The Human Continuity Activity Semi-Supervised Recognizing Model for Multi-View IoT Network

Ruiwen Yuan, JunPing Wang

***Abstract*—With advances in spatial-temporal internet of things (IoT) technologies, human activity recognition (HAR) has played a major role in human safety and medical health. Recently, most works focus on activity deep feature extraction, offering promising alternatives to manually engineered features. However, how to extract the effective and distinguishable continuity activity features and meanwhile reduce the heavy dependence on labels still remains the key challenge for human activity recognition. This paper proposes the human continuity activity semi-supervised recognizing method in multi-view IoT network scenarios. Our innovation combines supervised activity feature extraction with unsupervised encoder-decoder modules, which can capture continuity activity features from sensor data streams. To be more specific, our work applies convolutional neural network (CNN) to capture the local dependence of sensor data and designs an encoder-decoder architecture to extract temporal features in an unsupervised manner. Then we fuse these two features to recognize activities and train them with manual labels, thereby refining the temporal feature extraction and training CNN module. Experiments on five public datasets demonstrate the exceptional performance of our proposed method, which can achieve a higher recognition accuracy on almost all the datasets and is more robust and adaptive among different datasets.**

***Index Terms*—Human activity recognition, Deep learning, Activity feature extraction, Semi-supervised learning**

I. Introduction

WITH the development of the multi-view sensor network, 5G, and battery-powered network technologies, the future Internet of Things (IoT) will aggressively extend its coverage by integrating communications in different spatial domains to form the space, air, ground, and ocean integrated network (SAGOI-Net). Human activity recognition (HAR) [1] is an increasingly important application model for the SAGOI-Net, such as human safety [2, 3], industrial security [4, 5], and medical health [6–8]. The native of HAR records people's movement behaviors with data that allows computing systems to monitor, analyze, and assist their daily life in the SAGOI-Net, after which signal processing and machine learning techniques are applied to automatically recognize the activities [9]. Human activity recognition has been approached in different ways: video-based HAR [10] and sensor-based HAR [11, 12]. The video-based HAR uses cameras to take images or videos to recognize people's behaviors [10]. In a multi-view IoT network environment where multiple types of devices and sensors are used to record human activities, the sensor-based HAR utilizes on-body or ambient sensors to dead reckon people's motion details or log their activity tracks. However, video-based HAR always involves human privacy problems and suffers from some conditions such as low illumination and resolution factors [10]. In contrast, with the fast development of smart devices and the Internet of Things, multiple smart devices and sensors in SAGOI-Net can log human activity information conveniently and non-intrusively [13], which further enhances or enriches the understanding of human continuity activity recognition.

The HAR task assumes that the sensor's signal pattern contains important features of human activities. Therefore, activity feature extraction is crucial for sensor-based HAR in the SAGOI-Net and remains a key challenge [14]. Most existing methods are typically supervised models in the SAGOI-Net, which require the large-scale labeling of what activities have been performed after the data collection is completed [15]. However, these methods will not work on some annotation-scarce tasks. Therefore, to reduce the dependence on manual labels, current researches in the recognition of human activities have significantly improved deep learning techniques, including a semi-supervised deep learning approach [16], and self-supervised contrastive predictive coding [17]. Zhu et al. have proposed a semi-supervised deep learning approach, using temporal Deep Long Short-Term Memory (DLSTM) [16], to recognize human activities with smartphone inertial sensors. Harish et al. proposed contrastive predictive coding (CPC) [17]. It uses an autoregressive model to predict future timesteps of sensor data via a context vector derived from past data. Through this pre-training process, the model can extract features from the temporal structure of sensor data streams in a self-supervised manner. The pre-trained representations can be integrated into standard activity chains to complete classification in the fine-tuning stage with a small amount of labeled data.

The CPC is a powerful self-supervised framework for sensor-based HAR in the SAGOI-Net. The framework takes the best advantage of the effective use of small amounts of labeled data and the exploitation of unlabeled data collected in mobile and ubiquitous scenarios. The pre-trained representations encode the high-level semantic feature information between temporally separated parts of the time-series sensor data [17], thus leading to improved recognition performance. This approach has been successful and applied in human mobile activity recognition [17]. Unfortunately, applying the CPC to train the human continuity activity model has two

R. Yuan and J. Wang are with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China, and School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 101408, China. Email: yuanruiwen2021@ia.ac.cn; junping.wang@ia.ac.cn

problems in multi-view IoT networks. On the one hand, the CPC method uses the past data in a window to predict future data during training, but directly inputs all the data in a window during the testing process, resulting in a discrepancy between the lengths of the training data and the test data. This discrepancy will make the pre-trained model unable to effectively extract the temporal structural features of the test data because of the different distribution of training and test data, thereby reducing the accuracy of the model. On the other hand, the CPC method uses a fully self-supervised pre-training method to learn feature extraction. Even if labeled data is used in the fine-tuning task, the parameters of the pre-trained feature extraction module are frozen and cannot be refined with the labels to learn direct semantic information. Therefore, when the temporal features of the data are not obvious, the method is prone to overfitting in the training process, making the feature extraction less robust and difficult to use directly in some complex scenarios.

Based on this observation, this paper proposes the human continuity activity semi-supervised recognizing method in multi-view IoT network scenarios, considering the effective use of labeled data and feature extraction of unlabeled data. Our work focuses on two goals: 1) improving the accuracy of activity recognition, and 2) facilitating the robustness and universality of the method. More specifically, our model uses CNN to extract the local temporal features of the data and creates the unsupervised LSTM encoder-decoder module to extract the unsupervised global temporal features without manual labels. The model combines the local features with unsupervised global temporal features and then feeds them into the classifier. By training them using the activity labels, the CNN module is effectively trained with labels. The temporal features extracted by the unsupervised LSTM auto-encoder will be refined to integrate semantic information from labels, thus enriching the temporal features. Two advantages of our model alleviate the problems of CPC: firstly, our LSTM encoder-decoder architecture leverages the input data itself as a supervision signal to restore it from temporal features. Therefore, train data and test data have the same distribution so that the trained model can effectively work on test data. Secondly, we used the manual labels to further enhance the unsupervised temporal features. When faced with more complex scenarios, the temporal features will be more robust, thereby leading to high accuracy in different multi-view IoT network scenarios.

The main contributions of this paper can be summarized as follows:

1) This paper proposes a novel semi-supervised deep learning method that combines the local temporal feature extraction module using CNNs with a global temporal feature extraction module using encoder-decoder architecture. The model combines features from these two parts to enrich the extracted activity features while reducing the heavy reliance on manual labels.

2) The unsupervised encoder-decoder architecture is trained in a novel unsupervised manner which avoids the discrepancy between train and test data. Then we refine it with activity labels. After refining, the extracted temporal features will contain more semantic features from prior supervision labels, thus making feature extraction more robust even faced with some sensor data with indistinct temporal characteristics.

3) We compare the performance of our proposed model with the state-of-the-art methods on five public datasets, and the experimental results show that our proposed model remains higher recognition accuracy and more robust performance on multiple datasets from different IoT scenarios. Moreover, our model improves the performance of some activities that are difficult to distinguish.

The remainder of this paper is as follows: Section II provides a brief review of the most related work. Section III describes our proposed human continuity activity semi-supervised recognizing method in multi-view IoT network scenarios. Experimental analysis and completion results are shown in Section IV to verify our method. Finally, we conclude the proposed method in Section V.

II. Related work

Human activity recognizing methods have been extensively developing in recent years. We roughly divide them into three categories in accordance with their feature extraction method: 1) semi-supervised activity feature extraction, 2) supervised deep learning-based activity feature extraction and 3) manual feature extraction.

A. Semi-supervised Deep Learning based Feature Extraction

To reduce the reliance on manual labels, many researchers have introduced semi-supervised activity feature extraction into human activity recognition [15, 17, 18]. Harish et al. [17] focus on the effective use of small amounts of labeled data and the opportunistic exploitation of unlabeled data. The proposed model: CPC [17] is pre-trained with unlabeled data by self-supervised training and then fine-tuned with labeled data. Zhu et al. propose a semi-supervised deep learning framework [15]: to better solve the problem of the inadequately labeled sample, an intelligent auto-labeling scheme is developed, which efficiently uses and analyzes the weakly labeled sensor data to train the classifier model. zhu et al. propose a semi-supervised deep learning approach, using a temporal ensemble of Deep Long Short-Term Memory (DLSTM) [16] to recognize human activities with smartphone inertial sensors. By combining the supervised and unsupervised losses together, it made good use of the unlabeled data. Saeed et al. proposed a self-supervised approach based on wavelet transform [19] to learn useful representations from unlabeled sensor inputs. In our proposed model, we use a novel encoder-decoder architecture to extract global temporal features in an unsupervised way and use CNN to extract local temporal features, then we fuse these two features and train them with activity labels to refine the LSTM network and effectively train the CNN module. Through this process, the extracted features will be more robust and contain richer semantic information of activity, thus leading to better performance.

B. Supervised Deep-Learning-based Activity Feature Extraction

In contrast, with the fast development of deep neural networks in recent years, more and more researches used supervised deep learning models [20–23] to automatically extract hidden and abstract activity features from raw data without relying on domain knowledge, which are more robust and generalized.

Due to the ability to effectively extract local dependence in the sub-regions of the data, CNN has been widely used for HAR [24–27]. Yang et al. apply CNNs for feature extraction [24]. Ignatov et al. [25] use CNN for local feature extraction and supplement it with global statistical features that preserve the global properties of time series sensor data. The model is evaluated on two public datasets and achieves better results. Zhen et al. [28] encode the time series sensor data as images and use computer vision techniques: fusion ResNet for image recognition.

In addition, because the time series sensor data may contain the temporal characteristics of the activity, there are also some studies using Recurrent Neural Networks (RNNs) with long short-term memory [29] to extract the temporal features of original data. Guan et al. use hierarchical deep LSTM architecture [30] to extract temporal features. Zhao et al. propose the Deep Residual Bidir-LSTM model [31] which sets up a bidirectional connection and residual connections to improve the extracted temporal features and gets satisfactory results. Hammerla et al. [20] compare all kinds of deep learning algorithms including DNNs, CNNs, and various types of LSTMs. Furthermore, because Transformers is shown to outperform these architectures for sequence analysis tasks, Yoli et al. present an activity recognition model based on Transformers [32] to extract temporal features.

What is more, some recent researches [33–36] combine CNN with LSTM units to extract features simultaneously because they extract features from different aspects. Ordonez et al. propose the Deep ConvLSTM [33] algorithm composed of convolutional and LSTM recurrent layers, which can automatically model the temporary dependency of sensor data. Chen, L et al. [34] introduce a deep learning model which also combines CNNs and LSTM. Ihianle et al. use a combination of convolutional neural network (CNN) and Bidirectional long short-term memory (BLSTM) [37] to automatically extract features. Mohamed et al. [38] propose a supervised dual-channel model that combines LSTM and CNN, followed by an attention mechanism for the temporal fusion of sensor data.

However, the supervised deep-learning-based activity feature extraction heavily depends on the manual labels, which will not work well in some label-scarce scenarios.

C. Manually Engineered Activity Feature Extraction

Many traditional machine learning algorithms use manually designed features [7, 28], which represent various aspects of information hidden in the original data. The extracted features always involve the statistical features of the data segment such as mean, variance, root mean square, and spectral entropy [37]. And then the features will be fed into traditional machine learning classifiers including support vector machine (SVM) [39], random forest (RF) [40], and K-nearest neighbor (KNN) [41] to recognize the human activity. However, the features are generally redundant or too large. To facilitate more accurate and faster learning, Principal Component Analysis (PCA) [42], Linear Discriminant Analysis (LDA) [43], or other methods are required for feature selection and feature dimensionality reduction. Through the feature selection process, the machine learning methods have improved their generalization and interpretability and get satisfactory results.

However, these methods of manual activity extracting often rely on domain knowledge to design specific features for specific scenarios, which are very labor-intensive and difficult to characterize complex activities.

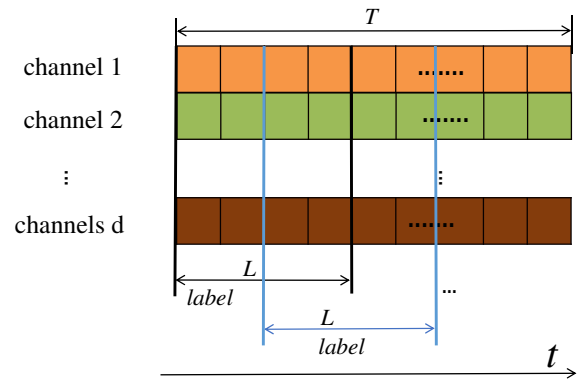

Fig. 1: Human continuity activity recognizing problem setting for multi-view IoT. Different color in every row denotes different channels of sensors and the overall number of channels is d. The horizontal direction represents the time axis. The black and blue vertical lines represent two sliding windows of length L respectively, which are allowed to overlap. They are used to segment the sensor data in the time range $[0, T]$ into different samples $\boldsymbol{X}_i$ and each sample is labeled as $\boldsymbol{y}_i, (i = 1, 2..., N)$.

III. Human Continuity Activity Semi-supervised Recognizing Model

A. Problem Formulation

In multi-view IoT network scenarios, each human activity is recorded by multi-channel sensing devices. In the problem setting of human continuity activity recognizing in Fig 1, the sensor-based human activity features in the time range $[0, T]$ are often cut into N segments by a sliding window of a certain length L at certain time token t_i [44], and each segment is regarded as a sample. Each sample is assigned an activity label or is sometimes unlabeled. We denote the sample as $\{\boldsymbol{X}_i, \boldsymbol{y}_i\}, (i = 1, 2..., N)$, and N is the number of the segmented samples. The unlabeled activity features $\boldsymbol{X}_i$ is denoted by a $d \times L$ matrix, which means the overall number of sensor channels is d and L is the number of timestamps in a window. The one-hot activity label of the sample is denoted by the $\boldsymbol{y}_i$. Our task is to predict $\boldsymbol{y}_i$ of each sample according to the extracted features of $\boldsymbol{X}_i$ from multi-channel sensor devices.

Because different participants will perform varying patterns even at the same activity, the key part of the activity recogni-

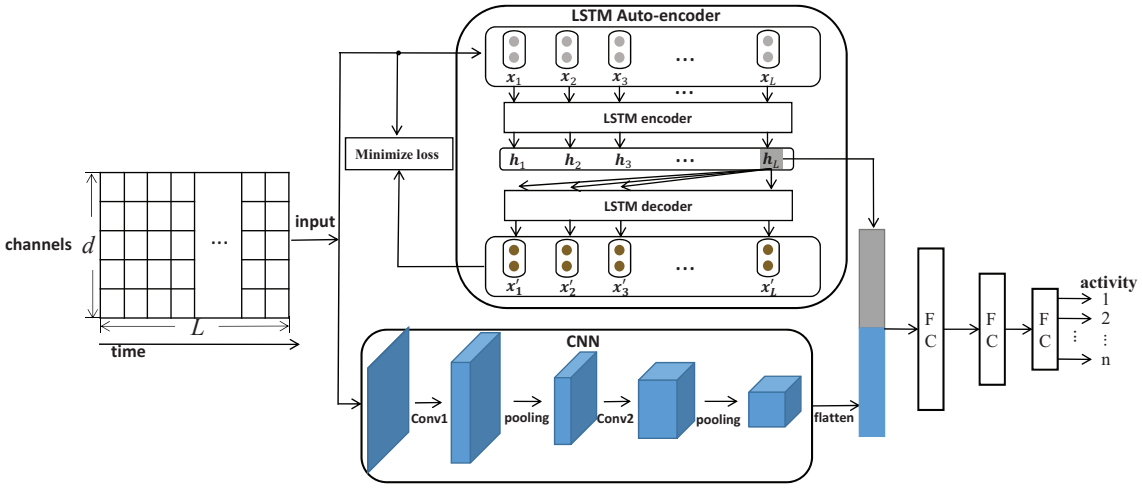

Fig. 2: The overall semi-supervised HAR architecture is shown above. The input data is a sensor data sample as shown in Fig1. An LSTM encoder-decoder module is proposed to extract temporal features and a CNN module is proposed to extract local features from the data sample. Then the features from these two modules marked glue and gray are fed into the classifier for human activity classification. The encoder-decoder is first trained in an unsupervised way through reconstruction loss to extract temporal features from data patterns, and then activity labels as the prior knowledge are used to train CNN modules and refine the unsupervised encoder-decoder module with the classification loss.

TABLE I: Notations and descriptions used in this paper.

Notations	Descriptions
$\boldsymbol{X}, \boldsymbol{X}_i$	Multi-channel sensor data sample, the ith sample
L	The number of timestamps in a sliding window
$\boldsymbol{y}, \boldsymbol{y}_i$	The one-hot category label, the label of the ith sample
N	The number of samples
$\boldsymbol{f}_1$	Extracted features by CNN
$\boldsymbol{f}_2$	Extracted features by LSTM encoder
$\boldsymbol{h}_l$	LSTM encoder hidden states at the lth timestamp
$\boldsymbol{x}_l$	The sensor data at the lth timestamp of $\boldsymbol{X}$.
$\boldsymbol{x}_{il}$	The sensor data at the lth timestamp of the ith sample $\boldsymbol{X}$
$\hat{\boldsymbol{X}}$	The reconstructed sensor data
$\boldsymbol{f}_c$	The fused features
C	The number of predefined classes

tion is the general semantic feature extraction of the original sensor data from multiple subjects. In the training process, our proposed model encodes the training data from two different aspects and defined two tasks on top of the encodings. One is to extract temporal features by LSTM encoder and use temporal features to restore the raw data by using an LSTM decoder. The other task is the classification task. The CNN module extracts local features of the data, and it is fused with temporal features extracted by LSTM and fed into the fully-connected network to predict the label of the activity. The reconstruction loss of task one is trained in an unsupervised way and the classification loss of the second task trains the CNN module effectively and extensively refines the LSTM network to make extracted temporal features contain richer semantic information. The overall architecture of our proposed model is shown in Fig 2 and the main notations used in this paper are summarized in Table I.

B. Local Activity Feature Extraction Module

The local temporal features of sensor data will always contain the key patterns that will lead to the specific type of activity. Given the importance of learning the local dependence of the multi-channel sensor data, the module aims to effectively extract local temporal features from the sensor data of different channels. Due to the powerful ability to extract local features, we use CNN in this module to capture the local temporal features of the multi-channel sensor data. The CNN is composed of convolutional layers and pooling layers. The convolutional layer has many neurons with a trainable kernel filter. It extracts local patterns through the convolution of the kernel and the data. The output will be added to the bias and then input into the activation function. The pooling layer can converge the nearby features in the feature map into only one local feature. Through the pooling operation, the dimension of the feature map will be reduced. In the meantime, it increases the invariance of features to enhance them.

The module uses a two-layer 1-D CNN and extracts local features by using the kernel filter running over the sensor data. In this way, the output feature map will represent the useful local temporal features that may contain specific movement patterns. The extracted features can be represented by:

$$\boldsymbol{f}_1 = f_{CNN}(\boldsymbol{W}, \boldsymbol{X}, b) \tag{1}$$

where $\boldsymbol{W}$ denotes the parameters of the networks, and b denotes the bias. To prevent overfitting problems, we introduce the dropout operation after the pooling layer [34]. It can set the output of some neurons to zero with the probability of p_{drop} during the training process. The extracted local temporal features will be fused with the global features extracted by the LSTM-autoencoder module introduced in Section III-C.

C. Unsupervised Encoder-Decoder Module

The unsupervised encoder-decoder module uses an encoder to extract temporal activity features from input sensor data and then uses a decoder to reconstruct the sensor data by leveraging low-dimensional temporal features as input. By minimizing the reconstruction loss, the activity temporal feature extraction module is trained in an unsupervised way. Because LSTM can model sequential data and learn high-level temporal representations [30, 45], our encoder and decoder have been implemented by LSTM. The output of LSTM at each timestamp depends on not only the input but also the hidden layer state at each time.

Given the vector of input sample $\boldsymbol{X}$ at each timestamp $\boldsymbol{x}_l, (l = 1, 2, ...L)$ measured by all sensors, we will get the hidden layer state $\boldsymbol{h}_l = f(\boldsymbol{x}_l, \boldsymbol{h}_{l-1})$ at each timestamp. Finally, the output at the last timestamp $\boldsymbol{h}_L$ can be obtained as the temporal features of the entire sample, and is denoted as:

$$\boldsymbol{f}_2 = \boldsymbol{h}_L = f_{LSTM}(\boldsymbol{X}; \theta) \tag{2}$$

where θ refers to the parameters of the network.

In this paper, we use LSTM as the encoding part of the auto-encoder, which is denoted as $f_e(\cdot)$. The output of the last timestamp $\boldsymbol{h}_L$ is regarded as the hidden temporal features in the auto-encoder and is denoted as $\boldsymbol{h}_L = f_e(\boldsymbol{X})$. Then to ensure that the features $\boldsymbol{f}_2$ extracted by LSTM can reserve more information from the original data, the decoder $f_d(\cdot)$ is also composed of LSTM and is used to restore raw input data by processing the hidden features. We set $\boldsymbol{h}_L$ as the initial hidden state of the decoder. The decoder processes the temporal features extracted by the encoder part so that the output sample $\hat{\boldsymbol{X}} = f_d(f_e(\boldsymbol{X}))$ can correspond to the entire input sample $\boldsymbol{X}$ at each timestamp, through which we can achieve the unsupervised training process.

At the same time, to prevent the occurrence of over-fitting, the encoder and the decoder are each set to a single-layer LSTM. Besides, the model introduces a dropout operation after the LSTM output layer during training. In the training process, the effectiveness of temporal features $\boldsymbol{f}_2$ is improved by continuously reducing the reconstruction loss between the restored data and the input data [46], which is denoted by Eq.(3):

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{L}\sum_{l=1}^{L}\|\boldsymbol{x}_l - \hat{\boldsymbol{x}}_l\|_2^2 \tag{3}$$

where $\boldsymbol{x}_l$ and $\hat{\boldsymbol{x}}_l$ denote the lth timestamp of the original sensor data and reconstructed data respectively.

D. End-to-End Fusion Training Module

There are many ways of feature fusion such as feature-level, decision-level, etc. [47, 48]. In this paper, we simply adopt feature-level fusion [47]. We assume that $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are extracted features from module one and module two respectively. The extracted feature map $\boldsymbol{f}_1$ is a 2-D matrix, while $\boldsymbol{f}_2$ is a 1-D vector, therefore, we must reshape $\boldsymbol{f}_1$ into a 1-D vector before it is fused with $\boldsymbol{f}_2$. After that, we directly concatenate $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ to obtain the fused feature $\boldsymbol{f}_c$, which is denoted by Eq.(4).

$$\boldsymbol{f}_c = [\boldsymbol{f}_1; \boldsymbol{f}_2] \tag{4}$$

Algorithm 1 Training algorithm of our proposed model.

Require: $\{\boldsymbol{X}_{train}, \boldsymbol{y}_{train}\}$
Initialize the parameters of the convolutional neural networks and LSTM auto-encoder f_{CNN}, f_e, f_d, and other hyper-parameters such as epoches, learning rate, and batch size;
While not convergence **do**
  $j = 0$;
  While $j \cdot batchsize \leq N$ **do**
    get batch size data from train set $\boldsymbol{X}_{batch}$;
    Extract local feature $\boldsymbol{f}_1 = f_{CNN}(\boldsymbol{W}, b; \boldsymbol{X}_{batch})$;
    Encode the input data with LSTM $\boldsymbol{f}_2 = f_e(\boldsymbol{X}_{batch})$;
    Decode the hidden feature $\boldsymbol{f}_2$ with another LSTM to reconstruct $\boldsymbol{X}_{batch}$, $\hat{\boldsymbol{X}}_{batch} = f_d(f_e(\boldsymbol{X}_{batch}))$;
    Calculate reconstruction loss $\mathcal{L}_1$ according to (2);
    Fuse the features $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ to get $\boldsymbol{f} = [\boldsymbol{f}_1; \boldsymbol{f}_2]$;
    Feed the fused feature $\boldsymbol{f}$ into the fully-connected network to get the probability distribution of the predefined activities $\hat{p}(\boldsymbol{y}|\boldsymbol{f})$;
    Calculate the cross-entropy loss function $\mathcal{L}_2$;
    Calculate total loss according to (4): $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$;
    Update the parameters of neural networks by applying Adam optimizer;
    $j = j + 1$;
  end While
end While
return the parameters f_{CNN} and f_e.

The fused features $\boldsymbol{f}_c$ combine the prior knowledge of manual labels and the comprehensive temporal features from multi-view sensor data patterns, which can effectively model the feature distribution of each type of activity. Therefore, after the features are fused, we feed the fused feature $\boldsymbol{f}_c$ into the fully-connected network to classify the predefined activities. The number of neurons in the output layer is the same as the number of predefined activities. The output of the last layer is fed into the Softmax activation function to normalize the output and get the probability distribution $\hat{p}(\boldsymbol{y}|\boldsymbol{f})$ of all categories, then we select the activity with the highest probability. We need to compare the predicted activity with the ground truth and use the cross-entropy loss function to calculate the loss function value $\mathcal{L}_2$:

$$\mathcal{L}_2 = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} log(p_{ij}) \tag{5}$$

where y_{ij} means that if the label of the ith sample belongs to the jth class, the value is 1. Otherwise, it is 0. C refers to the number of predefined categories, and p_{ij} represents the ith sample's output probability of the jth category. The overall

TABLE II: The overall statistical information of 5 publicly available datasets.

Name	# SR	# WL	# Subjects	# Channels	# Train	# Test	# Activities
PAMAP2	100Hz	160	9	40	14791	2833	12
UCI-HAR	50Hz	128	30	9	7352	2947	6
UCI-OPPRTUNITY	30Hz	30	4	113	47718	8090	18
WISDM	20Hz	180	29	46	8235	2746	6
Mhealth	50Hz	250	10	23	1817	738	12

loss function is the sum of the cross-entropy loss and MSE loss in the second module, so we obtain the following optimization goal:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_1 + \mathcal{L}_2 \\ &= \frac{1}{N}\sum_{i=1}^{N}(-\sum_{j=1}^{C} y_{ij} log(p_{ij}) + \lambda \frac{1}{L}\sum_{l=1}^{L}\|\boldsymbol{x}_{il} - \hat{\boldsymbol{x}}_{il}\|_2^2)\end{aligned} \tag{6}$$

where λ is the tradeoff parameter. The overall training algorithm is summarized in Algorithm 1.

After training, the CNN module can extract local temporal features effectively. The LSTM encoder can be trained in an unsupervised way and then refined with activity labels, which enriches the temporal features. After combining these two features, the final feature representation will become more effective and distinguishable to better represent the sensor data.

In the test process, we can combine the features extracted from CNN and LSTM encoder to predict the activity label of the unseen data.

IV. Experimental Results and Analysis

In this section, we will present the experimental setup and compare the performance of our proposed model with other state-of-the-art algorithms on five public datasets, the results will be displayed in the last part of this section.

A. Datasets from Multi-View IoT

We have conducted the experiments based on five public datasets from multi-view IoT, which are usually used for training the sensor-based HAR models. A Multi-view IoT network is composed of multiple types of sensors and devices to reflect the changing characteristics of people from different views as they conduct their activities, which is beneficial for accurately analyzing the human activity category. The statistical information is shown in detail in Table II, which contains the sensor sampling rate (#SR), the length of the sliding window (#WL), and so on.

1) *PAMAP2* [49]: This dataset records 12 daily activities collected from 9 elderly participants. Each subject wears a heart rate and three IMUs on their chest, ankle, and wrist respectively. Each IMU contains two 3-axis accelerometers, a 3-axis gyroscope, and a 3-axis magnetometer. The 5-second sliding window with 20% overlap is applied to the data [50].

2) *UCI-HAR* [51]: The UCI-HAR dataset is composed of 6 simple activities collected from 30 subjects aged 19-48. Subjects wear the smartphone with the IMU to measure the data during activities. Then the sensor data is labeled with the video that records the activities of the subjects. The data provider uses a 2.56s window with a 50% overlap to segment the data [52].

3) *Wisdom* [53]: It is created for 6 daily activities including sitting, working, standing, jogging, and so on. Volunteers in this experiment also carry a mobile phone with only one embedded accelerometer during the activities. The 9s window is used for dividing the data into samples [54].

4) *UCI-OPPORTUNITY* [55]: It comprises naturalistic activities collected from four subjects with ambient environment sensors and 19 sensors on the body and is more complex than others. Sliding windows of 1-second duration with 50% overlap are applied to the sensor data [56]. During the recordings, subjects are asked to perform daily activities without many restrictions, they performed 20 repetitions of a predefined set of 17 activities. The dataset has provided two tasks, and we choose task B: recognize 17 different right-arm gestures in a drill session. Considering the NULL class, it is an 18-class classification problem.

5) *MHealth* [57]: It comprises body motion and some signs recordings. 10 volunteers who wear sensors on their right wrist, left ankle, and chest perform 12 physical activities. The data provider has segmented the sensor data with a 5s sliding window [50].

B. Experiment Setup

We follow the experimental setup in [58]. For PAMAP2 dataset, we only focus on 12 protocol activities. The sensor data of subject 5 is used as the test set, and the rest makes up the train set. For the OPPERTUNITY dataset, run 3 and 4 for subject 2 and 3 are regarded as the test set with the rest being the train set. In WISDM, the subject's data are randomly divided into the train set and test set with a ratio of 3:1. As for UCI-HAR and Mhealth datasets, we all use the data from 70% of the participants as the train set, and the rest as the test set.

In our architecture, data from every dataset will be normalized before it's used for training. The dropout rate is set to 0.1.

In the CNN module, due to the importance of the number of convolutional layers, we set it to be {1,2,3,4,5,6,8} and then conduct experiments to choose the best one by evaluating their performance. The result is shown in Fig 3. We can clearly observe that there exist two peaks when the number of layers is set as 2 and 5 and the F1-score of 2 convolutional layers reaches the maximum value. When the number of layers continues to increase, the F1-score shows a downward trend. A possible reason for this phenomenon is that when there is

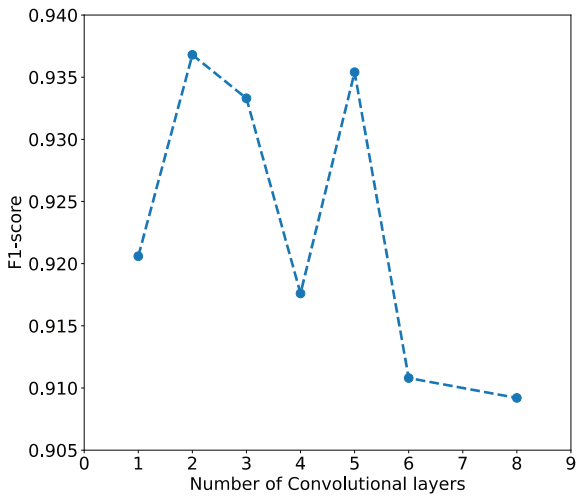

Fig. 3: The influence of the number of convolutional layers.

only one layer, it can not extract enough information; when the number of layers becomes too large, it is too complex for the problem and may lead to overfitting.

The kernel size of the convolutional layer is set to (12,1) and (6,1) respectively, with stride 1 and no zero-padding. Then it is followed by the ReLU activation function and the pooling layer. We adopt the none-overlap max-pooling, so both the pooling size and its stride are set to (2,1). In the LSTM auto-encoder module, there is one hidden layer in the encoder and decoder. As for the dimension of hidden representations of LSTM, because it is important to our model, we set it to be {32,64,128,256,512} and evaluate the F1-score for each of them. The results are shown in Fig4, which indicate that when the dimension of LSTM hidden representations increases, the F1 score rises first and then falls, reaching the maximum value at 128. So we set the hidden dimension as 128.

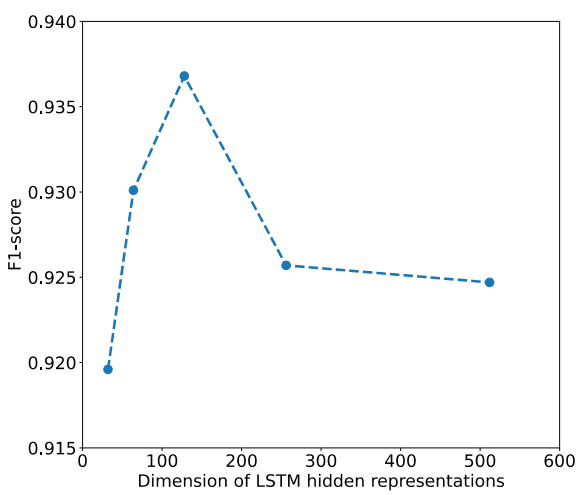

Fig. 4: The influence of the LSTM hidden dimension.

Finally, the tradeoff parameter λ is set to 1. The classifier is composed of 3-layer fully-connected network to obtain the recognition result.

After the overall loss function is calculated, the Adam optimizer is applied for training with the initial learning rate$1e^{-4}$ and weight decay $1e^{-5}$. The batch size is set to 1024 in OPPORTUNITY dataset and 256 in others. The maximum training epoch is 150. Our algorithm is implemented on the Tensorflow 2.0 platform, using two NVIDIA RTX2080 GPUs with 12G memory.

In this paper, we evaluate the classification performance of our proposed model using four metrics: accuracy, precision, recall, and F1-score, which are widely used in measuring the performance of HAR [33, 58, 59]. Accuracy directly indicates the overall performance of the classification for all classes. Moreover, we also need to evaluate the precision and recall of the model, especially on datasets with imbalanced categories. Because F1-score is the harmonic average of the precision and recall of the classification, we also use the weighted F1-score for evaluating the performance of the model. These four metrics are denoted as Eq.(7)(8)(9)(10). When all four metrics get a high value, the model performs well.

$$Acc = \frac{\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} TN_i}{\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} TN_i + \sum_{i=1}^{C} FP_i + \sum_{i=1}^{C} FN_i} \tag{7}$$

$$Pre = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} FP_i} \tag{8}$$

$$Rec = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} TP_i + \sum_{i=1}^{C} FN_i} \tag{9}$$

$$F1 = \frac{(2 * precision * recall)}{(precision + recall)} \tag{10}$$

where C denotes the number of classes, TP_i, TN_i, FN_i, FP_i represent the true positives, true negatives, false positives, and false negatives of class i respectively.

C. Compared Algorithms

For human activity recognition, there exist several methods that use traditional machine learning or deep learning based methods to extract activity features and then recognize the activity. We thus compare our proposed framework with the baselines listed below and conduct the ablation study to validate the effectiveness of each module in our framework.

1) *SVM*[39]: It firstly extracts handcrafted features from the original multi-channel sensor data, and then the extracted features are fed into the SVM classifier to recognize the activity categories.

2) *Random Forest* [40]: Similar to SVM, as a traditional machine learning classifier, it also uses manually extracted features to complete human activity recognition.

3) *Deep ConvLSTM* [33]: A state-of-the-art architecture based on convolutional layers and LSTM recurrent units. There are 4 convolutional layers and 2 LSTM layers to automatically extract features and temporal dependence. The model is trained in a supervised way.

4) *H-LSTM* [30]: The hierarchical deep LSTM architecture with 2 layers of LSTM can automatically extract the temporal feature and use the fully-connected network to output the recognition results by training in a supervised way.

TABLE III: The experimental results of all methods on UCI

	Dataset: UCI			
Method / Metrics	Acc	Pre	Rec	F1
SVM	0.8873	0.8872	0.8873	0.8866
Random Forest	0.8025	0.8039	0.8025	0.8023
Deep ConvLSTM	0.9314	0.9322	0.9324	0.9314
H-LSTM	0.9053	0.9100	0.9053	0.9063
CNN+statistical	**0.9712**	**0.9723**	**0.9713**	**0.9714**
Bidir-residual-LSTM	0.9118	0.9114	0.9118	0.9112
#Distribution-Embedded	0.9053	-	-	0.9058
Transformer Encoder	0.8836	0.8847	0.8836	0.8832
Our proposed model	0.9372	0.9386	0.9372	0.9368

TABLE IV: The experimental results of all methods on Mhealth

	Dataset: Mhealth			
Method / Metrics	Acc	Pre	Rec	F1
SVM	0.8808	0.8905	0.8808	0.8874
Random Forest	0.9051	0.9142	0.9051	0.9042
Deep ConvLSTM	0.8591	0.8553	0.8632	0.8457
H-LSTM	0.8848	0.8945	0.8848	0.8845
CNN+statistical	0.9417	0.9171	0.9417	0.9279
Bidir-residual-LSTM	0.8997	0.8986	0.8947	0.8937
Transformer Encoder	0.9187	0.9216	0.9187	0.9145
Our proposed model	**0.9607**	**0.9625**	**0.9607**	**0.9607**

5) *CNN+statistical features* [25]: This architecture combines local features extracted by convolutional neural networks with manual statistical features that preserve information about the global form of time series.

6) *Bidir-residual-LSTM* [31]: A deep network architecture using residual bidirectional LSTM cells, which builds bidirectional temporal connection and residual connections between stacked cells and trains the architecture with manual labels.

7) *Distribution-Embedded Neural Network* [58]: The semi-supervised state-of-the-art architecture which contains three modules can extract features including statistical features, temporal features, and spatial correlation features for activity recognition in a unified framework.

8) *Transformer Encoder* [32]: The Transformer encoder is applied to extract the global features of the sensor data and recognize the human activity type. The output of the Transformer Encoder at the position of the class token is regarded as the overall representation of the sample and is used to classify the activity category.

D. Experimental Results and Discussion

The experimental results of all the algorithms on five datasets in terms of four metrics are summarized in Table III-VII respectively. The best metrics on each dataset have been marked in bold. '#OPPOR' indicates the UCI-OPPORTUNITY dataset. The result of the '#Distribution-Embedded' algorithm is directly copied from [58]. The 'CNN + statistical features' algorithm is not applied to OPPORTUNITY dataset because of unrealistic time consumption. In addition, we further conduct an ablation study to validate the effectiveness and necessity of each feature extraction module in our proposal.

TABLE V: The experimental results of all methods on PAMAP2

	Dataset: PAMAP2			
Method/ Metrics	Acc	Pre	Rec	F1
SVM	0.9167	0.9205	0.9167	0.9170
Random Forest	0.9379	0.9416	0.9379	0.9378
Deep ConvLSTM	0.8522	0.8332	0.8339	0.8288
H-LSTM	0.8735	0.8883	0.8735	0.8723
CNN+statistical	0.9035	0.9049	0.9035	0.9029
Bidir-residual-LSTM	0.8697	0.8796	0.8697	0.8706
#Distribution-Embedded	0.9323	-	-	0.9338
Transformer Encoder	0.8726	0.8787	0.8726	0.8714
Our proposed model	**0.9407**	**0.9427**	**0.9407**	**0.9404**

TABLE VI: The experimental results of all methods on OPPORTUNITY

	Dataset: #OPPOR			
Method/ Metrics	Acc	Pre	Rec	F1
SVM	0.8907	0.8840	0.8907	0.8711
Random Forest	0.8739	0.8769	0.8739	0.8387
Deep ConvLSTM	0.8272	0.6843	0.8272	0.7490
H-LSTM	0.8915	0.8786	0.8915	0.8761
Bidir-residual-LSTM	0.8675	0.8374	0.8675	0.8597
#Distribution-Embedded	0.8366	-	-	0.8601
Transformer Encoder	0.8760	0.8799	0.8760	0.8754
Our proposed model	**0.9130**	**0.9095**	**0.9130**	**0.9093**

Overall performance: Firstly, compared with traditional machine learning classifiers (SVM and Random Forest) with our proposal, it can be observed that our model can outperform these algorithms with manually extracted features and show more steady performance among different datasets. The possible reason is that the handcrafted features heavily depend

TABLE VII: The experimental results of all methods on WISDM

	Dataset: WISDM			
Method/ Metrics	Acc	Pre	Rec	F1
SVM	0.7957	0.7596	0.7957	0.7409
Random Forest	0.8744	0.8812	0.8744	0.8394
Deep ConvLSTM	0.8807	0.8539	0.8613	0.8576
H-LSTM	0.9325	0.9325	0.9323	0.9324
CNN+statistical	0.9494	0.9510	0.9494	0.9493
Bidir-residual-LSTM	0.9374	0.9452	0.9374	0.9392
Transformer Encoder	0.9432	0.9417	0.9432	0.9418
Our proposed model	**0.9618**	**0.9616**	**0.9618**	**0.9614**

TABLE VIII: The ablation experimental results

algorithm	PAMAP2		UCI-HAR		OPPOR		Mhealth		WISDM	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Our proposed model-$f1$	0.8924	0.8842	0.9137	0.9121	0.8955	0.8922	0.9025	0.8899	0.9049	0.9045
Our proposed model-$f2$	0.9280	0.9276	0.9087	0.9087	0.9000	0.8986	0.9512	0.9515	0.9508	0.9494
Our proposed model	**0.9407**	**0.9404**	**0.9372**	**0.9368**	**0.9130**	**0.9093**	**0.9607**	**0.9607**	**0.9618**	**0.9614**

on domain knowledge and are less general because of the weak adaptability to different datasets. By contrast, our model uses the LSTM auto-encoder and CNNs to automatically extract temporal features both from the knowledge of prior labels and the sensor data patterns, which have stronger generalization.

Moreover, compared with other deep-learning based baselines, experimental results clearly show that our proposed semi-supervised model keeps the best performance on all datasets except UCI-HAR. As for the UCI-HAR dataset, it only provides 9 measurements at each timestamp, so it is relatively convenient to manually extract statistical features. However, on other datasets that provide more measurements, such as PAMAP2 and OPPORTUNITY datasets, the 'CNN + statistical features' method will be extremely time-consuming and laborious, and the performance on different datasets can not keep as stable as our method.

Note that our model outperforms the state-of-the-art Transformer encoder that is proven to be powerful in temporal feature extraction in the natural language processing field. It may be because the Transformer encoder tends to be overfitting due to the insufficient training data in HAR. Another possible reason is that compared with our proposal, the Transformer encoder only leverages the label information to learn the feature extraction. However, our model uses both supervised and unsupervised modules to learn better features from both the original data characteristics and the manual labels. Moreover, Transformer pays more attention to global feature extraction. By contrast, our model adopts CNN for extracting local temporal features that are combined with the global temporal features extracted by the LSTM encoder. By this means, the temporal features will contain more useful patterns and become more discriminative.

Therefore, the above experimental results demonstrate the superiority of our proposed model and verify that the features extracted by CNN and LSTM auto-encoder are meaningful and distinguishable for human activity recognition.

Ablation Study: To verify the effectiveness of each module in our model, we conduct the ablation experiment, which is shown as 'Our proposed model-$\boldsymbol{f}_1$' and 'Our proposed model-$\boldsymbol{f}_2$' in Table VIII. 'Acc' represents accuracy. 'Our proposed model-$\boldsymbol{f}_1$' and 'Our proposed model-$\boldsymbol{f}_2$' denote the proposed model without the CNN module and our proposed model without the LSTM auto-encoder module respectively. Therefore, we separately compare the performance of basic CNN and LSTM auto-encoder with our proposed model. The experimental results show that the accuracy and F1-score of our proposed model are better on all datasets. It obviously indicates that both CNN and LSTM auto-encoder modules in our model are effective and indispensable, and the combination of these two modules can further enhance the temporal feature extraction and improve the generalization ability of the model.

Ability to deal with datasets with the unbalanced number of samples in each category: Among the evaluation metrics we adopt, accuracy is mainly responsible for the model's overall classification performance of all categories. However, precision, recall, and F1 score pay more attention to the classification of each category, especially for categories with a small number of samples. It can be seen from the results that in most cases, the precision and F1-score of the compared algorithms are lower than accuracy when tested on the same dataset, especially on the dataset with category imbalance such as OPPORTUNITY. These metrics of the compared algorithms decrease more significantly on OPPORTUNITY, but our algorithm can still maintain a high F1 and precision, indicating that our algorithm has a stronger ability to identify categories with a small number of samples in the category-imbalanced dataset.

Model robustness and generalization: Experimental results obviously show that the performance of our architecture is more stable and adaptive on all datasets compared to other methods. It can keep relatively high accuracy and F1-score on all datasets, while the performance of other algorithms fluctuates greatly on different datasets. For relatively simple datasets, such as UCI-HAR, Wisdm, and Mhealth which only include simple human activities, such as 'running', 'jumping', etc., the data do not contain too many complex motion patterns, so both the compared algorithms and our algorithm show high accuracy and F1 score. However, for the more complex datasets, such as PAMAP2 and OPPORTUNITY, it shows different results. There are more types of activities in the PAMAP2 dataset which contains some complex activities, such as 'working', 'doing housework', etc. The identification of such activities often requires simultaneous gestures and body movement recognition. Besides, the OPPORTUNITY dataset contains many similar gestures that may be composed of similar dynamic features. So the activities of these two datasets are more difficult to recognize. As a result, the accuracy and F1-score of the compared algorithms show varying degrees of attenuation, but our model can still maintain high accuracy and F1 score. Therefore, it demonstrates the robustness and generalization ability of our proposal when faced with different scenarios.

Moreover, to further validate the robustness of our proposed model, we conduct experiments to verify the model's resistance to noise. Specifically, we add Gaussian white noise with different variances to the sensor data in the training set of UCI and PAMAP2 datasets. Then we observe the accuracy and F1-score of our proposed model, CNN, and 'Bidir-residual-

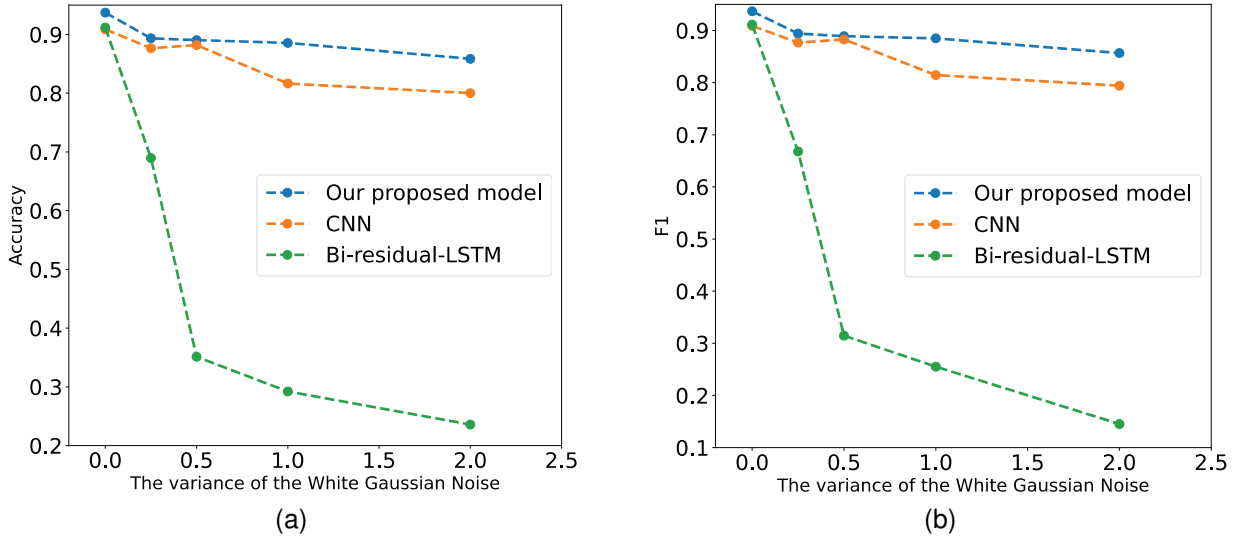

Fig. 5: The accuracy and F1 score of three models against different variances of noise on UCI dataset

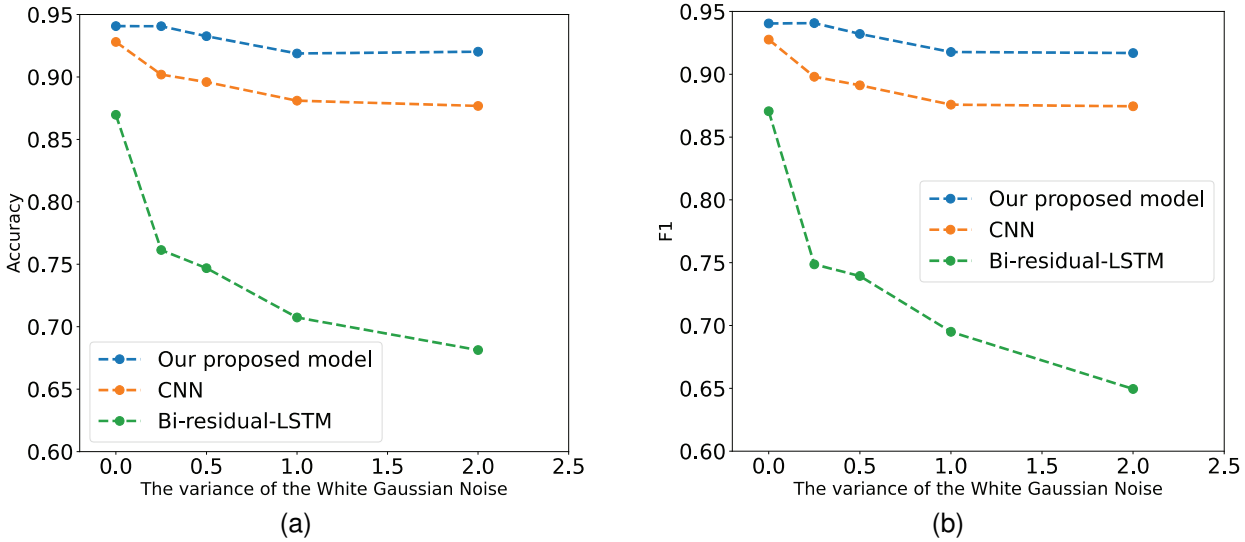

Fig. 6: The accuracy and F1 score of three models against different variances of noise on PAMAP2 dataset

LSTM' when they are affected by different levels of noise. The experimental results of this part are shown in Fig 5 and Fig 6. It can be clearly observed that with the increasing influence of noise on sensor data, the performance of our proposed model and compared methods all show a downward trend. However, the decrease in performance of our model is significantly less than other methods, whether on UCI or PAMAP2 datasets. Therefore, it can prove that our model has a more powerful ability to resist noise, thus further validating the robustness of our proposal.

Improvements of specific activities: To further explore the improvement of our proposed model on some activities that are difficult to distinguish, we also adopt the Confusion Matrix to display the accuracy of all kinds of activities in UCI-HAR and PAMAP2 datasets. We have conducted experiments on our proposed model, CNN, and LSTM. The results are shown in Fig7. In Fig7, numbers 1-6 in the UCI dataset represent 'walking', 'walking upstairs', 'walking downstairs', 'sitting', 'standing', and 'lying', and numbers 1-12 in PAMAP2 dataset represent 'rope jumping', 'lying', 'sitting', 'standing', 'walking', 'running', 'cycling', 'Nordic walking', 'ascending stairs', 'descending stairs', 'vacuum cleaning', 'ironing'.

The result of (a)(b)(c) shows that the activity 2 and 3, namely 'walking downstairs' and 'walking upstairs' are a pair of activities that are difficult to distinguish, as marked by the red box in Fig7. The result displays that activity 3 is often mistaken for activity 2. Compared to LSTM and CNN, our proposed model has improved the accuracy of activity 3 evidently, the improved accuracy is as high as 10% and 12% respectively.

In the PAMAP2 dataset, activities 3 and 4, i.e., 'sitting' and 'standing' are activities that are difficult to recognize, as

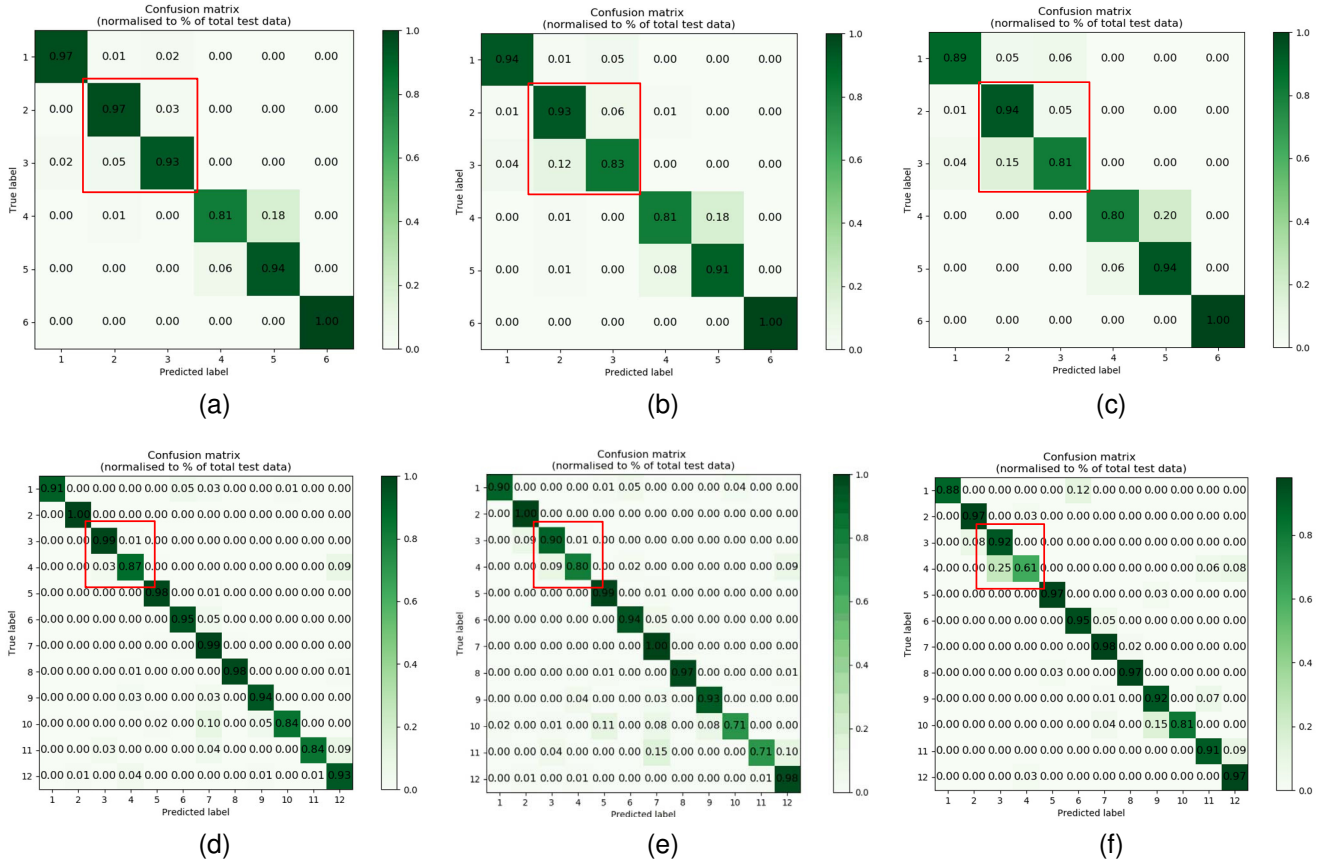

Fig. 7: (a)(b)(c): From left to right are the Confusion Matrix of our proposed model, CNN, and LSTM respectively on UCI dataset. (d)(e)(f): From left to right are the Confusion Matrix of our proposed model, CNN, and LSTM respectively on PAMAP2 dataset. The vertical axis represents the true class label, and the horizontal axis represents the predicted class label. The value in the i-th row and the j-th column represents the proportion of the samples whose true class is the i-th class being classified into the class j. Therefore, the values on the diagonal predict the correct proportion of each class of samples. The higher the value, the darker the color of the block it is in. By comparing the accuracy values on the diagonal, these three figures are used to show the improvement of some specific types of activities that are difficult to distinguish (shown in the red box).

marked by the red box in Fig 7. Perhaps because they are inverse activities and have similar body movement patterns. The result of (d)(e)(f) shows that our proposed model has improved the performance of these two activities compared to the other two basic methods. Moreover, the complex activity 11 'vacuum cleaning' is also improved to a certain extent by our model.

Therefore, the improved accuracy of these composite and inverse activity pairs shows that our model can further improve the feature extraction by using the semi-supervised training method, thus making extracted features more effective and discriminative.

V. Conclusion

In this work, we study the problem of continuity activity feature extraction. We present a novel human continuity activity semi-supervised recognizing method in multi-view IoT network scenarios. In our architecture, we design the supervised activity feature extraction module and unsupervised encoder-decoder feature extraction module and then combine the features extracted from these modules to improve the accuracy of activity recognition and robustness of the method. Furthermore, we refine the unsupervised encoder using activity labels to make extracted features contain richer semantic information. The advantage of this model is that it uses the semi-supervised method to complete robust feature extraction and maintains high performance in a variety of scenarios. Extensive experiments demonstrate that our proposed architecture can improve the accuracy and F1 score compared with other state-of-the-art models, and it is more adaptive and general with its stable performance on all datasets. In the future, we may try to use more weakly-labeled data for training. In addition, we may combine sensor-based data and vision-based data together for multi-modal human activity recognition.

Acknowledgments

The authors would like to thank editor and anonymous reviewers who gave valuable suggestion that has helped to improve the quality of the paper. This work was supported in part by the National Key R&D Program of China

(No.2022YFF0903304);National Natural Science Foundation of China under Grant 92167109;by the Dadu River cascade hydropower station safety early warning project.

REFERENCES

[1] M. Ziaeefard and R. Bergevin, "Semantic human activity recognition: A literature review," *Pattern Recognition*, 2015.

[2] S. Sennan, "Internet of things based ambient assisted living for elderly people health monitoring," *Research Journal of Pharmacy and Technology*, vol. 11, no. 9, pp. 1–5, 2018.

[3] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I. De Munari, "Iot wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8553–8562, 2019.

[4] T. Stiefmeier, D. Roggen, G. Troster, G. Ogris, and P. Lukowicz, "Wearable activity tracking in car manufacturing," *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 42–50, 2008.

[5] A. Nj, B. Nb, and C. Msb, "A new hybrid deep learning model for human action recognition," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 4, pp. 447–453, 2020.

[6] C. Aviles-Cruz, E. Rodriguez-Martinez, J. Villegas-Cortez, and A. Ferreyra-Ramirez, "Granger-causality: An efficient single user movement recognition using a smartphone accelerometer sensor," *Pattern Recognition Letters*, vol. 125, pp. 576–583, 2019.

[7] Y. Wang, S. Cang, and H. Yu, "A survey on wearable sensor modality centred human activity recognition in health care," *Expert Systems with Applications*, vol. 137, pp. 167–190, 2019.

[8] A. R. Javed, R. Faheem, M. Asim, T. Baker, and M. O. Beg, "A smartphone sensors-based personalized human activity recognition system for sustainable smart cities," *Sustainable Cities and Society*, vol. 71, p. 102970, 2021.

[9] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–40, 2021.

[10] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[11] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Activity recognition with evolving data streams: A review," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–36, 2018.

[12] S. Mekruksavanich and A. Jitpattanakul, "Biometric user identification based on human activity recognition using wearable sensors: An experiment using deep learning models," *Electronics*, vol. 10, no. 3, p. 308, 2021.

[13] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey," in *23th International conference on architecture of computing systems 2010*. VDE, 2010, pp. 1–10.

[14] D. Bouchabou, S. M. Nguyen, C. Lohr, B. LeDuc, and I. Kanellos, "A survey of human activity recognition in smart homes based on iot sensors algorithms: Taxonomies, challenges, and opportunities with deep learning," *Sensors*, vol. 21, no. 18, p. 6037, 2021.

[15] X. Zhou, W. Liang, I. K. Wang, H. Wang, and Q. Jin, "Deep learning enhanced human activity recognition for internet of healthcare things," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2020.

[16] Q. Zhu, Z. Chen, and Y. C. Soh, "A novel semisupervised deep learning method for human activity recognition," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3821–3830, 2018.

[17] H. Haresamudram, I. Essa, and T. Ploetz, "Contrastive predictive coding for human activity recognition," vol. 5, no. 2, pp. 1–26, 2021.

[18] D. Wang, J. Yang, W. Cui, L. Xie, and S. Sun, "Multimodal csi-based human activity recognition using gans," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17 345–17 355, 2021.

[19] A. Saeed, F. D. Salim, T. Ozcelebi, and J. Lukkien, "Federated self-supervised learning of multi-sensor representations for embedded intelligence," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2020.

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

[21] F. Xiao, L. Pei, L. Chu, D. Zou, W. Yu, Y. Zhu, and T. Li, "A deep learning method for complex human activity recognition using virtual wearable sensors," in *International Conference on Spatial Data and Intelligence*. Springer, 2020, pp. 261–270.

[22] P. Vepakomma, D. De, S. K. Das, and S. Bhansali, "A-wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities," in *2015 IEEE 12th International conference on wearable and implantable body sensor networks (BSN)*. IEEE, 2015, pp. 1–6.

[23] L. Pei, S. Xia, L. Chu, F. Xiao, Q. Wu, W. Yu, and R. Qiu, "Mars: Mixed virtual and real wearable sensors for human activity recognition with multidomain deep learning model," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9383–9396, 2021.

[24] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *6th international conference on mobile computing, applications and services*. IEEE, 2014, pp. 197–205.

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

[26] N. Rashid, B. U. Demirel, and M. Abdullah Al Faruque, "Ahar: Adaptive cnn for energy-efficient human activity recognition in low-power edge devices," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 041–13 051, 2022.

[27] X. Zhou, W. Liang, J. Ma, Z. Yan, I. Kevin, and K. Wang, "2d federated learning for personalized human activity recognition in cyber-physical-social systems," *IEEE Transactions on Network Science and Engineering*, 2022.

[28] Q. A. Zhen, A. Yz, A. Sm, A. Zq, and B. Kkrc, "Imaging and fusing time series for wearable sensor-based human activity recognition," *Information Fusion*, vol. 53, pp. 80–87, 2020.

[29] Y. Guan and T. Plötz, "Ensembles of deep lstm learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–28, 2017.

[30] L. Wang and R. Liu, "Human activity recognition based on wearable sensor using hierarchical deep lstm networks," *Circuits, Systems, and Signal Processing*, vol. 39, no. 2, pp. 837–856, 2020.

[31] Y. Zhao, R. Yang, G. Chevalier, X. Xu, and Z. Zhang, "Deep residual bidir-lstm for human activity recognition using wearable sensors," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[32] Y. Shavit and I. Klein, "Boosting inertial-based human activity recognition with transformers," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2021.

[33] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[34] L. Chen, X. Liu, L. Peng, and M. Wu, "Deep learning based multimodal complex human activity recognition using wearable devices," *Applied Intelligence*, vol. 51, no. 6, pp. 4029–4042, 2021.

[35] O. Nafea, W. Abdul, G. Muhammad, and M. Alsulaiman, "Sensor-based human activity recognition with spatio-temporal deep learning," *Sensors*, vol. 21, no. 6, p. 2141, 2021.

[36] K. Muhammad, A. Ullah, A. S. Imran, M. Sajjad, M. S. Kiran, G. Sannino, V. H. C. de Albuquerque *et al.*, "Human action recognition using attention based lstm network with dilated cnn features," *Future Generation Computer Systems*, vol. 125, pp. 820–830, 2021.

[37] I. K. Ihianle, A. O. Nwajana, S. H. Ebenuwa, R. I. Otuka, and M. O. Orisatoki, "A deep learning approach for human activities recognition from multimodal sensing devices," *IEEE Access*, vol. 8, pp. 179 028–179 038, 2020.

[38] M. Abdel-Basset, H. Hawash, R. K. Chakrabortty, M. Ryan, M. Elhoseny, and H. Song, "St-deephar: Deep learning model for human activity recognition in ioht applications," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4969–4979, 2020.

[39] L. Pei, J. Liu, R. Guinness, Y. Chen, H. Kuusniemi, and R. Chen, "Using ls-svm based motion recognition for smartphone indoor wireless positioning," *Sensors*, vol. 12, no. 5, pp. 6155–6175, 2012.

[40] K. BhanuJyothi, K. H. Bindu, and D. Suryanarayana, "A comparative study of random forest & k-nearest neighbors on har dataset using caret," *IJIRT*, vol. 3, pp. 6–9, 2017.

[41] S. Pirttikangas, K. Fujinami, and T. Nakajima, "Feature selection and activity recognition from wearable sensors," in *International symposium on ubiquitious computing systems*. Springer, 2006, pp. 516–527.

[42] M. Yang, H. Zheng, H. Wang, S. McClean, J. Hall, and N. Harris, "A machine learning approach to assessing gait patterns for complex regional pain syndrome," *Medical engineering & physics*, vol. 34, no. 6, pp. 740–746, 2012.

[43] M. Uray, D. Skocaj, P. M. Roth, H. Bischof, and A. Leonardis, "Incremental lda learning by combining reconstructive and discriminative approaches." in *BMVC*, vol. 2007. Citeseer, 2007, pp. 272–281.

[44] D. Triboan, L. Chen, F. Chen, and Z. Wang, "A semantics-based approach to sensor data segmentation in real-time activity recognition," *Future Generation Computer Systems*, vol. 93, pp. 224–236, 2019.

[45] J. Schmidhuber, S. Hochreiter *et al.*, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.

[46] C. Jia, M. Shao, S. Li, H. Zhao, and Y. Fu, "Stacked denoising tensor auto-encoder for action recognition with spatiotemporal corruptions," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1878–1887, 2017.

[47] J. Yang, J.-y. Yang, D. Zhang, and J.-f. Lu, "Feature fusion: parallel strategy vs. serial strategy," *Pattern recognition*, vol. 36, no. 6, pp. 1369–1381, 2003.

[48] U. G. Mangai, S. Samanta, S. Das, and P. R. Chowdhury, "A survey of decision fusion and feature fusion strategies for pattern classification," *IETE Technical Review*, vol. 27, no. 4, pp. 293–307.

[49] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th international symposium on wearable computers*. IEEE, 2012, pp. 108–109.

[50] I. P. Machado, A. L. Gomes, H. Gamboa, V. Paixão, and R. M. Costa, "Human activity data discovery from triaxial accelerometer sensor: Non-supervised learning sensitivity to feature extraction parametrization," *Information Processing & Management*, vol. 51, no. 2, pp. 204–214, 2015.

[51] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*. Springer, 2012, pp. 216–223.

[52] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, "A robust human activity recognition system using smartphone sensors and deep learning," *Future Generation Computer Systems*, vol. 81, pp. 307–313, 2018.

[53] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[54] H. Kalantarian, N. Alshurafa, T. Le, and M. Sarrafzadeh, "Monitoring eating habits using a piezoelectric sensor-based necklace," *Computers in biology and medicine*, vol. 58, pp. 46–55, 2015.

[55] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. d. R. Millán, and D. Roggen, "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2033–2042, 2013.

[56] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.

[57] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mhealthdroid: a novel framework for agile development of mobile health applications," in *International workshop on ambient assisted living*. Springer, 2014, pp. 91–98.

[58] H. Qian, S. J. Pan, B. Da, and C. Miao, "A novel distribution-embedded neural network for sensor-based activity recognition." in *IJCAI*, vol. 2019, 2019, pp. 5614–5620.

[59] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[60] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, "A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1384–1393, 2018.