IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 23, 2020, accepted April 3, 2020, date of publication April 7, 2020, date of current version April 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986246

A Hybrid Network Based on Dense Connection and Weighted Feature Aggregation for Human Activity Recognition

TIANQI LV, XIAOJUAN WANG, LEI JIN, YABO XIAO, AND MEI SONG
School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
Corresponding author: Xiaojuan Wang (wj2718@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61871046.

ABSTRACT Human activity recognition (HAR) using body-worn sensors is an active research area in human-computer interaction and human activity analysis. The traditional methods use hand-crafted features to classify multiple activities, which is both heavily dependent on human domain knowledge and results in shallow feature extraction. Rapid developments in deep learning have caused most researchers to switch to deep learning methods, which extract features from raw data automatically. Most of the existing works on human activity recognition tasks involve multimodal sensor data, and these networks mainly focus on the top representation extracted from bottom-up feedforward process without reusing other features from bottom layers. In this paper, we present a novel hybrid deep learning network for human activity recognition that also employs multimodal sensor data; however, our proposed model is a ConvLSTM pipeline that makes full use of the information in each layer extracted along the temporal domain. Thus, we propose a dense connection module (DCM) to ensure maximum information flow between the network layers. Furthermore, we employ a multilayer feature aggregation module (MFAM) to extract features along the spatial domain, and we aggregate the features obtained from every convolutional layer according to the importance of features in different spatial locations. The output of the MFAM is input into two LSTM layers to further model the temporal dependencies. Finally, a fully connected layer and a softmax function are used to compute the probability of each class. We demonstrate the effectiveness of our proposed model on two benchmark datasets: Opportunity and UniMiB-SHAR. The results illustrate that our designed network outperforms the state-of-the-art models. We also conduct experiments on efficiency, multimodal fusion and different hyperparameters to analyze our proposed network. Finally, we carry out ablation and visualization experiments to reveal the effectiveness of the two proposed modules.

INDEX TERMS Human activity recognition, deep learning, dense connection, multilayer feature aggregation, multimodal sensor data.

I. INTRODUCTION

The growing popularity of smart, wearable devices has greatly expanded the availability of time-series sensor data related to human activities. Therefore, wearable sensor-based human activity recognition (HAR) has attracted considerable research attention in the areas of pervasive computing and artificial intelligence. The main goal of HAR is to automatically detect and recognize activities based on analyzing data acquired by sensors [1]. Applications that benefit from HAR include health support [2], [3], smart homes [4], [5] and rehabilitation [6]. Compared with recognition using computer vision, wearable sensor-based HAR approaches offer low cost, high performance, and portability [7].

A typical HAR system includes data acquisition, data preprocessing, segmentation, feature extraction, and classification. Smartwatches, smartphones, and other devices supply data from multiple sensors. Pre-processing consists of segmentation (e.g., with sliding windows) and partitioning. Each segment provides features that can be useful in identifying different activities. The system then trains a classifier to make predictions based on these features. Fig. 1 illustrates this process.

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

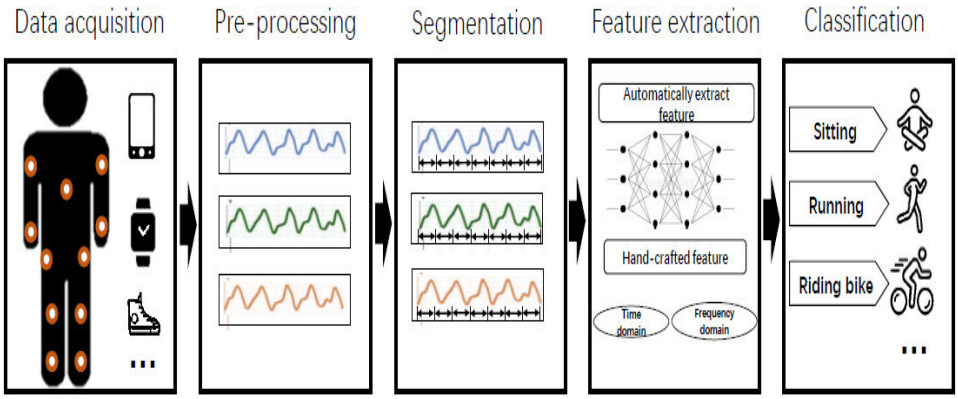


FIGURE 1. The processing flow of the human activity recognition system.

Most existing research focuses on feature extraction methods because the discriminative features are important for ensuring the generalizability of the HAR system. There are two primary ways to extract features from sensor-based data. One employs hand-crafted features based on the statistical knowledge, while the other automatically extracts features using neural networks [8]. The extraction of meaningful hand-crafted features from the time and frequency domains relies heavily on human experience and domain knowledge. In addition, hand-crafted features are usually designed for a specific task and are not suited for more general environments and tasks. Deep learning advances have been widely applied in HAR [9] because deep learning models can automatically extract high-dimensional features and are not dependent on domain knowledge.

Convolutional neural networks (CNNs) [10] and recurrent neural networks (RNNs) [11] are among the most popular deep learning methods. When used for classifying large-scale time-series data such as HAR, CNNs have the advantages of local dependency and scale invariance, making them the best candidates for use in classification problems. RNNs additionally consider long-term time dependencies, which is beneficial for time-series data. However, RNNs suffer from gradient exploding or vanishing problems. To address these problems, a variant of the standard neuron called the long short-term memory (LSTM) cell was proposed [12]. Thus, to obtain high-dimensional features that have both short- and long-term time dependencies, we combine CNN and LSTM networks to form a hybrid deep-learning architecture.

Most of the existing works on human activity recognition use CNN models with 1D or 2D kernels. For multimodality time series data, a 1D convolution operation captures only local dependencies over time but does not make full use of the dependency between different channels of multiple sensors. CNNs with 2D kernels can capture local dependency along time and spatial domains for unimodal sensor data, but they require large numbers of parameters, making them unsuitable for mobile devices with limited memory. Inspired by [13], any $N \times N$ convolution can be replaced by a $1 \times N$ convolution followed by a $N \times 1$ convolution, and this two-layer solution is considerably cheaper than the same square convolution. Therefore, in this paper, we use a 3×1 convolution followed by a 1×3 convolution to replace the 3×3 convolution. To prevent the decrease in accuracy caused by this operation, we use a dense connection after each 3×1 convolution. Then, the output of all the preceding layers is used as the input to each layer, and its output is used as input into all the subsequent layers. Dense connections help ensure maximum information flow between layers in the network. In addition, networks designed for human activity recognition focus mainly on the top representation extracted from the bottom-up feedforward process and ignore other features from the bottom layers. Therefore, we collect feature maps after each 1×3 convolution and aggregate these feature maps according to their importance in different spatial locations. We demonstrate the effectiveness of our method on two open human activity datasets, Opportunity [14] and UniMiB-SHAR [15]. Our contributions are as follows.

- To design network with fewer parameters, we replace 3×3 convolutional operations in our proposed model with a 3×1 convolution followed by a 1×3 convolution. The 3×1 convolution and the 1×3 convolution capture local dependencies along the temporal dimension of a single sensor and among multiple sensors, respectively.
- We design a dense connection module to collect the output of each 3×1 convolutional layer and promote information flow in the model. To reuse the information from each layer, we collect feature maps after each 1×3 convolution and employ a softmax function to aggregate the feature maps of each layer according to their importance in each spatial location. The aggregation module is called the multilayer feature aggregation module.
- By combining the dense connection module and the multilayer feature aggregation module, we propose a novel hybrid network for human activity recognition based on an underlying ConvLSTM network.
- We show that the proposed model outperforms other state-of-the-art models designed for human activity recognition on different recognition tasks on the Opportunity dataset under different data division methods

(i.e., 5-fold cross-validation and leave-subject-out cross-validation) on the UniMiB-SHAR dataset.
- We analyze the efficiency of our proposed network and discuss the influence of different numbers of sensor channels and hyperparameters on the network. In addition, we also conduct an ablation experiment and a visualization experiment to show the effectiveness of the two proposed modules.

The rest of this paper is organized as follows. Section II provides a brief overview of related works on HAR, including both traditional methods and deep learning methods. In Section III, we introduce the three main parts of our proposed network: a dense connection module, a multilayer feature aggregation module and a fundamental ConvLSTM framework. In Section VI, we introduce the two benchmark datasets, the performance metrics, and the settings used for model training. Section V provides a comparative analysis of the proposed network. We present conclusions and future work in Section VI.

II. RELATED WORK

Early research into HAR uses traditional sensor-based HAR systems with hand-crafted features extracted from the time and frequency domains to predict the class labels. The most popular traditional methods applied to recognize human activities include k-nearest neighbor (kNN), support vector machine (SVM), and decision tree (DT) models. Janidarmian *et al.* [16] conducted an extensive analysis among 293 classifiers, including DTs, KNNs, and SVMs, on the most complete dataset available, which includes data from accelerometers and various heterogeneous sources. The average classification accuracies achieved were 96.44% $\pm$ 1.62% with under 10-fold evaluation and 79.92% ś 9.68% under leave-subject-out cross-validation. The results indicate that KNN and its ensemble methods yield stable results across different positions and window sizes. Xie *et al.* [17] proposed a multilayer strategy based on inertial sensors and barometers to recognize eight human activities that adopted random forests (RFs) and SVMs for different classifications, and in which different feature sets were selected for the different classifiers.

Many achievements have been made by deep learning in fields such as visual object recognition, natural language processing, and logical reasoning [18]. Generally, the deep learning architectures for HAR fall into three categories. The first category consists of CNNs. Panwar *et al.* [19] designed a generalized CNN-based model to recognize three fundamental human forearm movements collected from a single accelerometer sensor on the wrist. The experimental results showed that the CNN-based model outperformed SVM, K-means and latent Dirichlet allocation (LDA). The authors of [20] investigated the effectiveness of proposed CNN-extracted features compared with hand-crafted features for the paroxysmal atrial fibrillation (PAF) screening problem. The use of a CNN structure to extract features in combination with other classifiers can significantly improve the resulting classification performance. Andrey and Ignatov [21] presented a CNN model for online HAR, and their experiments showed that a CNN combined with hand-crafted features yields significantly improved performance and can be executed on mobile phones in real time. Wang *et al.* [22] proposed a novel attention-based human activity recognition method to process weakly labeled activity data. Compared with a CNN and DeepConvLSTM, their experiments showed that the designed model worked well on the traditional UCI HAR dataset and outperformed them on the weakly labeled dataset in terms of accuracy.

The second category uses RNN models to capture the time dependencies of time-series data. Edel and Koppe proposed a binarized long short-term memory network (B-BLSTM-RNN) that is especially suitable for resource-constrained environments; it outperforms other recent methods by large margins on three tested datasets [23]. To tackle the challenges of imbalanced datasets and problematic data quality, Guan and Ploetz [24] designed a model through ensembles of deep LSTM networks that improved the recognition accuracy on the Opportunity, PAMAP2 and Skoda datasets. Inoue *et al.* [25] investigated several models and then proposed a good architecture that can perform mobile HAR with high throughput.

The third category consists of hybrid models that combine deep models to address HAR tasks. Ordóñez and Roggen showed that a hybrid architecture based on both convolutional and LSTM recurrent units functions better than do deep non-recurrent networks, and confirmed the improved performance on two benchmark datasets [6]. Xi *et al.* [26] presented a novel deep learning framework for human activity recognition problems. The model includes dilated CNN and SRU networks that exponentially expand the receptive field with no loss of resolution or coverage and model the long-temporal dependencies. Yi *et al.* [27] designed a novel deep learning framework called multi-channel deep convolutional neural networks (MC-DCNN) that learns features from the individual univariate time series in each channel and then applies the learned features in a multilayer perceptron (MLP) for classification. Extensive experiments on real-world data sets show that the model is competitive in accuracy. To improve the performance of the HAR system and design a smaller network for use in mobile devices, we propose using a novel hybrid model that fully aggregates features along both temporal and spatial domains; it also requires fewer parameters when combined with a DeepConvLSTM [6].

III. ARCHITECTURE

To analyze multimodal sensor data and obtain multivariate time series, the existing works on HAR using CNNs often use large convolution kernels to enlarge the receptive field. In addition, they primarily employ only the top-level information extracted from the bottom-up feedforward process, neglecting the use of other features from the lower-level layers and failing to consider the importance of multiple features

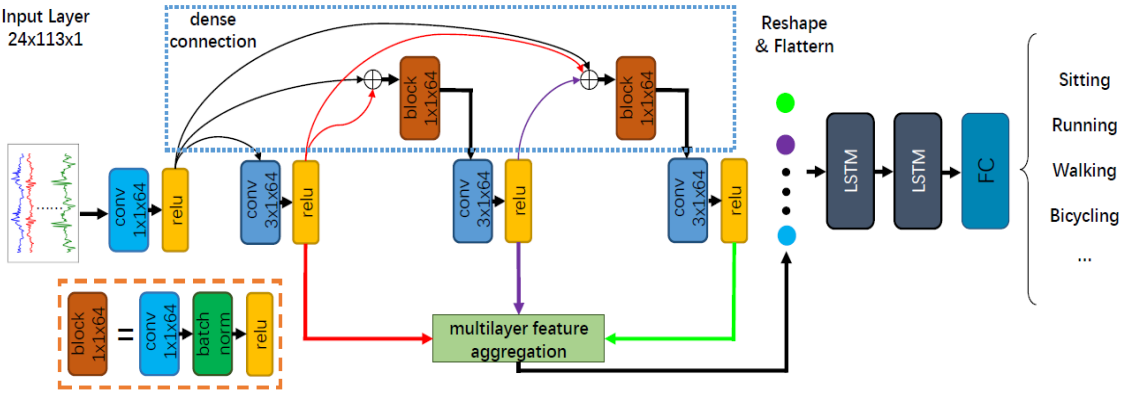


FIGURE 2. An overview of the proposed model for human activity recognition. The numbers shown inside each convolutional layer denote the conv kernel size, and the output channels. The architecture of the brown block is shown in the lower left corner. "FC" stands for "Fully Connected". We provide detailed descriptions of the dense connection module and the multilayer feature aggregation module in Sections III-A and III-B, respectively.

in the same spatial location. Therefore, we propose a novel hybrid network with two modules designed to address these problems. The overall architecture of our network is shown in Fig. 2. The network includes a base ConvLSTM pipeline, a dense connection module (DCM) and a multilayer feature aggregation module (MFAM). In this section, we introduce these two modules and the underlying ConvLSTM in detail.

A. DENSE CONNECTION MODULE

Traditional convolutional networks with L layers only have L connections. However, our dense connection module includes $\frac{L(L+1)}{2}$ direct connections. The output of the preceding layers are used as the input to each layer, and the output from each layer is sent into all subsequent layers. This operation promotes information flow in the model and ensures that each layer can directly access the gradients from the loss function.

CNN networks commonly consist of S layers; here, we denote the output of each layer as l_i. In our dense connection module, the s-th layer obtains the features from its preceding layers as input and uses a nonlinear function $\mathcal{F}_d(\cdot)$ to obtain l_s. The process is formulated as follows:

$$l_s = \mathcal{F}_d(\sum_{i=1}^{s-1} l_i, \theta_d), , \tag{1}$$

where $\sum_{i=1}^{s-1}$ is the addition of the feature maps obtained from layers $1, \ldots, s-1$. Inspired by [28], we define $\mathcal{F}_d(\cdot)$ as a block of three stacked layers: a 1×1 convolutional layer, batch normalization [29] and rectified linear units (RELU) [30], and θ_d represents its parameters. This block is illustrated in the lower-left corner of Fig. 2. Compared with the concatenation operation, using addition to aggregate information saves parameters, which reduces the number of channels and achieves better results in our experiments.

Considering the computational efficiency of the human activity recognition task, we use only two dense connection operations in our network. Along the temporal domain of multivariate time series data, both dense layers collect information produced by their preceding layers and pass on their own features to the next 3×1 convolutional layers. Over time, this model can acquire rich information via the dense connection module.

B. MULTILAYER FEATURE AGGREGATION MODULE

We propose a multilayer feature aggregation module to collect the features from each convolutional layer and aggregate them in different spatial locations according to their importance, as illustrated in Fig. 3.

Specifically, we formulate the forward process in the multilayer feature aggregation module as follows. We denote $L_S = \{l_1, l_2, l_3\}$ as the set of feature maps obtained by the three 3×1 convolutional layers, where l_i is the i-th feature map, which has 64 channels. For each l_i, we capture the spatial dependency among the sensors by generating the feature maps I_i:

$$I_i = \mathcal{F}_s(l_i, \theta_s), \tag{2}$$

where $\mathcal{F}_s$ is a composite function of two layers (a 1×3 convolutional layer and a RELU), and θ_s is its parameters. To aggregate information from different layers, we concatenate the outputs of $\mathcal{F}_s$. After concatenation, we apply a 1×1 convolutional layer to reduce the number of channels. The output of the 1×1 convolutional layer contains 3 channels, corresponding to the number of L_S. The concatenation and the 1×1 convolutional layer calculation is as follows:

$$H = \mathcal{F}_r([I_1, I_2, I_3], \theta_r), \tag{3}$$

where $[I_1, I_2, I_3]$ refers to the concatenation operation, $\mathcal{F}_r$ is the 1×1 convolutional layer, and θ_r is its parameters. Next, H is normalized to $A = \{a_1, a_2, a_3\}$ along the channel dimension by a softmax function:

$$a_i = \frac{exp(H_i)}{\sum_{j=1}^{3} exp(H_j)}. \tag{4}$$

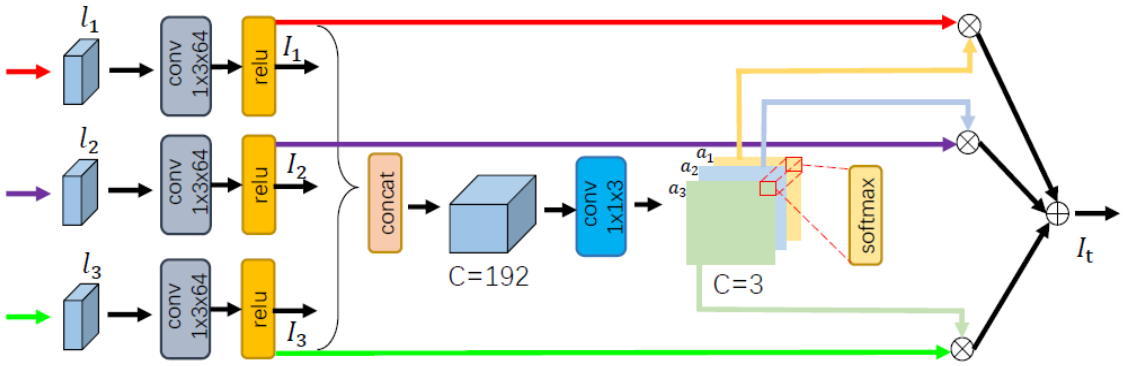


FIGURE 3. The multilayer feature aggregation module: the first stage of MFAM extracts features along the spatial domain and concatenates features along the channel dimension. Then, the second stage aggregates these features in different spatial locations according to their importance.

Finally, the normalized compatibility scores A and the feature maps $\{I_1, I_2, I_3\}$ are used to produce the output I_t by element-wise weighted averaging:

$$I_t = I_1 \cdot a_1 + I_2 \cdot a_2 + I_3 \cdot a_3. \tag{5}$$

A small value in position p of feature map a_i means that the information is irrelevant and should be suppressed. Therefore, the softmax function controls the contribution made by each of the three feature maps to the global feature map. Through weighted aggregation, the model learns to extract the rich features of both the temporal and spatial domains from multimodal data.

C. FUNDAMENTAL ConvLSTM

The fundamental model on which the DCM and MFAM are based is a ConvLSTM pipeline that consists of convolutional, LSTM and fully connected layers. The fundamental model is similar to the network in [6], but our model has fewer parameters. The input data are extracted from multimodal time series data using a sliding window approach and then turned into a two-dimensional matrix. The input data are passed into a 1×1 convolutional layer that can cast input into hidden spaces to create better information representations. Next are three 3×1 convolutional layers that capture dependency over time. The input to each of the three convolutional layers comes from the DCM, and the output of every layer is passed into the MFAM. Thus, the feature maps of MFAM include rich global information regarding both the time and spatial dimensions. We use RELU as the activation function after each of the three convolutional layers. Based on the experiments in [31], two stacked LSTM layers are beneficial for processing sequential data. Therefore, we employ two LSTM layers to process the output of MFAM. The last layer is a fully connected layer that maps the features obtained from the last LSTM layer into the output classes. After this layer, we apply a softmax function to obtain model output. Following the expression in [32], the shorthand description of our fundamental model is $C(1) - C(64) - C(64) - C(64) - R(64) - R(64) - D(64)$, where $C(n_c)$ denotes a convolutional layer with n_c feature maps, $R(n_l)$ is an LSTM layer with n_l cells and $D(n_d)$ is a fully connected layer with n_d units. Moreover, we include three dropout layers before the two LSTM and fully connected layers for regularization.

IV. EVALUATION

In this section, we first introduce two benchmark datasets. Then, to address the problem of the imbalanced classes, we adopt weighted $F1$ score and macro average accuracy as our metrics to assess the model performances. Finally, we describe the training parameters for our network.

A. BENCHMARK DATASET

Human activities are commonly defined as periodic (e.g., running and jumping), static (e.g., standing still), or sporadic (e.g., watching TV or driving a car) motions. A benchmark dataset should include all these types of activities. Researchers have created several datasets for HAR, including the Opportunity [14], UniMiB-SHAR [15], PAMAP2 [33], and Skoda [34] datasets. The two datasets we employ to evaluate the performance of our model are described below.

The Opportunity Dataset contains data from several on-body sensors of 17 different activities performed by 4 subjects in a kitchen scenario. It also includes a Null class, which is unrelated to any of the other activities, for a total of 18 classes. The data were acquired from 12 body parts at a frequency of 30 Hz. Each subject was asked to perform each action 6 times to record the data. All the subjects in the first 5 trials performed all the activities according to a script and then repeated each activity 20 times in the final trial. The data are stored in 5 ADL files and 1 drill file. We used 113-dimensional data for our experiments; missing values were inserted using linear interpolation. We chose runs 4 and 5 from subjects 2 and 3 as the testing dataset, and used the remaining data for training. For the frame-by-frame

analysis, the length of the sliding windows was 0.8 *s* and the sliding stride was 0.4 *s*. The resulting training set included approximately 61k frames.

The UniMiB-SHAR Dataset consists of annotated data obtained by a Samsung Galaxy Nexus I9250 smartphone from 30 volunteers (6 males and 24 females). Data from the smartphone's 3-axis accelerometer were captured at a constant rate of 50 Hz. Each subject placed the smartphone in his or her left or right pocket and performed 17 activities. For this dataset, the data were sliced with a fixed-width sliding window of approximately 3 s using a segmentation technique [15]. The total dataset includes approximately 11k frames. For the experiments with the UniMiB-SHAR dataset, we conducted both 5-fold and leave-subject-out cross-validation [15].

B. PERFORMANCE METRICS

Because human activity datasets often have unbalanced classes, reasonable performance metrics are required to measure human activity recognition algorithms. For example, the NULL class of the Opportunity dataset represents over 75% of the data. Therefore, when using classification accuracy as a performance assessment metric, the majority class will have a significant influence on the total accuracy. For this reason, we assess models using the weighted $F1$ score, which is the harmonic mean of precision and recall that provides a better evaluation than can precision alone. Precision and recall are defined as $\frac{TP}{TP+FP}$ and $\frac{TP}{TP+FN}$, respectively, where TP, FP, and FN represent the number of true positives, false positives, and false negatives, respectively. The weighted $F1$ score calculates the $F1$ score for each class and then multiplies it by a weight value. We compute the weighted $F1$ score using

$$F_w = \sum_c 2 * w_c \frac{precision_c \cdot recall_c}{precision_c + recall_c}, \tag{6}$$

where c represents the class index, and $w_c = n_c/N$ designates the proportion of samples belonging to the c-th class. We also use macro average accuracy (MAA) to evaluate the classification performances. The MAA is defined as follows:

$$MAA = \frac{1}{c} \sum_c \frac{TP_c}{n_c}. \tag{7}$$

C. MODEL TRAINING

We implemented our deep-learning models in Python using the PyTorch [35] framework, trained fully supervised models with the time-series data and calculated gradients by back-propagation from the softmax layers. We then employed the Adadelta optimizer and the gradient descent algorithm for all the trainable parameters. The recorded data were sampled as mini-batches with a size of 100 in the training and testing phases. We used the categorical cross-entropy function to calculate the loss between predictions and targets. In addition, all the parameters were randomly orthogonally initialized. The dropout probability was set to 0.5. Each model was trained for 150 epochs. All the experiments were performed on a workstation equipped with an Intel E5-2620 at 2.10 GHz, 9.6 GB RAM and a 11 GB NVIDIA 1080 Ti GPU.

V. RESULTS

A. CLASSIFICATION PERFORMANCE

To evaluate the recognition performances, we compared our proposed model with some other recognition models on the Opportunity and UniMiB-SHAR datasets in terms of the weighted $F1$ score (F_w) and performed leave-subject-out cross-validation on the UniMiB-SHAR dataset. We evaluated the following recognition models.

1) CONVOLUTIONAL NEURAL NETWORKS WITH A 1D KERNEL (1D CNN) [1]

In this model, each convolutional layer uses a 1D convolution operation along the temporal axis of an individual channel. In addition, the layer adopts RELU as its activation function and includes a max pooling operation. The shorthand description is $C(50) - C(40) - C(30) - D(1000) - Sm$, where Sm is a softmax layer.

2) LSTM [1]

This model, which is based on previous experiments, uses two stacked LSTM layers. Similar to a CNN models, the output of the second LSTM layer is sent to dense and softmax layers. The LSTM cells use a sigmoid function for gate activations and a hyperbolic tangent for other activations. The shorthand description is $R(64) - R(64) - D(512) - Sm$.

3) HYBRID NETWORKS AND DEEPCONVLSTM [1], [6]

This is a combined architecture consisting of several convolutional layers and LSTM layers. In [1], the author calls the model the Hybrid Network but name it DeepConvLSTM in [6]. Both models use convolutional layers with 1D kernels. The shorthand descriptions are $C(50) - R(27) - R(27) - D(512) - Sm$ and $C(64) - C(64) - C(64) - C(64) - R(128) - R(128) - Sm$.

4) DilatedSRU NETWORK [26]

This is a novel model for human activity recognition that introduces a dilated convolutional layer to avoid the information loss caused by pooling and padding operations. In addition, a novel RNN model called the dilatedSRU is proposed to model the temporal dependencies at different time scales.

The results of our model and the mentioned recognition models are listed in Table 1. We highlight the best score in bold. In terms of performance, the proposed model achieves the highest scores on both datasets. Our proposed model outperforms the other models with a 92.2% weighted $F1$ score on the Opportunity dataset and achieves the best performance with a 78.4% on the UniMiB-SHAR dataset. The DilatedSRU network also achieves a F_w score above 92%, but we find that the dilated convolutional operation reduces the time efficiency, which is an important aspect of online real-time sensor-based human activity recognition. To further

TABLE 1. Weighted $F1$ score performances of different recognition models on the Opportunity and UniMiB-SHAR datasets.

	UniMiB-SHAR	Opportunity
Performance	F_w	F_w
1D CNN	0.7429	0.9019
LSTM	0.7082	0.9116
Hybrid	0.7365	0.9156
DeepConvLSTM	0.776	0.915
DilatedSRU	-	0.9207
the proposed model	**0.784**	**0.922**

illustrate the effectiveness of our network, we conducted experiments on both tasks of the Opportunity challenge, either including or ignoring the Null class. The performances are shown in Table 2. Our proposed model outperforms DeepConvLSTM on all the tasks except modes of locomotion without the Null class. This result occurs because the locomotion task has fewer classes and is easier to recognize than the gesture recognition task; consequently, our novel modules cannot realize their potential.

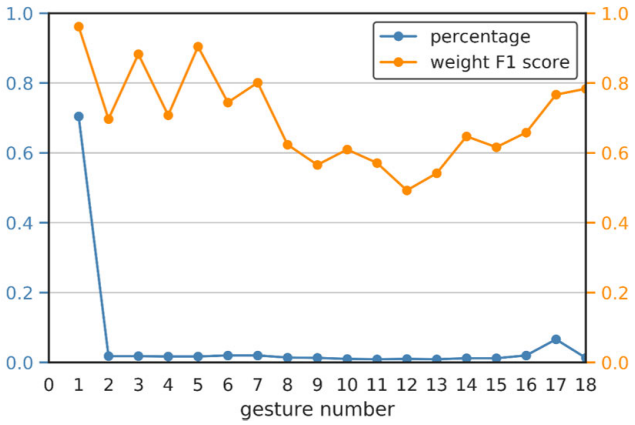


FIGURE 4. Weighted $F1$ scores on different classes of the Opportunity dataset. The blue line shows the percentage of each class in the dataset, and the orange line represents the performance on every class. The horizontal axis represents the number of gestures in the Opportunity dataset. 1: "Null", 2: "Open Door 1", 3: "Open Door 2", 4: "Close Door 1", 5: "Close Door 2", 6: "Open Fridge", 7: "Close Fridge", 8: "Open Dishwasher", 9: "Close Dishwasher", 10: "Open Drawer 1", 11: "Close Drawer 1", 12: "Open Drawer 2", 13: "Close Drawer 2", 14: "Open Drawer 3", 15: "Close Drawer 3", 16: "Clean Table", 17: "Drink from Cup", and 18: "Toggle Switch".

For the gesture recognition task, we depict the F_w for each gesture to reveal the influence of training data size on the recognition performance. As shown by the blue line in Fig. 4, the Opportunity dataset has a serious imbalance problem. The Null class (class 0) represents almost 70% of the items in the dataset, while the other classes rarely represent more than 2%. This phenomenon reveals that most of this dataset involves uninteresting human activities. Although the dataset is imbalanced, the performances on these classes are quite different. As shown by the orange line plots in Fig. 4, the Null class achieves the best performance (above 95%). Surprisingly, however, we find that the "Open Door 2" class (class 3) and "Close Door 2" class (class 5) achieve high performances (above 88%), while class 5 even achieves an F_w of 90%. This experiment shows good human activity recognition performance can be achieved from only a small amount of training data, which motivates us to seek even better deep learning models for HAR. We also compare our model's performance in this experiment with that of [26]. That model's worst performance on this dataset is below 40%, while our model's worst performance is approximately 50%, further demonstrating the effectiveness of our proposed model.

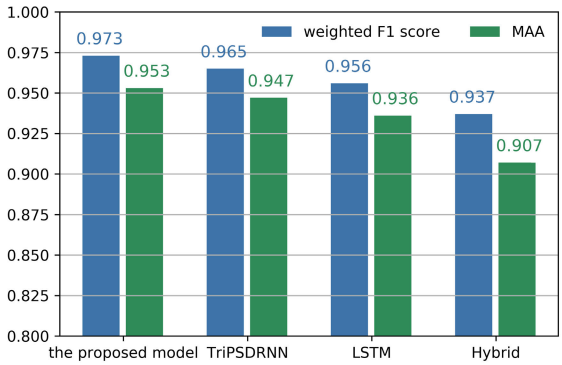


FIGURE 5. Five-fold cross-validation evaluation results of different models on the UniMiB-SHAR dataset.

To further examine the effectiveness of our proposed method, we performed a 5-fold cross-validation on the UniMiB-SHAR dataset and compared its results with the TriPSDRNN [36], LSTM [25] and Hybrid [1] models, as shown in Fig. 5. The proposed model achieves the highest weighted $F1$ score and MAA (97.3% and 95.3%, respectively). Compared with the TriPSDRNN, whose hand-crafted features include both time and frequency characteristics, our proposed network extracts discriminative features from acceleration data and outperforms TriPSDRNN by a margin of 0.8% on weighted $F1$ score and 0.6% on the MAA. We can also see that our proposed model outperforms the LSTM and Hybrid models, which further demonstrates the effectiveness of the two proposed modules.

B. EFFICIENCY

Because collecting data from body-worn sensors typically has high temporal resolution, human activity recognition is a time-critical issue. Therefore, we analyzed the recognition efficiency of our model compared to the DeepConvLSTM model from three aspects: the number of parameters, the computational complexity, and the recognition time per activity.

Table 3 presents the numbers and sizes of the parameters required by our model and by the DeepConvLSTM model. Both models have 8 fundamental layers: one input layer, four convolutional layers, two LSTM layers and a single fully connected layer. In addition, our network includes the two proposed modules; we also count their parameters. The last row of Table 3 lists the total number of parameters. Our model

TABLE 2. Weighted $F1$ scores of the proposed model and DeepConvLSTM on the Opportunity dataset for the gesture and modes of locomotion recognition tasks when including or omitting the Null class.

Method	Modes of Locomotion (No Null Class)	Modes of Locomotion	Gesture Recognition (No Null Class)	Gesture Recognition
DeepConvLSTM	0.930	0.895	0.866	0.915
the proposed model	0.930	0.90	0.873	0.922

TABLE 3. The numbers and sizes of parameters required by the DeepConvLSTM model and our proposed model. The final number of parameters depends on the number of classes in the classification task, denoted as n_c.

	our proposed model		DeepConvLSTM	
Layer	Size Per Parameter	Size Per Layer	Size Per Parameter	Size Per Layer
2	K: 1*64*1*1 b: 64	128	K: 64*5 b: 64	384
3-5	K: 64*64*3*1 b: 64	12,352	K: 64*64*5 b: 64	20,544
6	W_{ai},W_{af},W_{ac},W_{ao}: 4928*64 W_{hi},W_{hf},W_{hc},W_{ho}: 64*64 b_i,b_f,b_c,b_o: 64 W_{ci},W_{cf},W_{co}: 64 c: 64 h: 64	319,872	W_{ai},W_{af},W_{ac},W_{ao}: 4928*128 W_{hi},W_{hf},W_{hc},W_{ho}: 128*128 b_i,b_f,b_c,b_o: 128 W_{ci},W_{cf},W_{co}: 128 c: 128 h: 128	647,680
7	W_{ai},W_{af},W_{ac},W_{ao}: 64*64 W_{hi},W_{hf},W_{hc},W_{ho}: 64*64 b_i,b_f,b_c,b_o: 64 W_{ci},W_{cf},W_{co}: 64 c: 64 h: 64	8,448	W_{ai},W_{af},W_{ac},W_{ao}: 128*128 W_{hi},W_{hf},W_{hc},W_{ho}: 128*128 b_i,b_f,b_c,b_o: 128 W_{ci},W_{cf},W_{co}: 128 c: 128 h: 128	33,280
8	W: 64*n_c b: n_c	(64*n_c) + n_c	W: 128*n_c b: n_c	(128*n_c) + n_c
DCM	K: 2*(64*64*1*1) b: 2*64	8,320	-	-
MFAM	K: 3*(64*64*1*3)+192*3*1*1 b: 3*64+3	37,635	-	-
Total	**386,755** + (64*n_c) + n_c		**996,800** + (128*n_c) + n_c	

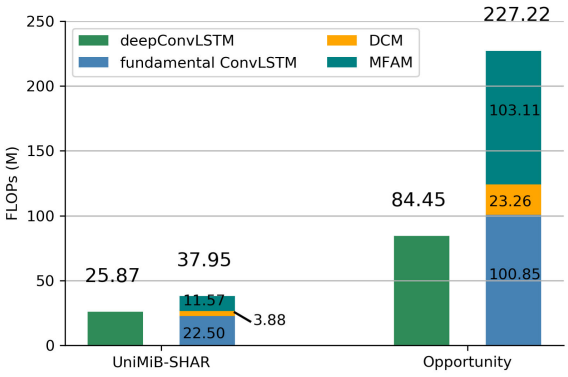


FIGURE 6. FLOPs of the proposed network and deepConvLSTM on the Opportunity dataset and UniMiB-SHAR dataset.

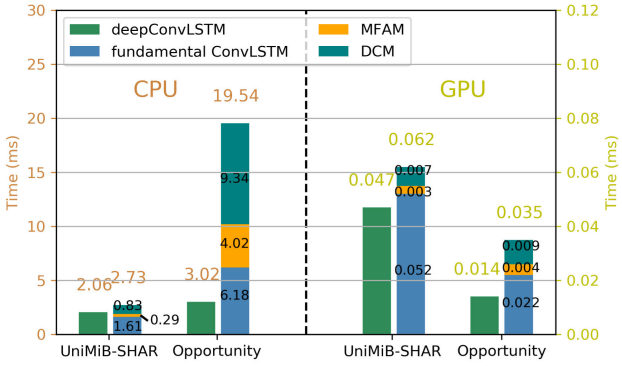


FIGURE 7. Inference times for CPU and GPU versions of the proposed network and deepConvLSTM on the Opportunity and UniMiB-SHAR datasets.

requires approximately 2 times fewer parameters than those of DeepConvLSTM, which indicates that the proposed network is more suitable for devices with limited memory.

To further reveal the computational complexity and the time efficiency, we calculated the floating-point operations (FLOPs) and the inference time of our network and deepConvLSTM. Measuring FLOPs can help reveal the computational complexity, while inference time is the time that models require to recognize a data segment on a computing device. We conducted this experiment on an Intel(R) Xeon(R) Gold 6130 CPU and a RTX 2080 Ti GPU. The results are shown in Fig. 6 and Fig. 7. As shown in Fig. 6, the total FLOPs are given at the top of each bar. The FLOP values for our proposed model are further individually divided into those used by the fundamental ConvLSTM, the DCM and the MFAM. Compared with the deepConvLSTM, our fundamental ConvLSTM (whose architecture is similar to that of the deepConvLSTM) achieves inconsistent performances on the

two datasets. This inconsistency is caused by the following two factors. On the one hand, our proposed model uses a 1×1 convolutional layer and a padding operation, which helps align the output size of the multilayer features in the 3×1 and 1×3 convolutional layers. However, the input data matrix of the Opportunity dataset is larger than that of the UniMiB-SHAR dataset, and the FLOPs are more affected by the 1×1 convolutional layer and padding operation, which increases the number of FLOPs required by the fundamental ConvLSTM on the Opportunity dataset. On the other hand, the longer the input time steps of the data segment on the UniMiB-SHAR dataset, the smaller are the FLOPs of the fundamental ConvLSTM compared with those of the deepConvLSTM, because the LSTM layers require fewer parameters. Regarding the two modules, the MFAM FLOP values are larger than those of the DCM because the MFAM includes more convolutional layers. On CPU, the inference time performance is consistent with the results of the FLOPs. However, on the GPU, which is more suitable for parallel computing, the proportion of inference time spent in the DCM and MFAM modules is small. Overall, although our model's human activity recognition speed is slower than that of the DeepConvLSTM, our model is still fast enough to meet the application requirements because the sliding window lengths of the Opportunity dataset and the UniMiB-SHAR dataset are 800 *ms* and 3,000 *ms*, respectively.

C. MULTIMODAL FUSION ANALYSIS

Sensor-based human activity recognition often uses multimodal sensor data with many sensor channels, which makes device setups complex. It is important for a network designed for human activity recognition to be robust to variations of different modalities and different sensor channels. Therefore, we conducted experiments with various modalities and various sensor channels. In Fig. 8, we show the weighted $F1$ score of our model on the Opportunity dataset for different modalities. In this experiment we used all the classes in this dataset except the Null class to match the setting of [26]. The results show that when using only several accelerometers with 15 channels, our model achieves a performance of 74%, and it improves the F_w by 6% when using gyroscopes with 15 channels. The performance exceeded 85% when we combined accelerometers and gyroscopes. When magnetic sensors are also added, the F_w is decreased slightly. In addition, we tested using all sensors (113 channels), but the performance improved by only 3%. This result reveals that model performance is not linear based on the number of sensor modalities.

We also conducted experiments with different subsets of the 113 sensor channels on the Opportunity dataset. We used the minimal-redundancy maximal-relevance (mRMR) algorithm [37], which selects a sensor channel based on mutual information, to select different sensor channel subsets, and we set the channel number to 5, 10, 20, 50, 80, and 113. The experimental settings are the same as those used in the experiment with various modalities. The results are shown in Fig. 9. As expected, increasing the number of sensor channels can achieve better performances. The best performance occurs when using all the sensor channels. However, because the amount of redundant information also increases, the rate of classification performance growth decreases.

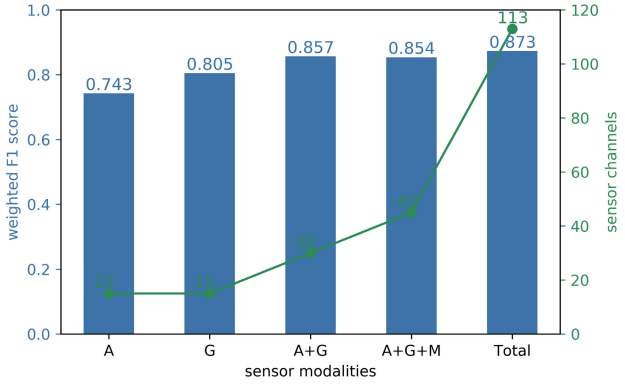


FIGURE 8. Performances using different sensor modalities on the Opportunity dataset. The blue histogram represents the weighted $F1$ scores by the proposed model when employing different sensor modalities. The green line represents the total number of sensor channels. "A", "G" and "M" represent accelerometer, gyroscope and magnetic data, respectively. The "Total" represents the complete Opportunity sensor set, which includes 113 channels.

D. HYPERPARAMETER EVALUATION

We conducted experiments on the influences of the four key hyperparameters in our network: sliding window length, kernel size of the convolutional layers, fusion mode of the DCM, and the number of convolutional layers. We performed 5-fold cross-validation for these hyperparameter evaluation experiments, which were conducted on the UniMiB-SHAR dataset.

1) SLIDING WINDOW LENGTH

To enable a fair comparison with the DeepConvLSTM, the default sliding window length on the Opportunity dataset was

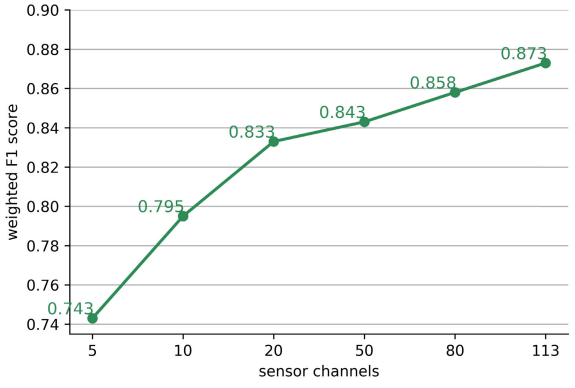


FIGURE 9. Performances using different numbers of sensor channels on the Opportunity dataset (selected by the mRMR algorithm).

set to 0.8 *s*. The UniMiB-SHAR dataset was sliced using a fixed-width sliding window of approximately 3 *s*. However, regarding different sliding window lengths, a too-short window is insufficient to extract effective features, while a too-long window leads to excessive amounts of redundant information. Therefore, we wanted to reveal the influences of various sliding window lengths. We conducted the experiments with length sequences of 0.4 *s*, 0.8 *s*, 1 *s*, and 1.5 *s* on the Opportunity dataset and 1.5 *s*, 2 *s*, 2.5 *s*, and 3 *s* on the UniMiB-SHAR dataset.

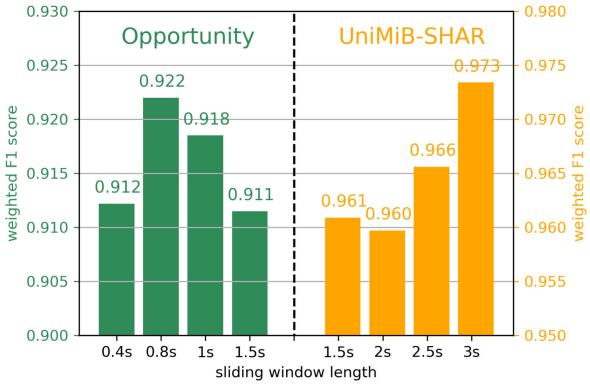


FIGURE 10. Weighted $F1$ scores of the proposed network on the Opportunity dataset and the UniMiB-SHAR dataset with different sliding window lengths. The green and orange histograms represent the classification performances for sliding window lengths of 0.4 *s*, 0.8 *s*, 1 *s*, and 1.5 *s* and 1.5 *s*, 2 *s*, 2.5 *s*, and 3 *s*, respectively.

Fig. 10 illustrates the performances under different sequence lengths on the Opportunity and UniMiB-SHAR datasets. Lengths of 0.8 *s* and 3 *s* achieve the best weighted $F1$ scores (92.2% and 97.3%) for the two datasets, respectively. For the Opportunity dataset, when the sliding window length is longer or shorter than 0.8 *s*, the performance begins to decrease. For the 0.4 *s* and 1.5 *s* cases, the weighted $F1$ scores fell below 91.5%. On the UniMiB-SHAR dataset, the performance decreases when the sliding window length is shorter than 3 *s*, and the worst performance (96.0%) occurs at a window length of 2 *s*. From these results we can observe that the recognition performance of the network for human activity recognition is strongly affected by the sliding window length setting.

2) KERNEL SIZES OF CONVOLUTIONAL LAYERS

We know that large convolution kernels for CNNs can enlarge receptive field; thus, intuitively they should extract better features. Using a square (N × N) convolution simultaneously captures local dependencies along the time and spatial domains for unimodal sensor data, but this setting requires high numbers of parameters. A two-layer convolution operation (a N × 1 convolution followed by a 1 × N convolution), can also capture local dependency along time and spatial domains but requires fewer parameters. Therefore, we conducted experiments to reveal the classification performances of our proposed model under different kernel sizes

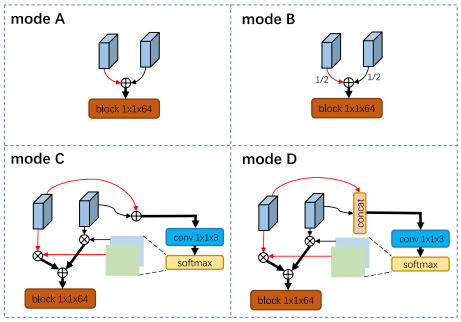


FIGURE 11. Different fusion modes of the DCM.

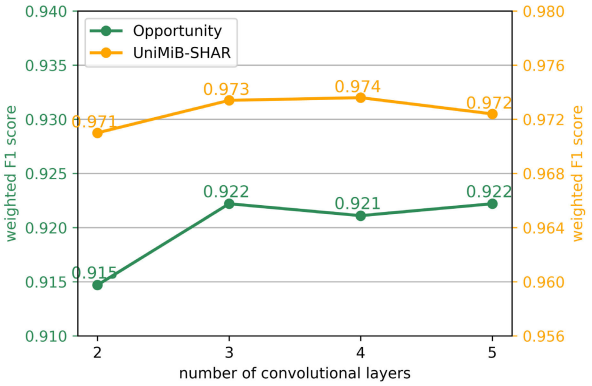


FIGURE 12. Classification performances of our proposed network on the Opportunity dataset and the UniMiB-SHAR dataset with different numbers of convolutional layers.

on the Opportunity dataset and the UniMiB-SHAR dataset. We embedded different kernel sizes in the fundamental ConvLSTM and the MFAM, which use 3 × 1 and 1 × 3 kernel sizes, respectively, in our proposed network. The MFAM uses a 1 × 3 kernel size on the UniMiB-SHAR dataset because it includes data only from a 3-axis accelerometer.

TABLE 4. Weighted $F1$ score performance of the proposed network on the Opportunity dataset and the UniMiB-SHAR dataset with different kernel sizes. 'F' and 'M' represent the fundamental ConvLSTM and the MFAM, respectively.

Opportunity		UniMiB-SHAR	
kernel size	F_w	kernel size	F_w
F: 3 × 1 M: 1 × 3	**0.922**	F: 3 × 1 M: 1 × 3	0.973
F: 3 × 3 M: 3 × 3	0.921	F: 3 × 3 M: 3 × 3	0.968
F: 5 × 1 M: 1 × 5	0.921	F: 5 × 1 M: 1 × 3	**0.974**
F: 5 × 5 M: 5 × 5	0.920	F: 7 × 1 M: 1 × 3	0.973

As Table 4 shows, we can observe that the kernel sizes (3 × 1 and 1 × 3) on the Opportunity dataset and the kernel sizes (5 × 1 and 1 × 3) on the UniMiB-SHAR dataset achieve the best weighted $F1$ scores (92.2% and 97.4%, respectively). From this experiment, we can make two inferences. First, the two-layer convolution operation is suitable for our proposed

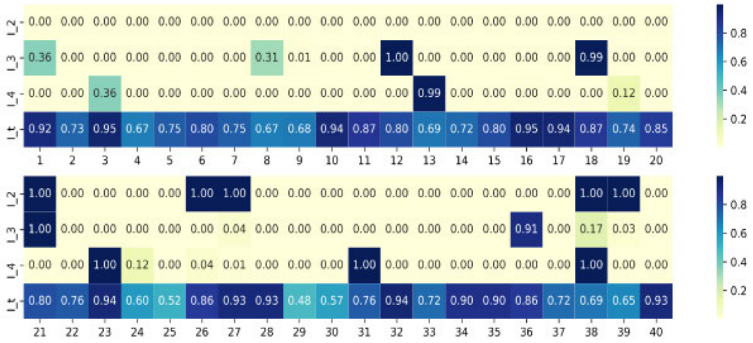


FIGURE 13. Probability estimations of I_1, I_2, I_3, I_t for the true class of 40 sample data on the Opportunity dataset (I_1, I_1, I_3, I_t are illustrated in Fig. 3). Each row represents the probability estimation for the corresponding true class of each I_i, and each column represents the probability estimation for each sample.

network and has fewer parameters. Second, compared with the Opportunity dataset, the longer input time steps of the UniMiB-SHAR dataset require a relatively larger kernel size to extract the most effective features along the temporal dimension.

3) FUSION MODE OF THE DCM

The fusion mode of the features extracted by multilayers networks affects their classification performances; their architectures are designed to aggregate multilayer features. Therefore, we conducted an experiment to investigate the influence of using different fusion modes in the DCM. Our proposed network uses mode A of Fig. 11, which also corresponds to Equation 1. For comparisons with mode A, we employ three other fusion modes. Mode B is an average weighted fusion mode similar to mode A. Mode C is the fusion method we use in the MFAM, and mode D is similar to mode C except for the concatenation operation.

TABLE 5. Weighted $F1$ score performance of the proposed network on the Opportunity and UniMiB-SHAR datasets when using different fusion modes in the DCM.

	Opportunity	UniMiB-SHAR
Performance	F_w	F_w
mode A	0.922	**0.973**
mode B	0.920	0.972
mode C	**0.923**	**0.973**
mode D	0.920	0.972

As Table 5 illustrates, the proposed network simultaneously achieves its best performances on both the Opportunity and UniMiB-SHAR datasets when using mode C (92.3% and 97.3%, respectively), illustrating the following three points. First, using a better fusion mode for the DCM improves the classification performance of our proposed network. Second, the results further prove the effectiveness of the aggregation method designed for the MFAM. Third, because the performance of mode D is lower than that of mode C on the two datasets, in our proposed model, the addition operation is more suitable for aggregating information than is the concatenation operation.

4) THE NUMBER OF CONVOLUTIONAL LAYERS

We also show how the performance of our model changes when using different numbers of convolutional layers. Because the DCM and MFAM require at least two 3×1 convolutional layers, we set the range for the number of layers to [2, 5]. As the results in Fig. 12 show, the F_w score improves by 0.7% and 0.2%, respectively, on the Opportunity dataset and the UniMiB-SHAR dataset when a new layer is added to the two-convolutional-layer model. The recognition performance was highest when using 3 and 4 convolutional layers on the Opportunity and UniMiB-SHAR datasets, respectively.

We conducted an ablation experiment to show the effectiveness of the two proposed modules on the final recognition performance on the two datasets. As listed in Table 6, we achieved F_w scores of 91.4% and 96.6%, respectively when employing the fundamental ConvLSTM (M_1). Then, when we added the DCM to the baseline model (M_2), the F_w scores improved to 91.8% and 97.0%, respectively, as listed in the second row in Table 6. Next, we added only the MFAM to the model, and the performance decreased slightly, as shown in the M_3 row. Finally, our complete network obtains F_w scores of 92.2% and 97.3% on the two datasets, respectively. These results demonstrate that the DCM and MFAM both help improve the recognition performance.

E. MULTILAYER FEATURE AGGREGATION ANALYSIS

To further clarify the effectiveness of the MFAM, we choose 40 sample data points and extracted feature maps from I_1, I_2, I_3, I_t (see Fig. 3). Then, we separately input these feature

TABLE 6. Weighted $F1$ scores on the Opportunity and the UniMiB-SHAR datasets for our complete proposed model and three ablated models: the fundamental ConvLSTM and the fundamental ConvLSTM with either the DCM or the MFAM.

Model	fundamental ConvLSTM	DCM	MFAM	Oppo	Uni
M_1	√			0.914	0.966
M_2	√	√		0.918	0.970
M_3	√		√	0.917	0.969
the proposed model	√	√	√	**0.922**	**0.973**

maps into the two LSTM layers and the fully connected layer of our model to obtain output probability estimations for each sample representation. Each column of Fig. 13 reflects one data sample, and the four rows for each data point represent the probability estimations of I_1, I_2, I_3, I_t for the true class. We intuitively observe from Fig. 13 that multilayer feature aggregation (I_t) can select the most discriminant weighted feature maps and prevent incorrect predictions. For example, the probability of I_1, I_2, I_3 for No.16 is approximately 0%; however, the multilayer feature aggregation (I_t) extracts fused information from the other three feature maps and obtains a probability of 95%. Although multilayer feature aggregation may reduce the probability of the true class for some samples (such as No.18 and No.23), it still produces a reasonable correct probability for these samples.

VI. CONCLUSION

For multimodal sensor-based human activity recognition applications, most existing works use large convolution kernels and employ only the top-level information extracted by the bottom-up feedforward process. These operations often achieve only low recognition efficiency and ignore considerable amounts of rich information. In this paper, we propose a novel hybrid network by designing a fundamental ConvLSTM pipeline with a dense connection module (DCM) and a multilayer feature aggregation module (MFAM). The DCM promotes information flow in the model and ensures that each layer can directly access the gradients of the loss function. The MFAM collects the features of each layer and aggregates them according to their importance. Compared with DeepConvLSTM and other state-of-the-art methods, our proposed network achieves the best performances on the Opportunity and UniMiB-SHAR datasets. To fully reveal the effectiveness of our network, we conducted experiments to test its efficiency and the effects of multimodal fusion and hyperparameter settings on the two datasets. In addition, we show the performances of the two modules on the two datasets separately and visualize the probability output by the MFAM for some samples.

In future work, to verify the robustness and practicality of our model, we plan to conduct experiments on additional datasets and apply our modules to other state-of-the-art deep learning models.

REFERENCES

[1] F. Li, K. Shirahama, M. Nisar, L. Köping, and M. Grzegorzek, "Comparison of feature learning methods for human activity recognition using wearable sensors," *Sensors*, vol. 18, no. 3, p. 679, 2018.

[2] 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 *Proc. 23th Int. Archit. Comput. Syst. Conf.*, Feb. 2010, pp. 1–10.

[3] S. Mazilu, U. Blanke, M. Hardegger, G. Tröster, E. Gazit, and J. M. Hausdorff, "GaitAssist: A daily-life support and training system for parkinson's disease patients with freezing of gait," in *Proc. 32nd Annu. ACM Conf. Hum. Factors Comput. Syst. CHI*, 2014, pp. 2531–2540.

[4] A. Tolstikov, X. Hong, J. Biswas, C. Nugent, L. Chen, and G. Parente, "Comparison of fusion methods based on DST and DBN in human activity recognition," *J. Control Theory Appl.*, vol. 9, no. 1, pp. 18–27, Feb. 2011.

[5] J. Hong and T. Ohtsuki, "A state classification method based on space-time signal processing using SVM for wireless monitoring systems," in *Proc. IEEE 22nd Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2011, pp. 2229–2233.

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

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

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

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

[10] S. Ha and S. Choi, "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 381–388.

[11] D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist, and A. Holzinger, "Human activity recognition using recurrent neural networks," in *Mach. Learn. Knowl. Extraction*, vol. 2017, pp. 267–274.

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

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015, *arXiv:1512.00567*. [Online]. Available: http://arxiv.org/abs/1512.00567

[14] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Forster, G. Troster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. D. R. Millan, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proc. 7th Int. Conf. Networked Sens. Syst. (INSS)*, Jun. 2010, pp. 233–240.

[15] D. Micucci, M. Mobilio, and P. Napoletano, "UniMiB SHAR: A new dataset for human activity recognition using acceleration data from smartphones," 2016, *arXiv:1611.07688*. [Online]. Available: http://arxiv.org/abs/1611.07688

[16] M. Janidarmian, A. Roshan Fekr, K. Radecka, and Z. Zilic, "A comprehensive analysis on wearable acceleration sensors in human activity recognition," *Sensors*, vol. 17, no. 3, p. 529, 2017.

[17] L. Xie, J. Tian, G. Ding, and Q. Zhao, "Human activity recognition method based on inertial sensor and barometer," in *Proc. IEEE Int. Symp. Inertial Sensors Syst. (INERTIAL)*, Mar. 2018, pp. 1–4.

[18] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[19] M. Panwar, S. Ram Dyuthi, K. Chandra Prakash, D. Biswas, A. Acharyya, K. Maharatna, A. Gautam, and G. R. Naik, "CNN based approach for activity recognition using a wrist-worn accelerometer," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 2438–2441.

[20] B. Pourbabaee, M. J. Roshtkhari, and K. Khorasani, "Deep convolutional neural networks and learning ECG features for screening paroxysmal atrial fibrillation patients," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 12, pp. 2095–2104, Dec. 2018.

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

[22] K. Wang, J. He, and L. Zhang, "Attention-based convolutional neural network for weakly labeled human activities recognition with wearable sensors," 2019, *arXiv:1903.10909*. [Online]. Available: http://arxiv.org/abs/1903.10909

[23] M. Edel and E. Koppe, "Binarized-BLSTM-RNN based human activity recognition," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2016, pp. 1–7.

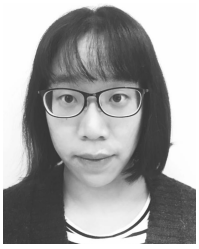
[24] Y. Guan and T. Ploetz, "Ensembles of deep LSTM learners for activity recognition using wearables," 2017, *arXiv:1703.09370*. [Online]. Available: http://arxiv.org/abs/1703.09370

[25] M. Inoue, S. Inoue, and T. Nishida, "Deep recurrent neural network for mobile human activity recognition with high throughput," *Artif. Life Robot.*, vol. 23, no. 2, pp. 173–185, Jun. 2018.

[26] R. Xi, M. Li, M. Hou, M. Fu, H. Qu, D. Liu, and C. R. Haruna, "Deep dilation on multimodality time series for human activity recognition," *IEEE Access*, vol. 6, pp. 53381–53396, 2018.

[27] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification," *Frontiers Comput. Sci.*, vol. 10, no. 1, pp. 96–112, Feb. 2016.
[28] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.
[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167
[30] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. conf. Artif. Intell. Statist.*, Jun. 2011, pp. 315–323.
[31] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and understanding recurrent networks," 2015, *arXiv:1506.02078*. [Online]. Available: http://arxiv.org/abs/1506.02078
[32] L. Pigou, A. van den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *Int. J. Comput. Vis.*, vol. 126, nos. 2–4, pp. 430–439, Apr. 2018.
[33] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.
[34] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster, "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," in *Wireless Sensor Networks*. Berlin, Germany: Springer, 2008, pp. 17–33.
[35] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS) Workshop*, 2017.
[36] X. Li, Y. Wang, B. Zhang, and J. Ma, "PSDRNN: An efficient and effective HAR scheme based on feature extraction and deep learning," *IEEE Trans. Ind. Informat.*, early access, Jan. 23, 2020, doi: 10.1109/TII.2020.2968920.
[37] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

TIANQI LV received the B.E. degree in electronic information engineering from Yanshan University, Hebei, China, in 2015. He is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include deep learning and artificial intelligence.



XIAOJUAN WANG received the Ph.D. degree in electronic science and technology from the Beijing University of Posts and Telecommunications. She is currently an Associate Professor with the School of Electronic Engineering, Beijing University of Posts and Telecommunications. Her research interests include deep learning, complex networks, and human gesture recognition.

LEI JIN received the B.E. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include complex networks and deep learning.

YABO XIAO received the B.E. degree from Jilin University, China, in 2017. He is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications. His research interests include computer version and deep learning.

MEI SONG is currently a Professor with the School of Electronic Engineering, Beijing University of Posts and Telecommunications. She has conducted research and development for key technologies in future communication and integration networks, mobile Internet, integrated circuit and communication systems, next-generation networks, the Internet of Things, and modern service science. Under her leadership, the ICN & CAD Laboratory has planned and undertaken high-level scientific research projects for the National Science and Technology Support Plan, the National 863 Plan, the National Natural Science Foundation Project, the International Cooperation Project of Ministry of Science and Technology, China Mobile, China Unicom, and other enterprise cooperation projects. She is a member of the Teaching Steering Committee of the Ministry of Education, the Semiconductor and Integration Technology Branch, China Electronics Society, and the Device Professional Group of the Microcomputer Professional Committee, China Computer Society.