m \in \{1, 2, \ldots, M\}$. The activities that appear in different stages, such as sitting, running, and climbing up, are assumed to be random and repeatable considering the general situation; incremental learning must be used to distinguish these activities. For each stage, the data from different sensors in each time window can be represented as $X_{k,v} = \{x^1_{k,v}, x^2_{k,v}, \ldots, x^N_{k,v}\}$, where $k \in \{1, 2, \ldots, K\}$ and $v \in \{1, 2, \ldots, V\}$, and N indicates the total amount of

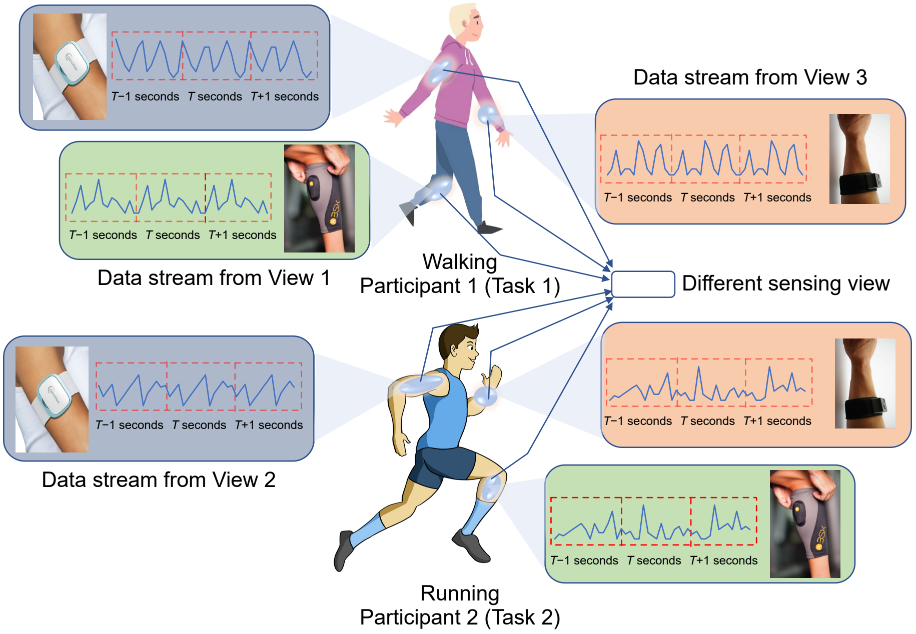


Fig. 2 Assumption indicates that each participant represents a specific task, each sensor corresponds to a view, and the collected data by the sensor are the data stream. Task heterogeneity and view heterogeneity arise because each participant or sensor behaves differently for the same activity. Thus, the multi-task and multi-view incremental learning method is used to solve these problems.

generated data. In the proposed model, $x^n_{k,v}$ can be generat- ed by different sensor data at the same moment through the same attention layer, as shown in Fig. 3, where $n \in \{1, 2, \ldots, N\}$. Within the m-th stage, the data of the k-th task can be denoted as $\{X^t_k\}^{t_m}_{t=0}$. The goal is to learn an adaptive multi-task multi-view incremental model to achieve the mapping from data collected by different sensors to y^m_k, where y^m_k represents the label of the k-th task at the m-th stage,

$$f\left(\left\{X^t_k\right\}^{t_m}_{t=0}\right) \to y^m_k \tag{1}$$

3.2 Data alignment

In the collected datasets, the input data come from different sensors, which have various effects on the prediction results of the same activity simultaneously. In addition, different measurement values generated in the same time interval may have various effects on the prediction results. The hierarchical attention mechanism is introduced to perform the above situations separately to address heterogeneity. The attention mechanism can assign weights based on the importance of different parts and is already widely used in computer vision tasks. Specifically, in the first part of the attention layer, the data of different sensors located in the same part of the body are aligned, and a fixed-length feature vector is generated during each time interval with T seconds. In the second part of the attention layer, the different measurement values generated in the same time interval are weighted to achieve the alignment of different data generated in the same time interval.

The details of each part of the attention layer can be formalized as follows:

$$a^{n,s}_{k,v} = \frac{\exp\left(\tanh\left(w^1_{k,v} \cdot x^{n,s}_{k,v} + b^1_{k,v}\right)\right)}{\sum\limits_{s=1}^{S} \exp\left(\tanh\left(w^1_{k,v} \cdot x^{n,s}_{k,v} + b^1_{k,v}\right)\right)},$$
$$x^n_{k,v} = \sum_{s=1}^{S} a^{n,s}_{k,v} \cdot x^{n,s}_{k,v} \tag{2}$$

$$a^n_{k,v} = \frac{\exp\left(\tanh\left(w^2_{k,v} \cdot x^n_{k,v} + b^2_{k,v}\right)\right)}{\sum\limits_{n=1}^{N} \exp\left(\tanh\left(w^2_{k,v} \cdot x^n_{k,v} + b^2_{k,v}\right)\right)},$$
$$\hat{x}_{k,v} = \sum_{n=1}^{N} a^n_{k,v} \cdot x^n_{k,v} \tag{3}$$

where $a^{n,s}_{k,v}$ is the attention value representing the significance of the data collected by the$s$-th sensor, and $s \in \{1, 2, \ldots, S\}$. $x^{n,s}_{k,v}$ is the s-th feature representation vector of the sensor, and $x^n_{k,v}$ is the sum of the

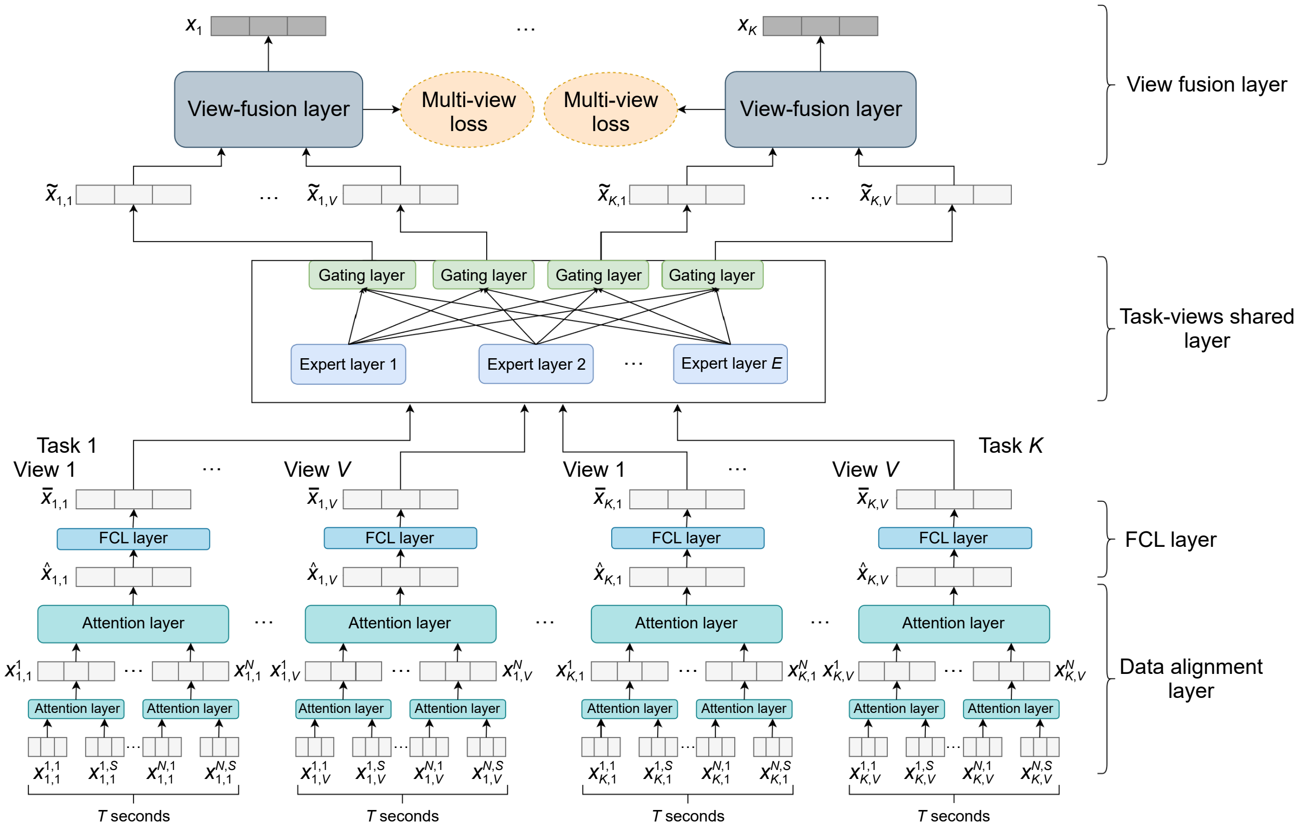


Fig. 3 Multi-task multi-view deep neural network of the proposed model MTMVIS. The MTMVIS model initially aligns the data to facilitate optimal feature extraction. Subsequently, this model employs MMoE to choose the most suitable expert network for diverse task views and establishes a specialized view fusion layer for each task. The notations appearing in this figure are shown in Table 1.

representation feature vectors of all sensors weighted by the above attention values. $a^n_{k,v}$ is the attention value, which evaluates the significance of the n-th measurement, and $\hat{x}_{k,v}$ is the sum of the representation feature vectors of all measurements weighted by the above attention values. Function tanh(·) is used as the activation function to enhance the nonlinear capability. $w^1_{k,v}$, $b^1_{k,v}$, $w^2_{k,v}$, and $b^2_{k,v}$ are parameters that must be determined by learning.

The learning capability of the model is further improved by using a fully connected neural network with two hidden layers at the output,

$$\bar{x}_{k,v} = \mathrm{FCL}(\bar{x}_{k,v}) \tag{4}$$

where FCL(·) represents the fully connected neural network with two hidden layers, and $\bar{x}_{k,v}$ denotes the feature representation of the v-th view of k-th task.

3.3 Task-views shared layer with multi-gate mixture-of-experts

After computing the representation of each view of each task within the k-th stage, we then model the relationship among task-views. Most prior studies utilized soft-sharing constraints and are based on the division of the Multi-Task Multi-View Learning (MTMVL) problem into multiple multi-task learning problems, which are considered as separate approaches. Moreover, in these studies, it is assumed that all views are conditionally independent[11, 42, 43], and finer-grained task-view-interaction correlations are ignored. Inspired by MMoE and the recent MoE layer, we propose an MMoE network with feature representation obtained by FCL(·) for each view of each task. The MMoE network consists of $K \times V$ gates and E experts, as shown in Fig. 3. Here, the gating is a trainable gated network that selects different combinations of experts to process the input data. The gated network for each task view achieves selective use of experts by different final output weights, and the gated network for different task views can learn different combinations of experts. Based on the output of multiple experts, each task view can get customized feature representations which will help in the training of specific tasks.

MMoE networks can formally be identified as follows:

$$\tilde{x}^{e}_{k,v} = \mathrm{ReLU}\left(w^{e}_{k,v} \cdot \bar{x}_{k,v} + b^{e}_{k,v}\right),$$
$$c^{e}_{k,v} = \frac{\exp\left(\mathrm{ReLU}\left(w^{e}_{k,v} \cdot \bar{x}_{k,v} + b^{e}_{k,v}\right)\right)}{\sum\limits_{e=1}^{E} \exp\left(\mathrm{ReLU}\left(w^{e}_{k,v} \cdot \bar{x}_{k,v} + b^{e}_{k,v}\right)\right)},$$
$$\tilde{x}_{k,v} = \sum_{e=1}^{E} \left(c^{e}_{k,v} \cdot \tilde{x}^{e}_{k,v}\right) \tag{5}$$

where $\bar{x}_{k,v}$ is the representation generated by the FCL layer, $\tilde{x}^{e}_{k,v}$ represents output representation of the e-th expert, $w^{e}_{k,v}$ is the weight of the e-th expert, $\tilde{x}_{k,v}$ represents the gate's output representation of the v-th view of the k-th task, and $\{w^{e}_{k,v}, b^{e}_{k,v}\}$ are the parameters that need to be learned. We utilize the function ReLU as the activation function.

3.4 View fusion layer

Based on the assumption that all views are conditionally independent, the traditional method transforms the multi-task multi-view learning problem into the multi-task learning problem; that is, the previous multi-task multi-view learning model simply represented the average of the output of all views as the task output. In our work, after getting a better representation of task-view features, we use an additional attention layer to fuse view features from each task. Considering that different views may contribute differently to different tasks, we utilize an attention layer to calculate the importance of each view.

For a single task, its attention layer can be formalized as follows:

$$d_{k,v} = \frac{\exp(\tanh(w_{k,v} \cdot \tilde{x}_{k,v} + b_{k,v}))}{\sum\limits_{v=1}^{V} \exp(\tanh(w_{k,v} \cdot \tilde{x}_{k,v} + b_{k,v}))},$$
$$\vec{X}_{k} = \sum_{v=1}^{V} d_{k,v} \cdot \tilde{x}_{k,v} \tag{6}$$

where $d_{k,v}$ is the importance measure of the v-th view for the k-th task, and $\{w_{k,v}, b_{k,v}\}$ are the parameters that need to be learned. We use function tanh as our activation function to enhance the nonlinear capability.

Thus, we utilize regularization term to represent multi-view loss as follows:

$$L_{\mathrm{MV}} = \lambda_{\mathrm{MV}} \sum_{k=1}^{K} \sum_{v=1}^{V} \frac{1}{V} \left\| f_{k}(\cdot) - f_{k,v}(\cdot) \right\|_{2} \tag{7}$$

where $f_{k}(\cdot)$ represents the view fusion function for the k-th task with all views fusion, we utilize $f_{k,v}(\cdot)$ to denote the task-view-specific model function for the v-th view of k-th task, λ_{MV} is a hyperparameter.

3.5 Adaptive output layer

A considerable amount of stream data is constantly evolving in nature. That is, the joint distribution between the ground truth and the input features varies due to concept drift[54]. If the changes in the distribution are ignored, then the performance of the previously obtained distribution would have been markedly degraded, consequently leading to a catastrophic forgetting problem[55]. Some works recently utilize the Fisher information matrix[56] to address the above problem. However, these works ignore the significance of the different components in the Fisher information matrix. Therefore, inspired by IADM[18], a special adaptive output layer is proposed for each task. Unlike the original network that uses the final feature representation for prediction, in MTMVIS, Fig. 1 shows that customized feature representations in the adaptive output layer are obtained at the end of the attention layer as the input part of the final output layer by training each network layer in the model. Each task obtains a specific output representation that will help improve the overall prediction performance by weighing the combination of the multilayer network output.

The details of the adaptive output layer for each task can be formalized as follows:

$$\hat{\beta}_{k,v} = \frac{\exp\left(\tanh\left(\widehat{w}_{k,v} \cdot \hat{x}_{k,v} + \hat{b}_{k,v}\right)\right)}{\sum\limits_{k=1}^{K} \sum\limits_{v=1}^{V} \exp\left(\widehat{w}_{k,v} \cdot \hat{x}_{k,v} + \hat{b}_{k,v}\right)\right)},$$
$$\widehat{X} = \sum_{k=1}^{K} \sum_{v=1}^{V} \hat{\beta}_{k,v} \hat{x}_{k,v} \tag{8}$$

$$\bar{\beta}_{k,v} = \frac{\exp\left(\tanh\left(\bar{w}_{k,v} \cdot \widehat{x}_{k,v} + \bar{b}_{k,v}\right)\right)}{\sum\limits_{k=1}^{K} \sum\limits_{v=1}^{V} \exp\left(\bar{w}_{k,v} \cdot \bar{x}_{k,v} + \bar{b}_{k,v}\right)\right)},$$
$$\bar{X} = \sum_{k=1}^{K} \sum_{v=1}^{V} \bar{\beta}_{k,v} \cdot \bar{x}_{k,v} \tag{9}$$

$$\widetilde{\beta}_{k,v} = \frac{\exp\left(\tanh\left(\widetilde{w}_{k,v} \cdot \tilde{x}_{k,v} + \tilde{b}_{k,v}\right)\right)}{\sum\limits_{k=1}^{K}\sum\limits_{v=1}^{V} \exp\left(\widetilde{w}_{k,v} \cdot \tilde{x}_{k,v} + \tilde{b}_{k,v}\right)\Big)},$$

$$\widetilde{X} = \sum_{k=1}^{K}\sum_{v=1}^{V} \widetilde{\beta}_{k,v} \cdot \tilde{x}_{k,v} \tag{10}$$

$$\gamma_{k_l} = \frac{\exp\left(\tanh\left(w_k \cdot X_{k_l} + b_k\right)\right)}{\sum\limits_{l=1}^{L} \exp\left(\tanh\left(w_k \cdot X_{k_l} + b_k\right)\right)}, \tag{11}$$

$$F_k = w_{\text{out}}\left(\sum_{l=1}^{L} \gamma_{k_l} \cdot X_{k_l}\right) + b_{\text{out}} \tag{12}$$

where $\widehat{\beta}_{k,v}$, $\bar{\beta}_{k,v}$, and $\widetilde{\beta}_{k,v}$ denote the attention value that evaluates the significance of the input feature vector $\hat{x}_{k,v}$, $\bar{x}_{k,v}$, and $\tilde{x}_{k,v}$, respectively; $\widehat{X}$, $\bar{X}$, and X denote the sum of the representation of all input feature vectors $\hat{x}_{k,v}$, $\bar{x}_{k,v}$, and $\tilde{x}_{k,v}$, respectively, weighted by the attention values. γ_{k_l} is the attention value that assesses the significance of the l-th layer, where $X_{k_1} = \widehat{X}$, $X_{k_2} = \bar{X}$, $X_{k_3} = \widetilde{X}$, and $X_{k_4} = \vec{X}_k$, and F_k denotes the k-th task output. $\widehat{w}_{k,v}, \hat{b}_{k,v}, \bar{w}_{k,v}, \bar{b}_{k,v}, \widetilde{w}_{k,v}, \tilde{b}_{k,v}, w_k, b_k, w_{\text{out}}$, and b_{out} are the parameters that must be learned.

Cross entropy is used in the current work. The classification loss can be formalized as follows:

$$L_{\text{CL}} = \sum_{k=1}^{K} \|y_k - F_k\|_2 \tag{13}$$

3.6 Modeling tasks relationship with homoscedastic uncertainty

Simultaneously, MTMVIS is concerned with the joint optimization of multiple related tasks. The conventional method aims to define a total loss function, which is a linear combination of the loss of each individual task. However, manually adjusting the weight parameters is expensive and tricky. If the weight values of the loss of different tasks are quite different, then a certain task will dominate the value of the overall loss, and the final network learning process will eventually be affected by the loss of other tasks. Inspired by Ref. [19], the loss for each task is automatically traded off by homoscedastic uncertainty during the model training to address the above problems. The relative weights of each task in the total loss function are adjusted by deriving a multi-task loss function based on the maximum likelihood estimation of the normal distribution of task-dependent uncertainties. Specifically, the multi-task loss function is rewritten as follows:

$$L_{\text{MT}} = \lambda_{\text{MT}} \sum_{k=1}^{K} L_k\left(X_k, \theta_k, \sigma_k\right) \tag{14}$$

where $L_k(\cdot)$ is the classification loss function for the k-th task, X_k is the input data for the k-th task, θ_k denotes the model parameters for k-th task, and σ_k is the standard deviation of the Gaussian distribution as the noise of the k-th task. λ_{MT} is a hyperparameter.

The similarity likelihood is adjusted to compress a scaled version of the model output via the softmax function,

$$p\ \left(y | F_k\left(X_k\right), \sigma_k\right) = \text{softmax}\left(\frac{1}{\sigma_k^2} F_k\left(X_k\right)\right) \tag{15}$$

The output can be written as

$$p\ \left(y = c | X_k, \theta_k, \sigma_k\right) = \frac{\exp\left(\frac{1}{\sigma_k^2} f_k^c\left(X_k\right)\right)}{\sum\limits_{c=1}^{C} \exp\left(\frac{1}{\sigma_k^2} f_k^c\left(X_k\right)\right)} \tag{16}$$

where $f_k^c\left(X_k\right)$ is the c-th element of the vector $F_k\left(X_k\right)$ for the k-th task. The classification loss with uncertainty is represented by using the negative log likelihood as follow:

$$L_k\left(X_k, \theta_k, \sigma_k\right) = \sum_{c=1}^{C} -\log p\left(y = c | X_k, \theta_k, \sigma_k\right) =$$

$$\frac{1}{\sigma_k^2} \sum_{c=1}^{C} \log p\left(y = c | X_k, \theta_k\right) + \log \sigma_k \tag{17}$$

where σ_k represents the relevant weight loss for the k-th task. The corresponding weight increases when σ_k decreases, which effectively balances the loss weights of different tasks and helps improve overall performance.

3.7 Adaptive Fisher regularization for overcoming catastrophic forgetting

A model that adapts to new data often suffers from catastrophic forgetting, where it forgets what it has learned before. This phenomenon poses a practical challenge for incremental learning, i.e., minimally forgetting previous knowledge. Previous work has focused on protecting old knowledge from being overwritten by new knowledge by imposing constraints on the loss function of the new task to address the abovementioned challenge. This study takes the following view: the distribution of instances does not change significantly within a short period, such that the interest of users in following an online information stream remains stable. Moreover, a drift detection algorithm can be employed in highly complex cases to partition the stream into epochs with relatively smooth underlying distributions. Therefore, using the Fisher information matrix is proposed to regularize the conditional likelihood distribution at each stage as a forgetting metric. This matrix can be represented as follows:

$$L_{\mathrm{IL}}\left(\theta_k^{m+1}\right)=\sum_{p=1}^{P} F_k^{m_p}\left(\theta_k^{(m+1)_p}-\theta_k^{m_p}\right)^2 \tag{18}$$

where $F_k^{m_p}(\,)$ denotes the p-th diagonal value of the Fisher matrix computed after the training of the m-th stage for the k-th task, while $\theta_k^{m_p}$ is the p-th parameter in the model after the training of the m-th stage for the k-th task.

Notably, Eq. (18) only considers the Fisher information matrix of the previous stage and disregards all previous stages, which may still lead to interval forgetting. Moreover, in the incremental setting, different hidden layers of the neural network may have various importance due to the network structure evolution with an attention mechanism, resulting in different parts of the Fisher information matrix with various significance in successive stages. The attention weights are incorporated into the relevant parameters of the Fisher regularization to address this issue and gradually align the posterior distribution of the neural network trained at each stage. Therefore, Eq. (18) can be further expressed as

$$L_{\mathrm{IL}}\left(\theta_k^{m+1}\right)=\lambda_{\mathrm{IL}}\sum_{p=1}^{P}\gamma_k^m\odot F_k^{m_p}\left(\theta_k^{(m+1)_p}-\theta_k^{m_p}\right)^2 \tag{19}$$

where $\gamma_k^m=\left[\gamma_{k_1}^m,\gamma_{k_2}^m,\ldots,\gamma_{k_l}^m,\ldots,\gamma_{k_L}^m\right]$, "$\odot$" represents multiplying γ_k^m with the parameters of the corresponding layer in the Fisher matrix, and λ_{IL} is a hyperparameter that determines the significance of the first m stages compared to the new stage. This continuous average shows that the previously learned stages have less impact than the most recently learned stages. $F_k^{m_p}$ can be calculated in the following:

$$F_k^{m_p}=\sum_{t=1}^{t_m}\left(\frac{\partial L\left(X_k^t\right)}{\partial\theta_k^{m_p}}\right)^2+F_k^{(m-1)_p} \tag{20}$$

where $L\left(X_k^{t_m}\right)$ represents the loss value evaluated on the t-th sample of the m-th stage in the k-th task, and $F_k^{(m-1)_p}$ is the saved diagonal Fisher matrix computed after the training of the $(m-1)$-th stage.

Parameters related to multi-view loss and multi-task loss by integrating parameters relates to classification loss. In the next training iteration, the proposed MTMVIS uses a method that preferentially preserves the above parameters to overcome the catastrophic forgetting problem. The formula for calculating the loss function is as follows:

$$L_F\left(\theta_k^{m+1}\right)=L_{\mathrm{CL}}\left(\theta_k^{m+1}\right)+L_{\mathrm{MV}}\left(\theta_k^{m+1}\right)+L_{\mathrm{MT}}\left(\theta_k^{m+1}\right)+\sum_{k=1}^{K}L_{\mathrm{IL}}\left(\theta_k^{m+1}\right) \tag{21}$$

4 Experiment

Experiments using two real-world human activity recognition datasets are conducted in this section to demonstrate that the proposed model outperforms many existing baselines. A detailed introduction to the two datasets used, the baseline, and the experimental setup are first presented. The effectiveness of the proposed MTMVIS is then demonstrated.

4.1 Dataset descriptions

(1) RealWorld-HAR[20]: The dataset comprises motion sensor data for various activities, including stair climbing and jumping. Four males and four females are selected from a pool of 15 participants to join in the experiment. The acceleration of seven body positions during each activity is recorded using sensors: chest, forearm, head, calf, thigh, upper arm, and waist. Each participant is regarded as a separate task, and each body position is considered a distinct view to perform behavioral recognition for each participant. Thus, this dataset has eight tasks and seven views.

(2) GLEAM[21]: The GLEAM dataset is a collection of head motion tracking data gathered from Google Glass, comprising 2 hours of data spanning various activities, such as eating and walking. These labeled data obtained from the head-mounted sensor can be utilized to identify diet and other activities, ultimately aiding patients with chronic diseases. The dataset includes data from all the sensors of Glass, which include acceleration, gravity, linear acceleration, rotation, gyroscope, magnetic field, and light. Table 2 provides a brief description of these sensors. The data

Table 2 Properties of different sensors.

Sensor	What it measures	Unit
Linear acceleration	Acceleration forces along 3 axes, excluding gravity	m/s^2
Gravity	Forces of gravity along 3 axes	m/s^2
Accelerometer	Acceleration forces along 3 axes, including gravity	m/s^2
Rotation vector	Rotation vector component along 3 axes	–
Gyroscope	Rate of rotation	rad/s
Geomagnetic	Geomagnetic field strength along 3 axes	Micro-Tesla (μT)
Light	Environment luminance	lux

are collected from 38 participants aged 18–21, and eight of them are randomly selected as experimenters, each having six views (light sensor data are not used).

4.2 Baselines

Some baselines are presented in this study and compared with the proposed method.

- **DMTMV[11]:** This baseline is a deep multi-task multi-view approach in the unified framework, which learns not only nonlinear feature representation and classifier but also the relationship between tasks in nonlinear models.
- **ASM2TV[57]:** This baseline is a semi-supervised multi-task multi-view learning algorithm. ASM2TV presents a new perspective named gating control policy, a learnable task-view-interacted sharing policy that adaptively selects the most desirable candidate shared block for any view across any task.
- **IADM[18]:** An incremental adaptive model, provides an additional attention model to the hidden layers which prevents forgetting by exploiting the attention-based Fisher information matrix to address the capacity sustainability problem. Specifically, the method incorporates the attention weights obtained through learning into the corresponding model parameters of the abovementioned matrix when computing the Fisher information matrix.
- **MTIS:** A multi-task multi-view neural network and adaptive Fisher regularization are used to train the model sequentially. MTIS is the same as the MTMVIS structure, except that the attention mechanism is not used in the view fusion layer. Instead, the average of the output results of all views is used as the output vector of each task.
- **MVIS:** A multi-task multi-view neural network and adaptive Fisher regularization are used to train the model sequentially. MVIS is the same as the MTMVIS structure, except that only one expert is used in the MMoE layer.

4.3 Experiment settings

The model is trained one after another with stages of sequential style. The model no longer stores training data after completing each training stage. Each activity occurs thrice, and the sequence of activities in each stage is randomly arranged. Specifically, the measurement values from different sensors in a certain time window ($T = 2.5$ s in default) are used as the input for action recognition. For both datasets, 70% of the data are randomly selected as the training set, 10% of the data are the validation set, and the rest of the data is chosen for testing. Adam is chosen as the optimizer due to the use of adaptive Fisher regularization to tackle the catastrophic forgetting problem. $\{10^{-6}, 10^{-5}, \ldots, 10^{6}\}$ is utilized for hyperparameter λ, 10^{-4} for learning rate. Accuracy, F1-score (F1), precision, and AUC are used to measure the model performance.

4.4 Experiment results

4.4.1 Numerical results

Table 3 and Fig. 4 display the results of the proposed framework and all baselines under four metrics on the dataset RealWorld-HAR[20]. Meanwhile, Table 4 and Fig. 5 display the results of the proposed framework and all baselines under four metrics on dataset GLEAM[21]. The evaluation metrics, which are derived from eight tasks on two datasets, RealWorld-HAR [20] and GLEAM[21] (Tables 3 and 4, respectively), are reported, and the change curves of four metrics (see Figs. 4 and 5) with the number of training stages on two datasets RealWorld-HAR[20] and GLEAM[21] are presented. The following observations are provided.

- Compared with baselines, the proposed MTMVIS reaches the best predictive performance on four metrics of all tasks on two real-world datasets.
- DMTMV and ASM2TV are not designed for incremental learning scenarios. Thus, the performances of DMTMV and ASM2TV sharply decrease when faced with the training of sequential stages. The results show that when completing all stages of training, compared with DMTMV and ASM2TV, the four metrics are improved by approximately 45%.
- The performance of IADM is worse than that of MTMVIS, indicating that the attention mechanism has

Table 3 Experimental results on the dataset RealWorld-HAR[20] are shown in bold for the best model performance and underlined for the second-best results. MTMVIS generally achieves the best performance and classification results on all evaluation metrics for all presented tasks.

Method	Task	Metric			
		Accuracy	F1-score	Precision	AUC
DMTMV	Task 1	0.3291	0.2619	0.2619	0.3319
	Task 2	0.3412	0.2810	0.2721	0.3547
	Task 3	0.2893	0.2456	0.2713	0.3019
	Task 4	0.3281	0.2711	0.2814	0.3320
	Task 5	0.2697	0.2191	0.2557	0.2714
	Task 6	0.2772	0.2453	0.2133	0.2937
	Task 7	0.3803	0.3385	0.3513	0.3919
	Task 8	0.3815	0.2703	0.2917	0.4003
	Average	0.3246	0.2666	0.2748	0.3347
ASM2TV	Task 1	0.3858	0.3172	0.3169	0.3913
	Task 2	0.3549	0.2923	0.2713	0.3615
	Task 3	0.2958	0.2491	0.2729	0.3030
	Task 4	0.3217	0.2983	0.2713	0.3312
	Task 5	0.2739	0.2451	0.2715	0.3019
	Task 6	0.2815	0.2513	0.2302	0.3161
	Task 7	0.3769	0.3297	0.3521	0.3927
	Task 8	0.3979	0.3581	0.3879	0.4289
	Average	0.3361	0.2926	0.2968	0.3533
IADM	Task 1	0.5138	0.4416	0.4349	0.5387
	Task 2	0.5107	0.4603	0.4951	0.5417
	Task 3	0.4711	0.4571	0.4329	0.5015
	Task 4	0.5096	0.4505	0.4618	0.5125
	Task 5	0.4658	0.4024	0.3997	0.4884
	Task 6	0.5139	0.3617	0.3917	0.4808
	Task 7	0.5611	0.4949	0.5029	0.5892
	Task 8	0.5790	0.5127	0.5215	0.5875
	Average	0.5156	0.4477	0.4551	0.5300
MTIS	Task 1	0.5938	0.5486	0.5601	0.6377
	Task 2	0.6016	0.5812	0.5711	0.6424
	Task 3	0.6283	0.5410	0.5329	0.6595
	Task 4	0.5962	0.5415	0.5430	0.6101
	Task 5	0.5358	0.5124	0.5207	0.5674
	Task 6	0.5639	0.5310	0.5409	0.6198
	Task 7	0.6110	0.5593	0.5619	0.6382
	Task 8	0.5813	0.5411	0.5512	0.6165
	Average	0.5890	0.5445	0.5477	0.6240
MVIS	Task 1	0.5819	0.5261	0.5315	0.6239
	Task 2	0.5718	0.5081	0.5106	0.6115
	Task 3	0.5816	0.5203	0.5311	0.6058
	Task 4	0.5618	0.4828	0.5101	0.6131
	Task 5	0.4962	0.4533	0.4716	0.5666
	Task 6	0.5115	0.4810	0.4619	0.5691
	Task 7	0.5729	0.5116	0.5041	0.6032
	Task 8	0.5713	0.5463	0.5317	0.5976
	Average	0.5561	0.5037	0.5066	0.5989

(To be continued)

Table 3 Experimental results on the dataset RealWorld-HAR[20] are shown in bold for the best model performance and underlined for the second-best results. MTMVIS generally achieves the best performance and classification results on all evaluation metrics for all presented tasks. (Continued)

Method	Task	Metric			
		Accuracy	F1-score	Precision	AUC
MTMVIS	Task 1	**0.6774**	**0.6539**	**0.7661**	**0.7885**
	Task 2	**0.6511**	**0.5963**	**0.6319**	**0.7619**
	Task 3	**0.6530**	**0.6243**	**0.6172**	**0.7519**
	Task 4	**0.6456**	**0.5191**	**0.6267**	**0.7388**
	Task 5	**0.5497**	**0.5186**	**0.5262**	**0.7341**
	Task 6	**0.5763**	**0.5615**	**0.5749**	**0.7128**
	Task 7	**0.6581**	**0.6243**	**0.6296**	**0.7517**
	Task 8	**0.6524**	**0.5601**	**0.6153**	**0.7413**
	Average	**0.6330**	**0.5823**	**0.6235**	**0.7476**

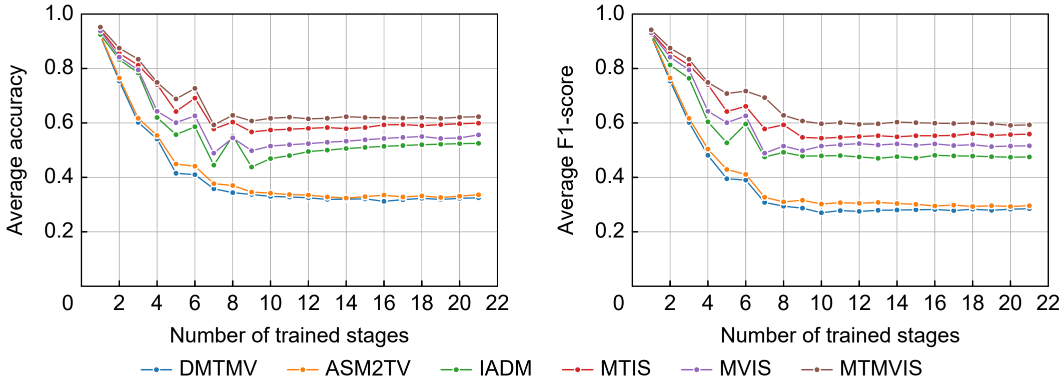


Fig. 4 Experimental results with metrics accuracy and F1-score on dataset RealWorld-HAR.

superior performance in processing time series perception data in incremental learning. The idea of multi-task multi-view learning also plays a role in improvement.

- The performance of MTIS is lower than that of the proposed MTMVIS, because MTIS discards the attention mechanism of the view fusion layer and directly averages the output of all views. This approach ignores the importance of different sensors for the same activity and the impact of sensor data at different times. By contrast, considering the different importance of various views in the same task is beneficial for learning the invariant features between views and improving overall performance.

- The performance of MVIS is worse than that of MTMVIS, because harmful interference or negative feedback cannot be avoided when all task-views share the same expert network.

- The performance of MTMVIS can be maintained at a stable level under repetitive activities. For instance, on dataset GLEAM[21] shown in Fig. 5, from the first to the third stage, all metrics decrease due to the catastrophic forgetting problem. However, the performance of the model improves from the third to the fourth stage. This phenomenon is because the activities in the first and fourth stages are both talking, and the proposed MTMVIS consolidates important parameters for talking.

4.4.2 Ablation studies

A series of ablation studies is further conducted on the effect of each component to validate the effectiveness of the proposed MTMVIS. Table 5 reports performance improvement results (in percent) on both datasets compared to IADM. Specifically, the following observations are presented.

- Compared to single-task single-view models, MTMVIS generally achieves significant improvements of over 23% and 15% for all metrics on the RealWorld-HAR and GLEAM datasets, respectively.

- MTIS can also achieve 17% and 11% significant improvements on all metrics on two datasets. This finding indicates that the appropriate number of experts often has a considerable effect on model improvement.

- MVIS has not significantly enhanced with 8% and

Table 4 Experimental results on the dataset GLEAM[21] are shown in bold for the best model performance and underlined for the second-best results. MTMVIS generally achieves the best performance on all valuation metrics for all presented tasks.

Method	Task	Metric			
		Accuracy	F1-score	Precision	AUC
DMTMV	Task 1	0.4713	0.4164	0.4377	0.4781
	Task 2	0.4651	0.4301	0.4201	0.4739
	Task 3	0.4328	0.4027	0.4152	0.4653
	Task 4	0.4218	0.3934	0.4455	0.4539
	Task 5	0.4918	0.4772	0.4801	0.4918
	Task 6	0.4129	0.4239	0.4011	0.5063
	Task 7	0.4152	0.3712	0.3731	0.4498
	Task 8	0.4390	0.4256	0.4318	0.4435
	Average	0.4437	0.4176	0.4256	0.4703
ASM2TV	Task 1	0.4993	0.4617	0.4812	0.5101
	Task 2	0.5106	0.4694	0.4719	0.5017
	Task 3	0.4937	0.4414	0.4501	0.5072
	Task 4	0.5075	0.4673	0.4797	0.5017
	Task 5	0.5161	0.4979	0.4898	0.5129
	Task 6	0.4910	0.4713	0.4687	0.5007
	Task 7	0.4637	0.4281	0.4196	0.4967
	Task 8	0.4790	0.4443	0.4506	0.4845
	Average	0.4951	0.4602	0.4640	0.5019
IADM	Task 1	0.6635	0.6108	0.6246	0.7015
	Task 2	0.6946	0.6317	0.6291	0.7016
	Task 3	0.6038	0.6046	0.6182	0.6892
	Task 4	0.6014	0.6252	0.6351	0.6139
	Task 5	0.6922	0.6314	0.6467	0.7086
	Task 6	0.6933	0.6741	0.6406	0.6120
	Task 7	0.6725	0.6009	0.6012	0.6702
	Task 8	0.6253	0.6052	0.6111	0.6811
	Average	0.6558	0.6230	0.6258	0.6723
MTIS	Task 1	0.7796	0.7315	0.7512	0.8015
	Task 2	0.7401	0.7266	0.7187	0.7804
	Task 3	0.7193	0.6843	0.6801	0.7417
	Task 4	0.7341	0.7014	0.7020	0.7582
	Task 5	0.7926	**0.7414**	0.7307	0.8034
	Task 6	0.7613	0.7134	0.7306	0.7819
	Task 7	0.7051	0.6950	0.6705	0.7221
	Task 8	0.7112	0.7063	0.6881	0.7312
	Average	0.7429	0.7125	0.7090	0.765
MVIS	Task 1	0.7501	0.7226	0.7403	0.7901
	Task 2	0.7226	0.7015	0.7163	0.7453
	Task 3	0.6880	0.6461	0.6218	0.7011
	Task 4	0.7116	0.6416	0.6198	0.7031
	Task 5	0.7413	0.7210	0.7114	0.7506
	Task 6	0.7211	0.7198	0.6912	0.7114
	Task 7	0.6584	0.6424	0.6283	0.6827
	Task 8	0.6543	0.6279	0.6330	0.7107
	Average	0.7059	0.6779	0.6703	0.7244

(To be continued)

Table 4 Experimental results on the dataset GLEAM[21] are shown in bold for the best model performance and underlined for the second-best results. MTMVIS generally achieves the best performance on all valuation metrics for all presented tasks. (Continued)

Method	Task	Metric			
		Accuracy	F1-score	Precision	AUC
MTMVIS	Task 1	**0.8012**	**0.7681**	**0.7590**	**0.8423**
	Task 2	**0.7993**	**0.7902**	**0.7825**	**0.8304**
	Task 3	**0.7501**	**0.7406**	**0.7669**	**0.7945**
	Task 4	**0.7885**	**0.7616**	**0.7641**	**0.8125**
	Task 5	**0.8199**	0.7383	**0.7873**	**0.8353**
	Task 6	**0.8016**	**0.7921**	**0.7998**	**0.8405**
	Task 7	**0.7533**	**0.7423**	**0.7413**	**0.8047**
	Task 8	**0.7729**	**0.7988**	**0.7677**	**0.7892**
	Average	**0.7859**	**0.7665**	**0.7711**	**0.8187**

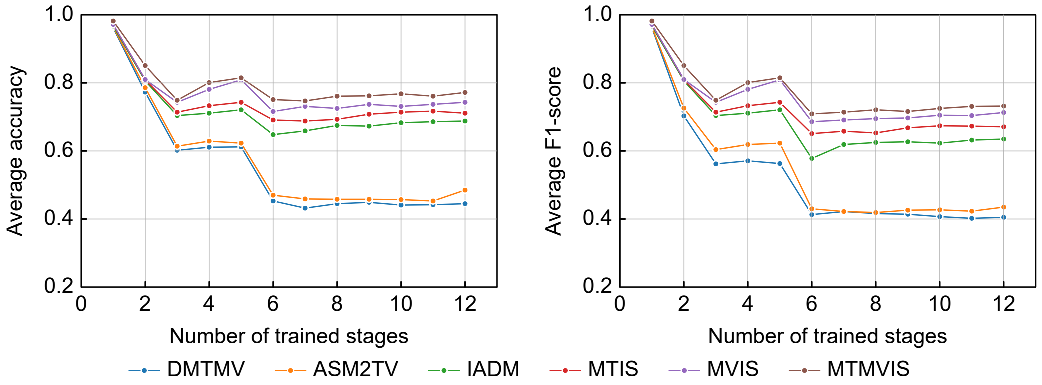


Fig. 5 Experimental results with metrics accuracy and F1-score on dataset GLEAM.

Table 5 Ablation studies on two datasets. The overall percentage improvement via all quantitative metric compared to IADM is presented.

(%)

Dataset	Method	Result			
		ΔAccuracy	ΔF1-score	ΔPrecision	ΔAUC
RealWorld-HAR	MTIS	14%	21%	20%	18%
	MVIS	8%	13%	11%	13%
	MTMVS	20%	23%	24%	23%
	MTMVIS	**23%**	**27%**	**27%**	**28%**
GLEAM	MTIS	13%	19%	13%	14%
	MVIS	7%	8%	7%	7%
	MTMVS	15%	16%	17%	16%
	MTMVIS	**19%**	**23%**	**23%**	**21%**

4% significant improvements on all metrics on two datasets. This finding shows that negative feedback is often unavoidable when only one expert network is shared.

- MTMVS has the same structure as MTMVIS, except that it only uses the output of the last layer as the final output. Moreover, the performance of MTMVS is substantially close to the proposed MTMVIS, with 21% and 11% significant improvements on all metrics on the two datasets. This finding indicates that the combination of the outputs of different layers as the final output performs better than only the output of the last layer as the final output.

- Overall, each component contributes to the performance achieved by the final model.

4.4.3 Parameter analysis

Further experiments are conducted considering the parameters presented below to explore the effect of different learning parameters on the effectiveness and efficiency of the proposed MTMIS.

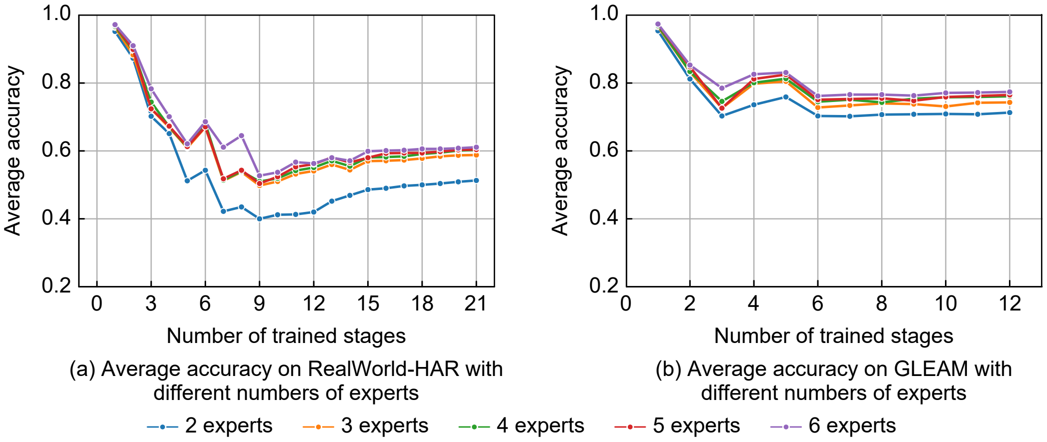
(1) Number of experts: The influence of the number of experts on the model performance (Fig. 6) is investigated on two datasets, namely RealWorld-HAR[20] and GLEAM[21]. The best model performance is observed on both datasets when the number of experts is 5. The model performance does not improve significantly, and the curve tends to be flat as the number of experts increases. The model performance on datasets slightly improves as the number of experts increases; thus, no significant difference is observed under statistical testing. With this exciting discovery, the model can achieve the best performance with the least number of experts, which will help reduce the number of network parameters, increase the convergence speed, and reduce memory consumption.

(2) Attention weight: As depicted in Fig. 7, it shows the attention weight matrix for the output layer on two datasets. The results show that the largest weights are in the shallow layers in the initial stage. Deep layers obtain additional weights as the stages change. Thus, the weight evolution indicates that MTMVIS can perform model selection. Moreover, different stages have various attention weight matrices, indicating that MTMVIS learns additional discriminative features in task-view sharing layer and view fusion layers. Thus, multi-task multi-view learning can play a significant role in incremental learning to solve catastrophic forgetting.

In addition, the adaptive multi-task multi-view incremental learning framework can reflect excellent effects on highly complex network structures. For traditional models, if the network structure is too complex, then the overall convergence speed of the model will be markedly reduced, and the natural performance will also decrease. The proposed model can adaptively make the suitable choice along with the training process.

5 Conclusion

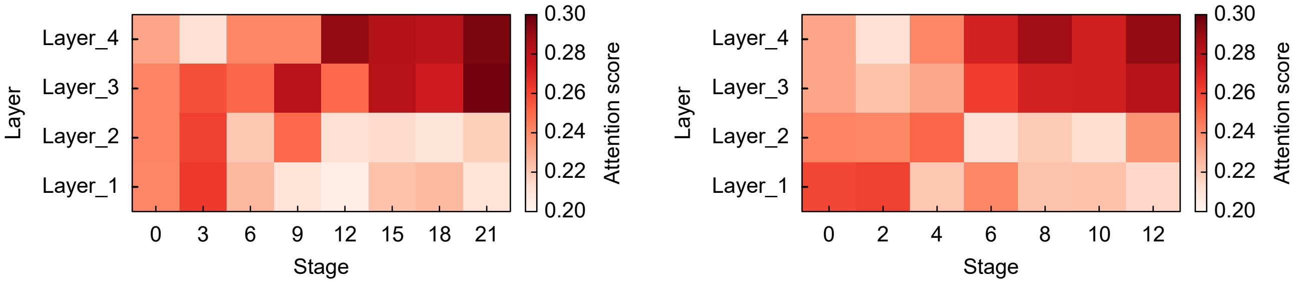
An adaptive multi-task multi-view incremental learning framework for data stream classification, namely MTMVIS, is proposed in the current work to address



(a) Average accuracy on RealWorld-HAR with different numbers of experts

(b) Average accuracy on GLEAM with different numbers of experts

Fig. 6 Model performance with different numbers of experts.



(a) Attention weight matrix of RealWorld-HAR[20]

(b) Attention weight matrix of GLEAM[21]

Fig. 7 Attention weight matrix of the output layer with the change in stage on two datasets.

the catastrophic forgetting problem in incremental learning. MTMVIS heuristically uses a hierarchical attention mechanism to align data collected by different sensors in different views. In addition, MTMVIS utilizes adaptive Fisher regularization from the perspective of multi-task multi-view learning to preserve knowledge in previous stages. The experimental results demonstrate its superiority, achieving better model performance than other advanced models on two public datasets. The proposed MTMVIS has achieved several exciting results, which can effectively alleviate the catastrophic forgetting problem in incremental learning. Numerous unlabeled samples exist in real-world scenarios. Thus, future work aims to extend the proposed model using semi-supervised learning to improve its overall performance.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (No. 2022YFF0712100), the National Natural Science Foundation of China (Nos. 62202273 and 62072279), the National Science Fund for Excellent Young Scholars of China (No. 62122042), the Shandong Provincial Natural Science Foundation of China (No. ZR2021QF044), and the Fundamental Research Funds for the Central Universities.

References

[1] X. Zhang, Q. Wang, Z. Ye, H. Ying, and D. Yu, Federated representation learning with data heterogeneity for human mobility prediction, *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 6, pp. 6111–6122, 2023.

[2] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, Learning graph structures with transformer for multivariate time-series anomaly detection in IoT, *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9179–9189, 2021.

[3] R. S. Peres, A. D. Rocha, P. Leitao, and J. Barata, IDARTS–towards intelligent data analysis and real-time supervision for industry 4.0, *Comput. Ind.*, vol. 101, pp. 138–146, 2018.

[4] W. C. Abraham and A. Robins, Memory retention – the synaptic stability versus plasticity dilemma, *Trends Neurosci.*, vol. 28, no. 2, pp. 73–78, 2005.

[5] B. Goodrich and I. Arel, Unsupervised neuron selection for mitigating catastrophic forgetting in neural networks, in *Proc. 2014 IEEE 57th Int. Midwest Symp. on Circuits and Systems* (*MWSCAS*), College Station, TX, USA, 2014, pp. 997–1000.

[6] Y. Yang, Z. Q. Sun, H. Zhu, Y. Fu, Y. Zhou, H. Xiong, and J. Yang, Learning adaptive embedding considering incremental class, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2736–2749, 2023.

[7] H. Yu, Z. Chen, X. Zhang, X. Chen, F. Zhuang, H. Xiong, and X. Cheng, FedHAR: Semi-supervised online learning for personalized federated human activity recognition, *IEEE Trans. Mob. Comput.*, vol. 22, no. 6, pp. 3318–3332, 2023.

[8] J. He and R. Lawrence, A graph-based framework for multi-task multi-view learning, in *Proc. 28th Int. Conf. Int. Conf. Machine Learning*, Bellevue, WA, USA, 2011, pp. 25–32.

[9] J. Baxter, A Bayesian/information theoretic model of learning to learn via multiple task sampling, *Mach. Learn.*, vol. 28, no. 1, pp. 7–39, 1997.

[10] S. Ruder, An overview of multi-task learning in deep neural networks, arXiv preprint arXiv: 1706.05098, 2017.

[11] Y. F. Wu, D. C. Zhan, and Y. Jiang, DMTMV: A unified learning framework for deep multi-task multi-view learning, in *Proc. 2018 IEEE Int. Conf. Big Knowledge* (*ICBK*), Singapore, 2018, pp. 49–56.

[12] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, *Proc. Natl. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

[13] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, DeepSense: A unified deep learning framework for time-series mobile sensing data processing, in *Proc. 26th Int. Conf. World Wide Web*, Perth, Australia, 2017, pp. 351–360.

[14] P. Siirtola, H. Koskimäki, and J. Röning, Personalizing human activity recognition models using incremental learning, in *Proc. 26th European Symp. on Artificial Neural Networks*, Bruges, Belgium, 2019, pp. 627–632.

[15] K. Demertzis, L. Iliadis, and V. D. Anezakis, MOLESTRA: A multi-task learning approach for real-time big data analytics, in *Proc. 2018 Innovations in Intelligent Systems and Applications* (*INISTA*), Thessaloniki, Greece, 2018, pp. 1–8.

[16] S. Li, Y. Li, and Y. Fu, Multi-view time series classification: A discriminative bilinear projection approach, in *Proc. 25th ACM Int. Conf. Information and Knowledge Management*, Indianapolis, IN, USA, 2016, pp. 989–998.

[17] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, Modeling task relationships in multi-task learning with multi-gate mixture-of-experts, in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery* & *Data Mining*, London, UK, 2018, pp. 1930–1939.

[18] Y. Yang, D. W. Zhou, D. C. Zhan, H. Xiong, and Y. Jiang, Adaptive deep models for incremental learning: Considering capacity scalability and sustainability, in *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 2019, pp. 74–82.

[19] A. Cipolla, Y. Gal, and A. Kendall, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in *Proc. 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 7482–7491.

[20] T. Sztyler and H. Stuckenschmidt, On-body localization of wearable devices: An investigation of position-aware

activity recognition, in *Proc. 2016 IEEE Int. Conf. Pervasive Computing and Communications* (*PerCom*), Sydney, Australia, 2016, pp. 1–9.

[21] S. A. Rahman, C. Merck, Y. Huang, and S. Kleinberg, Unintrusive eating recognition using google glass, in *Proc. 2015 9th Int. Conf. Pervasive Computing Technologies for Healthcare* (*PervasiveHealth*), Istanbul, Turkey, 2015, pp. 108–111.

[22] S. Akaho, A kernel method for canonical correlation analysis, arXiv preprint arXiv: cs/0609071, 2006.

[23] R. K. Ando and T. Zhang, A framework for learning predictive structures from multiple tasks and unlabeled data, *J. Mach. Learn. Res.*, vol. 6, no. 61, pp. 1817–1853, 2005.

[24] A. Argyriou, T. Evgeniou, and M. Pontil, Convex multi-task feature learning, *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.

[25] A. Argyriou, T. Evgeniou, and M. Pontil, Multi-task feature learning, in *Proc. Int. Conf. Neural Information Processing Systems 19*, Vancouver, Canada, 2006, pp. 41–48, 2007.

[26] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, Adaptive mixtures of local experts, *Neural Comput.*, vol. 3, no. 1, pp. 79–87, 1991.

[27] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet, A multi-task deep learning architecture for maritime surveillance using AIS data streams, in *Proc. 2018 IEEE 5th Int. Conf. Data Science and Advanced Analytics* (*DSAA*), Turin, Italy, 2018, pp. 331–340.

[28] Y. Z. Shi, A. D. Li, Z. H. Deng, Q. S. Yan, Q. D. Lou, H. R. Chen, K. S. Choi, and S. T. Wang, Double-coupling learning for multi-task data stream classification, *Inf. Sci.*, vol. 613, pp. 494–506, 2022.

[29] J. Zhang, Y. Zheng, J. Sun, and D. Qi, Flow prediction in spatio-temporal networks based on multitask deep learning, *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, 2020.

[30] Z. Kang, K. Grauman, and F. Sha, Learning with whom to share in multi-task feature learning, in *Proc. 28th Int. Conf. Int. Conf. Machine Learning*, Bellevue, WA, USA, 2011, pp. 521–528.

[31] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, Multi-view clustering via canonical correlation analysis, in *Proc. 26th Annu. Int. Conf. Machine Learning*, Montreal, Canada, 2009, pp. 129–136.

[32] S. Bickel and T. Scheffer, Multi-view clustering, in *Proc. 4th IEEE Int. Conf. Data Mining*, Brighton, UK, 2004, pp. 19–26.

[33] Y. Wang, L. Wu, X. M. Lin, and J. B. Gao, Multiview spectral clustering via structured low-rank matrix factorization, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4833–4843, 2018.

[34] Y. Wang and L. Wu, Beyond low-rank representations: Orthogonal clustering basis reconstruction with optimized graph structure for multi-view spectral clustering, *Neural Netw.*, vol. 103, pp. 1–8, 2018.

[35] L. Wu and Y. Wang, Robust hashing for multi-view data: Jointly learning low-rank kernelized similarity consensus and hash functions, *Image Vis. Comput.*, vol. 57, pp. 58–66, 2017.

[36] A. Blum and T. Mitchell, Combining labeled and unlabeled data with co-training, in *Proc. 11th Annu. Conf. Computational Learning Theory*, Madison, WI, USA, 1998, pp. 92–100.

[37] V. Sindhwani, P. Niyogi, and M. Belkin, A co-regularization approach to semi-supervised learning with multiple views, presented at the 22nd Workshop on Learning with Multiple Views, Bonn, Germany, 2005.

[38] H. Hotelling, Relations between two sets of variates, in *Breakthroughs in Statistics*, S. Kotz, N. L. Johnson, eds. New York, NY, USA: Springer, 1992, pp. 162–190.

[39] S. M. Kakade and D. P. Foster, Multi-view regression via canonical correlation analysis, in *Proc. 20th Annu. Conf. Learning Theory*, San Diego, CA, USA, 2007, pp. 82–96.

[40] V. M. Devagiri, V. Boeva, and E. Tsiporkova, Split-merge evolutionary clustering for multi-view streaming data, *Procedia Comput. Sci.*, vol. 176, pp. 460–469, 2020.

[41] J. Zhang and J. Huan, Inductive multi-task learning with multiple view data, in *Proc. 18th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Beijing, China, 2012, pp. 543–551.

[42] X. Jin, F. Zhuang, S. Wang, Q. He, and Z. Shi, Shared structure learning for multiple tasks with multiple views, in *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases*, Prague, Czech Republic, 2013, pp. 353–368.

[43] X. Jin, F. Zhuang, H. Xiong, C. Du, P. Luo, and Q. He, Multi-task multi-view learning for heterogeneous tasks, in *Proc. 23rd ACM Int. Conf. Conf. Information and Knowledge Management*, Shanghai, China, 2014, pp. 441–450.

[44] C. T. Lu, L. He, W. Shao, B. Cao, and P. S. Yu, Multilinear factorization machines for multi-task multi-view learning, in *Proc. 10th ACM Int. Conf. Web Search and Data Mining*, Cambridge, UK, 2017, pp. 701–709.

[45] G. Cauwenberghs and T. Poggio, Incremental and decremental support vector machine learning, in *Proc. 13th Int. Conf. Neural Information Processing Systems*, Denver, CO, USA, 2001, pp. 409–415.

[46] P. Laskov, C. Gehl, S. Krüger, and K. R. Müller, Incremental support vector learning: Analysis, implementation and applications, *J. Mach. Learn. Res.*, vol. 7, no. 69, pp. 1909–1936, 2006.

[47] G. Zhou, K. Sohn, and H. Lee, Online incremental feature learning with denoising autoencoders, in *Proc. 15th Int. Conf. Artificial Intelligence and Statistics*, La Palma, Spain, 2012, pp. 1453–1461.

[48] Z. Li and D. Hoiem, Learning without forgetting, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, 2018.

[49] S. A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, iCaRL: Incremental classifier and representation learning, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 2001–2010.

[50] G. Ditzler and R. Polikar, Incremental learning of concept drift from streaming imbalanced data, *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2283–2301, 2013.

[51] P. Domingos and G. Hulten, Mining high-speed data streams, in *Proc. 6th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Boston, MA, USA, 2000, pp. 71–80.

[52] R. Pecori, P. Ducange, and F. Marcelloni, Incremental learning of fuzzy decision trees for streaming data classification, presented at the 11th Conf. European Society for Fuzzy Logic and Technology, Prague, Czech Republic, 2019.

[53] Y. Li, Y. Wang, Q. Liu, C. Bi, X. Jiang, and S. Sun, Incremental semi-supervised learning on streaming data, *Pattern Recognit.*, vol. 88, pp. 383–396, 2019.

[54] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, A survey on concept drift adaptation, *ACM Comput. Surv.*, vol. 46, no. 4, p. 44, 2014.

[55] R. Ratcliff, Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions, *Psychol. Rev.*, vol. 97, no. 2, pp. 285–308, 1990.

[56] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr, Riemannian walk for incremental learning: Understanding forgetting and intransigence, in *Proc. 15th European Conf. Computer Vision*, Munich, Germany, 2018, pp. 532–547.

[57] Z. Chen, X. Zhang, and X. Cheng, ASM2TV: An adaptive semi-supervised multi-task multi-view learning framework, arXiv preprint arXiv: 2105.08643, 2021.

Jun Wang received the BEng degree from Shandong Normal University, China in 2004, and the MEng degree from Shandong University, China in 2006. She is currently a PhD candidate at Shandong University, China. Her main research interests include data mining and big data analysis.

Maiwang Shi received the BEng degree from Shandong University, China in 2019, and the MEng degree from Shandong University, China in 2022. He is currently a master student at Shandong University, China. His main research interests are multitasking and multi-view learning.

Xiao Zhang received the BEng degree from Central South University, China in 2013, and the PhD degree from Nanjing University, China in 2019. He is now an assistant professor at School of Computer Science and Technology, Shandong University, China. His research interests include data mining, intelligent sensing, multi-task learning, and federated learning.

Dongxiao Yu received the BEng degree from Shandong University, China in 2006, and the PhD degree from University of Hong Kong, China in 2014. He became an associate professor at School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2016. He is currently a professor at School of Computer Science and Technology, Shandong University, China. His research interests include edge intelligence, distributed computing, and data mining.

Yan Li is an undergraduate student in computer science and technology at Shandong University, China. His main research interests are machine learning and data mining.

Yunsheng Yuan is an undergraduate student in computer science and technology at Shandong University, China. His main research interests are data mining and federated learning.

Chenglei Yang received the BEng, MEng and PhD degrees from Shandong University, China in 1995, 1998, and 2004, respectively. He is now a professor at School of Software, Shandong University, China. His main research interests are human-computer interaction, virtual reality, and computational geometry.