IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 12, 2019, accepted May 30, 2019, date of publication June 5, 2019, date of current version June 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2920969

Human Activity Recognition Based on Motion Sensor Using U-Net

YONG ZHANG[1], ZHAO ZHANG[1], YU ZHANG[1], JIE BAO[1], YIFAN ZHANG[2], AND HAIQIN DENG[3]

[1]School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]School of Information, Production and Systems, Waseda University, Tokyo 169-8050, Japan
[3]AIdong Super AI (Beijing) Co., Ltd., Beijing 100007, China

Corresponding author: Yong Zhang (yongzhang@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61871046, and in part by the Fab. X Artificial Intelligence Research Center, Beijing, China.

ABSTRACT Traditional human activity recognition (HAR) based on a motion sensor adopts sliding window labeling and prediction. This method faces the multi-class window problem, which mistakenly labels different classes of sampling points within a window as a class. In this paper, we propose a novel HAR method based on U-Net to overcome the multi-class problem, performing activity labeling and prediction of each sampling point. The motion sensor data collected from the wearable sensors are mapped into an image with the single-pixel column and multi-channel, and then, it is input into the U-Net network to complete the pixel-level activity recognition function. We design a complete HAR framework based on U-Net to realize the dense prediction of motion sensor data, including data preprocessing, dense prediction, and post analysis. In order to further improve the dense prediction performance, we propose the post-correction algorithm for the dense prediction results on the basis of the activity misalignment analysis. The extensive experimental results demonstrate that our U-Net method performs better than the traditional machine learning and deep learning methods based on the sliding window prediction. And it also outperforms full convolutional network (FCN), SegNet, and Mask R-CNN based on the dense prediction on the four datasets. Moreover, it also shows the better robustness and excellent performance of recognition on the short-term activities and minority classes. We release a new dataset named Sanitation, which includes seven types of daily work activity data of sanitation workers to evaluate the HAR algorithm's performance.

INDEX TERMS Human activity recognition, U-Net, neural networks, deep learning.

I. INTRODUCTION

Human activity recognition (HAR) is the key technology of human-computer interaction and human activity analysis. The basic task of HAR is to select the appropriate sensor and deploy it to monitor and capture the user's activity [1]. HAR can be divided into two categories: video-based HAR and sensor-based HAR. With the wide use of portable and wearable sensors in our daily life, HAR based on sensor data has become a research hotspot, the HAR systems have been used in sleep state detection [2], behavior monitoring [3], [4], health monitoring [5], smart home [6], medical care [7], [8] and so on.

Data collected from the portable and wearable sensors are usually time series data. Human activity recognition for time series is a complex process, which usually involves the following steps. First, preprocess and segment the time series data, extract the features of the data, and then classify by using the classification algorithm. It requires manual extraction of features for human activity recognition based on traditional machine learning methods. With the development of deep learning, deep learning has been widely used in HAR [9], which can automatically learn and extract features without the more complicated steps of manual feature extraction. The workload of feature engineering is greatly reduced by automatically learning and extracting features [10], [11]. Therefore, deep learning based HAR is superior to traditional machine learning method, in which Convolutional Neural Network (CNN) is often used to analyze simple and complex activities.

Continuous segmentation of input sensor sequence data is a challenging task since the duration of human activities is different and the exact boundaries of activities are difficult to define. At present, both the machine learning-based

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

and the deep learning-based HAR methods use fixed sliding window technology to divide the sensor signal into fixed-length windows and label all samples in the window with the labeling strategies. Then the classification algorithm is applied to generate a predicted label for all samples in each window [12]. There are two common sliding window labeling strategies [13], one is to select the most frequent sample class in the window as the label of the window, the other is to select the sample class of the last time step in the window as the label of the window. Considering that all samples of a window may not always share the same label, some samples will be mislabeled. Both the two methods may lead to incorrect labeling, thereby reducing the recognition accuracy. This problem caused by fixed sliding window segmentation and labeling is called multi-class window problem [14]. Multi-class window problem is a common problem in HAR based on the motion sensor, which has a significantly negative impact on the recognition of short-term activity sequences. It makes short-term activity classification challenging. In order to improve the recognition accuracy, the common method is to apply a small-size window to segment time series data [15], which is time-consuming by sacrificing speed in exchange for accuracy.

The most obvious way to solve the multi-class window problem is to label and predict the activities based on sampling points directly. Therefore, we adopt dense labeling by labeling every sampling point instead of sliding window labeling so as to preserve the correct label information for each sample and improve the recognition accuracy of the classifier. The maximum pooling operations in the traditional CNN architecture result in reducing the resolution of top-level output and a size mismatch between the output and the input. Due to the size mismatch, the existing HAR model based on the CNN architecture cannot achieve dense prediction for each sampling point's label in the time series data, only suitable for sliding window prediction. In recent years, there are several new deep neural network architectures emerged in the field of image semantic segmentation, such as the Full Convolutional Network (FCN) [16], U-Net [17], SegNet [18], Mask R-CNN [19]. These architectures can realize the image pixel level classification, by introducing the up-sampling operation, thus achieving the top level output with the same resolution as the input, namely the size of input matching with the output. U-Net network is a more efficient network for the image pixel level classification through merging the information of the up-sampling and down-sampling block, compared with the other architectures [17]. And multi-channel time series data can be regarded as a single pixel column, multi-channel image, we propose the human activity recognition algorithm based on U-Net, and realize that the input sensor data can be predicted densely based on the sampling points, so as to solve the multi-class problem caused by the fixed sliding window labeling.

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

- We propose a novel HAR framework based on U-Net to overcome the multi-class problem. To the best of our knowledge, U-Net network is applied to HAR for the first time, which includes the down-sampling and up-sampling operations and achieves dense prediction by predicting each sampling in the time series data.
- In order to further improve the dense prediction performance, we propose the post-correction algorithm for the dense prediction results on the basis of the activity misalignment analysis.
- To evaluate the proposed HAR method based on U-Net, we also apply the existing image semantic segmentation methods to HAR based on dense prediction as comparative experiments for the first time, including SegNet and Mask R-CNN. Different from the comparative methods, the U-Net combines the shallow and deep network information to offset the information loss caused by the down-sampling operation, showing better robustness and recognition performance on the short-term activities and minority classes.

What's more, a new dataset named Sanitation is released to evaluate the HAR algorithm's performance and benefit the researchers in this field, which collects seven types of daily work activity data from sanitation workers.

The structure of the rest of the paper is as follows. Section II provides an overview of HAR methods. The HAR algorithm based on U-Net is proposed in Section III. The experimental results and analysis are presented in Section IV. The conclusion is given in Section V.

II. RELATED WORK

For the early research of HAR, the feature is manually extracted from the motion sensor data. Then, the extracted features are classified by various classification algorithms. The current machine learning algorithms for HAR can be divided into two categories: the classification based on discriminative model and the classification based on the generative model [20].

The classification method of HAR based on discriminative model mainly includes Support Vector Machine (SVM), Decision Tree (DT), k-Nearest Neighbor (kNN) and Artificial Neural Network (ANN). In [21], He and Jin extracted the autoregressive coefficients of the accelerometer data as the features of activity recognition and used SVM to classify human activities, which achieved good recognition performance on running, standing, jumping and walking. Fan *et al.* [22] used the built-in accelerometer to classify five activities and constructed a location-independent activity recognition model based on the DT algorithm. Khan *et al.* [23] established a hierarchical scheme for the classification of 15 specific activities, in which the upper layer uses the autoregressive model of the acceleration signal to generate the augmented feature vector and the lower layer processes the eigenvector of the triaxial accelerometer data through Linear Discriminant Analysis (LDA) and ANN.

Preece S J et al. constructed the nearest neighbor classifier to classify and analyzed the daily activities based on acceleration data. They adopted a robust individual-based cross-validation method. The classification accuracy on the best feature set reached 95% [24].

The classification methods of HAR based on generative model mainly include HMM and naive Bayes method. Lester J et al. proposed a dynamic activity recognition method, capturing temporal regularity and smoothness by HMM [25]. Lee S et al. proposed a HAR algorithm based on the semi-Markov random domain for accelerometer data, and it worked well for complicated activities like eating and driving a car [26]. Long X et al. proposed a Bayesian classification algorithm for human physical activity recognition based on acceleration data, which are collected from a single tri-axial accelerometer placed on the waist, and it used Principal Component Analysis (PCA) to reduce the dimension of the feature vector. The recognition performance is better than that of DT [27].

In recent years, deep learning has made great achievements in static image feature extraction and has been gradually extended to the study of time series data. The deep learning methods for HAR can be summarized into three categories. The first category is through using Convolutional Neural Network to automatically extract features from sensor data for recognition. Song-Mi Lee et al. proposed a One-Dimensional CNN to identify human behavior for triaxial acceleration sensors collected by smartphones, which input a vector magnitude data transformed by the raw accelerometer data, and the proposed 1D CNN-based method which reduced the possible rotational interference present in the raw data, showing the effectiveness compared with the baseline random forest approach [28]. Panwar M et al. investigated a deep learning framework for predicting the arm motion in daily activity by using a hand-mounted triaxial accelerometer. CNN is used to automatically extract useful features. The average recognition accuracy is better than clustering, LDA and SVM method [10]. The second category is to use Recurrent Neural Network (RNN) to capture the time dependence of sensor data. Guan Y and Ploetz proposed a HAR model based on Ensembles of deep Long Short Term Memory, which has achieved a good recognition effect on Opportunity, PAMAP2 and Skoda datasets [29]. In [30], Edel and Köppe proposed a HAR model based on a Binarized Long Short-Term Memory Network (B-BLSTM-RNN), processing sensor data gathered from different positions and keeping invariant to transformations and distortions of the input patterns. The third category is to use a hybrid model to identify human activity. Ordonez and Roggen D proposed a deep framework of HAR based on CNN and LSTM hybrid neural network. The accuracy of recognition on the Opportunity and Skoda dataset is higher than that on the previous report by 9%. It is suitable for multimodal wearable sensors and can accurately model the feature of real-time dynamic changes without using professional knowledge to design features [13]. NY Hammerla and Shane Halloran et al proposed a HAR scheme based on deep convolution and recurrent hybrid model for wearable sensor data [31].

More recently, Rui Yao et al. proposed a human activity recognition algorithm based on FCN [14], which realized the dense prediction of human activity sequences from wearable devices and conducted extensive experiments on three datasets. It achieved 88.7% with weighted F1-measure on Opportunity Locomotion dataset, 59.6% on Opportunity Gesture dataset, 89.3% on subject 1 for Hand Gesture dataset, 88.3% on subject 2 for Hand Gesture dataset, and 79.0% on the self-collected Hospital dataset.

Different from the above work, the U-Net based on HAR algorithm predicts the labels of each sample in the input time series data precisely so as to overcome the multi-class window problem existing in the sliding window method.

III. HAR FRAMEWORK BASED ON U-NET

A. DATA PREPROCESSING

In this part, the considerable details regarding the conduct of data preprocessing are offered, including data acquisition and standardization, dense labeling, and generating subsequences.

1) DATA ACQUISITION AND STANDARDIZATION

In our work, the experimental raw data are collected from the wearable sensor devices, and the obtained two-dimensional time series data is denoted as $X_{L,C} = \{(x_{11}, x_{12}, \ldots, x_{1C}), (x_{21}, x_{22}, \ldots, x_{2C}), \ldots, (x_{L1}, x_{L2}, \ldots, x_{LC})\}$, where L is the length of time series data and C is the number of sensor channels. Considering that the classifier's performance will deteriorate when the raw data distribution does not conform to the standard normal distribution with wide fluctuations. Then, we use z-score normalization to standardize the raw data:

$$x'_{ij} = \frac{x_{ij} - E(x_j)}{\sqrt{D(x_j)}} \tag{1}$$

where, x_{ij} represents the sampling point in the time series data, $E(x_j)$ represents the mean value of all the sampling points data under the j-th channel, and $\sqrt{D(x_j)}$ represents the standard deviation of all the sampling points data under the j-th channel.

2) DENSE LABELING

As a terminology, the term "dense labeling" is usually used for image semantic segmentation. The concept of dense labeling was introduced into human activity recognition to distinguish recognition methods based on sliding windows by some researchers [14]. The traditional sliding window labeling method assigns all samples in the subwindow with one label, but the samples within one subwindow may not share the same label, then the multi-class window problem will occur, as shown in Fig.1. Dense labeling means that the human activity is recognized on a sample-by-sample basis instead of a sliding window. To avoid the multi-class window problem, we adopt the dense labeling method, providing a single label

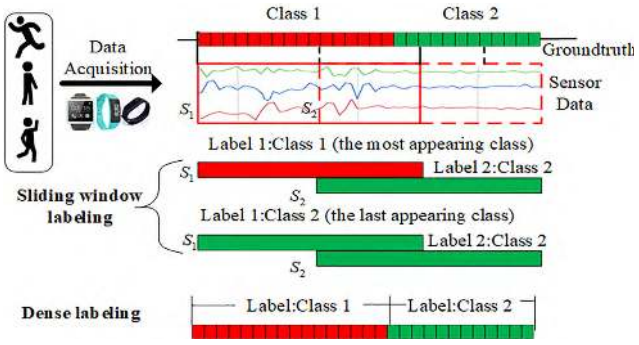


FIGURE 1. Sliding window labeling and dense labeling.

for each sample in the time series data, that is, labeling each timestamp rather than each subwindow. The dense labeling result is denoted as $Y_L = \{a_1, a_2, \ldots, a_i, \ldots, a_L\}$, where a_i represents the i-th sample's activity class. The dense labeling result has the same length of the time series data.

Fig.1 describes the two different sequence labeling methods: sliding window labeling and dense labeling. Red solid line frame represents the current sliding window and red dashed line frame represents the next one, denoted as S_1 and S_2, respectively. For example, according to the sliding window labeling strategy with the most appearing class, all data of S_1 are labeled as class 1. Whereas virtually S_1 contains class 1 and class 2 information. The sliding window labeling method results in the label information loss of class 2, learning incorrect context information, and lowering recognition accuracy. Labeling with the last appearing class also encounters the same problem.

3) GENERATING SUBSEQUENCES

Considering that the whole long time series data cannot be regarded as network input, the input continuous time series data has to be divided into several long subsequences. Each subsequence is regarded as a training sample. Due to the overlap between adjacent subsequences, the prediction of sample labels in overlapping parts may produce ambiguity. So we adopt the non-overlap fixed-length sliding segmentation, the $t-$th subsequence is expressed as $\{(x, y)|x[1, p_t : p_t+N, C], y[1, p_t : p_t+N, N_c]\}$, where N is the subsequence length, N_c represents the number of human activity classes, p_t represents the starting point of the t-th subsequence, the starting point of the $(t+1)$-th subsequence is presented as $p_{t+1}=p_t+N$.

It is worth noting that the generated subsequences of the densely labeled data here are different from traditional sliding subwindows. For each training subsequence, the output labels $\hat{y}_t$ has the same length of the input data x_t. However, for the training sliding subwindows, the length of the output labels is shorter than that of the input data since all samples in a subwindow just correspond to a single label.

B. DENSE PREDICTION MODLE BASED ON U-NET

In this section, we firstly apply U-Net for HAR based on dense prediction to predict every sampling point in time series data collected by the sensors. Firstly, we will give a brief introduction of dense prediction and how the U-Net works for it. Then, the architecture of the U-Net network for HAR model based on dense prediction is presented.

1) DENSE PREDICTION AND U-NET

In order to avoid the multi-class window problem, we adopt dense labeling instead of window labeling. Our method is to train the densely labeled data and predict each sampling point of the input sequence. The prediction of each sample's label is called dense prediction. The traditional CNN architecture is suitable for the window labeling prediction, which outputs a single prediction label for each segment of window data.

U-Net is the development and extension on CNN. Olaf Ronneberger et al. proposed an end-to-end U-Net network to achieve pixel level classification in microscopic images [17]. The U-Net structure consists of two paths. One is the encoder (down-sampling) path on the left side to capture contextual information, which is composed of several down-sampling blocks. Each block includes convolution, pooling, activation and so on. The other is the decoder (up-sampling) path on the right side for improving the resolution of the network layer gradually by up-sampling, which consists of the same number of the up-sampling blocks as the down-sampling block. By stacking the up-sampling block and the corresponding down-sampling block on the left side, the shallow and deep network information can be merged to offset the loss of information caused by the previous pooling operation. Finally, when the resolution of the network output layer and input layer is the same, the pixel level classification is realized.

As for the two-dimensional time series data that we input, the first dimension is the sample sequence time dimension. The second dimension is the number of channels of the sensor, that is, the number of sensor data axes. So it can be regarded as a single pixel column and multi-channel image which is input into the U-Net network. Each sampling point of the time series data can be predicted by means of the ability of the U-Net pixel level classification.

2) NETWORK ARCHITECTURE

The proposed dense prediction network architecture is adapted from the U-Net architecture presented by Olaf Ronneberger. Our network inputs two-dimensional time series data collected by the sensor. It can be regarded as a single pixel column and multi-channel image with the size of $(1, N, C)$, where N represents the sampling points of the input subsequence, C represents the number of the sensor's channels. Our network architecture consists of the encoder network and the decoder network. Each down-sampling block is composed of two convolutional layers with a kernel size of $1 \times S_c$ and a pooling layer with a kernel size of $1 \times S_p$. The size of the feature map remains the same by setting the

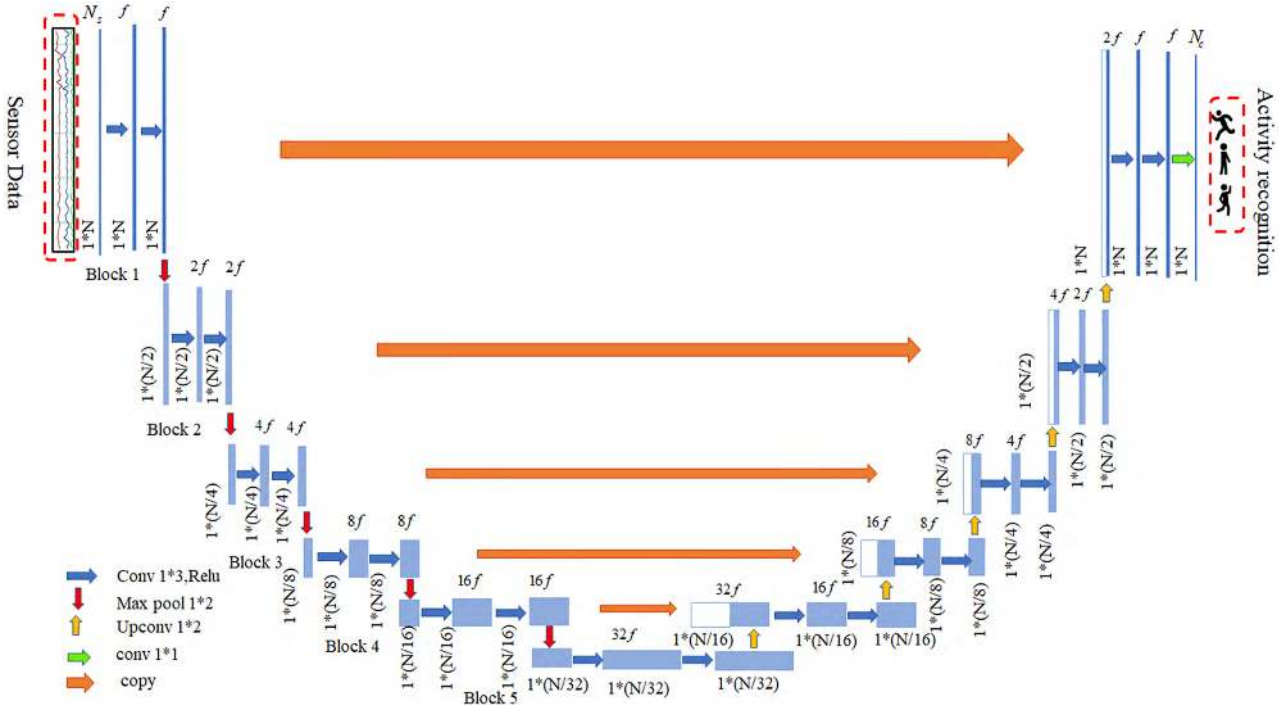


FIGURE 2. Network architecture of the U-Net for HAR.

appropriate filling after each convolution operation. And it is activated by the Restricted Linear Unit (ReLU) function. Then the size of the feature map is reduced by half after the pooling operation. The number of feature maps in each block of the encoder network is constant, but the number of feature maps at the next block is twice than that of the previous one. Each up-sampling block of the decoder network corresponds to the down-sampling block in the same level of the encoder network, which is mainly composed of one up-convolutional layer with a kernel size of $1 \times S_p$ and two convolutional layers with a kernel size of $1 \times S_c$. The output feature map of the convolution operation doubles the size of the output feature map so that it has the same resolution as the output feature map of the corresponding down-sampling block to achieve the merging. Then two convolution operations and the activation of the ReLU function are performed. After the last block of the decoder network, we add a Dropout layer to prevent overfitting. Finally, by mapping each feature vector to the corresponding class through a convolutional network with a kernel size of 1×1 and Softmax classifier, the dense prediction results of the input time sequence are obtained and the dimension is $(1, N, N_c)$.

Considering that the network output should have the same size as the input, each convolutional layer adopts the same padding. We adopt the same settings of the convolution kernel size S_c, the pooling kernel size S_p and its stride, and the filters' number f as in [17], where S_c is set to 3, S_p is fixed to 2, its stride is set to 2, and f is set to 32. The value range of the subsequence length N depends on the network structure. $N = k \cdot 2^L$, where $k \in N^+$, L is the number of the pooling layers which is also the number of U-Net blocks. The length of the subsequence can be set longer to speed up the training speed. The deeper the network is, the stronger the feature extraction ability will be, but the computational complexity will also increase. As for the parameter tuning of the network block number and the subsequence length, we select the optimal parameters based on the validation procedures on WISDM dataset. The subsequence length is set to 224, the network block number is set to 5. Considering the good robustness of the deep network, the network settings of our U-Net are consistent across all datasets. We will further discuss the influence of network depth and subsequence length on each dataset in Section IV-E. The proposed network structure is shown detailedly in Fig.2.

3) NETWORK TRAINING

The input feature map of each layer in the network can be regarded as a three-dimensional tensor of $1 \times w_l \times f_l$ size. w_l represent the length of the input feature map of layer l. f_l represents the number of the input feature maps in layer l. The i-th input data vector in a particular layer is denoted x_i. Then y_i denotes the corresponding output vector of the layer. It can be calculated by (2),

$$y_i = f(\{x_{i+i'}\}_{-\frac{k_i-1}{2} \le i' \le \frac{k_i-1}{2}}) \tag{2}$$

where the size of the convolution kernel is represented by k_i. $f(\cdot)$ represents the type of layer: matrix multiplication of convolutional layer, maximum pooling operation, up-convolution, nonlinear operation with activation function and etc. y_i is the output of layer l and the input of layer $l+1$.

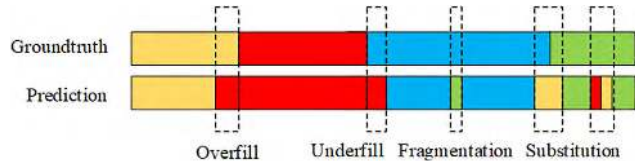


FIGURE 3. Different types of activity misalignments occurred in dense prediction results.

Its size can be expressed as $1 \times w_{l+1} \times f_{l+1}$, among which $w_{l+1} = (w_l - k_i)/s_i + 1$. The stride of kernel movement is represented by s_i. By performing proper filling operations on input feature maps, the output feature map can have the same resolution as the input feature map.

Given a set of input sequences and labels, the goal of network training is to estimate the appropriate parameters (W, b) of U-Net network to achieve accurate dense prediction. This is achieved by minimizing the loss values of all samples from each subsequence in the training dataset. The negative logarithmic likelihood function as the loss function can be written as follows:

$$l(x, y; W, b) = \sum_j^N l'(x, y_j; W, b) \tag{3}$$

where $l'(x, y_j; W, b) = -\log(p(y_j|x, W, b))$ represents the loss function of the j-th sample in a subsequence.

C. RESULTS ANALYSIS AND POST CORRECTION

To analyze and evaluate the continuous activity recognition on time series data, the activity-based misalignment measure is proposed in [32], where the HAR model focuses on the classification of the interesting classes and the NULL class, all the misalignment types are related to the wrong classification of the NULL class. Different from the evaluation of the HAR model in [32], the HAR model based on dense prediction is aimed at the multi-class recognition problem in which the NULL class is just regarded as a common class like the other activity classes, so that the misalignment types are related to the misclassification of all classes and are mainly divided into four types: Overfill, Underfill, Fragmentation, and Substitution, as shown in Fig.3. Overfill is defined as the misalignment errors that the recognition result of the next activity begins before the end of the previous activity. Underfill denotes the misalignment that the next activity has already started, but the dense prediction results still remain the previous activity class. Fragmentation measures the errors that misclassified as the other activities in the middle of one uninterrupted activity class. Substitution represents the errors of assigning wrong classes distinct from the previous and the next activity class, which usually occurs during the two activities transition period.

In order to further improve the performance of dense prediction model, we propose a post-correction algorithm to correct the misalignment errors in dense prediction results. Considering that the errors of Overfill and Underfill are difficult to be observed in the dense prediction results without knowing the groundtruth, our algorithm focuses on correcting the two types of errors: Fragmentation and Substitution. According to the continuity of activity, each activity frame in the dense prediction results can be represented as a continuous activity window $AW_i = \{y[B : F]|\forall y = a_i\}$, where B and F denote as the start and end index of activity class a_i. According to the duration of the action, when the length of the continuous activity window AW_i is less than the given threshold length ρ_L, AW_i may be the misaligned sequence, and we introduce the error correction window to represent the possible misalignment sequence, denoted as W. The error type of W is determined by the definitions of Fragmentation and Substitution. If W belongs to the Fragmentation errors, all recognition results output by U-Net model within W will be corrected to the activity class adjoined W directly by the post-correction algorithm. As for the Substitution error, our algorithm corrects all results within W to the adjacent activity class before or after the window. Considering that the boundary and adjacent prediction probability of W play an important role in determining the activity class for correcting Substitution errors, we define the boundary correlation coefficient for the correction window, denoted as:

$$\rho(p_i, p_j) = p_i \cdot p_j^T \tag{4}$$

where p_i and p_j represent the two adjacent probability vectors for each class in the boundary of W. When the correlation coefficient of the start boundary denoted as $\rho(p_{B-1}, p_B)$ is greater than that of the end boundary denoted as $\rho(p_F, p_{F+1})$, all results in the error correction window will be corrected into the adjacent activity class before the window, or otherwise corrected into the class after the window. The whole flowchart for the proposed post-correction algorithm is shown in Fig.4.

IV. EXPERIMENT

This section focuses on the experiments of HAR using U-Net. Compared with other algorithms on different datasets, the experiments evaluate the effectiveness of applying U-Net to HAR. Firstly, the datasets and experimental configurations used in the experiments are introduced. Then, a new unified evaluation index is proposed to suit the dense labeling scene. Finally, the performance of each algorithm on each dataset is compared synthetically.

A. DATASET

In this paper, we conduct experiments on the four datasets, including WISDM dataset [33], UCI HAPT dataset (HAPT) [34] UCI OPPORTUNITY Gesture dataset (OPP Gesture) [35] and the self-collected Sanitation dataset. All the datasets provide the densely labeled data which contain every sampling's label. The brief introduction of four datasets for HAR is shown in Table 1, and more information of the first three datasets can be found in their original papers.

TABLE 1. The brief introduction of five datasets for human activity recognition.

Dataset	Collection Device	Sampling Rate(Hz)	Activity
WISDM	Smartphone	20	Walk, Jog, Upstairs, Downstairs, Sit, Stand
UCI HAPT	Smartphone	50	Walk, Upstairs, Downstairs, Sit, Stand, Lie, Stand-to-Sit, Sit-to-Stand, Sit-to-Lie, Lie-to-Sit, Stand-to-Lie, Lie-to-Stand
OPP Gesture	Wearable sensors	30	Null, Open Door 1, Open Door 2, Close Door 1, Close Door 2, Open Fridge, Close Fridge, Open Dishwasher, Close Dishwasher, Open Drawer 1, Close Drawer 1, Open Drawer 2, Close Drawer 2, Open Drawer 3, Close Drawer 3, Clean Table, Drink from Cup, Toggle Switch
Sanitation	Smartwatch	25	Walk, Run, Bweep(broom sweep), Sweep, Clean, Dump, Daily

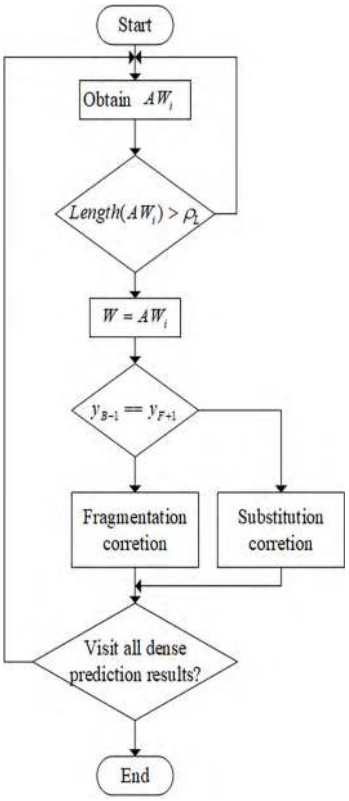


FIGURE 4. Flowchart of the post-correction algorithm.

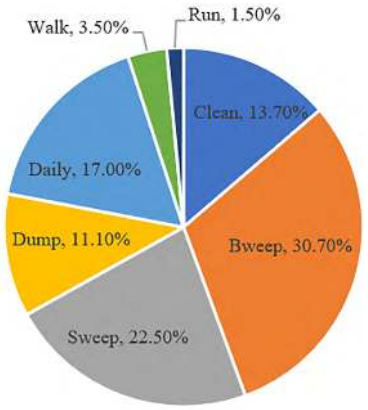


FIGURE 5. Percentage of activity of the Sanitation dataset.

In the following, we introduce the self-collected Sanitation dataset.[1] The self-collected Sanitation dataset is collected from the open environment. A triaxial accelerometer worn in a wrist smartwatch is used to collect seven types of daily work activity data of sanitation workers. The sampling frequency is 25 Hz. These seven types of activity are: Walk, Run, Sweep, Bsweep (sweep using a big broom), Clean, Dump and Daily activities (like sitting and smoking). The whole dataset contains 266555 samples, in which each sample contains X, Y, and Z three-axis acceleration values. The proportion of various types of activity samples is shown in Fig.5. We also provide the preprocessed dataset by dividing the whole time series data into 5026 windows by sliding window segmentation and generating 57 features for each window data. The time-domain and frequency-domain features are both extracted.

[1] https://ieee-dataport.org/documents/sanitation-dataset

B. EXPERIMENT CONFIGURATION

In order to train U-Net network, the learning rate is set to 0.001, the training batch size is set to 32, and the training epoch is set to 100. The dropout rate is set to 0.2. The network is optimized and updated by the Adam algorithm.

To evaluate the effectiveness of our U-Net method, we also conduct the comparative experiments of the HAR methods based on sliding window prediction and dense prediction, using machine learning and deep learning on the four datasets. We adopt the simple cross validation on the four datasets. The whole dataset is randomly divided into the training set and the testing set, in which the training set accounts for 70%. We split the validation set from the training set, and the split ratio is 0.3. The performance on the validation set is used to select the optimal parameters of traditional machine learning methods and tune the hyper-parameters of deep learning methods. Considering that the deep learning method can extract feature automatically, the raw data of the four datasets can be feed to the deep networks without

feature engineering. The optimal hyper-parameters of each deep network are determined by the validation procedures on WISDM dataset. The setting of each deep network is consistent in the four datasets. The implementations of all comparative methods are based on the descriptions in their papers and we have released the codes online.[2]

We compare the proposed U-Net method with the following two HAR baselines based on machine learning, namely SVM [34] and DT [22]. Machine learning-based HAR methods rely on feature engineering. Both WISDM dataset and HAPT dataset include the raw data and the extracted features data. 242 features from OPP Gesture dataset and 57 features from Sanitation dataset are extracted manually, including the means, variances and so on. The extracted features datasets are available online.[2]

- SVM. In this paper, SVM with radial basis function (RBF) kernel is used. The validation procedure on each dataset is used to tune the penalty parameter c on a scale from 0 to 10. We adopt the One-versus-One Strategy (OVO) to construct a set of binary classifiers and achieve multi-class classification on the four datasets.
- DT. In this baseline, we adopt the Gini impurity as the criteria and choose the best split at each node on the four datasets.

The three deep HAR methods based on the sliding window method are also conducted on the datasets as the comparative experiments, including CNN [10], LSTM [30], and CovLSTM [13]. Similar to the U-Net parameter setting, the optimal length of sliding subwindow is set to 96 according to the validation results.

- CNN. The CNN architecture consists of three 1D convolutional layers, three 1D max-pooling layers, and three fully connected layers in this paper. In each convolutional layer, the kernel size is fixed to 3 and its stride is set to 1. The number of filters in the convolutional layer in order is 64, 128, and 256. Each convolutional layer is followed by a max-pooling layer where the pooling size is 4 and its stride is 2. The same padding is adopted. At the top level, a softmax function is employed.
- LSTM. In this paper, the LSTM architecture contains two layers of forward recurrent LSTM units, followed by a fully connected layer and a softmax layer. The number of hidden units is set to 32.
- CovLSTM. This architecture combines two convolutional layers and two LSTM layers with 32 hidden units. In each convolutional layer, the kernel size is 5, its stride is 1, the number of filters is 32, and the same padding is used. At the top level, a fully connected layer and a softmax function are employed.

To evaluate the effectiveness of the proposed U-Net method, we also conduct comparative experiments with HAR methods based on dense prediction, namely FCN [14], SegNet [18], and Mask R-CNN [19], among which SegNet and Mask R-CNN are first applied to HAR. The comparative methods adopt the same data preprocessing procedure as our U-Net method, the subsequence length also keeps consistent with the U-Net's setting, except the Mask R-CNN.

- FCN. In this baseline, we reproduce a 2D convolutional layer and a 2D max-pooling layer four times, where the convolutional kernel size is 1×3, the pooling size is 1×4, the pooling stride is 1×2 and the number of convolutional filters is 32. A convolutional layer with kernel size of 1×1 is employed, followed by a 2D deconvolutional layer to implement the up-sampling operation and achieve the same length as the input.
- SegNet. The SegNet architecture consists of the encoder and the decoder network. The encoder network contains 13 convolutional layers which correspond to the first 13 convolutional layers in the VGG16 network, followed by a max-pooling layer with the pooling size of 1×4 and its stride of 1×3. Each convolutional kernel size is 1×3. The preserved max-pooling indices are utilized by MaxUnpooling operation in the decoder network. The decoder has the same architecture with the encoder, except the MaxUnpooling operation. The number of the filters in the first convolutional layer is set to 64.
- Mask R-CNN. Considering that the Mask R-CNN network contains two branches for object detection and image semantic segmentation respectively, among which the Mask branch is used for semantic segmentation of the proposal image regions with a size of 28×28, we apply the Mask branch network to HAR for the first time and realize dense prediction. The subsequence length is set to 28. The Mask network has a similar architecture with FCN and contains a stack of four consecutive convolutional layers with kernel size of 1×3, a deconvolutional layer with kernel size 1×2, and a convolutional layer with kernel size of 1×1. Before the Mask network, we construct a simple CNN network as the backbone, which includes a convolutional layer and a pooling layer with its pooling stride of 1×2. The number of filters is set to 32.

In addition, the threshold length ρ_L is set to 25, 25, 10 and 20 on the WISDM, Sanitation, OPP Gesture and HAPT dataset for the evaluation of the post-correction algorithm's performance.

The hardware used in the experiment is equipped with an NVIDIA GeForce GTX 1060 6G GPU. The programming language used in the experiment is Python3.6.4. The implementations of machine learning methods are based on sklearn. The deep learning frameworks we used are Tensorflow and Keras.

C. UNIFIED EVALUATION INDEX

The algorithms based on the sliding window method divide the time series data into several subwindows and predicts a single label for each subwindow, including SVM, DT, CNN, LSTM, and CovLSTM. The evaluation indexes for them are calculated by comparing the given label and the predicted

[2]https://github.com/zhangzhao156/Human-Activity-Recognition-Codes-Datasets

label for each subwindow. Whereas the evaluation indexes for dense prediction are obtained by comparing the given label and the predicted label for each sampling point in the whole time series data, including FCN, SegNet, Mask R-CNN, and U-Net. Due to the difference of the evaluation index calculation method between the sliding window-based algorithm and the dense prediction-based algorithm, a unified model evaluation index is defined. The dense labeling evaluation index calculation method is adopted uniformly. For the sliding window-based algorithm, each predicted label in the sliding window prediction results is assigned to all sampling points within the corresponding subwindow. Then the sampling points' labels in all subwindows are connected together and the sliding window prediction results are converted to the dense prediction results. Considering that the overlap between successive subwindows for the sliding window-based algorithm will lead to multi-class prediction conflicts when assigning the predicted label to the sampling points within the overlap, thus we set the overlap to 0 in all sliding window segmentation.

To evaluate the performance of the proposed dense prediction method based on U-Net and the comparison methods, the evaluation indexes are shown as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{5}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precsion} + \text{Re}\,call} \tag{8}$$

$$\text{Fw-Score} = \sum_i w_i \cdot \text{F1-Score}_i \tag{9}$$

where TP is true positive, TN is true negative, FP is false positive, and is false negative.w_i is the proportion of samples with label i.

D. EXPERIMENT RESULTS

We summarize the performance of the nine algorithms on four datasets and mark the highest score in bold, as shown in Table 1. The results clearly indicate that our U-Net method achieves the highest scores on all datasets in terms of Accuracy (Acc) and Fw-Score (Fw). U-Net outperforms the comparative methods. Most notably, for the OPP Gesture dataset with a large number of short-term activities, the accuracy of the U-Net method is up to 94.7%, 3.3% higher than the second highest score, while the accuracies of the sliding window-based HAR methods are all not ideal and lower than 90%. Even though LSTM and CovLSTM have proven to be of good performance in time series recognition and perform well in all the experiments except on OPP Gesture dataset, but U-Net still has a slight advantage over LSTM and CovLSTM. Furthermore, U-Net achieves far higher accuracy than LSTM and CovLSTM on OPP Gesture. The reason is that LSTM and CovLSTM adopt the sliding window method, which brings the multi-class window problem and reduces accuracy, especially in short-term activity recognition. Our U-Net method is based on dense prediction and avoids the multi-class problem, thus it further improves the recognition accuracy and is also more suitable for short-term activity recognition. For WISDM dataset with six kinds of simple long-term activities, our U-Net method still has an advantage over the other methods, 1.6% higher than the second highest accuracy. This proves that U-Net method is not only suitable for simple long-term activity recognition, but also more suitable for short-term activity recognition, showing strong robustness.

It can be found that although FCN, SegNet, and Mask R-CNN can also realize dense prediction, U-Net's performance is still better than that of these comparative methods. Most notably, FCN, SegNet, and Mask R-CNN perform better on OPP Gesture and HAPT dataset than the sliding window-based methods, which further proves the advantage of dense prediction. For WISDM and Sanitation dataset, the performance of FCN and Mask R-CNN are even worse than that of CNN. The FCN architecture used in [14] is to add one up-sampling layer after the traditional CNN architecture, the low-resolution feature map produced by CNN is directly linearly expanded so that the output has the same size as the input and realizes dense prediction. So does the Mask R-CNN used in this paper. Theoretically, the dense prediction results of FCN and Mask R-CNN are rough and not as good as U-Net and even worse than traditional CNN based on sliding window prediction. Our experimental result is consistent with this theoretical conclusion.

Compared with FCN, the SegNet architecture in this paper expands the output size of the decoder network gradually, by adopting multiple deconvolution operation and utilizing the memorized max-pooling indices from the corresponding encoder feature map. More useful information from the encoder network is utilized in the decoder network, thus, SegNet performs better than FCN on the four datasets. However, its performance is still worse than that of U-Net. Unlike the SegNet method for dense prediction, our U-Net utilizes the high-resolution feature map information from the encoder instead reusing pooling indices only, this indicates that the entire features from the encoder contain fine-grained information which plays an important role in dense prediction.

Remarkably, SVM sometimes performs better than the deep learning methods based on the sliding window prediction. For example, the performance of SVM is even better than that of CNN and CovLSTM on HAPT dataset, but the performance of SVM is not stable enough with inferior robustness.

In order to show the differences between these algorithms more clearly, it is necessary to calculate the evaluation index on its different classes of each dataset. For the sake of brevity and without losing generality, F1-score on different classes is selected as evaluation index only. Fig. 6 shows F1-Score on different classes of the four datasets for the nine algorithms.

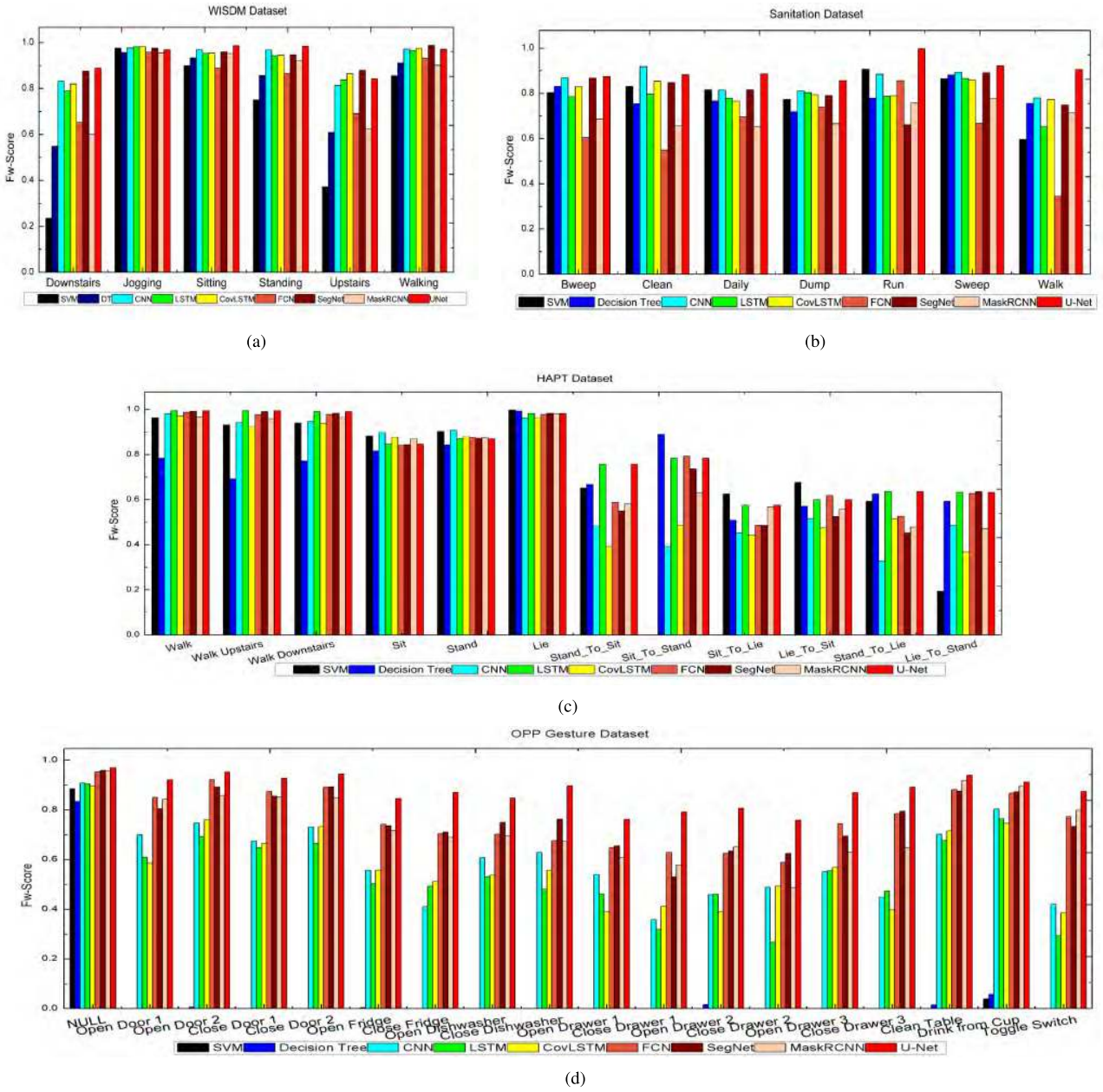


FIGURE 6. The F1-scores on different classes of the U-Net method and the comparison methods on four datasets. (a) WISDM Dataset. (b) Sanitation Dataset. (c) HAPT Dataset. (d) OPP Gesture Dataset.

In general, our U-Net method still performs better than other algorithms for specific classes classification. Most notably, as for Sanitation dataset, our F1-Score is 0.998 and 0.905 for Run and Walk, in contrasts, the second highest F1-Score for Run is 0.906 achieved by SVM, and that for Walk is 0.78 achieved by CNN. Considering that Run and Walk are both the minority classes of Sanitation dataset shown in Fig.5, which account for 1.5% and 3.5% respectively. This also proves that our U-Net method is also suitable for the unbalanced dataset with reliable performance on the minority classes. From Fig. 6 (d), the traditional machine learning methods can hardly recognize these short-term activities on OPP Gesture dataset with an F1-score of almost zero on these activities, leading to the lower recognition accuracy and Fw-score shown in Table 2. In contrasts, our U-Net method shows an absolute advantage in specific short-term activities recognition. For the six common activities of HAPT, there is a little difference of F1-Score between U-Net and other deep learning-based comparative methods, but for six types of transition activities of HAPT, U-Net has obvious advantages over them in F1-score. Though FCN outperforms our U-Net method in activity Sit_to_Stand and Lie_to_Sit, the performance is not stable and still has the potential for significant improvement. This shows that U-NET can detect more details of the transition activities and has more stable performance.

TABLE 2. The Accuracies and Fw-Scores of the U-Net method and the comparison methods on four datasets.

Dataset	Indicator	SVM	DT	LSTM	CNN	CovLSTM	FCN	SegNet	Mask R-CNN	U-Net
WISDM	Acc	0.813	0.853	0.938	0.941	0.948	0.879	0.957	0.862	**0.964**
	Fw	0.848	0.850	0.937	0.941	0.947	0.881	0.957	0.863	**0.965**
UCI HAPT	Acc	0.918	0.807	0.920	0.903	0.893	0.912	0.915	0.908	**0.921**
	Fw	0.922	0.806	0.917	0.901	0.889	0.912	0.913	0.906	**0.922**
OPP Gesture	Acc	0.785	0.700	0.8316	0.831	0.817	0.911	0.914	0.908	**0.947**
	Fw	0.827	0.698	0.8414	0.934	0.818	0.910	0.912	0.907	**0.947**
Sanitation	Acc	0.817	0.803	0.8024	0.864	0.825	0.640	0.846	0.697	**0.889**
	Fw	0.820	0.804	0.8028	0.863	0.823	0.637	0.846	0.696	**0.889**

E. PERFORMANCE ANALYSIS

To further evaluate our U-Net method, we conduct the activity misalignment analysis, the parameter sensitivity analysis and computation analysis for the proposed method.

1) MISALIGNMENT AND POST-CORRECTION ANALYSIS

We perform activity misalignment analysis on the dense prediction results obtained by our proposed U-Net method on the four datasets, comparing with the other deep learning methods, including LSTM, CNN, CovLSTM, FCN, SegNet, and Mask R-CNN. The obtained experimental results are shown in Figure 7. For the four datasets, U-Net has the higher accuracy and shows a significant reduction on Underfill, Overfill, Substitution, and Fragmentation errors, compared with the deep learning methods based on sliding window prediction, such as LSTM, CNN, and CovLSTM. Especially for the OPP Gesture data, the Underfill and Overfill error rates of LSTM are 5.54% and 2.74%, while U-Net has the lower error rates of Underfill and Overfill, which account for 1.42% and 1.39%, respectively. As we know, the errors of Underfill and Overfill are caused by the inability to identify the activity boundary correctly. This indicates that U-Net has an advantage over LSTM in identifying the boundary of activities more accurately, since the sliding window-based HAR method will result in the loss of useful activity boundary information while U-Net will preserve more boundary information by dense labeling and its specific network architecture of combing the shallow and deep network information. This is why U-Net performs better than LSTM, especially for short-term activity recognition. Though FCN, SegNet, and Mask R-CNN can be also used for dense prediction, the error rates of Underfill, Overfill, Substitution, and Fragmentation on each dataset are always higher than those of U-Net. In particular, there are a large number of substitution and fragmentation errors in the self-collected Sanitation dataset, which leads to a sharp decrease in the accuracy of FCN and Mask R-CNN. Since the Mask R-CNN has similar network architecture with FCN, the misalignment error rates are also similar to FCN.

The errors of Substitution and Fragmentation account for the largest proportion of misalignments occurred in U-Net dense prediction results of various datasets. The two misalignment errors are the main causes of U-Net misclassification. And FCN, SegNet, and Mask R-CNN also face the same problem as dense prediction methods. The post-correction algorithm we proposed in Section III-C is mainly aimed at the correction of these two types of errors. We apply the post-correction method for the dense prediction results of U-Net, FCN, SegNet, and Mask R-CNN on four datasets and analyze the misalignment errors, the obtained results are shown in Fig.7, denoted as U-Net_PC, FCN_PC, SegNet_PC, and MaskRCNN_PC respectively. Compared with the original dense prediction results of the four methods, U-Net_PC, FCN_PC, SegNet_PC, and MaskRCNN_PC have improved the accuracy on each dataset and reduced the Fragmentation error rate significantly, which proves the effectiveness of the proposed post-correction method for the dense prediction-based HAR methods. In particular, FCN_PC and MaskRCNN_PC demonstrate the largest improvement in Sanitation datasets. But the improvement of OPP Gesture and HAPT dataset is small. As for the OPP Gesture, the accuracy of U-Net is 94.75%, while the accuracy of U-Net_PC is 94.78%. Although the Fragmentation error rate for U-Net_PC decreases, the Underfill error rate for U-Net_PC still increases. Therefore, the accuracy of U-Net increases slightly using the post-correction algorithm. The reason is that determining the optimal shortest threshold of the activity length is difficult due to too many short-term activities in OPP Gesture dataset.

2) PARAMETER SENSITIVITY ANALYSIS

In order to evaluate the effect of different settings (network depth, subsequence length) on our proposed approach, we conduct several experiments on the four datasets, in which the U-Net's settings keeps the same as what are shown in Section III-B(2), except the network depth or the subsequence length. Our U-Net network architecture consists of the same number of the down-sampling blocks and the up-sampling blocks. The more blocks it contains, the deeper the network will be. We investigate the performance of our U-Net network with different numbers of U-Net blocks to evaluate the influence of different network depths, the results are shown

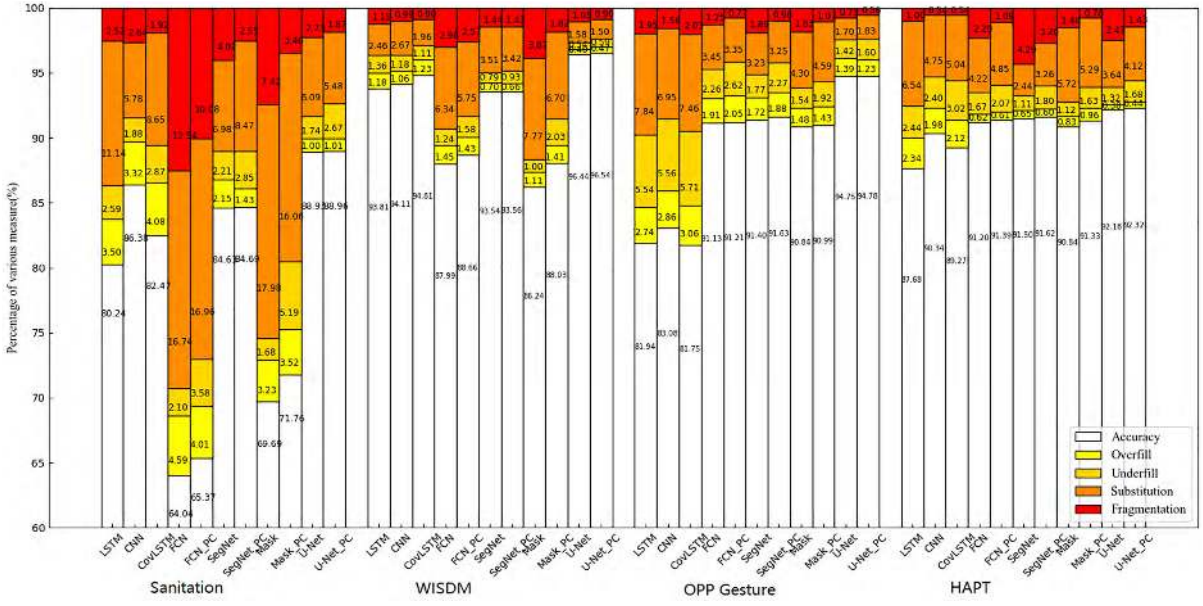


FIGURE 7. Activity misalignment and the post-correction performance analysis on four datasets.

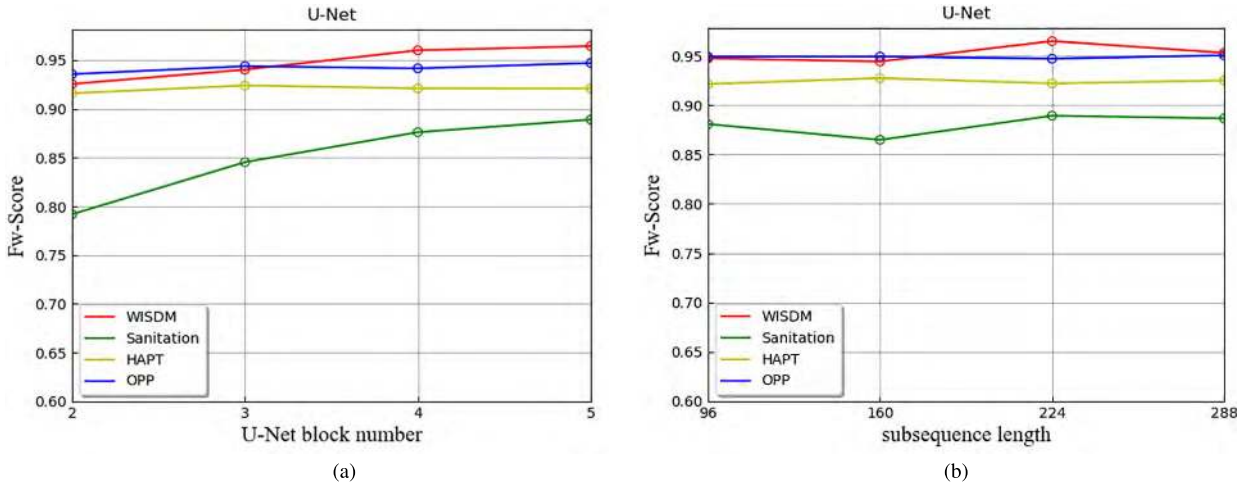


FIGURE 8. Test performance of U-Net with different block number and sequence length. (a) Network depth. (b) Subsequence length.

in Fig.8(a). For WSIDM and Sanitation, the recognition Fw-Score improves with the increase of block number. Especially on WISDM, U-Net with the block number of 5 achieves the highest Fw-Score and has a tendency to remain stable. For HAPT and OPP Gesture, U-Net with the block number of 5 also achieves the highest Fw-Score, but the overall fluctuation range is small with the increase of block number. It proves that it's optimal to set the network block number to 5 in this paper. We also investigate the effect of our approach with the different subsequence length on the four datasets. The results are shown in Fig.8(b). For WISDM, the recognition Fw-Score firstly improves with the increase of the subsequence length and achieves the highest when the subsequence length grows to 244 sampling points, then it decreases. U-Net with the subsequence length of 224 also achieves the highest Fw-Score on the other datasets and the results are not greatly affected by the length of the subsequence, showing the effectiveness of our design and great robustness.

3) COMPUTATION ANALYSIS

The prediction time consumption of each method on different datasets is used to evaluate the computational complexity. What really affects the prediction efficiency of the model is not the training time, but the actual prediction speed. Even if the model training time is short, as long as the prediction time is too long, it is not an efficient model. As shown in Table 3, it is the prediction time consumption of nine algorithms on four datasets. The time unit is in seconds (s).

It can be found that SVM takes a longer time on HAPT dataset and OPP Gesture dataset, indicating that the time

TABLE 3. The prediction time consumption (s) of the U-Net method and the comparison methods on four datasets.

DataSet	SVM	DT	CNN	LSTM	CovLSTM	FCN	SegNet	Mask R-CNN	U-Net
WISDM	0.179	0	0.024	0.049	0.034	0.128	8.663	2.699	0.508
UCI HAPT	6.75	0.008	0.023	0.043	0.038	0.100	6.158	1.879	0.343
OPP Gesture	53.2	0.008	0.097	0.102	0.090	0.162	6.759	2.153	0.426
Sanitation	0.158	0	0.051	0.030	0.037	0.062	2.146	0.652	0.129

efficiency of SVM is not very high. Especially the DT's prediction time is very short because it only needs to perform branch judgment according to each feature. CNN, LSTM, CovLSTM, FCN, SegNet, Mask R-CNN, and U-Net use GPU to accelerate the prediction speed in all experiments, which is not the same as the traditional algorithms that run on CPU. We find that the prediction time of CNN, FCN, U-Net, Mask R-CNN, and SegNet increase in turn. In addition, U-Net's prediction time is less than 1 second, which can be further improved on better performance machines. Remarkably, although Mask R-CNN has a similar architecture with FCN, the prediction time of Mask R-CNN is significantly longer than that of FCN. This is because its subsequence length is 28, much shorter than that of FCN, thus leading to the extension of prediction time. It proves that longer subsequence can speed up the inference speed. In addition, SegNet has a good performance on the four datasets, but it takes a longer time since its encoder network contains more convolutional layers and more kernels than U-Net.

V. CONCLUSION

In this paper, we propose a HAR method based on motion sensor using U-Net. Different from the existing machine learning-based and deep learning-based HAR methods which use the sliding window labeling and prediction, our U-Net method overcomes the multi-class window problem inherent in the sliding window method and realizes the prediction of each sampling point's label in time series data. The experimental results demonstrate that our U-Net outperforms all the comparative methods on the four datasets with better performance on short-term activity recognition and better robustness. Although our U-Net method is designed for human activity recognition base on the activity sensor data, our U-Net method has the potential to realize dense prediction for the other types of time series data, our research will provide the foundation for applying U-Net method to the recognition of other types of time series data.

REFERENCES

[1] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 790–808, Nov. 2012.

[2] A. Sathyanarayana, F. Ofli, L. Fernandez-Luque, J. Srivastava, A. Elmagarmid, T. Arora, and S. Taheri, "Robust automated human activity recognition and its application to sleep research," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops (ICDMW)*, Barcelona, Spain, Dec. 2016, pp. 495–502.

[3] H. Rezaie and M. Ghassemian, "An adaptive algorithm to improve energy efficiency in wearable activity recognition systems," *IEEE Sensors J.*, vol. 17, no. 16, pp. 5315–5323, Aug. 2017.

[4] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 1, pp. 156–167, Jan. 2006.

[5] M. Arif, M. Bilal, A. Kattan, and S. I. Ahamed, "Better physical activity classification using smartphone acceleration sensor," *J. Med. Syst.*, vol. 38, no. 9, p. 95, Sep. 2014.

[6] J. Sarkar, L. T. Vinh, Y.-K. Lee, and S. Lee, "GPARS: A general-purpose activity recognition system," *Appl. Intell.*, vol. 35, no. 2, pp. 242–259, Oct. 2011.

[7] V. K. Verma, W.-Y. Lin, M.-Y. Lee, and C.-S. Lai, "Levels of activity identification & sleep duration detection with a wrist-worn accelerometer-based device," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Seogwipo, South Korea, Jul. 2017, pp. 2369–2372.

[8] S. González, J. Sedano, J. R. Villar E. Corchadoc, Á. Herrerod, and B. Baruqued, "Features and models for human activity recognition," *Neurocomputing*, vol. 167, pp. 52–60, Nov. 2015.

[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] M. Panwar, S. R. Dyuthi, K. C. 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. EMBC*, Seogwipo, South Korea, Jul. 2017, pp. 2438–2441.

[11] Y. Chen and Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," in *Proc. SMC*, Kowloon, China, Oct. 2015, pp. 1488–1492.

[12] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer, and R. Crompton, "Activity identification using body-mounted sensors—A review of classification techniques," *Physiol. Meas.*, vol. 30, no. 4, p. R1, Apr. 2009.

[13] 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, Jan. 2016.

[14] R. Yao, G. Lin, Q. Shi, and D. C. Ranasingheb, "Efficient dense labelling of human activity sequences from wearables using fully convolutional networks," *Pattern Recognit.*, vol. 78, pp. 252–266, Jun. 2018.

[15] Y. Zheng, W.-K. Wong, X. Guan, and S. Trost, "Physical activity recognition from accelerometer data using a multi-scale ensemble method," in *Proc. 25th Conf. (IAAI)*, Jun. 2013, pp. 1575–1581.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR*, Jun. 2015, pp. 3431–3440.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, Oct. 2015, pp. 234–241.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[19] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

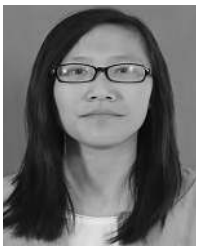
[20] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes," in *Proc. NIPS*, 2002, pp. 841–848.

[21] Z.-Y. He and L.-W. Jin, "Activity recognition from acceleration data using AR model representation and SVM," in *Proc. ICMLC*, Kunming, China, vol. 4, Jul. 2008, pp. 2245–2250.

[22] L. Fan, Z. Wang, and H. Wang, "Human activity recognition model based on decision tree," in *Proc. CBD*, Nanjing, China, Dec. 2013, pp. 64–68.

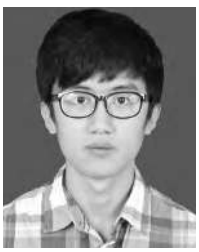
[23] A. M. Khan, Y.-K. Lee, S. Y. Lee, and T.-S. Kim, "A triaxial accelerometer-based physical-activity recognition via augmented-signal features and a hierarchical recognizer," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 5, pp. 1166–1172, Sep. 2010.
[24] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 3, pp. 871–879, Mar. 2009.
[25] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A hybrid discriminative/generative approach for modeling human activities," in *Proc. IJCAI*, Jul. 2005, pp. 722–766.
[26] S. Lee, S. Lee, H. X. Le, H. Q. Ngo, H. I. Kim, M. Han, and Y.-K. Lee, "Semi-Markov conditional random fields for accelerometer-based activity recognition," *Appl. Intell.*, vol. 35, no. 2, pp. 226–241, Oct. 2011.
[27] X. Long, B. Yin, and R. M. Aarts, "Single-accelerometer-based daily physical activity classification," in *Proc. EMBC*, Minnesota, MN, USA, Sep. 2009, pp. 6107–6110.
[28] S.-M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jeju, South Korea, Feb. 2017, pp. 131–134.
[29] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 2, p. 11, Mar. 2017.
[30] M. Edel, and E. Köppe, "Binarized-BLSTM-RNN based human activity recognition," in *Proc. IPIN*, Alcala de Henares, Spain, Oct. 2016, pp. 1–7.
[31] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," 2016, *arXiv:1604.08880*. [Online]. Available: https://arxiv.org/abs/1604.08880
[32] J. A. Ward, P. Lukowicz, and H. W. Gellersen, "Performance metrics for activity recognition," *Trans. Intell. Syst. Technol.*, vol. 2, no. 1, p. 6, Jan. 2011.
[33] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Mar. 2011.
[34] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, Jan. 2016.
[35] 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 Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, Nov. 2013.

YONG ZHANG received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2007, where he has been an Associate Professor with the School of Electronic Engineering and is currently the Director of the Fab. X Artificial Intelligence Research Center. He is also the Deputy Head of the Mobile Internet Service and Platform Working Group, China Communications Standards Association. He has authored or coauthored more than 70 papers and holds 29 granted Chinese patents. His research interests include artificial intelligence, wireless communication, and the Internet of Things. His awards and honors include the Second Prize of the Science and Technology Award from the Chinese Institute of Electronics, in 2017, and the First Prize of the Power Innovation Award from the China Electricity Council, in 2018.



ZHAO ZHANG received the bachelor's degree from Yanshan University, Qinhuangdao, China, in 2018. She is currently pursuing the master's degree with the Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include deep learning and artificial intelligence.

YU ZHANG has been a Graduate Student with the School of Electronic Engineering, Beijing University of Posts and Telecommunications, China. His research interests include artificial intelligence and satellite communication.



JIE BAO is currently pursuing the master's degree with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include artificial intelligence and the Internet of Things.

YIFAN ZHANG received the bachelor's degree from the Beijing Institute of Technology. He is currently pursuing the master's degree with the School of Information, Production and Systems, Waseda University, Japan. His research interests include artificial intelligence, automation, and EDA algorithm.

HAIQIN DENG is the Founder and the CEO of AIdong Super AI (Beijing) Company, Ltd., which is a company focused on the artificial intelligence technology. He spent nine years at Nokia as a Senior Technology Expert and a Product Manager. He possesses over 18 years of telecommunication and software development experiences in mobile Internet and product operation. He was the Final Winner of the Nokia Innovation and Excellence Award 2009.