DeepFusion: A Deep Learning Framework for the Fusion of Heterogeneous Sensory Data

Hongfei Xue[1], Wenjun Jiang[1], Chenglin Miao[1], Ye Yuan[2], Fenglong Ma[1], Xin Ma[1], Yijiang Wang[1], Shuochao Yao[3], Wenyao Xu[1], Aidong Zhang[4], Lu Su[1*]

[1] State University of New York at Buffalo, Buffalo, NY USA [2] Beijing University of Technology, Beijing, China
[3] University of Illinois at Urbana-Champaign, Urbana, IL USA [4] University of Virginia, Charlottesville, VA USA
Email: [1] {hongfeix, wenjunji, cmiao, fenglong, xma24, yijiangw, wenyaoxu, lusu}@buffalo.edu,
[2] yuanye91@emails.bjut.edu.cn, [3] syao9@illinois.edu, [4] aidong@virginia.edu

ABSTRACT

In recent years, significant research efforts have been spent towards building intelligent and user-friendly IoT systems to enable a new generation of applications capable of performing complex sensing and recognition tasks. In many of such applications, there are usually multiple different sensors monitoring the same object. Each of these sensors can be regarded as an information source and provides us a unique "view" of the observed object. Intuitively, if we can combine the complementary information carried by multiple sensors, we will be able to improve the sensing performance. Towards this end, we propose DeepFusion, a unified multi-sensor deep learning framework, to learn informative representations of heterogeneous sensory data. DeepFusion can combine different sensors' information weighted by the quality of their data and incorporate cross-sensor correlations, and thus can benefit a wide spectrum of IoT applications. To evaluate the proposed DeepFusion model, we set up two real-world human activity recognition testbeds using commercialized wearable and wireless sensing devices. Experiment results show that DeepFusion can outperform the state-of-the-art human activity recognition methods.

CCS CONCEPTS

• **Computing methodologies → Supervised learning by classification**; • **Computer systems organization → Sensor networks**.

KEYWORDS

Sensor Fusion, Deep Learning, Internet of Things

ACM Reference Format:
Hongfei Xue[1], Wenjun Jiang[1], Chenglin Miao[1], Ye Yuan[2], Fenglong Ma[1], Xin Ma[1], Yijiang Wang[1], Shuochao Yao[3], Wenyao Xu[1], Aidong Zhang[4], Lu Su[1]. 2019. DeepFusion: A Deep Learning Framework for the Fusion of Heterogeneous Sensory Data. In *The Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '19), July 2–5, 2019, Catania, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3323679.3326513

*Lu Su is the corresponding author.

Mobihoc '19, July 2–5, 2019, Catania, Italy

ACM ISBN 978-1-4503-6764-6/19/07…$15.00
https://doi.org/10.1145/3323679.3326513

1 INTRODUCTION

In recent years, significant research efforts have been spent towards building intelligent and user-friendly IoT (Internet of Things) systems upon interconnected computing and sensing devices, so as to enable a new generation of applications capable of performing complex sensing and recognition tasks that can support a new realm of interactions between humans and their physical surroundings. In many of such applications, there are usually multiple different sensors monitoring the same object. Consider a real world health-care scenario where smart sensors are deployed to monitor the daily activities of patients, not only the devices carried by the monitored subject, such as smartphone, smart watch, and smart glasses, but also the wireless devices placed in the room where the subject is living, such as laptop and iPad, can provide useful information characterizing the subject's activities. Each of these devices can be regarded as an information source and provides us a unique "view" of the observed subject. Intuitively, if we can combine the complementary information carried by multiple sensors, we will be able to improve the activity recognition accuracy.

However, to unleash the power of multi-sensor information, we have to address a series of challenges. First, different sensors may provide *heterogeneous data*. On one hand, different modalities of sensory data (e.g., acceleration readings, WiFi signal, ultrasound, and visible light) may be collected concurrently for the recognition of same activities. On the other hand, difference sensors may have different data collecting patterns (e.g., sampling rate, transmission rate, signal strength), and this will add further heterogeneity to the information extracted from different devices. Second, different sensors may carry different amount of information, due to various reasons such as the quality of hardware, the location where the sensor is placed, as well as background noise. An ideal data fusion approach should be able to capture the variance in data quality among the sensors and rely on more informative ones. Third, the information provided by different sensors may be correlated with each other, and thus it is necessary to capture and incorporate such cross-sensor correlation into the data fusion model.

To tackle the above challenges, we propose to adopt deep learning techniques, which have been proved to be effective on noisy and heterogeneous big data. In this paper, we develop a deep learning framework, named DeepFusion, to integrate heterogeneous sensory data. In this model, we utilize a CNN-based Sensor-Representation module to preserve the unique characteristics of

each sensor view while uniform the dimensionality of heterogeneous inputs. A Weighted-Combination module is developed to estimate the quality of information contributed by each sensor and combine multi-sensor features in a weighted manner. We also construct a Cross-Sensor module to extract and incorporate cross-sensor correlation features into our model. Taking the advantage of multi-sensor structure, the proposed model is able to not only make full use of data collected by the sensors with different levels of quality, but also characterize different patterns of the data across different sensor views.

In this paper, we will use a representative sensing task: Human Activity Recognition (HAR), which plays an important role in a wide range of real-world IoT applications such as smart home, health care and fitness tracking, as an example to illustrate our proposed DeepFusion framework. Though being focused on the HAR application in our presentation, the proposed framework can be generalized to benefit many other IoT applications that involve classification or recognition tasks. In order to evaluate the proposed DeepFusion framework, we conduct extensive real-world experiments on both wearable device based and device-free human activity recognition applications. Experimental results demonstrate that our model outperforms the state-of-the-art algorithms significantly, which illustrates the effectiveness of the proposed DeepFusion model. We summarize the contributions of this paper as follows:

- We identify the opportunities as well as challenges in the fusion of heterogeneous multi-sensor data.
- We propose DeepFusion, a unified multi-sensor deep learning framework, to learn informative representations of heterogeneous sensory data. DeepFusion can combine different sensors' information weighted by the quality of their data and incorporate cross-sensor correlations to improve the sensing performance.
- We set up two testbeds using COTS (i.e., commercial off-the-shelf) wearable (smartphone, smart watch, and Shimmer sensor[1]) and wireless (WiFi and acoustic) sensing devices, and collect real-world human activity data. We empirically show that the proposed DeepFusion model can effectively recognize activities and outperform the state-of-the-art human activity recognition methods on the collected dataset.

2 SYSTEM OVERVIEW

In this section, we provide an overview of the DeepFusion system using human activity recognition as an illustrative example. The DeepFusion system takes the raw sensing data (e.g., accelerometer data, WiFi/Ultrasound signals) as the input and outputs inferred activities of the monitored subject. As shown in Figure 1, DeepFusion consists of three major components: (1) data collection, (2) data preprocessing, and (3) classification.

2.1 Data Collection

The major function of the data collection component is to collect various raw sensory data from heterogeneous wearable sensors or wireless sensing devices and forward them to the data preprocessing component. In our prototype DeepFusion system, we build two testbeds using COTS wearable (smartphone, smart watch, and Shimmer sensor) and wireless (WiFi and acoustic) sensing devices to collect human activity data. In real practice, a subject may be equipped with multiple wearable sensing devices, and different wireless sensing devices may be placed in different places of an indoor environment. Each of these devices can be regarded as an information source that monitors the observed subject.

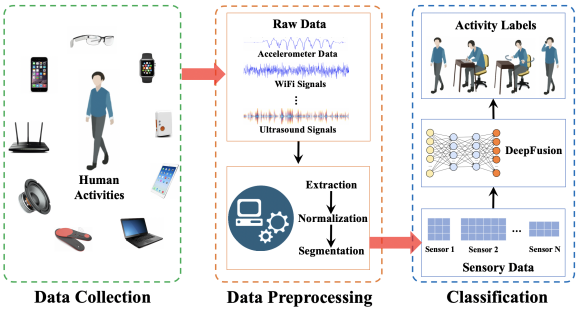

Figure 1: The Overview of DeepFusion System.

2.2 Data Preprocessing

The function of this component is to generate the data that can be directly fed to the proposed deep learning model. After being collected, the raw sensory data or signals are stored in some specifically formatted files (e.g., wav format for audio data) by the sensing devices. The raw data are first extracted and aligned from those files. Then to remove noise, we take several procedures to preprocess the extracted data, such as resizing, normalization, standardization, etc. After that, to generate a dataset with a predefined data size, we also segment the data into non-overlapping pieces and make Fourier transform. The generated dataset is then fed to the proposed deep learning model for classification.

2.3 Classification

The processed activity data are still very complex, i.e, high-dimensional, noisy and heterogeneous. This makes it extremely difficult for traditional machine learning algorithms to characterize the underlying patterns of such data. To address this challenge, we make use of deep learning techniques which have been proved effective for extracting representations from complex data. In particular, we propose a deep learning framework that not only considers the quality of data collected from different sensors, but also incorporates the relationships among different sensor views. The details of the proposed DeepFusion model are described in Section 3. With the processed heterogeneous sensory data, our model is able to significantly improve the recognition performance by learning informative representations of different human activities.

3 METHODOLOGY

In this section, we introduce DeepFusion, a unified multi-sensor deep learning framework for human activity recognition with heterogeneous data as inputs. The architecture of the proposed model is illustrated in Figure 2. It includes three modules: Sensor-Representation (SR) Module, Weighted-Combination (WC) Module and Cross-Sensor (CS) Module. In the following subsections, we detail these modules, respectively.

For the rest of this paper, all matrices and tensors are denoted by bold upper-case characters (e.g., weight matrix $\mathbf{W}$), vectors

[1]http://www.shimmersensing.com/

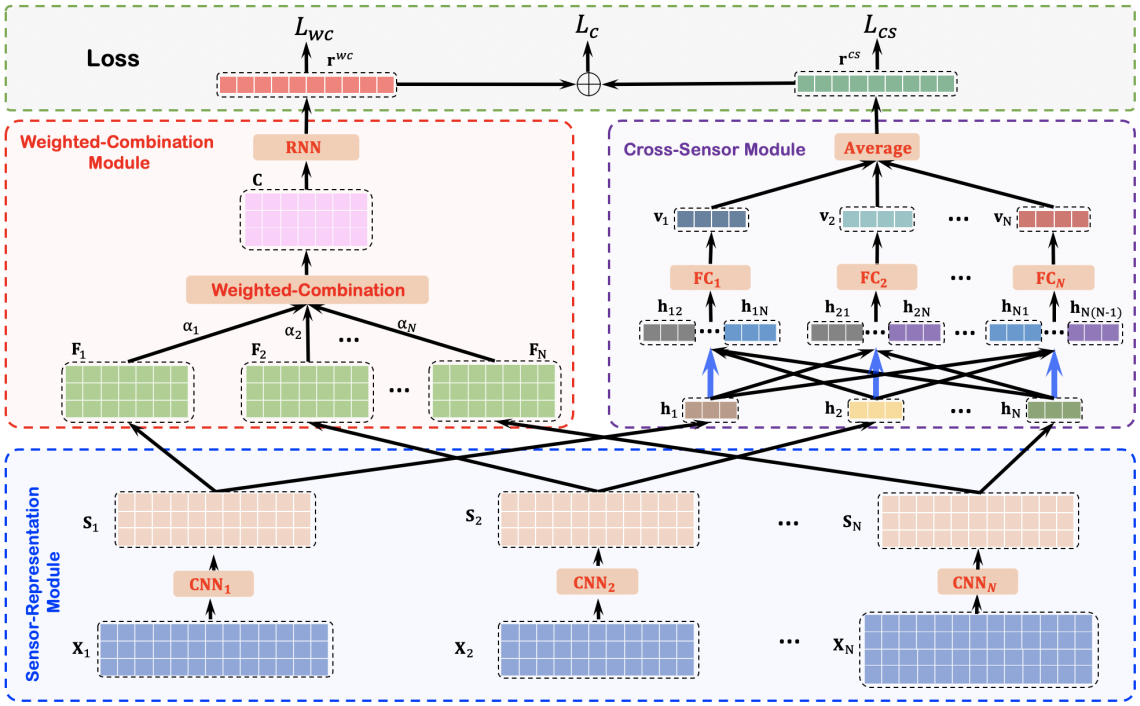

Figure 2: The Architecture of DeepFusion Model.

are represented by bold lower-case characters (e.g., bias vector **b**), and scalar numbers are denoted by regular characters (e.g., hyper-parameter a).

3.1 Sensor-Representation Module

In practice, the collected multi-sensor data can be represented as a collection of heterogeneous continuous time series consisting of several non-uniformly sampled signals. In order to *preserve the unique characteristics of each sensor while uniform the dimensionality of different sensors' inputs*, we present a CNN-based module to learn sensor-representations from raw heterogeneous data. The efficiency and effectiveness of CNN block make it an ideal building component for our activity recognition framework. The CNN blocks are able to transform the raw data into a low dimensional space representation with the same size, meanwhile enhance the ability of feature expression. In CNN, the convolutional layers with sets of flexible filters and strides are the most important parts for reshaping the data, and the pooling layers can also be appended to down-sampling the input data. In our model, we need to set several parameters when constructing the stacked CNN blocks for different sensors. Specifically, we assume that the input data contains N sensing node $\{\mathbf{X}_1, \cdots, \mathbf{X}_i, \cdots, \mathbf{X}_N\}$. Given the input data of the i-th sensing node, denoted as $\mathbf{X}_i$, we first obtain its sensor representation through stacked CNN blocks (i.e., CNN_i), as follows:

$$\mathbf{S}_i = \mathrm{CNN}_i(\mathbf{X}_i; \theta_i),$$

where function CNN_i denote the stacked CNN blocks specified for each input matrix $\mathbf{X}_i$, and θ_i is the parameters to be learned for the corresponding stacked CNN blocks. In our model, the activation function *ReLU*, batch normalization and dropout technique are also leveraged in CNN blocks. The sensor representations extracted from all the sensing nodes, denoted as $\{\mathbf{S}_1, \ldots, \mathbf{S}_i, \ldots, \mathbf{S}_N\}$, are used to represent the heterogeneous inputs in a high-level feature space with the same size.

Subsequently, in order to unify the dimensions of features as inputs for the other two modules, we further reformat the learned hidden representations using *pooling* and $flatten$ operations. Regarding the input of Weighted-Combination Module, on one hand, we use the *pooling* operation as follows:

$$\mathbf{F}_i = \mathrm{Pooling}(\mathbf{S}_i).$$

The *pooling* operation helps us reduce the input feature size into a proper one to calculate numeral weights, which are then fed to *RNN*. For the Cross-Sensor Module, on the other hand, we adopt the $flatten$ operation to obtain a cross-sensor input vector $\mathbf{h}_i$ of the i-th sensor, as follows:

$$\mathbf{h}_i = \mathrm{Flatten}(\mathbf{S}_i).$$

In this way, even though the dimensionality of raw data is different across sensors, the proposed model is able to adopt the corresponding CNN blocks for different sensors with various sets of parameters to uniform the heterogeneous inputs. It finally leads the model to learning informative sensor representations.

3.2 Weighted-Combination Module

Different sensors may carry different amount of information, due to various reasons such as the type of the sensing signal, the quality of hardware, the distance and angle to the observed object, as well as the ambient noise and setting. An ideal activity recognition approach should be able to *capture the variance in data quality among different sensors and rely on more informative ones to achieve better*

performance. Toward this end, we propose a Weighted-Combination module to estimate the quality of information (referred to as *quality weight*) contributed by each sensor and combine multi-sensor information in a weighted manner.

The basic idea of our Weighted-Combination module is derived based on the attention mechanism [10, 13, 18, 37, 46]. The attention mechanism is a weighted aggregation method that is widely used for the application of machine translation [5], computer vision [14, 35], and disease prediction [29, 49]. However, traditional attention mechanism is based on the assumption that *only a few views are related to the task goal*. As a result, it tends to assign close-to-zero weight to most of the views. In the scenario of human activity recognition, however, such assumption does not hold, since *a significant portion of the sensors may provide informative observations*.

To address this challenge, we propose a new weight-assignment strategy to fully utilize the multi-sensor information. Given the learned weight-combination input matrices $\{\mathbf{F}_1, \cdots, \mathbf{F}_i, \cdots, \mathbf{F}_N\}$, to calculate the quality weight of each sensor, we first employ a pooling layer and the flatten operation to obtain their encoding vectors $\{\mathbf{u}_1, \cdots, \mathbf{u}_i, \cdots, \mathbf{u}_N\}$, respectively. In particular, the encoding vector of the i-th sensor $\mathbf{u}_i$ can be obtained by:

$$\mathbf{u}_i = \text{Flatten}(\mathbf{F}_i) .$$

Next, the quality weight of the i-th sensor e_i can be calculated using the following formula:

$$e_i = \left(\mathbf{w}^{wc\intercal}\mathbf{u}_i + b^{wc}\right)/l_{\mathbf{w}^{wc}} , \tag{1}$$

where $\mathbf{w}^{wc}$ and b^{wc} are the parameters to be learned, and $l_{\mathbf{w}^{wc}}$ denotes the length of the encoding vector $\mathbf{u}_i$. According to Eq. (1), we can then obtain a normalized quality weight α_i, as follows:

$$\alpha_i = \frac{\tilde{\alpha_i}}{\sum_j \tilde{\alpha_j}} ,$$

where $\tilde{\alpha_i}$ denotes the rescaled quality weight using a sigmoid-based function:

$$\tilde{\alpha_i} = \frac{a}{1+\exp\left(-e_i/b\right)} + c , \tag{2}$$

where a, b, and c are the predefined hyper-parameters. In Eq. (2), the upper-bound value and lower-bound value of the rescaled weights are $a + c$ and c, respectively. b determines the slope of the function near zero value. In such a way, the variance of normalized quality weights among all the sensors can be reduced by setting appropriate hyper-parameters. Thus, our model can incorporate more sensors to recognize activities. Based on the normalized quality weights of all the sensors $\{\alpha_1, \cdots, \alpha_i, \cdots, \alpha_N\}$, the sensor combination matrix $\mathbf{C}$ can be calculated using weighted aggregation:

$$\mathbf{C} = \sum_i \alpha_i \odot \mathbf{F}_i , \tag{3}$$

where $\odot$ denotes the element-wise multiplication.

To further represent sensor combination as a vector, we applied a 2-layer stacked Gated Recurrent Unit (GRU) to calculate the output vector of the Weighted-Combination Module $\mathbf{r}^{wc}$, as follows:

$$\mathbf{H}_{1:L} = \text{GRU}(\mathbf{C}_{1:L}; \phi) , \tag{4}$$

$$\mathbf{r}^{wc} = \sum_l \mathbf{H}_l , \tag{5}$$

where L denotes the column length of matrix, $\mathbf{H}_l$ is the output vectors of GRU, and ϕ denotes all the parameters of GRU. Then $\mathbf{r}^{wc}$ is obtained by adopting the summation operation on the output vectors of GRU. In this way, our model can fully utilize the multi-sensor information by capturing the variance among different senros and relying on more informative ones.

3.3 Cross-Sensor Module

The above module, though being able to combine multi-sensor information, treats each sensor independently and ignores the correlations among them. In the task of human activity recognition, different sensors may provide correlated information. If we can capture and incorporate the cross-sensor correlation into the deep learning framework, we will be able to recognize more general patterns and further improve model performance. Clearly, simply concatenating the raw input features together may not be enough for models to yield robust and accurate results. To tackle this challenge, we propose a Cross-Sensor module to provide complementary information to the weighted-combination features.

In this module, given the cross-sensor input vectors $\{\mathbf{h}_1, \cdots, \mathbf{h}_i, \cdots, \mathbf{h}_N\}$, the correlation vector of the i-th sensor, denoted as $\mathbf{v}_i$, can be obtained as follows:

$$\mathbf{v}_i = f\left(\mathbf{W}_i^{v\intercal}[\tilde{\mathbf{h}}_{i,1} \oplus \tilde{\mathbf{h}}_{i,2} \oplus \cdots \oplus \tilde{\mathbf{h}}_{i,N}] + \mathbf{b}_i^v\right) ,$$

where $\oplus$ denotes the concatenation operator, $\mathbf{W}_i^v$ and $\mathbf{b}_i^v$ are the learnable parameters of FC_i^{cv} (i.e., a single layer fully connected neural network), and $\tilde{\mathbf{h}}_{i,j}$ denotes the correlation between sensor j and sensor i measured by element-wise difference [30]:

$$\tilde{\mathbf{h}}_{i,j} = \mathbf{h}_j - \mathbf{h}_i .$$

Note that there are $N-1$ corresponding correlations in total, since the self-correlation is ignored.

Based on the correlation vectors $\{\mathbf{v}_1, \cdots, \mathbf{v}_i, \cdots, \mathbf{v}_N\}$, we can then obtain the output vector of the Cross-Sensor Module $\mathbf{r}^{cv}$ by adopting the averaging operation:

$$\mathbf{r}^{cv} = \frac{1}{N}\sum_i \mathbf{v}_i .$$

In this way, we can capture the correlations across multi-sensor sensing data in a low dimensional space. Different from the Weighted-Combination Module using weighted combination operation to merge features among sensors, here we use averaging operation since each correlation vector encodes the correlation of sensors. Thus, it is unnecessary to distinguish the importance among them.

3.4 Model Objective

Given the output vectors of the Weighted-Combination and Cross-Sensor Modules, we concatenate them together and use the softmax layer to derive the prediction of the j-th training sample:

$$\tilde{\mathbf{y}}_j^c = \text{softmax}\left(\mathbf{W}^{s\intercal}\left[\mathbf{r}_j^{wc} \oplus \mathbf{r}_j^{cv}\right] + \mathbf{b}^s\right) ,$$

where $\mathbf{W}^s$ and $\mathbf{b}^s$ are the parameters to be learned. We then use cross-entropy to calculate the loss, the overall loss $\mathcal{L}_c$ is defined as:

$$\mathcal{L}_c = -\sum_j \left(\mathbf{y}_j \log\left(\tilde{\mathbf{y}}_j^c\right) + (1-\mathbf{y}_j)\log\left(1-\tilde{\mathbf{y}}_j^c\right)\right) ,$$

where $\mathbf{y}_j$ is the ground truth of the j-th training sample.

Note that the quality of sensor representations is highly dependent on the training of the Weighted-Combination Module. Since the Cross-Sensor Module only extract the correlations among sensors, the quality of correlation vectors rely highly on the quality of sensor representations. Facing the complicated module relationships, we take into account the local loss of each module to balance their interactions in the unified training procedure.

The local loss of the Weighted-Combination Module $\mathcal{L}_{wc}$ using cross-entropy is defined as:

$$\tilde{\mathbf{y}}_j^{wc} = \text{softmax}\left(\mathbf{W}^{y^{wc}\top}\mathbf{r}_j^{wc} + \mathbf{b}^{y^{wc}}\right) ,$$

$$\mathcal{L}_{wc} = -\sum_j \left(\mathbf{y}_j \log\left(\tilde{\mathbf{y}}_j^{wc}\right) + (1-\mathbf{y}_j)\log\left(1-\tilde{\mathbf{y}}_j^{wc}\right)\right) .$$

The local loss of the Cross-Sensor Module $\mathcal{L}_{wc}$ is defined as:

$$\tilde{\mathbf{y}}_j^{cs} = \text{softmax}\left(\mathbf{W}^{y^{cs}\top}\mathbf{r}_j^{cs} + \mathbf{b}^{y^{cs}}\right) ,$$

$$\mathcal{L}_{cs} = -\sum_j \left(\mathbf{y}_j \log\left(\tilde{\mathbf{y}}_j^{cs}\right) + (1-\mathbf{y}_j)\log\left(1-\tilde{\mathbf{y}}_j^{cs}\right)\right) .$$

To train a unified model, the final loss function of our DeepFusion model $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathcal{L}_c + \beta\mathcal{L}_{wc} + \gamma\mathcal{L}_{cv} , \tag{6}$$

where β and γ are hyper-parameters to control the contribution of each local loss function.

4 EXPERIMENTS

In this section, we use real-world human activity recognition testbeds to evaluate our proposed DeepFusion model. We first introduce the state-of-the-art human activity recognition approaches as baselines. Then, we conduct experiments on human activity data collected from two testbeds using COTS wearable (smartphone, smart watch, and Shimmer sensor) and wireless (WiFi and acoustic) sensing devices.

4.1 Baselines

To fairly evaluate the performance of the proposed DeepFusion model, we use the following models as baselines:

SVM *[11].* Support Vector Machine (SVM) is a widely adopted supervised machine learning model. There are some studies [41, 45, 50] employing SVM model for human activity recognition task. Since standard linear SVM model is a binary classifier, we use one-vs-all SVM for our multi-class classification task. In the experiments, we flatten data from all sensors into a single feature vector and feed it to the SVM model.

DeepSense *[47].* DeepSense is the state-of-the-art deep learning model for the classification of multi-sensor data. The architecture of DeepSense includes three layers of local CNN, three layers of global CNN and two layers of GRU. In our experiments, we follow the settings of original paper. Specifically, on each convolutional layer, the number of filters is 64 and the size of filters is set to 3×3. In addition, dropout and batch norm technologies are also used.

DeepFusion Variants. The proposed DeepFusion model considers not only the varying contributions of different sensors, but also the correlation across sensors. There are three modules in the proposed DeepFusion, including the Sensor-Representation module (referred to as SR), the Weighted-Combination module (referred to as WC) and Cross-Sensor module (referred to as CS). Thus, we propose two simplified models as baselines:

- **SR+Avg**. we use the Sensor-Representation module to obtain the representations of sensory data, and then average all the sensor representations. Moreover, a fully connected layer is employed to reduce its dimension. Finally, we use the reduced representation to make predictions.
- **SR+WC**. In this baseline, we use the Sensor-Representation module and the Weighted-Combination module together to recognize human activities.

4.2 Experiments On Wearable Sensor Data

In this section, we evaluate the proposed DeepFusion model based on a real-world activity dataset collected from multiple wearable sensors placed on different body areas.

Experiment Setups. We use three types of wearable sensing devices in this experiment: smartphone, smart watch and Shimmer sensor. Each device is regarded as a sensor that contains a triaxial accelerometer, a triaxial gyroscope and a triaxial magnetometer. We employ six volunteers (including both men and women) as the subjects to collect the data. Each subject wore four sensors (two smartphones, one smart watch and one Shimmer sensor) on four different regions of his/her body, i.e., left upper arm, left waist, right wrist and right ankle as shown in Figure 3. We consider 27 activities and the list of them are shown in Table 1. Each subject is asked to conduct each activity for one minute.

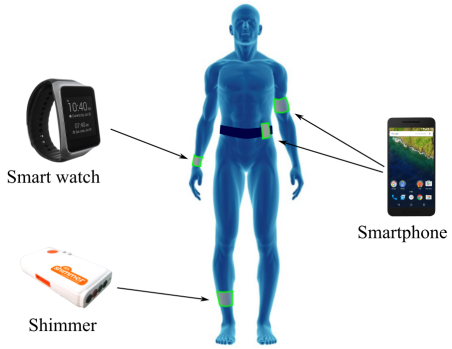

Figure 3: Experiment Setup for Wearable Sensor Data Collection.

Data Preprocessing. The data collected from each sensing node contain 9 signals, i.e., three axes of the accelerometer, three axes of the gyroscope and three axes of the magnetometer. Although different sensors have different sampling rate and value range, they were all downsampled to $25Hz$, and then scaled to values between 0.0 and 1.0 according to their magnitudes. The data is segmented without overlap via a window of 2 seconds, which corresponds to 50 points. Then each segment is combined with the FFT of itself as the input of the deep learning model. Therefore the final size of each data segment in one sensing node is $9 \times 50 \times 2$.

Table 1: Activities in Wearable Sensor Dataset.

ID	Activities	ID	Activities	ID	Activities
1	running	10	going downstairs and making a phone call	19	standing and washing hands
2	running in place	11	going upstairs	20	standing and wiping the blackboard
3	sitting and making a phone call	12	going upstairs and making a phone call	21	standing and wiping the table
4	sitting and keyboarding	13	standing and making a phone call	22	standing and writing
5	sitting and typing on the phone	14	standing and washing the dishes	23	standing and writing on the blackboard
6	sitting still	15	standing and keyboarding	24	walking backward
7	sitting and wiping the table	16	standing and typing on the phone	25	walking and making a phone call
8	sitting and writing	17	standing still	26	walking forward
9	going downstairs	18	standing and brushing the teeth	27	walking in place

For the traditional classification model SVM, we extract 36 features for each of the accelerometer, gyroscope and magnetometer signal on each sensor, which amounts to 432 features in total. These 36 features include mean, standard deviation, median absolute deviation (MAD), median, maximum, minimum, energy, signal magnitude area, interquartile range of each of the x, y, and z axes, as well as the magnitude of each signal, the angles between each signal and its three axes, pair-wise correlation of the axes, the energy of each signal, and each signal's magnitude area.

Model Settings. For the experiments on **wearable sensor** data, 6 CNN blocks are stacked to form a sensor representation extractor for each sensing node. For the convolutional layers, the sizes of filters are set to 3×5, 3×3, 3×3, 3×3, 1×3 and 1×3 respectively. The number of filters on each convolutional layer is set to 64. There is no padding operation in first four CNN blocks. And the max-pooling is leveraged to reduce the data size. For the GRU, the size of hidden state is set to 64. For fully connected neural networks in the Cross-Sensor Module, the size of condensed correlation vector $\mathbf{v}_i$ is set to 64. All the activation functions are *ReLU*. The dropout rate for CNN is set to 0.8 and is 0.7 for RNN. For the hyper-parameters, we set $a = 9.0$, $b = 0.01$, $c = 10.0$, $\beta = 0.1$ and $\gamma = 0.1$.

During the training process, ADAM optimization algorithm [22] is used to optimize the parameters. The learning rate is $1e-4$. Batch size we used is 100. We use the accuracy score as our performance criteria. We implement the proposed DeepFusion model using Tensorflow [1].

Performance Validation. In this experiment, we use left-one-subject-out strategy for the evaluation dataset, and the average accuracy score of all subjects is calculated as the performance measure. Table 2 shows the accuracy of all the approaches on the wearable sensor data. We can observe that the proposed DeepFusion model achieves the best performance. The performance of traditional classification approach SVM is the worst. This demonstrates the effectiveness of deep learning models for HAR task. Among the three deep learning baselines, the accuracy of SR+WC is the highest. The reason is that SR+WC takes into account the quality of the data from different sensors, while DeepSense and SR+Avg can not capture the variance in data quality across the sensors. The proposed DeepFusion models not only different sensors' weights but also the relationships among sensors, and thus it can achieve the best performance.

Learned Weight Analysis. In this experiment, we select 3 activities (4, 9 and 15 in Table 1) as case study samples to analyze the weights learned by our DeepFusion model. Table 3 shows the learned weight of each sensor. As we can see, for the 4-th activity, i.e., *sitting and keyboarding*, the sensor on the right wrist is assigned the highest weight. This is because the subjects' arms were held by the table and their legs were kept still, and the wrist has more movements than other body areas. The 9-th activity, i.e., *going downstairs*, has more effect on the sensor deployed on the ankle, and thus this sensor is assigned with the highest weight. As for the 15-th activity, i.e., *standing and keyboarding*, it is different from the activity of *sitting and keyboarding*. The subjects need to keep their arms in the air and the sensors on the wrist and the upper arm are both affected more. Therefore, the model assigns more weights to the sensors on wrist and arm. The above experiment results illustrate that the proposed DeepFusion model can dynamically adjust sensor weights on different activities and thus can achieve better performance than the approaches that treat all the sensors equally.

Table 2: Performance on the Wearable Sensor Data.

Model	Accuracy
SVM	0.350
DeepSense	0.862
SR+Avg	0.835
SR+WC	0.870
DeepFusion	**0.905**

Comparison with DeepSense Model. Since DeepSense model [47] and our model are both deep learning methods, we further analyze the two models from the perspectives of computational complexity and latency. For both models, over 96% of the calculations are conducted in CNN part. In our model, we reduce the size of CNN feature maps layer by layer, and thus dramatically reduce the needed calculations. To be specific, we compare the numbers of multiplication operations in both DeepSense model and our model. For each testing instance, the DeepSense model needs to conduct about $341M$ multiplication operations while our DeepFusion model only needs about $56M$ multiplication operations, which is much smaller than DeepSense. As for the latency evaluation, we conduct thousands of model inferences to estimate the average inference time on the same GPU. For DeepSense, it needs about $36ms$ to finish one batch. In contrast, DeepFusion only needs around $24ms$ to finish the same batch. So our model also has less latency than DeepSense.

Table 3: The Learned Weights on the Wearable Sensor Data

Activity ID	4	9	15
Smart Watch on Right Wrist	**0.276**	0.228	**0.273**
Shimmer on Right Ankle	0.236	**0.272**	0.217
Smartphone on Left Upper Arm	0.245	0.256	0.260
Smartphone on Left Waist	0.243	0.243	0.250

4.3 Experiments On Device-Free Human Activity Data

Although the device-based approaches provide an effective way to monitor the human activities, they have many limitations due to the extra burden and discomfort brought to the users who wear them. To address this challenge, significant efforts are recently made to explore device-free human activity recognition techniques that utilizethe information collected by existing indoor wireless infrastructures without the need for the monitored subject to carry a dedicated device. The philosophy behind these approaches is that the activities of a person located between a pair of wireless devices (e.g., smartphone, laptop, WiFi access point) can be infered by extracting and analyzing the information carried by the wireless signal transmitted between the sender and receiver. Here each pair of the sender and receiver provides us a unique "view" of the observed subject.

In this section, we conduct the experiments on the device-free human activity data collected from a real-world testbed. Specifically, we show the experimental results and analyze the learned weights on two datasets. The one is a homogeneous WiFi signal dataset, and the other is a heterogeneous dataset containing both WiFi and ultrasound signals.

Experiment Setups. In this experiment, we consider seven different human activities, which are shown in Table 4. There are eight subjects (including both men and women) involved in the data collection, and each subject is asked to take each activity for 51 seconds, and all the activities are repeated for 2 rounds.

We collect two kinds of signals: WiFi and Ultrasound. To collect **WiFi signals**, we use a TP-Link AC3150 Wireless WiFi Gigabit Router (Archer C3150 V1) to send packets to different receivers at a constant packet transmission rate, i.e., 30 packets per second, which is a typical transmission rate in practical wireless communication scenarios. Each receiver is configured with Intel Wireless Link 5300 NIC, Ubuntu 11.04 LTS with 2.2.36 kernel, and Linux 802.11n CSI extraction toolkit provided in [16]. For both 2.4 GHz and 5 GHz radio bands, the Linux 802.11n CSI extraction toolkit can report the CSI matrices of 30 sub-carriers. Figure 4 pictures the experiment environment, where the positions of both transmitter and part of receivers are marked.

In order to collect **ultrasound signals**, we use an Apple iPad mini 4 as the sound generator, which transmits near-ultrasound (i.e., 19 KHz) signals toward the subject. Since the sampling rate of the MICs on smartphones can reach as high as 44.1 KHz, we can use smartphones as receivers. In our experiments, three Huawei Nexus 6P's are used as receivers to record the ultrasound signals reflected by the body of the subject. These receivers are deployed at different positions in the room.

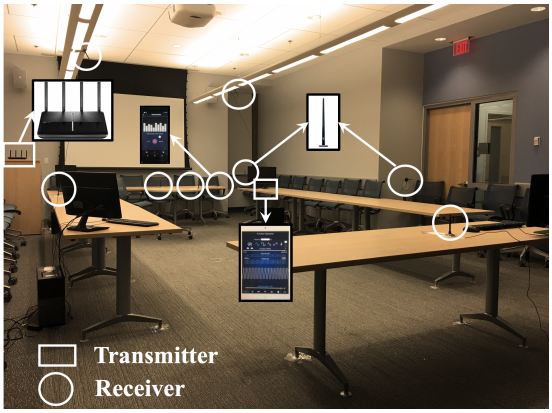

Figure 4: Experiment Setup of DeepFusion Human Activity Recognition System. (Note that not all the receivers are covered in this image.)

Data Preprocessing. Since the proposed DeepFusion model is able to deal with heterogeneous data, in the experiments, we include two kinds of data in this device-free scenario: WiFi signals and Ultrasound signals. For different kinds of signals, we employ different preprocessing approaches.

• **WiFi signals**. Given the powerful capability of representation learning, in the experiments, we directly feed the raw CSI data to the proposed DeepFusion model. Specifically, we concatenate the CSI matrices of all the 30 sub-carriers to create a two dimensional matrix which can take advantage of CNN to extract effective features. We segment the CSI data without overlapping via a fixed time window with the size of 1500ms. For each sample, we will obtain a matrix with size of 30×45. In order to make the neural network trainable, we need to normalize each sample value between 0.0 and 1.0 before we feed it to the proposed DeepFusion model.

• **Ultrasound signals**. The feasibility of employing ultrasound signals to identify human activities comes from the fact that the phase of the received ultrasound signal will get increased/decreased with the change of its propagation distance caused by the movement of the monitored subject. Thus, we can extract the phase information through demodulating the received signals as suggested in [43]. Assume that the transmitted signal can be represented by $T(t) = A\cos(2\pi ft)$, then we can represent the received signal as $R(t) = A'\cos(2\pi ft - 2\pi fd/c)$, where A and A' are the amplitude of the transmitted and received signal respectively, f is the frequency, c is the speed of sound, d is the length of the propagation which will be influenced by the movement of the subject. Then d/c is the propagation delay and $2\pi fd/c$ is the phase lag caused by the propagation delay. The demodulation algorithm is to multiply the received signal with $\cos(2\pi ft)$ to extract the signal around frequency f:

$$A'\cos(2\pi ft - 2\pi fd/c) \times \cos(2\pi ft)$$
$$= \frac{A'}{2}(\cos(-2\pi fd/c) + \cos(4\pi ft - 2\pi fd/c)).$$

After pass the output signal through a low pass filter of frequency f', we only keep the signal whose original frequency was between

Table 4: Activities in Device-Free Human Activity Dataset.

ID	Activities	ID	Activities	ID	Activities
1	Rotating the chair	4	Typing	7	Writing
2	Sitting during the phone call	5	Walking	-	-
3	Walking during the phone call	6	Sitting and wiping	-	-

$[f - f', f + f']$, which captures the influence of the human movement on the ultrasound signal. Using similar method, we multiply the received signal with $-\sin(2\pi ft)$ to get $\frac{A'}{2}(\sin(-2\pi fd/c))$. Then, we downsample the signal to 60 Hz and segment the signal for every 90 points without overlapping points. Finally, we use $\frac{A'}{2}(\cos(-2\pi fd/c))$ and $\frac{A'}{2}(\sin(-2\pi fd/c))$ as well as their FFT as the input to the deep learning model.

4.3.1 Experiments on the Homogeneous CSI Data. Since the baseline method DeepSense can not deal with the heterogeneous data, we first conduct experiments on a homogeneous CSI dataset to compare the performance of the proposed DeepFusion with that of the baselines. In this experiment, there are twelve WiFi receivers (considered as 12 sensors or views).

Model Settings. In this experiment, 3 CNN blocks are stacked to form a sensor representation extractor for each sensing node. For all the convolutional layers, the sizes of filters are all set to 3×3. The number of filters on each convolutional layer is set to 64. And the max-pooling is leveraged after each CNN block to reshape the input data. For the GRU, the size of hidden state is set to 128. For fully connected neural network in Cross-Sensor Module, the size of condensed correlation vector $\mathbf{v}_i$ is set to 64. All the activation functions are *ReLU*. The dropout rate for CNN is set to 0.8 and is 0.5 for RNN. For the hyper-parameters, we set $a = 19.0$, $b = 0.1$, $c = 1.0$, $\beta = 0.6$ and $\gamma = 0.4$.

During the training process, ADAM optimization algorithm [22] is used to optimize the parameters. The learning rate is $1e-4$. Batch size we used is 100. We first shuffle all the samples and split the datasets into training, validation and testing sets with the ratio $0.8 : 0.1 : 0.1$. Validation set is only used to select the best set of parameters.

Performance Validation. The accuracy of all the approaches on the CSI dataset is reported in Table 5. We can observe that the results are similar to that on the wearable sensor data. The traditional classification approach SVM has the lowest accuracy and our proposed DeepFusion model achieves the best performance. This further demonstrates the advantages of the deep learning models for HAR task. The results in Table 5 also confirm that taking into account different sensors' weights and the relationships among them can help improve the recognition performance.

Table 5: Performance on the CSI Dataset.

Model	Accuracy
SVM	0.520
DeepSense	0.860
SR+Avg	0.833
SR+WC	0.865
DeepFusion	**0.908**

Learned Weight Analysis. One of the DeepFusion's advantages is its ability to interpret the importance of each sensor through analyzing the learned weights. Intuitively, the learned weight on each sensor should be related to the individual performance of this sensor. To validate this point, we conduct experiments using the CSI data of each single sensor. Specifically, we first calculate the accuracy and the learned weight on each sensor, and then analyze the relationship between them. The results are shown in Figure 5, in which each point represents a sensor. X- denotes the accuracy of each sensor, and Y- is the learned weight by DeepFusion. We can observe that the accuracies on different sensors are different, which demonsrates that different sensors have different degrees of contributions for the HAR task. Ideally, the learned weights should be positively related to the accuracies on individual sensors. The results in Figure 5 show that the sensors with higher accuracy are mostly assigned with higher weights by DeepFusion. This confirms that the weights derived by DeepFusion truthfully reflect the quality of sensors.

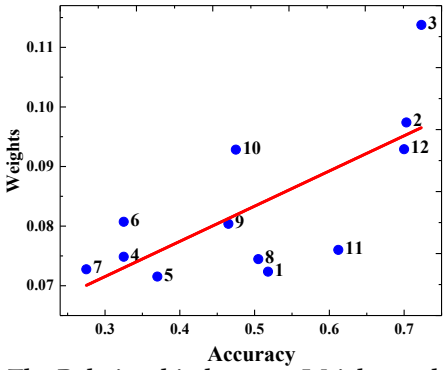

Figure 5: The Relationship between Weights and Accuracy on the CSI Dataset.

4.3.2 Experiments on the Heterogeneous Data. In order to demonstrate DeepFusion's ability of combining heterogeneous inputs, in this section, we conduct experiments on a heterogeneous dataset. Specifically, we set up four WiFi sensors and three ultrasound sensors, and evaluate the performance of the proposed and baseline models.

Model Settings. In this experiment, there are two kinds of signals: WiFi and ultrasound. The setting of WiFi sensors remains the same as aforementioned. For the ultrasound sensors, the filter sizes of convolutional layers are set as 6×2. Similarly, the max-pooling is also used among CNN blocks.

Performance Validation. In this experiment, we do not include DeepSense as a baseline, since it cannot deal with heterogeneous inputs. To evaluate the effectiveness of DeepFusion on the heterogeneous data, we conduct three experiments. The first two experiments employ all the approaches separately on WiFi and ultrasound signals, and the third experiment is done on the combined dataset.

Table 6: Comparison on the Heterogeneous Data.

Model	Accuracy		
	WiFi	Ultrasound	Heterogeneous
SVM	0.450	0.410	0.528
SR+Avg	0.713	0.623	0.833
SR+WC	0.795	0.635	0.835
DeepFusion	0.813	0.633	**0.885**

Table 6 shows the results of all the approaches on the three datasets. On the homogeneous WiFi dataset, DeepFusion can achieve the best performance, which is in accordance with the results shown in Table 5. On the ultrasound dataset, there are only three sensors, and the overall quality of these sensors is lower than that of WiFi sensors. Thus, the overall performance on the ultrasound dataset is worse than that on the WiFi dataset, and this brings negative effect when DeepFusion calculates the cross-sensor vector $\mathbf{r}^{cv}$. Nevertheless, DeepFusion still obtains comparable accuracy compared with SR+WC that does not consider the relationships among sensors.

On the heterogeneous dataset combining all the seven sensors' data, the results have similar patterns as those in Table 5. They confirm that the proposed DeepFusion model can effectively recognize the characteristics of activities from different types of signals. Also, the overall performance of all the approaches is better than that on the two homogeneous datasets. This implies that, though the overall quality of ultrasound signals is low, they still contain complementary information that can help the model to improve the prediction accuracy.

Learned Weight Analysis. To demonstrate the reasonableness of the learned weights by the DeepFusion model, we show the relationship between the learned weight and accuracy on each individual sensor in Figure 6. We can observe that on the heterogeneous dataset, there are two high-quality sensors. Correspondingly, DeepFusion assigns them higher weights. This again confirms that the proposed DeepFusion model is able to automatically capture the quality of different sensors. In fact, it also proves that DeepFusion can provide a high-level interpretability for the final prediction. As can be seen in Figure 6, the learned weights are highly correlated with the performance of individual sensors. Thus, we can conclude that the proposed DeepFusion can learn reasonable weights even with data from heterogeneous sensors.

5 RELATED WORK

The problem and methodologies presented in this paper are highly related to the following two research areas: human activity recognition and multi-sensor fusion.

5.1 Human Activity Recognition

Human activity recognition has been a hot topic for quite a long time. The human activity recognition systems can be roughly divided into three categories: vision based method [7, 32], wearable sensor based method [6, 20] and device-free method [2–4, 12, 15, 19, 23, 24, 26, 33, 36, 38, 39, 42, 44].

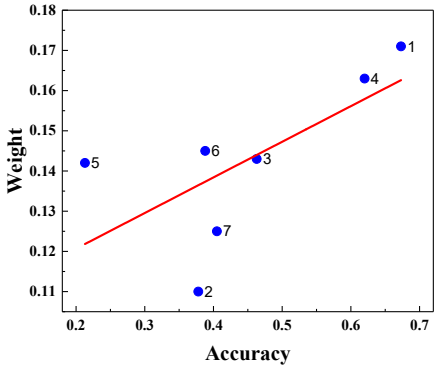

Figure 6: The Relationship between Weights and Accuracy on the Heterogeneous Dataset.

In our paper, we apply our model to both wearable sensor-based and device-free activity recognition. Orthogonal to these previous work, which uses only one kind of sensor or combine multiple sensors in a naive way, we explore to improve the human activity recognition though fusing multiple heterogeneous sensors' data, which can benefit a wide spectrum of IoT applications.

5.2 Multi-Sensor Fusion

Sensor fusion is the technique that involves gathering and combining information from multiple sensors in order to provide better information for target detection or recognition. The fusion methods can be divided into either signal, feature or decision level fusion [8]. Traditional sensor fusion methods include probabilistic fusion, evidential belief reasoning, fuzzy reasoning, possibilistic fusion, rough set based fusion, random set theoretic fusion, hybrid fusion, etc [21].

Recently, neural network based methods are applied to the sensor fusion problems because of its non-linear feature representation, self-learning adaptability and parallel processing [9]. [9, 25, 27] model the sensor fusion problems with a fully-connected neural network, which however cannot deal with input from heterogeneous sensors. Deep multimodal encoder is used to learn the shared representation from multiple sensing modalities [17, 28, 31, 34, 40, 48]. Most of these deep models mainly use the parameter sharing architecture to learn the joint representation. In contrast, we combine the representation of each sensor according to not only the relative significance of each sensor but also the correlation among different sensors, which cannot be captured by existing work. In addition, DeepSense [47] is a unified model to fuse multiple similar sensors to solve both classification and regression problems. It exploits the interactions among sensors and achieves state-of-the-art performance. As shown by the experiment results, our model outperforms DeepSense in three aspects. First, our model is able to handle heterogeneous sensor data due to its flexible architecture. Second, our model takes the varying quality of different sensors into consideration where DeepSense treats every single sensor equally. Third, our model incorporates the cross-sensor correlations which are ignored by DeepSense.

6 CONCLUSIONS

The proliferation of various IoT systems have facilitated a broad spectrum of classification and recognition applications in which

the same object can be monitored by multiple different types of sensing devices, each of which can be regarded as a unique information source. To unleash the power of such multi-sensor information, in this paper, we propose a unified deep learning framework, named DeepFusion, to extract informative features from heterogeneous sensory data. In order to improve the performance of classification/recognition tasks, the proposed DeepFusion model is able to combine the complementary information of multiple sensors by incorporating both the weighted-combination features and cross-sensor features. For validation, two real-world human activity recognition testbeds are built using commercialized wearable and wireless devices. Experimental results on the collected activity datasets justify the effectiveness of our proposed DeepFusion model for the fusion of heterogeneous sensory data.

ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation under Grants IIS-1218393, IIS-1514204 and CNS-1652503. And we gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. In *arXiv preprint arXiv:1603.04467*.
[2] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. 2015. Wigest: A ubiquitous wifi-based gesture recognition system. In *INFOCOM*.
[3] Fadel Adib, Zachary Kabelac, Dina Katabi, and Robert C Miller. 2014. 3D Tracking via Body Radio Reflections. In *NSDI*.
[4] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke recognition using wifi signals. In *MobiCom*.
[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).
[6] Sourav Bhattacharya and Nicholas D Lane. 2016. From smart to deep: Robust activity recognition on smartwatches using deep learning. In *PerCom Workshops*.
[7] Robert Bodor, Bennett Jackson, and Nikolaos Papanikolopoulos. 2003. Vision-based human tracking and activity recognition. In *MED*.
[8] Balasubramaniyan Chandrasekaran, Shruti Gangadhar, and James M Conrad. 2017. A survey of multisensor fusion techniques, architectures and methodologies. In *SoutheastCon*.
[9] Haixia Chen. 2015. Research on multi-sensor data fusion technology based on PSO-RBF neural network. In *IAEAC*.
[10] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *NIPS*.
[11] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. In *Machine learning*.
[12] Amit Das, Ivan Tashev, and Shoaib Mohammed. 2017. Ultrasound based gesture recognition. In *ICASSP*.
[13] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. *arXiv preprint arXiv:1601.01073*.
[14] Rohit Girdhar and Deva Ramanan. 2017. Attentional pooling for action recognition. In *NIPS*.
[15] Xiaonan Guo, Bo Liu, Cong Shi, Hongbo Liu, Yingying Chen, and Mooi Choo Chuah. 2017. WiFi-Enabled Smart Human Dynamics Monitoring. In *SenSys*.
[16] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool release: Gathering 802.11 n traces with channel state information. In *SIGCOMM*.
[17] Chaoqun Hong, Jun Yu, Jian Wan, Dacheng Tao, and Meng Wang. 2015. Multimodal deep autoencoder for human pose recovery. In *TIP*.
[18] Che-Wei Huang and Shrikanth Shri Narayanan. 2017. Deep convolutional recurrent neural network with attention mechanism for robust speech emotion recognition. In *ICME*.
[19] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, et al. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *MobiCom*.
[20] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. 2014. Bringing Gesture Recognition to All Devices. In *NSDI*.
[21] Bahador Khaleghi, Alaa Khamis, Fakhreddine O Karray, and Saiedeh N Razavi. 2013. Multisensor data fusion: A review of the state-of-the-art. In *Information fusion*.
[22] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*.
[23] Nicholas D Lane, Petko Georgiev, and Lorena Qendro. 2015. DeepEar: robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *UbiComp*.
[24] Hong Li, Wei Yang, Jianxin Wang, Yang Xu, and Liusheng Huang. 2016. WiFinger: talk to your smart devices with finger-grained gesture. In *UbiComp*.
[25] Yan-Hua Liang and Wei-Min Tian. 2016. Multi-sensor fusion approach for fire alarm using BP neural network. In *INCoS*.
[26] Jian Liu, Yan Wang, Yingying Chen, Jie Yang, Xu Chen, and Jerry Cheng. 2015. Tracking vital signs during sleep leveraging off-the-shelf wifi. In *MobiHoc*.
[27] Qiang Liu, Xin Wang, and Nageswara SV Rao. 2015. Artificial neural networks for estimation and fusion in long-haul sensor networks. In *Fusion*.
[28] Zuozhu Liu, Wenyu Zhang, Tony QS Quek, and Shaowei Lin. 2017. Deep fusion of heterogeneous sensor data. In *ICASSP*.
[29] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *KDD*.
[30] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2015. Natural language inference by tree-based convolution and heuristic matching. *arXiv preprint arXiv:1512.08422*.
[31] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *ICML*.
[32] Ronald Poppe. 2010. A survey on vision-based human action recognition. In *Image and vision computing*.
[33] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home gesture recognition using wireless signals. In *MobiCom*.
[34] Valentin Radu, Catherine Tong, Sourav Bhattacharya, Nicholas D Lane, Cecilia Mascolo, Mahesh K Marina, and Fahim Kawsar. 2018. Multimodal Deep Learning for Activity and Context Recognition. In *IMWUT*.
[35] Mengye Ren and Richard S Zemel. 2017. End-to-end instance segmentation with recurrent attention. In *arXiv preprint arXiv:1605.09410*.
[36] Wenjie Ruan, Quan Z Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. 2016. AudioGest: enabling fine-grained hand gesture detection by decoding echo signal. In *UbiComp*.
[37] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. LSTM-based deep learning models for non-factoid answer selection. In *arXiv preprint arXiv:1511.04108*.
[38] Sheng Tan and Jie Yang. 2016. WiFinger: leveraging commodity WiFi for fine-grained finger gesture recognition. In *MobiHoc*.
[39] Guanhua Wang, Yongpan Zou, Zimu Zhou, Kaishun Wu, and Lionel M Ni. 2016. We can hear you with wi-fi!. In *TMC*.
[40] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. 2015. On deep multi-view representation learning. In *ICML*.
[41] Wei Wang, Alex X Liu, and Muhammad Shahzad. 2016. Gait recognition using wifi signals. In *UbiComp*.
[42] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and modeling of wifi signal based human activity recognition. In *MobiCom*.
[43] Wei Wang, Alex X Liu, and Ke Sun. 2016. Device-free gesture tracking using acoustic signals. In *MobiCom*.
[44] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures. In *MobiCom*.
[45] Yuxi Wang, Kaishun Wu, and Lionel M Ni. 2017. Wifall: Device-free fall detection by wireless networks. *TMC*.
[46] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.
[47] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *WWW*.
[48] Ye Yuan, Guangxu Xun, Kebin Jia, and Aidong Zhang. 2017. A multi-view deep learning method for epileptic seizure detection using short-time fourier transform. In *BCB*.
[49] Ye Yuan, Guangxu Xun, Fenglong Ma, Qiuling Suo, Hongfei Xue, Kebin Jia, and Aidong Zhang. 2018. A Novel Channel-aware Attention Framework for Multi-channel EEG Seizure Detection via Multi-view Deep Learning. In *BHI*.
[50] Rui Zhou, Xiang Lu, Pengbiao Zhao, and Jiesong Chen. 2017. Device-Free Presence Detection and Localization With SVM and CSI Fingerprinting. In *IEEE Sensors Journal*.