applied sciences

MDPI

Article

Marfusion: An Attention-Based Multimodal Fusion Model for Human Activity Recognition in Real-World Scenarios

Yunhan Zhao [1], Siqi Guo [1], Zeqi Chen[1], Qiang Shen [1], Zhengyuan Meng [2] and Hao Xu [1,3,*]

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China; zhaoyh2119@mails.jlu.edu.cn (Y.Z.); guosq2119@mails.jlu.edu.cn (S.G.); chenzq2019@mails.jlu.edu.cn (Z.C.); shenqiang19@mails.jlu.edu.cn (Q.S.)
[2] College of Software Engineer, Jilin University, Changchun 130012, China; mengzy19@mails.jlu.edu.cn
[3] Chongqing Research Institute, Jilin University, Chongqing 401123, China
* Correspondence: xuhao@jlu.edu.cn

Abstract: Human Activity Recognition(HAR) plays an important role in the field of ubiquitous computing, which can benefit various human-centric applications such as smart homes, health monitoring, and aging systems. Human Activity Recognition mainly leverages smartphones and wearable devices to collect sensory signals labeled with activity annotations and train machine learning models to recognize individuals' activity automatically. In order to deploy the Human Activity Recognition model in real-world scenarios, however, there are two major barriers. Firstly, sensor data and activity labels are traditionally collected using special experimental equipment in a controlled environment, which means fitting models trained with these datasets may result in poor generalization to real-life scenarios. Secondly, existing studies focus on single or a few modalities of sensor readings, which neglect useful information and its relations existing in multimodal sensor data. To tackle these issues, we propose a novel activity recognition model for multimodal sensory data fusion: Marfusion, and an experimental data collection platform for HAR tasks in real-world scenarios: MarSense. Specifically, Marfusion extensively uses a convolution structure to extract sensory features for each modality of the smartphone sensor and then fuse the multimodal features using the attention mechanism. MarSense can automatically collect a large amount of smartphone sensor data via smartphones among multiple users in their natural-used conditions and environment. To evaluate our proposed platform and model, we conduct a data collection experiment in real-life among university students and then compare our Marfusion model with several other state-of-the-art models on the collected datasets. Experimental Results do not only indicate that the proposed platform collected Human Activity Recognition data in the real-world scenario successfully, but also verify the advantages of the Marfusion model compared to existing models in Human Activity Recognition.

Keywords: human activity recognition; multimodal data fusion; deep learning; pervasive computing

check for updates

Citation: Zhao, Y.; Guo, S.; Chen, Z.; Shen, Q.; Meng, Z.; Xu, H. Marfusion: An Attention-Based Multimodal Fusion Model for Human Activity Recognition in Real-World Scenarios. *Appl. Sci.* **2022**, *12*, 5408. https://doi.org/10.3390/app12115408

Academic Editors: Juan Ye and Gabriele Civitarese

Received: 20 April 2022
Accepted: 24 May 2022
Published: 26 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As an important research topic in ubiquitous computing, Human Activity Recognition (HAR) from wearable sensors is a key element of various intelligent applications, such as smart personal assistants [1], healthcare assessment [2–6], sports monitoring [7], and aging care [8]. HAR can recognize a variety of individual behaviors, such as running, and walking, and has a wide range of applications [9]. HAR mainly uses wearable devices to collect individual behavior data and model the data for individual behavior identification.

Deep neural networks perform well on HAR due to the ability to extract explicit feature design. Activity recognition models based on neural network technology often consist of three modules, which are the convolution module for extracting the correlation between sensor data, the convolution module for extracting the correlation among different kinds of sensor data, and the recurrent neural network module of extracting the correlation of data

in a different moment. The models use convolution networks for feature extraction in the spatial dimension and then use recurrent neural networks to play for feature extraction in the temporal dimension [10]. In addition, some models also use the Attention mechanism to further enhance the generalization of the model [11].

However, there are two major deficiencies in the current datasets used to train activity recognition models in the real world: First, sensor data and activity labels are mostly collected using special experimental equipment in a supervised experimental environment, which affects the normal life of users [12–14]. Second, the modalities of sensor data are relatively single [15,16], and existing studies focus on single or a few modalities of sensor readings, which neglects useful information and its relations existing in multimodal sensor data. Aiming at the two deficiencies of existing datasets, we build an experimental platform for activity recognition data collection and activity label collection: MarSense. MarSense can automatically collect a large amount of sensor data through a small number of user operations in the user's daily life. This data collection method takes full advantage of the powerful functions and widespread availability of smartphones, reduces the cost of data collection, and can collect richer data from a larger population.

We also build our activity recognition model Marfusion, and use the sensor data and labels collected by the MarSense experimental platform for user experiments to train the model. Marfusion is to extract features in each dimension and the associations between different features to extract hidden features. It extracts features from multimodal sensors by the Convolution Neural Network (CNN) structure and Attention mechanism. For each sensor, a set of CNN-based networks is used for feature extraction for each modality. After that, Dot-Product Scaled Self Attention is used to process and give weight to each feature. Then, the data were multiplied with corresponding weights and they were input into the fusion feature extraction module. Finally, the data were input into the classification subnet to obtain the probability values of different categories. This paper also compares the training results of the Marfusion model with several other representative models. The value of Precision, Recall, and F1 reach 0.944, 0.944, and 0.943, which are higher than other models and verify the advantages of the Marfusion model compared to existing models in activity recognition.

The rest of the paper is organized as follows. In the next Section 2, we introduce the existing approaches of HAR, HAR in the wild, and Data Fusion. Then, in Section 3 we introduce Marfusion and explain the principle and mechanism of vital modules. The data collection is presented in Section 5. In this section, we introduce our collection experiment, data collection platform (MarSense), and dataset construction. Section 6 presents the model training and model evaluation. The final Section 7 presents the concluding remarks of this paper.

2. Related Work

2.1. Human Activity Recognition

A lot of research has been conducted on human activity recognition (HAR) by sensors, and several models have been proposed to map the sensor data to specific activity categories. However, most of these methods require subjects to wear special equipment on specific parts for data collection and performing required behavior. There are two shortcomings of the existing data. One is the need to use special equipment, which is not convenient to collect the data. Second, it required the participants to do specific activities which have a great impact on daily life.

Human activity recognition requires time-series sensor data in most cases, such as accelerometers, gyroscopes, etc. Some of the models recognize the state of users by accelerometers and rotation sensor data on the mobile phone. The user's behavior can be identified by artificially designed features after collecting the data from mobile phones. However, noise is unavoidable when we use mobile phone sensors for activity prediction [10], which make it difficult to achieve high performance. Moreover, feature extraction based on artificially designed features performs steadily in some cases but is not good at identifying the vital

features, as well as the fact that it is strongly affected by noise. The complexity of human activity improves the difficulty of recognition.

In 2017, ShuoChao Yao et al. proposed the DeepSense model [10] to solve noise and feature extraction problems in a unified way. DeepSense combines the features of Convolution and Recurrent neural networks to improve the recognition capability of correlations and features between sensors. The model has better performance in integrating the features of different types of sensors by transforming the features into the form of the model on the temporal dimension. DeepSense mainly uses three functional modules, which are the Exclusion Convolution module, Fusion Convolution module, and Recurrent Neural Network, which can have better performance on HAR. Compared to DeepSense, the AttnSense proposed by Haojie Ma et al. in 2019 optimized the model by using the Attention mechanism [11]. In addition to using a Conventional Convolution network, AttnSense used Gated Recurrent Unit (GRU) in the Recurrent Neural Network, which simplifies the calculation and used the Attention mechanism to calculate weights in data fusion to better fuse the features of all types of sensors. In addition, in the output process of GRU, all states are fused through the Self-Attention mechanism, which can extract the correlation of data in the temporal dimension better and improve the accuracy of HAR.

In recent years, activity recognition is highly applied by many smartphone makers, such as Huawei and Xiaomi. Compared with existing applications such as the Huawei recognition system and the Xiaomi recognition system, we utilize more sensors than them and our system can recognize more complex activities, as theirs can only recognize some simple activities.

2.2. HAR in the Wild

HAR in the wild [17] is challenging because the variability in behavioral patterns and the data collection may be unbalanced and incomplete [18]. Vaizman, Y et al. [17] used smartphones and smartwatch sensors to collect label data. Moreover, Yonatan Vaizman et al. [18] proposed using the Multilayer perceptron (MLP) as a multi-task model for context recognition, and the model start by learning good representations for predicting a basic set of labels and using these representations to classify another set of labels. They adjust the MLP's objective function in a non-standard fashion to handle unbalanced, incomplete labeling. Compared to previous models, it improved recognition, even with fewer parameters than a linear model.

Yonatan Vaizman et al. [19] presented a framework that converged the feature fusion strategy with a teacher/learner approach over the active learning technique to automatize the self-training process of the learner models. The framework integrated the Teacher/Learner, active learning, and data fusion technique to optimize the performance of HAR.

As for context recognition, Shen, Q et al. [20] presented a novel ontological context model that supports both reasoning and recognition from a subjective perspective, which captured five dimensions and enables downstream context recognition tools to leverage correlations between these dimensions. They also incorporate three levels of description for each aspect, which supports different kinds of annotations. The results show that handling correlation across aspects improves recognition performance. Zhang, W et al. [1] used multi-modality and collected the data in real-world scenarios from mobile phone sensors and self-annotations. The results show that individuals are more easily identified by rarer, rather than frequent, context annotations.

2.3. Motion Sensors

The motion sensor is a sensor used in wearable devices and mobile phones to collect motion data such as acceleration and angular velocity. In recent years, there have been many studies using motion sensor data for human behavior recognition. Alireza Ghods and Diane J. Cook [21] investigated a method for automatically extracting universal and meaningful features that were applicable across similar time series-based learning tasks such as activity recognition and fall detection. Nils Y. Hammerla et al. [22] proposed a new

regularization method that improved the accuracy of human behavior recognition using sensor data. There are also many studies using neural networks to obtain highly accurate in human activity recognition by using sensor dataset [23–25].

2.4. Multimodal Fusion

Data Fusion is to combine data from multiple data sensors and finds the correlation between different features, which can improve the accuracy of recognition and lower the probability of detection error [26–28]. Compared with a single sensor, the advantage of multiple data sensors are to find the correlation among all the features and extract the hidden features.

Data Fusion has shown to be effective in HAR. For example, Wang, Y. et al. [29] propose a practical hybrid sensory HAR system, which combines wearable sensors and ambient sensors and achieves the 96.82%. Vidya, B. et al. [30] proposed a supervised machine learning-based activity recognition framework and they used discrete wavelet transform and the empirical mode decomposition to extract features. Han, J. et al. [31] proposed a scalable framework that models spatial relationship, local interaction, and long-term temporal dependency simultaneously and it achieved 1.75%, 2.68% improvements over existing methods. IsIam, M.M. et al. [32] developed a multimodal attention mechanism for disentangling and fusing the salient unimodal features to compute the multimodal features in the upper layer, and the multimodal features are used in a fully connected neural network to recognize the human activity. The model outperformed all other evaluated baselines. The results of the studies above showed that multiple sensors fusion has better performance on HAR.

3. Multimodal HAR Model: Marfusion

In this section, we introduced our proposed multi-modality sensor fusion model for Human Activity Recognition (HAR) Marfusion, which is used to classify the data by the embedded sensor of smartphones to identify the corresponding activities of the data. In the following subsections, the overall structure and each component of Marfusion are introduced extensively.

3.1. The Overall Structure of Marfusion

Marfusion aims to extract features in each dimension and the associations between different features to extract hidden features. The overall structure of Marfusion is shown in Figure 1.

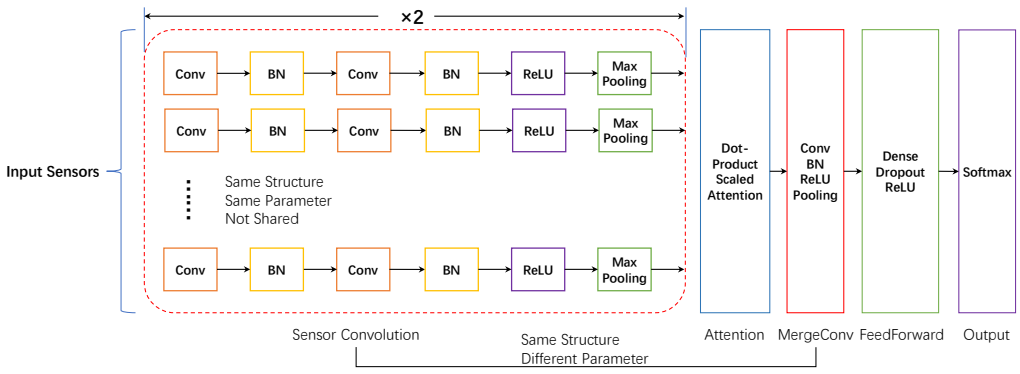


Figure 1. The Overall Structure of Marfusion.

Marfusion extracts features from multimodal sensors by leveraging both the Convolution Neural Network (CNN) structure and Attention mechanism. Specifically, for each sensor, a set of CNN-based networks is used for feature extraction for each modality. After that, we use Dot-Product Scaled Self Attention to process and give weight to each feature channel of each sensor. Then, we multiply the data of different channels with

corresponding weights and input them into the fusion feature extraction module. The structure of the fusion feature extraction module is the same as that of the sensor independent feature extraction module, but the parameters may be different. The fusion feature module uses a large convolution kernel for better performance on extracting features across multiple sensors. Finally, the data were input into the classification subnet composed of a full-connection layer, Batch Normalization layer, Dropout layer, ReLU activation function layer, and Softmax activation function layer to obtain the probability values of different categories. Next, the principle and mechanism of vital modules are introduced.

3.2. Data Augment Based on Fast Fourier Transform (FFT)

Considering the fact that noise is unavoidable in the sensor data, we capture the noise pattern and frequency features in the multidimensional sensing signals using data augmentation techniques. The sensory signal is essentially composed of functions of time and frequency features that represent the changes in the energy content of a signal [10]. To extract frequency features, therefore, we use 1-order FFT to transform the data on each axis.

The process of FFT is as follows. Set a polynomial $A(x)$ as shown in Equation (1).

$$A(x) = \sum_{i=0}^{n-1} a_i x^i = a_0 + a_1 x + a_2 x^2 + a_{n-1} x^{n-1} \tag{1}$$

It is divided into two parts according to its parity, as shown in Equation (2).

$$A(x) = \left(a_0 + a_2 x^2 + \cdots + a_{n-2} x^{n-2}\right) + x\left(a_1 + a_3 x^2 + \cdots + a_{n-1} x^{n-2}\right) \tag{2}$$

Add two polynomials $A_1(x)$ and $A_2(x)$, as shown in Equations (3) and (4).

$$A_1(x) = a_0 + a_2 x + a_4 x^2 + \cdots + a_{n-2} x^{\frac{n}{2}-1} \tag{3}$$

$$A_2(x) = a_1 + a_3 x + a_5 x^2 + \cdots + a_{n-1} x^{\frac{n}{2}-1} \tag{4}$$

Then we can obtain $A(x) = A_1(x^2) + xA_2(x^2)$. Suppose $k < \frac{n}{2}$, let $x = \omega_n^k, \omega_n^{k+\frac{n}{2}}$, respectively, the following equations will be deduced.

$$A\left(\omega_n^k\right) = A_1\left(\omega_n^{2k}\right) + \omega_n^k A_2\left(\omega_n^{2k}\right) = A_1\left(\omega_n^{2k}\right) + \omega_n^k A_2\left(\omega_n^{2k}\right) = A_1\left(\omega_{\frac{n}{2}}^k\right) + \omega_n^k A_2\left(\omega_{\frac{n}{2}} k\right) \tag{5}$$

$$A\left(\omega_n^{k+\frac{n}{2}}\right) = A_1\left(\omega_n^{2k+n}\right) + \omega_n^{k+\frac{n}{2}} A_2\left(\omega_n^{2k+n}\right) = A_1\left(\omega_n^{2k}\right) - \omega_n^k A_2\left(\omega_n^{2k}\right) = A_1\left(\omega_{\frac{n}{2}}^k\right) - \omega_n^k A_2\left(\omega_{\frac{n}{2}} k\right) \tag{6}$$

where we obtain the value of $A\left(\omega_n^k\right)$ and $A\left(\omega_n^{k+\frac{n}{2}}\right)$ according to the value of $A_1\left(\omega_{\frac{n}{2}}^k\right)$ and $A_2\left(\omega_{\frac{n}{2}}^k\right)$. In this way, the FFT sequence can be evaluated iteratively.

There are two parts of results in the FFT, which is the real part and the imaginary part. Taking the accelerometer as an example, we can obtain six data axes, which is three-dimension real part and three-dimension imaginary part after FFT. The shapes of different behaviors in the accelerometer before FFT are $[200, 3]$, which is visualized as shown in Figure 2.

As shown in the figure, data generated by accelerometers varies greatly when users behave differently. We collect the data by smartphone sensors in real-life scenarios, and we have no restrictions on the participants during data collection. Therefore, the participants may use their mobile phones when they are sitting and the data for sitting is fluctuating. After transforming the data from time-domain into frequency-domain with FFT, periodic noise data will show certain features from a frequency-domain perspective, which can be discovered easily by the neural network, and weaken the influence of the accuracy of recognition results.

After Fourier Transformation, the shapes of sensor data are $[2, 200, 3]$. The 2 represents both the time and frequency domain after Fourier transformation, 200 represents the length of the sequence and 3 represents three data-generating axes of each sensor. The transformed data shown in Figure 3 is the input of the neural network.

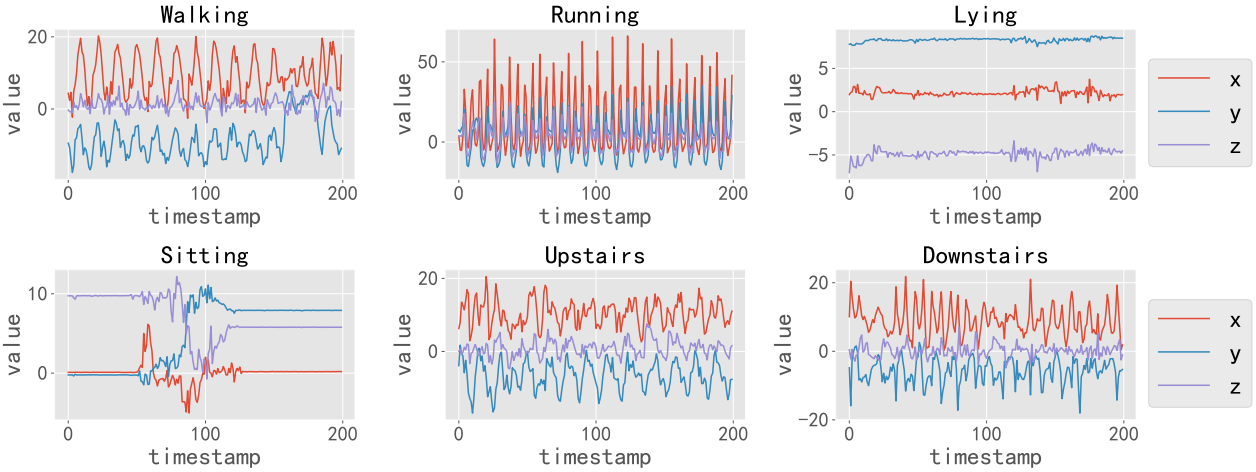


Figure 2. Accelerometer data for different behavior states.

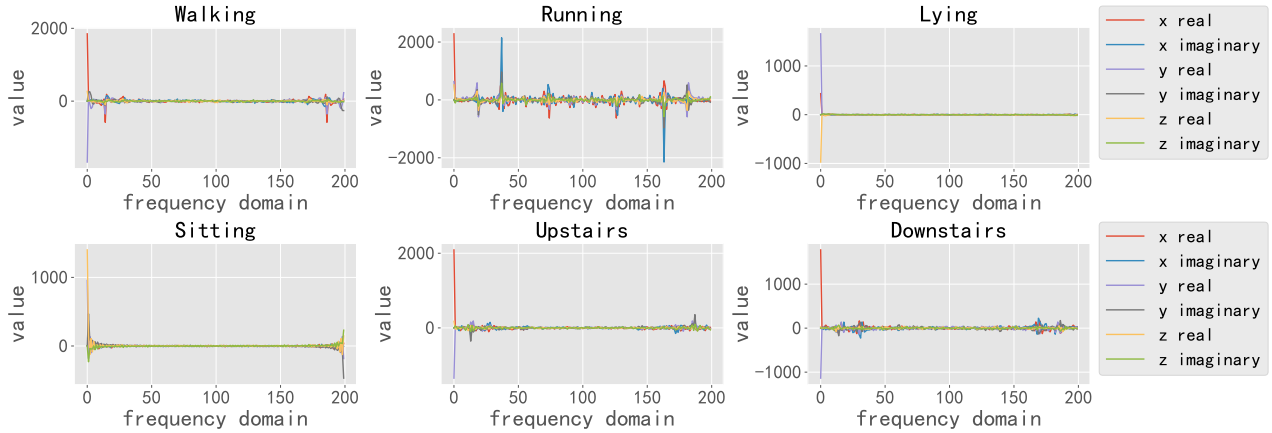


Figure 3. Accelerometer data after Fourier Transformation.

3.3. Feature Extraction Module Based on CNN

In Marfusion, each type of sensor has its independent feature extraction module for better performance on extracting features. The feature extraction module is mainly composed of the four-layer Convolution layer, Batch Normalization layer, ReLU activation function layer, and Max Pooling layer. The Convolution layer is used to extract the features in spatial dimension; the Pooling layer is mainly used to compress the features extracted from the Convolution layer and lower the feature vectors to reduce the parameters and calculation of the network. In the case of exploding gradient and vanishing gradient problems in the training process, we use the Batch Normalization Layer to transform the data so that the gradient of the model parameters of each layer can be stabilized in an appropriate range to prevent non-convergence or shaking repeatedly. To improve the fitting of data, there is a ReLU activation function layer before the Max Pooling layer in the feature extraction module. Non-linear activation and less computational cost are the main characteristics of the ReLU function, which further reinforces the fitting of input data. Note that we apply 2-D CNN architecture to extract the features of multi-dimension sensors such as the accelerometer, while for the single-dimension sensors such as linear accelerometer, 1-D CNN is applied to extract the features.

In our proposed method, there is multiple-modality sensor streaming to be introduced into MarFusion. Specifically, acceleration data has X, Y, Z axes. Linear acceleration is the combination of the three axes. While gyroscope contains three modalities. The rotation vector represents the current tilt of the phone and the data is hybrid computing by axes and angles. The coordinate system of the rotation vector is the same as that of the accelerometer. The magnetic field represents the magnetic field value in three directions of the space coordinate system. Orientation represents the orientation vector that the phone is currently pointing at and the data is the angle of each axis. In total, there are 19 modalities of data streams introduced into the proposed HAR model. In the SensorConv of Marfusion, we use a five-layer convolution network. The structure of MergeConv is the same as that of SensorConv, but the parameters may be different. The specific parameters and the changes in data shape of the convolution layer in SensorConv is shown in Figure 4.

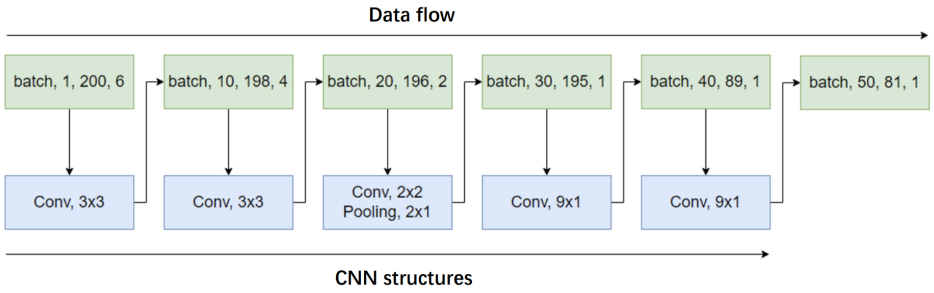


Figure 4. Parameters of Convolution Layer and Data Shape Change in SensorConv.

3.4. Attention Mechanism

For better performance, different weights have been evaluated on different sensors and we use Dot-product scaled Self-Attention to process the outputs of different sensors. By evaluating different sensors, Marfusion dynamically selects the sensors that impact the result and focuses on them. The Attention Mechanism improves the anti-interference capability of Marfusion, which makes higher accuracy in the test set, and enhances the generalization ability of different inputs. The equation of Self-Attention mechanism used in the paper is shown in (7).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

Q, K, V are all obtained from the input data X after a linear transformation, as Equations (8)–(10) show.

$$Q = XW_Q \tag{8}$$
$$K = XW_K \tag{9}$$
$$V = XW_V \tag{10}$$

W_Q, W_K, W_V are the weight matrix of linear transformation corresponding to Q, K, V.

3.5. Batch Normalization

In the case of exploding gradient and vanishing gradient problems in the training process, we use Batch Normalization to optimize the model. In the training process, if the parameter of a network layer tends to 0, after successive multiplication, the gradient value of some parameters may become very small so it fails to update the parameters, which is a vanishing gradient problem. An exploding gradient problem is the opposite to a vanishing gradient problem, that is, the parameter is too large, which leads to the update of the parameters being unstable. Batch Normalization can transform the input data to make it fluctuate in the small-scale near 0, in case of exploding gradient and vanishing

gradient problems, speeding up the training model and improving the stability of the training process.

The Batch Normalization layer makes statistics on the input data, obtaining the mean and variance of each batch, and varies the data by mean and variance, so that the mean and variance of the current batch are 0 and 1, respectively. However, such scaling leads to ignoring the features in the data, which is not conducive to feature extraction. Therefore, in the last step of Batch Normalization, the data will be transformed by trainable parameters γ and β to reserve the feature information. The process of data transformation by Batch Normalization is shown in Equations (11)–(14).

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{11}$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{12}$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{13}$$

$$y_i = \gamma \hat{x}_i + \beta \tag{14}$$

3.6. L2 Regularization

In the case of over-fitting and failing to predict categories of behavior from sensor data accurately, we use L2 regularization to restrict the loss function. With the restriction of L2 regularization, the loss function of the model consists of two parts: one is the Cross-Entropy loss function, whose value is determined by the difference between the predicted value and the actual value. The other is the L2 Regular penalty term determined by all the weight parameters.

Before adding the L2 regular penalty term, the loss function of the model is shown in Equation (15).

$$J = CrossEntropy(Y_{true}, Y_{pred}) \tag{15}$$

After adding, the loss function is shown in Equation (16). J represents the original Cross-Entropy loss function, and J_{L2} represents the loss function after adding the penalty term.

$$J_{L2} = J + \frac{\lambda}{2n}\sum_{i=1}^{n} w_i^2 \tag{16}$$

Before adding a penalty term, the model is optimized to minimize the loss of Cross-Entropy. When the L2 regular penalty term is added, the optimization of the model becomes a game of Cross-Entropy loss and L2 regular term. To an extent, the reduction in cross-entropy and the reduction in L2 regular penalty terms are mutually restrictive. To minimize the sum of the two, the model will try to find a balance between them to minimize the loss function.

The addition of L2 regular term makes the weight parameters tend to 0 in general, in the case of excessive parameters, which makes the model smoother and prevents the model from over-learning some interference features. Thus, the model has better performance on the test set and improves the generalization and stability performance of the model.

4. Data Collection Platform: MarSense

MarSense is a software platform specially developed for smartphone sensor data collection. In previous studies, sensor data were mostly collected by wearing special equipment, which has a great impact on users' daily life and is not convenient for more users to participate in data collection. With the large-scale popularization and rapid development of smartphones, smartphones are generally equipped with a variety of physical sensors. Therefore, using smartphones to collect data can reduce the interference in users' daily life,

since users can complete the data collection by some simple interaction, which reduces the cost of data collection and the impact on users' life. Therefore, more participants will participate in the data collection experiment, which is conducive to involving more users and improving the performance of the model in real-life scenarios. The overall architecture of the MarSense platform is shown in Figure 5.

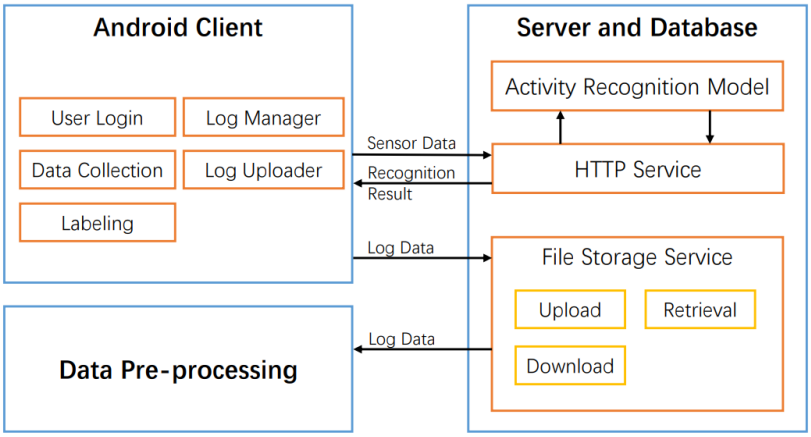


Figure 5. Architecture of MarSense.

MarSense is mainly composed of two parts:

1. **Android client**. Android client is the core of the whole platform, which is to collect sensor data and labels. Data collection is automatically completed by the built-in sensor of the mobile phone, while label collection will be completed by interacting with the users according to the experimental setting.
2. **Server and database**. The main function of the server is to provide the experimental data with storage, retrieval, and download services. In the MarSense client, the experimental data will be stored in a file in the form of a log. At an appropriate time, the log file will be uploaded to the Object Storage Service (OSS) of the server through the Hyper Text Transfer Protocol (HTTP). These log files are always stored in OSS. At the beginning of data pre-processing, the log data generated by the experiment will be retrieved and downloaded from OSS through corresponding tools, and these data will be used for filtering, segmentation, and transformation, and finally assembled into an experimental dataset for model training.

After the model training, we use the built-in smartphone sensors to collect the data and input the data into the model for activity recognition and finally display the results of the activity recognition on the smartphone. The mobile phone is only responsible for data collection, data uploading, and result display. The Android client calls the built-in sensors of the system for data collection and short-term storage. When the data volume reaches the length of the window, it will be uploaded to the server. The server inputs the data into the model and obtains the classification results of activity recognition. The result will be sent and displayed to the Android client to realize real-time recognition. The flowchart of activity recognition using MarSense is shown in Figure 6.

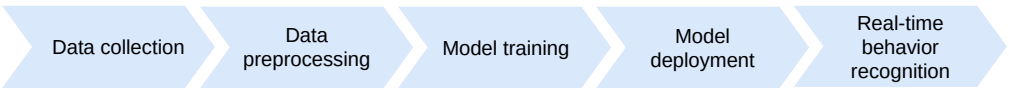


Figure 6. The flowchart of MarSense.

5. Data Collection

Considering the fact that the dataset used in existing work is collected in experimental scenarios, we aim at collecting real-world data to improve the generalization ability of the

Human Activity Recognition (HAR) model. In order to collect data from smartphone sensors in naturally-used conditions and real-world environments, we leverage the MarSense platform to conduct a large-scale data collection experiment. Through the MarSense client running on the Android mobile phone, the data are collected in the scene close to the user's real life as much as possible.

5.1. Data Collection Procedure

The purpose of the data collection experiment is to collect sensor data and corresponding behavior labels using MarSense developed in this paper for data collection so as to use these data for the training of the activity recognition model. The participants in the experiment were students at Jilin University. Students participating in the experiment must have an Android mobile phone that can be used online, and the system version must be later than Android 7. The dataset is available.

The experiment lasted for 1 week, with a total of seven participants. In the experiment, the user-defined mode is used. When users will perform some behavior in the future, they switch the activity label on the client to what will be performed next. After that, the phone will turn on the sensor to continuously collect data. During the data collection, the phone will generate behavior labels every 2 s. If the user ends the current behavior, the client will stop data collection and label generation. The selectable activities are common simple activities in daily life, such as walking, going upstairs, etc. During the experiment, the client on the mobile phone will automatically upload data and labels without user intervention.

As shown in Table 1, there are seven kinds of sensor data collected in this experiment, which are accelerometer, gyroscope, gravity, linear acceleration, magnetic field, direction, and rotation vector. These sensors are generally equipped in current mobile phones. Acceleration is the change in velocity per second along the X, Y, Z axes. Linear acceleration is the combination of the three axes. The gyroscope represents the current angular velocity. Gravity represents the local acceleration of gravity. The rotation vector represents the current tilt of the phone and the data is hybrid computing by axes and angles. The coordinate system of the rotation vector is the same as that of the accelerometer. The magnetic field represents the magnetic field value in three directions of the space coordinate system. Orientation represents the orientation vector that the phone is currently pointing at and the data is the angle of each axis. In total, there are 19 modalities of data streams introduced into the proposed HAR model. There are six kinds of data labels collected in this experiment, which are walking, running, lying, sitting, going upstairs, and going downstairs.

Table 1. List of sensors including sensor type, sampling rate, and unit.

Sensor	Sampling Rate	Unit
Acceleration	20 Hz	m/s^2
Linear Acceleration	20 Hz	m/s^2
Gyroscope	20 Hz	rad/s
Gravity	20 Hz	m/s^2
Rotation Vector	20 Hz	Unitless
Magnetic Field	20 Hz	μT
Orientation	20 Hz	Degrees

The data and labels collected in the experiment need to be further filtered and processed to generate the dataset for training the model. In order to show the structure and the form of the collected data more clearly, this paper visualizes the sensor data, and shows the specific meaning of the experimental data through graphics. In addition, this paper also introduces how to use the collected data to generate datasets for model training through pre-processing and transformation.

5.2. Data Set Construction

5.2.1. Data Structure

Taking the accelerometer on the mobile phone as an example, it can generate three axial data, namely X, Y and Z. The data of these three axes will change constantly when the mobile phone is moving, so as to generate the data reflecting the current acceleration state of the mobile phone continuously. The direction of the three axes of the accelerometer relative to the mobile phone is shown in Figure 7.

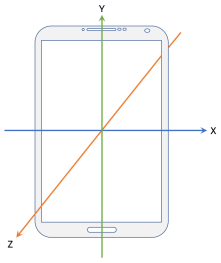


Figure 7. Accelerometer axes on the mobile phone.

The three data generated by the three axes of a sensor on the mobile phone at a certain time point form a data item, and these successively generated data items will form a data sequence, as shown in Figure 8.

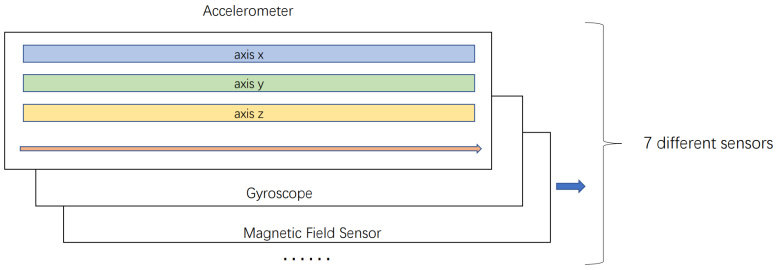


Figure 8. The data sequence.

All seven sensors used in the experiment have three axes so that the shape of the generated data are completely consistent. However, the sampling frequency of each sensor is slightly different, so it cannot guarantee that all the sensors have the same sampling time. Consequently, the data generated by different types of physical sensors on the mobile phone cannot be automatically aligned in time, so it is necessary to pre-process the data after the experiment to generate the data set for model training.

5.2.2. Data Pre-Processing

After data collection, this paper carries out format conversion and segmented filtering on the original data. The specific process is as follows:

1. Make the necessary format conversion of the original data. The original data from server are in CSV format, so we need to use python for pre-processing and we use the tensor to store the data.
2. Put each sensor data into the MongoDB database by sensor type to identify them easily.
3. After the data import, a joint index is established on two attributes of sensor data, which are the "timestamp" and "sensorName" to speed up the search speed when segmenting the data.

After importing the sensor data into the database and establishing the index, we traverse the activity labels in the questionnaire and search the sensor data in the corresponding time period according to the time when the label is generated.

In the process of data search, for all activity labels, Algorithm 1 is used to search the data of each sensor. For the searched data, if the length is greater than 200, the part greater

than 200 will be cut off and only the previous data will be retained. If the length is less than 200, zeros are added after the data until the length is equal to 200. If the length of the actual available data for a label is less than 180, they will be excluded from the training dataset to prevent too many zeros from affecting the accuracy of the model.

After the data of the seven sensors are searched, the data collected by seven sensors are merged in the sensor dimension and form the sensor data to the corresponding activity label. The data shape after merging is $[7, 300, 3]$, where 7 is the number of sensors, 200 is the length of sensor data, and 3 is the number of sensor axes.

The search process for certain sensor data is shown in Algorithm 1.

Algorithm 1 Search Certain Sensor Data According to Activity Labels

Require: The timestamp of the label A, the maximum length M of the window, the data quantity K inside the window, the current window C and the data length threshold T.
for $i = 1, \ldots, M$ **do**
  $C = [A, A + i]$
  $K = getSensorDataLengthOfWindow(C)$
  if $K > T$ **then**
    finish
  else
    continue
  end if
end for

Taking the label of "running" as an example, the corresponding accelerometer, gyroscope and gravity acceleration data are visualized as shown in Figure 9.

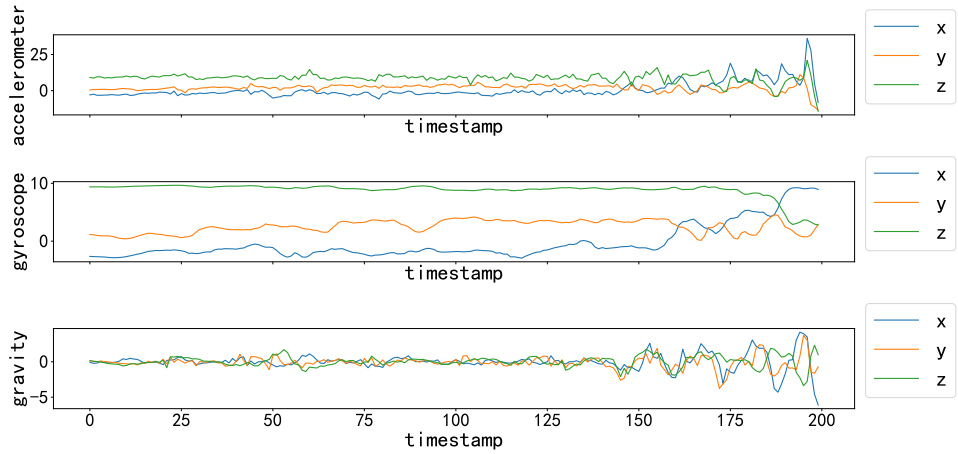


Figure 9. The triaxial data of accelerometer, gyroscope and gravity acceleration.

It can be seen from Figure 9, different types of sensors have significantly different characteristics with a certain periodicity. Therefore, the neural network model can accurately classify the behavior corresponding to the sensor data through feature extraction and fusion.

5.2.3. Dataset Exploration

Through a data preprocessing procedure, the dataset (We open source SmartJLU dataset and source code on Github: https://github.com/Super-Shen/marfusion-dataset (accessed on 17 May 2022)) is finally transformed into a tensor and the shape is $[x, 7, 2, 200, 3]$, which represents the number of available label instances, seven types of sensors, the real part and the imaginary part formed after Fourier transform, the length of the data sequence, and the three axes of the sensor, respectively. The population information of participants is shown Table 2.

Table 2. Statistics of the 7 participants in the dataset.

	Range	Mean (SD)
Age (years)	19–25	23.1 (1.7)
Height (cm)	159–182.7	174.3 (13.9)
Weight (kg)	49–76	63 (19.7)
Numbers of labeled data	501–982	721 (89)

The number and proportion of different types of labels are shown in Figure 10. Running, going upstairs and going downstairs is rare in daily life, so the data samples corresponding to these three behavior labels in the dataset are less than other labels.

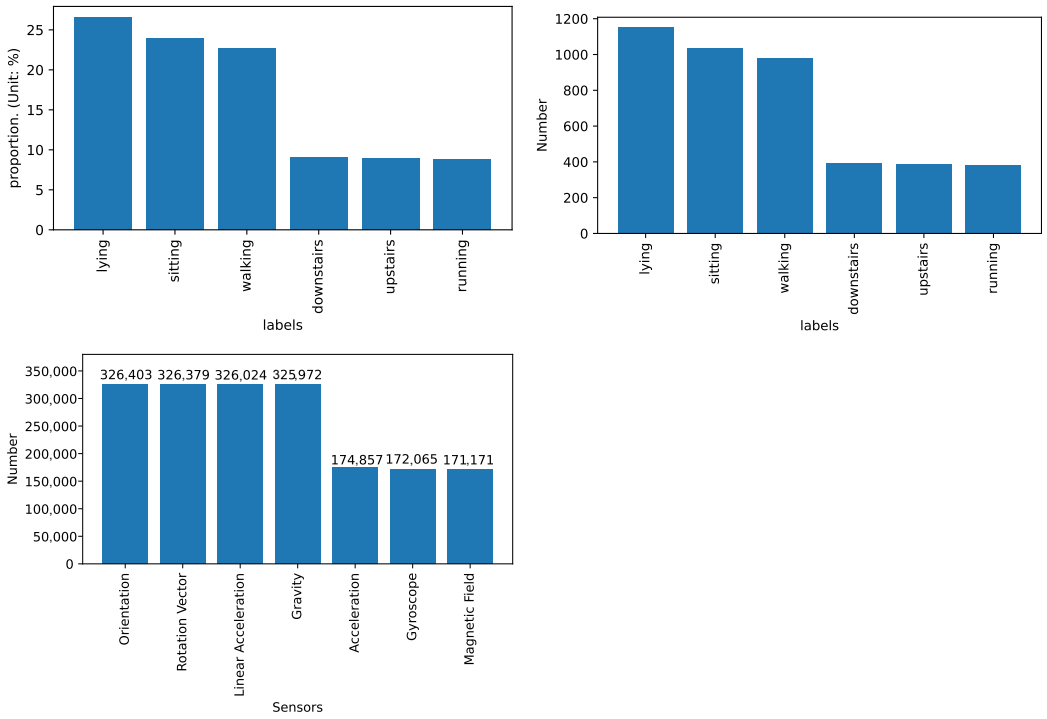


Figure 10. Proportion and number of different labels. Furthermore, the number of each sensor readings visualized.

6. Evaluation

6.1. Experimental Settings

In order to evaluate the effect more accurately, this paper adopts F1 value as the evaluation index of the model effect. F1 value is an improved evaluation index based on Precision and Recall. The specific meanings of these indicators are introduced below:

- Precision. The precision rate refers to the proportion between the number of truly true samples and the number of samples judged to be true by the model. From the perspective of prediction results, precision rate describes the specific proportion of samples that are positively predicted by the model.
- Recall. Recall rate refers to the proportion of all samples of a certain category that can be correctly found by the model. Recall rate describes the ability of the model to find such samples from all samples from the perspective of a certain category of samples. The higher the recall rate is, the more accurate the model can find such samples from all samples.
- F1 value comprehensively considers the precision rate and the recall rate and can describe the performance comprehensively. Compared with using the precision rate or recall rate alone, using F1 value as the evaluation index can more accurately and comprehensively describe the effect and prevent the model from cheating.

This paper will use the evaluation measurements mentioned above to monitor the training process of the model, and better display the changes of the model in the training process by drawing these evaluation indicators into images.

PyTorch framework is used to build and train the neural network model. The development and training environment of Marfusion activity recognition model is shown in the following Table 3. As for the Convolution Neural Network (CNN) structures, we applied 4-layer CNN and one-layer pooling to extract features of different modalities.

For fair comparison, we apply the manner of splitting dataset following the settings in the state-of-the-art approaches [10,11,23,33]. In addition, these state-of-the-art studies [10,11,23,33] evaluated their models on publicly available datasets, which are similar as our dataset with multiple individuals. Thus, we apply the same setting as these state-of-the-art studies to validate our model in a fair experimental setting. Before starting the model training, all samples in the dataset were divided into two parts, 80% of which were used for model training and 20% for testing the training effect and generalization ability. During the training, the batch size was set to 32, and the total loss in each epoch was recorded. For each epoch training, an accuracy calculation was performed. When the loss value becomes stable, the training is stopped and the model is saved on the hard disk.

Table 3. Training environment.

Environment	Version
CPU	Ryzen R9 3950 × 16Cores
GPU	Nvidia GTX 1070 8G
Neural Network Framework	PyTorch V1.10.1

The prediction value and true value will be used to calculate the current F1 value for each epoch training. The training was stopped when the epoch loss no longer decreased and the model tended to be stable.

6.2. Training Procedure

After each epoch is trained, we test the data fitting ability of the model on the training set and the test set. The higher the precision is on the training set or the test set, the better the model can fit the data. The average precision of each tag category changes with the training epoch, as shown in Figure 11.

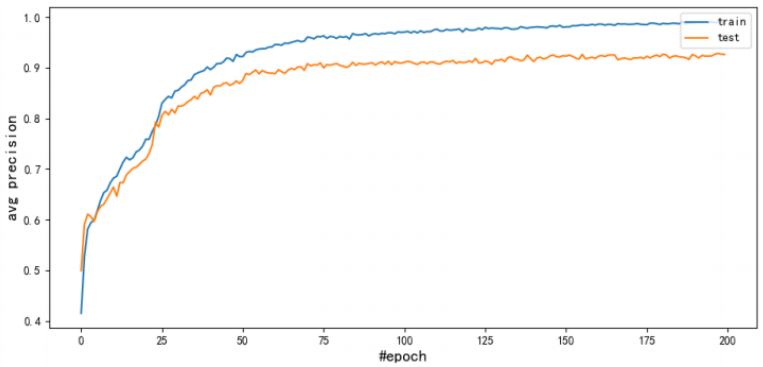


Figure 11. The change of average precision value of each label category with training epoch.

After the end of each epoch, the training state was detected by using the data of the training set and the test set. Each training sample was input into the model to obtain the prediction sequence, and the prediction sequence was compared with the real sequence to obtain the current recall of the model. The average value of recall of each label category changes with the training epoch, as shown in Figure 12.

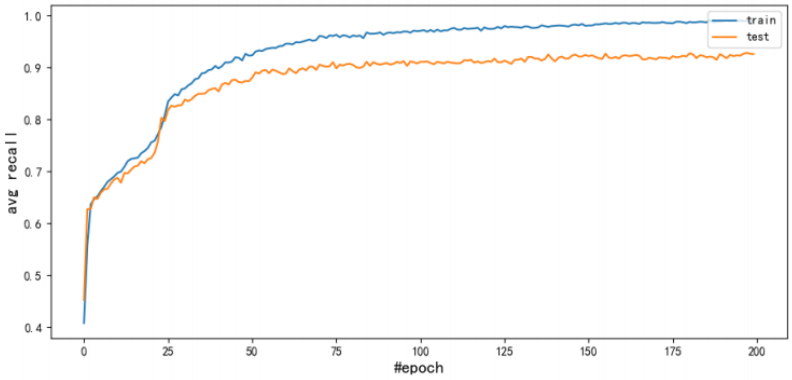


Figure 12. The change of average recall value of each label category with training epoch.

Precision and Recall can only reflect the performance of the model in a certain aspect, but not the overall performance. To better detect the real situation in the training process, at the end of each epoch, the current precision and recall rate were used to calculate the current F1 value. The change of F1 mean values of each label category with the training epoch is shown in Figure 13.

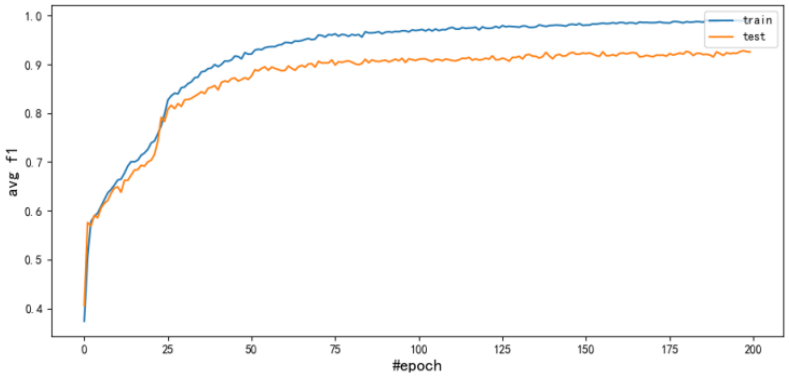


Figure 13. The change of average F1 value of each label category with training epoch.

After the model training, the confusion matrix visualization is used to visually display the model classification effect. The confounding matrix can better display the real and predicted category of the samples in the whole dataset. If all the samples in the dataset are correctly classified, the real and predicted categories of the samples will overlap diagonally in the confusion matrix. A sample that falls in a non-diagonal position is a misclassified sample.

Using the trained model to predict the training set, and the confusion matrix obtained is shown in Figure 14. It can be seen from the confusion matrix that the model has a good fitting effect on the training set, and few samples with predicted tags do not conform to the actual tags, which indicates that the capacity and expression ability of the model are sufficient and can perform well on predicting specific behavior.

After the model completely converges and the training is finished, the saved model is loaded from the hard disk, and the test dataset is input into the model to obtain the predicted value sequence of the model. Then, the predicted value sequence of the model is compared with the real tag value sequence to calculate the accuracy and generalization ability of the model. The corresponding evaluation indexes of labels of each category in the final model are shown in Table 4.

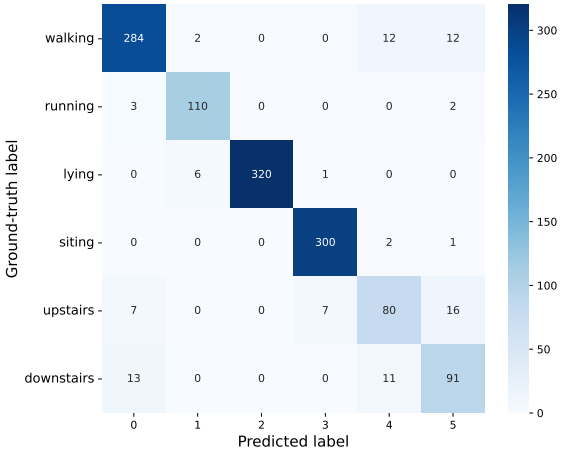


Figure 14. The confusion matrix obtained by using the MarFusion model to predict the training set.

Table 4. Final evaluation indicators corresponding to various labels.

Label	Precision	Recall	F1	Support
Walk	0.93	0.94	0.94	310
Run	0.94	0.98	0.98	115
Lying	1.0	0.98	0.99	327
Sit	0.99	0.99	0.99	303
Upstairs	0.73	0.81	0.77	110
Downstairs	0.78	0.70	0.74	115

As can be seen from the performance on the test set, the accuracy of the model on a certain label category is slightly affected by the number of samples corresponding to this label category. The reason for this may be that the number of samples in the dataset is relatively small, leading to a certain lack of fitting of the sensor data characteristics corresponding to a certain behavior tag. This situation can be improved in the future by increasing the size of the data.

Model Comparison

In order to better verify the accuracy of the Marfusion model in behavior recognition, the Marfusion model and existing classical models are trained together with the dataset. The final performance of the models is measured by Precision, Recall and F1 value. The following is a brief overview of the model used to compare Marfusion.

1. **Support vector machine (SVM)** [34]. SVM classifies data by obtaining a plane in linear space that can segment different types of data and maximize the distance from the data to the segmented plane. The goal of SVM is to maximize the interval, so it can be formalized to solve a convex quadratic programming problem. Support vector machine has the advantages of easy calculation and fast solution.
2. **Random forest** [35]. The Random forest classifies data by establishing multiple simple decision trees and gathering the opinions of all decision trees. The random forest classification algorithm fully shows the advantages of swarm intelligence, has the advantages of fast training speed and difficulty in over-fitting.
3. **Convolutional neural network (CNN)** [36]. CNN has an excellent performance in computer vision and image classification in recent years and has very strong feature extraction and self-learning ability. However, the CNN with too many layers may have gradient dispersion in the training process, which makes it difficult to converge. The CNN model used for comparison in this paper includes three convolution layers and one fully connected output layer.
4. **Recurrent neural network (RNN)** [37]. The RNN is a good solution to the extraction of contextual features in sequential data. LSTM, a variant of RNN, better solves the

problem of long-term memory by introducing a gating mechanism. It can carry out association and feature extraction for data with a long time span from the sequential data of a long sequence. The RNN model used for comparison in this paper includes two LSTM layers, two Batch Normalization layers and a fully connected output layer using Softmax as the activation function.

5. **DeepSense** [10]. By combining CNN and RNN, the DeepSense model solves the problem of feature extraction and noise interference in behavior recognition and has a good performance in different behavior recognition data sets.
6. **AttnSense** [11]. The AttnSense model uses CNN to extract spatial features, and also uses Gated Recurrent Unit (GRU), a variant of RNN, to extract temporal features, so as to better classify individual behaviors. In addition, the AttnSense model also introduces the Attention mechanism to better assign different weights to feature data from different sensors, which enhances the robustness and generalization performance of the model for individual behavior classification tasks.

In this paper, the above model is trained on our activity recognition dataset and compared with the Marfusion. Among them, Precision, Recall and F1 value are weighted mean according to the sample volume, and the comparison results are shown in Table 5.

Table 5. The evaluation indicators of each model obtained on the test set.

Model	Precision	Recall	F1
SVM	0.796	0.807	0.780
Random Forest	0.898	0.887	0.874
CNN	0.901	0.905	0.902
LSTM	0.892	0.861	0.860
DeepSense	0.903	0.905	0.903
AttnSense	0.914	0.916	0.914
Marfusion	**0.944**	**0.944**	**0.943**

Based on the above data, it can be concluded that the Marfusion model achieves better recognition effect in multimodal data collected by mobile phones. The reason may be that we have added the Attenion mechanism and weighted the values of each sensor. At the same time, we fuse multi-modal data to extract new features. In addition, we analyze the different contribution of each sensor as shown in Figure 15. Specifically, we average the weights of attention for each sensor and then normalized them for better comparison and visualization. The results indicate that linear acceleration and acceleration contribute more than other sensors. While magnetic field is the least useful sensor to recognize corresponding activity.

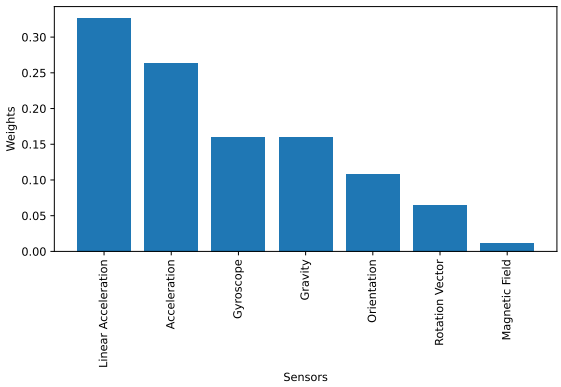


Figure 15. Attention weights of each sensor.

6.3. Discussion

The advantages of this experiment are as follows. We used MarSense to collect data without user intervention and build a new multimodal dataset. Moreover, based on this dataset, we developed the Marfusion. (1) Marfusion introduces the Attention mechanism. (2) Marfusion fuses multimodal data and extracts new features. Therefore, compared with other models, the identification precision of Marfusion on the multimodal dataset reached 0.944, and the F1 value also reached 0.943.

As for the open research questions with related proposed solutions, although our model has great performance on recognition, it has not been used in applications. Meanwhile, the participants in data collection are only students, which is one-fold in some way. Another shortcoming is the limited data collecting time.

As for the potential future work, we can combine the Marfusion and the MarSense to construct a recognition system for intelligent health systems, intelligent medical systems, motion monitoring systems, and aging systems. For example, in the field of the intelligent health system, MarSense will collect the sensor data by built-in sensors of mobile phones and using Marfusion to recognize the current activity of the user. If the user sits for a long time, our system will go off a beeping sound to remind him of taking a rest or standing up for a while. Furthermore, we will invite participants from different jobs, places, and age groups. For example, the workers that sit in their workstations for a long time and change their activities constantly will be involved in our experiment. Some elderly and children will also be invited to participate in the data collection to optimize the generalization of the model in real-world scenarios. Furthermore, we will extend the time of data collection.

7. Conclusions

In this study, We developed the MarSense experimental platform to collect and annotate user behavior data through smartphones. A large number of sensor data and behavior tags are collected that are closer to users' daily life scenarios. In addition, we design a human activity recognition model Marfusion, which is based on multimodal fusion and use the data to train the activity recognition model. Our proposed platform for data collection reduces user interference and has no restriction on user behavior, which becomes closer to real-world scenarios. Extensive empirical results on the collected dataset show that the Marfusion model can accurately predict the individual's current behavior state based on the data collected by smartphone sensors. In the future, we will improve our work by collecting more types of behavior labels and conducting data collection experiments among more individuals.

Author Contributions: Conceptualization, Q.S., S.G., Y.Z., Z.C., Z.M. and H.X.; methodology, Q.S. and Z.C.; software, Z.M. and S.G.; validation, Y.Z., Z.C. and H.X.; formal analysis, Q.S., S.G. and Z.M.; investigation, Y.Z. and H.X.; resources, Q.S. and Y.Z.; data curation, Z.C. and Z.M.; writing—original draft preparation, Y.Z. and H.X.; writing—review and editing, Q.S., Y.Z. and S.G.; visualization, H.X., S.G. and Z.C.; supervision, Q.S.; project administration, Q.S., Z.M. and S.G.; funding acquisition, Q.S., Z.M. and H.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China (62077027), the Ministry of Science and Technology of the People's Republic of China (2018YFC2002500), the Jilin Province Development and Reform Commission, China (2019C053-1), the Education Department of Jilin Province, China (JJKH20200993K), the Department of Science and Technology of Jilin Province, China (20200801002GH), and the European Union's Horizon 2020 FET Proactive project "WeNet-The Internet of us" (No. 823783).

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Jilin University.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, W.; Shen, Q.; Teso, S.; Lepri, B.; Passerini, A.; Bison, I.; Giunchiglia, F. Putting human behavior predictability in context. *EPJ Data Sci.* **2021**, *10*, 42. [CrossRef]
2. Intille, S. The Precision Medicine Initiative and Pervasive Health Research. *IEEE Pervasive Comput.* **2016**, *15*, 88–91. [CrossRef]
3. Hammerla, N.Y.; Fisher, J.; Andras, P.; Rochester, L.; Walker, R.; Plötz, T. PD disease state assessment in naturalistic environments using deep learning. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–29 January 2015.
4. Gao, Y.; Long, Y.; Guan, Y.; Basu, A.; Baggaley, J.; Ploetz, T. Towards reliable, automated general movement assessment for perinatal stroke screening in infants using wearable accelerometers. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2019**, *3*, 1–22. [CrossRef]
5. O'Brien, J.T.; Gallagher, P.; Stow, D.; Hammerla, N.; Ploetz, T.; Firbank, M.; Ladha, C.; Ladha, K.; Jackson, D.; McNaney, R.; et al. A study of wrist-worn activity measurement as a potential real-world biomarker for late-life depression. *Psychol. Med.* **2017**, *47*, 93–102. [CrossRef] [PubMed]
6. Yao, X.; Plötz, T.; Johnson, M.; Barbaro, K.D. Automated detection of infant holding using wearable sensing: Implications for developmental science and intervention. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2019**, *3*, 1–17. [CrossRef]
7. Nguyen, L.N.N.; Rodríguez-Martín, D.; Català, A.; Pérez-López, C.; Samà, A.; Cavallaro, A. Basketball activity recognition using wearable inertial measurement units. In Proceedings of the XVI International Conference on Human Computer Interaction, Vilanova i la Geltru, Spain, 7–9 September 2015.
8. Lee, M.L.; Dey, A.K. Sensor-based observations of daily living for aging in place. *Pers. Ubiquitous Comput.* **2015**, *19*, 27–43. [CrossRef]
9. Vrigkas, M.; Nikou, C.; Kakadiaris, I.A. A review of human activity recognition methods. *Front. Robot. AI* **2015**, *2*, 28. [CrossRef]
10. Yao, S.; Hu, S.; Zhao, Y.; Zhang, A.; Abdelzaher, T. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017.
11. Ma, H.; Li, W.; Zhang, X.; Gao, S.; Lu, S. AttnSense: Multi-level Attention Mechanism For Multimodal Human Activity Recognition. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019.
12. Zhang, M.; Sawchuk, A.A. USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, Pittsburgh, PA, USA, 5–8 September 2012.
13. De-La-Hoz-Franco, E.; Ariza-Colpas, P.; Quero, J.M.; Espinilla, M. Sensor-based datasets for human activity recognition—A systematic review of literature. *IEEE Access* **2018**, *6*, 59192–59210. [CrossRef]
14. Anguita, D.; Ghio, A.; Oneto, L.; Parra Perez, X.; Reyes Ortiz, J.L. A public domain dataset for human activity recognition using smartphones. In Proceedings of the 21th International European Symposium on Artificial Neural Networks, Bruges, Belgium, 24–26 April 2013.
15. Ramanujam, E.; Perumal, T.; Padmavathi, S. Human activity recognition with smartphone and wearable sensors using deep learning techniques: A review. *IEEE Sens. J.* **2021**, *21*, 13029–13040. [CrossRef]
16. Kim, E.; Helal, S.; Cook, D. Human activity recognition and pattern discovery. *IEEE Pervasive Comput.* **2009**, *9*, 48–53. [CrossRef]
17. Vaizman, Y.; Ellis, K.; Lanckriet, G. Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE Pervasive Comput.* **2017**, *16*, 62–74. [CrossRef]
18. Vaizman, Y.; Weibel, N.; Lanckriet, G. Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *1*, 1–22. [CrossRef]
19. Hernandez-Cruz, N.; Nugent, C.; Zhang, S.; McChesney, I. The Use of Transfer Learning for Activity Recognition in Instances of Heterogeneous Sensing. *Appl. Sci.* **2021**, *11*, 7660. [CrossRef]
20. Shen, Q.; Teso, S.; Zhang, W.; Xu, H.; Giunchiglia, F. Multi-modal subjective context modelling and recognition. *arXiv* **2020**, arXiv:2011.09671.
21. Ghods, A.; Cook, D.J. Activity2vec: Learning adl embeddings from sensor data with a sequence-to-sequence model. *arXiv* **2019**, arXiv:1907.05597.
22. Hammerla, N.Y.; Halloran, S.; Plötz, T. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv* **2016**, arXiv:1604.08880.
23. Ordóñez, F.J.; Roggen, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* **2016**, *16*, 115. [CrossRef]
24. Lv, M.; Xu, W.; Chen, T. A hybrid deep convolutional and recurrent neural network for complex activity recognition using multimodal sensors. *Neurocomputing* **2019**, *362*, 33–40. [CrossRef]
25. Otebolaku, A.; Enamamu, T.; Alfoudi, A.; Ikpehai, A.; Marchang, J.; Lee, G.M. Deep Sensing: Inertial and Ambient Sensing for Activity Context Recognition Using Deep Convolutional Neural Networks. *Sensors* **2020**, *20*, 3803. [CrossRef]
26. Sun, Z.; Ke, Q.; Rahmani, H.; Bennamoun, M.; Wang, G.; Liu, J. Human action recognition from various data modalities: A review. *arXiv* **2021**, arXiv:2012.11866.
27. Lahat, D.; Adali, T.; Jutten, C. Multimodal data fusion: An overview of methods, challenges, and prospects. *Proc. IEEE* **2015**, *103*, 1449–1477. [CrossRef]

28. Atrey, P.K.; Hossain, M.A.; El Saddik, A.; Kankanhalli, M.S. Multimodal fusion for multimedia analysis: A survey. *Multimed. Syst.* **2010**, *16*, 345–379. [CrossRef]
29. Wang, Y.C.; Shuang, H.Y. A data fusion-based hybrid sensory system for older people's daily activity and daily routine recognition. *IEEE Sens. J.* **2018**, *18*, 6874–6888. [CrossRef]
30. Vidya, B.; Sasikumar, P. Wearable Multi-sensor Data Fusion Approach for Human Activity Recognition using Machine Learning Algorithms. *Sens. Actuators A Phys.* **2022**, 113557. [CrossRef]
31. Han, J.; He, Y.; Liu, J.; Zhang, Q.; Jing, X. GraphConvLSTM: Spatiotemporal Learning for Activity Recognition with Wearable Sensors. In Proceedings of the 2019 IEEE Global Communications Conference, Waikoloa, HI, USA, 9–13 December 2019.
32. Islam, M.M.; Iqbal, T. Hamlet: A hierarchical multimodal attention-based human activity recognition algorithm. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 25–29 October 2020.
33. Chen, Y.; Zhong, K.; Zhang, J.; Sun, Q.; Zhao, X. LSTM networks for mobile human activity recognition. In Proceedings of the 2016 International Conference on Artificial Intelligence: Technologies and Applications, Bangkok, Thailand, 24–25 January 2016.
34. Schuldt, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 23–26 August 2004.
35. Liaw, A.; Wiener, M. Classification and regression by randomForest. *R News* **2004**, *2*, 18–22.
36. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
37. Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit. Lett.* **2019**, *119*, 3–11. [CrossRef]