Understanding and Improving Recurrent Networks for Human Activity Recognition by Continuous Attention

Ming Zeng[1]*, **Haoxiang Gao**[1]*, **Tong Yu**[1], **Ole J. Mengshoel**[1],
Helge Langseth[2], **Ian Lane**[1], **Xiaobing Liu**[3]
[1]Carnegie Mellon University, {ming.zeng, haoxiang.gao, tong.yu, ole.mengshoel, ian.lane}@sv.cmu.edu
[2]The Norwegian University of Science and Technology, helge.langseth@ntnu.no
[3]Google Brain, xbing@google.com

ABSTRACT
Deep neural networks, including recurrent networks, have been successfully applied to human activity recognition. Unfortunately, the final representation learned by recurrent networks might encode some noise (irrelevant signal components, unimportant sensor modalities, etc.). Besides, it is difficult to interpret the recurrent networks to gain insight into the models' behavior. To address these issues, we propose two attention models for human activity recognition: temporal attention and sensor attention. These two mechanisms adaptively focus on important signals and sensor modalities. To further improve the understandability and mean F1 score, we add continuity constraints, considering that continuous sensor signals are more robust than discrete ones. We evaluate the approaches on three datasets and obtain state-of-the-art results. Furthermore, qualitative analysis shows that the attention learned by the models agree well with human intuition.

Author Keywords
Human activity recognition; Recurrent neural networks; Attention mechanism; Interpretability.

ACM Classification Keywords
I.2.m Artificial Intelligence: Miscellaneous

INTRODUCTION
Wearable-based human activity recognition (HAR) systems are an indispensable component in mobile ubiquitous computing and human-computer interaction [5]. By integrating data from various sensor modalities (accelerometer, gyroscope, GPS, etc.), HAR systems are used in a large number of context aware applications such as daily life logging [6], and cross-device user recognition [23]. To recognize users' activities, various machine learning algorithms have been engineered for specific application contexts [4, 5].

Deep Neural Networks (DNNs), especially Recurrent Neural Networks (RNNs), are very good at discovering intricate structure in sequential data, and have proven their potential and pushed the state-of-the-art in HAR [11, 19, 8]. A DNN consists of multiple layers of neurons and is built for automatic feature extraction, which reduces the need for designing hand-crafted features. Recently, one of the RNN models, so-called Long Short Term Memory (LSTM), has been very successfully employed in HAR [8]. The LSTM encodes an input time series signals sequentially into a fixed length vector and then feeds it into a classifier. This sequential architecture is appealing, as it make it possible to capture long-range dependencies in a sequence.

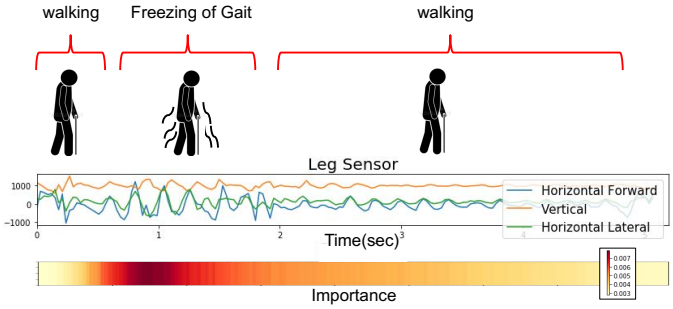

Figure 1. An example of a Freezing of Gait (FOG) detection for Parkinson disease from Daphnet Gait (DG) dataset [2]. The important acceleration signal components for FOG is shown in dark red.

However, there are some challenges in HAR with LSTM: (i) The sensor signals may contain some irrelevant components. For example, the abnormal signal of the freezing of gait (FOG) [2] of Parkinson disease (Figure 1) only shows up in a small time interval rather than in the entire window. Standard LSTM may not effectively detect FOG in this time series signal. (ii) Different sensor modalities play different roles for recognizing different activities [29, 26]. Using unimportant sensors modalities may introduce substantial noise into HAR. For example, to detect FOG during walking, the accelerometer is more important than the magnetometer

*equal contribution

and gyroscope. And the sensor attached to the leg is more important than that on the trunk [2]. (iii) The gains in good recognition results with RNNs have come at the cost of interpretability. It is desirable to understand the underlying basis for the decisions of neural models in HAR.

To address the first challenge, noisy signals (i), we apply temporal attention LSTM to automatically ignore the unimportant parts in the input signals and highlight the important parts. To deal with the second challenge, sensor modality importance (ii), we propose to use sensor attention on the input layer to fuse the sensor modalities based on their importance. However, when applying attention through time, we notice that continuous signals are more robust than discrete ones. Thus, we propose a continuity constraint for the two attention models. Visualizing where in the input the models are attending enables us to understand the models' behavior.

The contributions of this paper are the following:

- We propose temporal attention applied to the LSTM's hidden layer to highlight the important part of the time series signals.
- We propose sensor attention applied to the LSTM's input layer to reweight the sensor modality importance during training.
- To further improve the two attention LSTMs on time series sensor data, we propose continuous attention constraints on both time attention and sensor attention by using additional regularizations.
- By visualizing which parts of the input signals the models are attending to, we can get valuable insights into the models' behavior, thus improving interpreterbility.

RELATED WORK

The standard way of dealing with HAR relies on hand-crafted features. Many existing feature extraction approaches use statistical features of raw signals, such as mean, variance, entropy, and correlation coefficients [4]. Another branch of HAR feature extraction is transform coding, such as Fourier and wavelet transform [13].

Since designing hand-crafted features requires domain knowledge, it is desirable to develop a systematical feature learning approach to model the time series signals in HAR with multiple sensors [25]. Deep neural networks (DNNs) has had revolutionary impact in speech recognition [7], image classification [24], and machine translation [3]. DNNs also provided promising results in HAR domain (*e.g.,* [20, 22, 10]). Deep Belief Network (DBN) is used for feature extraction, but fail to consider the order of signal and cannot outperform PCA-ECDF features [20]. Similar to speech recognition, the combination of traditional sequence models (HMMs) and DNNs is also applied in HAR [31, 1]. However, the fully-connected DNNs fail to consider the order of time series signals. The 1D convolutional neural netowrks (CNN) employ more effective signal processing units, such as convolution (capturing local dependency), pooling (capturing time invariant features), and it also makes use of the available label information in feature extraction [28]. To fuse different sensor modalities [25, 17], the 2D CNN regards the set of signals as an image. The width is the length of the signal and the height is the number of sensors. The image-like CNN not only captures the salient features, but also takes advantage of more sensors to obtain higher accuracy. Ordóñez et al. [16] demonstrate the use of transfer learning to reduce the influence on randomly initialized CNN weights. Recently, Zeng et al. [30] propose the CNN encoder-decoder and CNN ladder for HAR in the semi-supervised setting.

In order to capture the long-term information in time series data, the RNN with long short-term memory (LSTM) is proposed by Hochreiter and Schmidhuber [12], and successfully applied in HAR [19, 11, 8]. Ordóñez et al. [19] combined convolutional and LSTM layers to provide promising results in recognition performance. Hammerla et al. [11] compared various DNN models in HAR, including LSTMs, and CNNs. Guan et al. [8] used ensemble LSTM model to capture diverse data during the training and significantly improved the recognition performance and robustness. However, a potential issue with LSTMs is that a neural network needs to be able to compress all the necessary information of a single input, which will involve noise or irrelevant parts. The attention approach is proposed to mitigate this problem for speech and natural language processing (NLP) [3, 14], and it offers means to explicate the inner workings of neural networks. Different from the speech and NLP, the dominant signals in HAR are continuous rather than discrete as discussed in the subsection Continuous Attentions. Sensors on different part of the body play different important roles for different activities. With this motivation, we built a new attention-based deep network architecture for HAR.

ATTENTION-BASED RNNS FOR HUMAN ACTIVITY RECOGNITION

We frame HAR as a sequence classification problem [4]. Given a sequence of sensor readings as input, namely $\mathbf{X} = (\mathbf{x}_1, ...\mathbf{x}_T)$, where T denotes the length of the signals, and $\mathbf{x}_t = [x_t^1, ..., x_t^D]$ ($t \in \{1, ..., T\}$) is a D dimensional vector denoting a sensor reading at time t for D channels (*e.g.* acceleration has three channels: x-axis, y-axis, z-axis). The learning problem is to map the input sequence $\mathbf{X}$ to a target $y \in \mathbf{C}$, where $C = |\mathbf{C}|$ is the number of activity classes. We first introduce the baseline model, called long and short term memory (LSTM). The LSTM is also treated as a basic building block of our sequence classier in this paper.

Standard LSTM

The LSTM is a recurrent network with four gates: $\mathbf{i}$ is the input gate, $\mathbf{f}$ is the forget gate, $\mathbf{o}$ is the output gate, and $\mathbf{c}$ is the cell activation vector [12]. They can be described by the following equations:

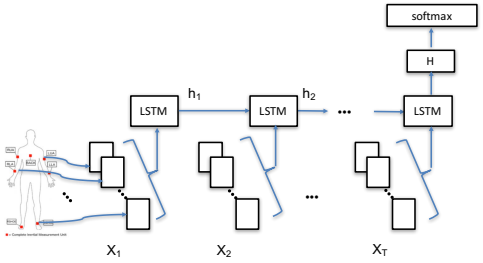

Figure 2. Structure of the standard LSTM [12].

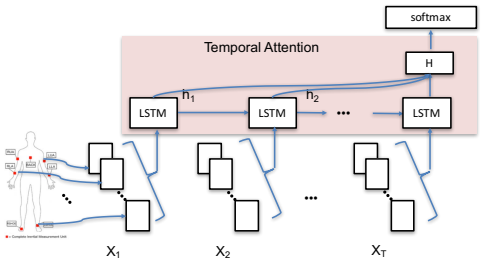

Figure 3. Structure of our LSTM + Temporal Attention for HAR

$$\mathbf{i}_t = \sigma(\mathbf{x}_t\mathbf{W}_{\mathbf{xi}} + \mathbf{h}_{t-1}\mathbf{W}_{\mathbf{hi}} + \mathbf{b}_t) \quad (1)$$
$$\mathbf{f}_t = \sigma(\mathbf{x}_t\mathbf{W}_{\mathbf{xf}} + \mathbf{h}_{t-1}\mathbf{W}_{\mathbf{hf}} + \mathbf{b}_{\mathbf{f}}) \quad (2)$$
$$\mathbf{c}_t = \mathbf{f}_t \cdot \mathbf{c}_{t-1} + \mathbf{i}_t \cdot tanh(\mathbf{x}_j\mathbf{W}_{\mathbf{xc}} + \mathbf{h}_{t-1}\mathbf{W}_{\mathbf{hc}}) \quad (3)$$
$$\mathbf{o}_t = \sigma(\mathbf{x}_j\mathbf{W}_{\mathbf{xo}} + \mathbf{h}_{t-1}\mathbf{W}_{\mathbf{ho}} + \mathbf{b}_{\mathbf{o}}) \quad (4)$$
$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{c}_t). \quad (5)$$

where the $\mathbf{W}$'s and $\mathbf{b}$'s are the parameters of the LSTM, $\mathbf{h}_t$ is a real-valued hidden-state vector at timestep t, $\sigma(\cdot)$ is a sigmoid function, and $\odot$ represents an element-wise multiplication. The whole network structure is described in Figure 2. In the last timestep T, $\mathbf{h}_T$ encodes all the previous information of the sequence and can be used for classification. This $\mathbf{h}_T$ is fed to a softmax layer whose target is the class label $\mathbf{y} \in [0,1]^C$, associated with the input sequence. Assuming that the LSTM predicts $\hat{\mathbf{y}} = \textbf{encoder}(\mathbf{X})$ =softmax($\mathbf{h}_T$), we will use the cross-entropy as the classification loss function:

$$\mathcal{L}_1(\mathbf{X}, \mathbf{y}) = -\mathbf{y}\log(\hat{\mathbf{y}}) = -\mathbf{y}\log(\textbf{encoder}(\mathbf{X})). \quad (6)$$

The **encoder**$(\cdot)$ encodes the whole sequence into a hidden representation. The encoder can be realized in may ways, such as RNN, or (Attention) LSTM.

Theoretically, a sufficiently large and well-trained neural network model should be able to perform sequence classification perfectly [18], because neural networks are universal function approximators [9]. However, in practice, it is necessary to learn these functions from limited data. A drawback of standard LSTM for classification is the long-distance dependencies problem. For example, the beginning of signals might have less impact on the decision. The LSTM memory cell are designed to mitigate this problem by allowing the LSTM memory cells to store and access information over long periods of time. Unfortunately, due to the noisy or irrelevant signals, it is hard to guarantee that we will learn to handle these properly.

LSTM with Temporal Attention

The basic idea of temporal attention is that instead of attempting to learn a single vector representation for the whole signal in the last timestep, we instead keep around vectors for every timestep in the input signal. These vectors are referenced at the final step for classification (Figure 3). As a result, we can express input signal in a more efficient way. To compute the vector $\mathbf{H}$, instead of using the last hidden state vector $\mathbf{h}_T$ in the LSTM [12], we consider a weighted sum of all the previous timesteps,

$$\mathbf{H} = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t, \quad (7)$$

where the α_t are the attention weights. In standard LSTM [12], the α_T is fixed to be 1, and $\alpha_t = 0$ when $t < T$. In the attention model, α_t are computed with a feed-forward neural network:

$$\alpha_t = \frac{\exp\{\text{score}(\mathbf{h}_T, \mathbf{h}_t)\}}{\sum_{s=1}^{T}\exp\{\text{score}(\mathbf{h}_T, \mathbf{h}_s)\}}. \quad (8)$$

Here, the score$(\cdot)$ is a bilinear function:

$$\text{score}(\mathbf{h}_t, \mathbf{h}_s) = \mathbf{h}_t^T\mathbf{W}_\alpha\mathbf{h}_s, \quad (9)$$

where, $\mathbf{W}_\alpha$ is a matrix of learnable parameter. In this way, the model is able to revisit the previous information and focus on more important parts to learn a better representation. The corresponding loss function is given by

$$\mathcal{L}_2(\mathbf{X}, \mathbf{y}, \boldsymbol{\alpha}) = -\mathbf{y}\log(\textbf{encoder}(\mathbf{X}, \boldsymbol{\alpha})). \quad (10)$$

LSTM with Sensor Attention

We would also like to capture the potentially varying importance of sensor modalities. To do so, we propose a sensor attention mechanism on the input layer. Different from the temporal attention, the sensor attention weights are calculated based on two different information sources: (i) previous attention history $\boldsymbol{\beta}_{t-1}$ and (ii) current input signal $\mathbf{x}_t$ (Figure 4). An unnormalized scalar energy value, $\mathbf{e}_t$, is produced for each memory entry:

$$\mathbf{e}_t = \mathbf{w}_e^T \tanh(\mathbf{W}_\beta\boldsymbol{\beta}_{t-1} + \mathbf{W}_x\mathbf{x}_t) \quad (11)$$
$$\boldsymbol{\beta}_t = \frac{\exp(\mathbf{e}_t)}{\sum_{c=1}^{C}\exp(\mathbf{e}_t^c)} \quad (12)$$
$$\mathbf{x}'_t = \boldsymbol{\beta}_t \odot \mathbf{x}_t. \quad (13)$$

Here, $\mathbf{x}'_t$ is the reweighted signal input at time t and $\odot$ is element-wise multiplication. The loss function is given by

$$\mathcal{L}_3(\mathbf{X}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = -\mathbf{y}\log(\textbf{encoder}(\mathbf{X}, \boldsymbol{\alpha}, \boldsymbol{\beta})). \quad (14)$$

Note that α_t at timestep t is a scalar for reweighting the hidden representations, while $\boldsymbol{\beta}_t$ is a vector whose dimension

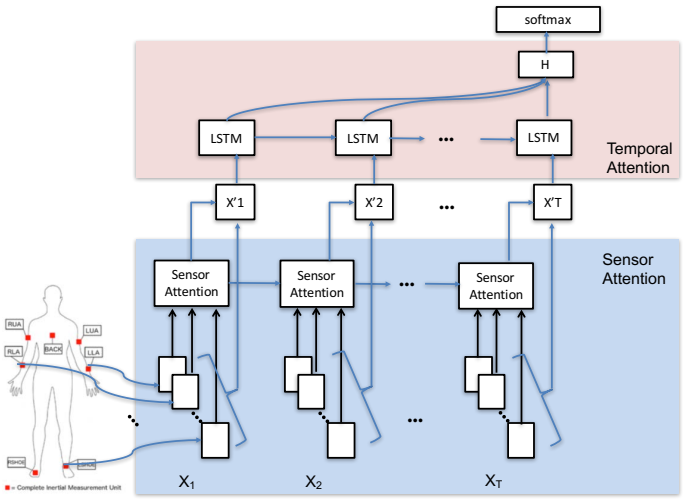

Figure 4. Structure of our novel Attention-based LSTM (Temporal Attention + Sensor Attention) framework for HAR. From the bottom, the signals coming from the multiple wearable sensors are re-weighted by attention LSTM, which also allows to capture time series information. On top of that, one more LSTM layer is stacked to capture more abstract features. The top layer is a softmax classifier.

equals the number of channels[1]. In standard LSTM [12], the sensor weight can be regarded as uniform distribution.

Improving Attention LSTM with Continuous Attention

One problem of the previous attention approaches is that the distribution of the attention weights is sharp. This is reasonable for NLP, because the discrete tokens in a sentence independently convey meanings. For example, in the sentence "This is a good restaurant," the word "good" shows a positive sentiment. However, when we recognize the activity from a series of signals, the selected signals should be a consecutive series rather than represent discrete, disconnected points. We therefore introduce additional regularization terms for temporal attention and sensor attention respectively.

Continuous Temporal Attention

The continuous temporal attention regularization encourages continuity, which is given by

$$\Omega_T(\boldsymbol{\alpha}) = \lambda_1 \sum_t |\alpha_t - \alpha_{t-1}|, \tag{15}$$

where the regularization forces the continuous attention.

Continuous Sensor Attention

The continuous sensor attention regularization discourages transitions. It is given by

$$\Omega_S(\boldsymbol{\beta}) = \lambda_2 \sum_t |\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}|, \tag{16}$$

where the regularization discourages switching sensor modalities back and forth.

[1]We have different granularity to apply the attention mechanism, e.g. on sensors, on sensor modalities, or on each channel for each modality. But the smaller granularity requires more data to train the model. Here we only focus on sensor modality, and use sensor attention and sensor modality attention interchangeably.

The final loss function is the combination of equation (14), (15) and (16), $\mathcal{L}(\mathbf{X}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathcal{L}_3(\mathbf{X}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}) + \Omega_T(\boldsymbol{\alpha}) + \Omega_S(\boldsymbol{\beta})$. Because the attention weights are not provided during training, we actually minimize the expected loss

$$\min_{\theta} \sum_{(\mathbf{X},\mathbf{y})\in D} \mathcal{L}(\mathbf{X}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{17}$$

where θ denotes the set of learnable parameters in the model, and D is the collection of training instances. Our continuous attention objective (17) encourages the model to compress the input signal into coherent representations that work well for the recognition. To minimize the loss, we can use mini-batch gradient descent.

EXPERIMENTS

We selected three publically available HAR datasets for our evaluation. All datasets reflect human activities in different contexts and have been recorded by various sensors (*e.g.*, accelerometers, gyroscope, etc.). Sensors were either worn or embedded into objects that subjects manipulated. The sensor data was segmented using a sliding window as described. All the machine learning experiments were carried out on a server equipped with a Tesla K20c GPU and 64G memory.

Training details include: 1 layer LSTM with 128 dimension hidden representation; optimization approach is ADMA with 0.05 learning rate; gradient normalization at 1.

Datasets and Setup

We use three publicly available datasets in our experiments with the same settings as previous works. The first is the **PAMAP2** dataset (Physical Activity Monitoring for Aging People 2) [21]. It consists of 12 lifestyle activities ("walking," "lying down," "standing" etc.) and sport exercises ("nordic walking," "running," etc.) by 9 participants. Accelerometer,

gyroscope, magnetometer, temperature, and heart rate data are recorded from inertial measurement units (IMUs) located on the hand, chest and ankle over 10 hours, resulting in 52 dimensions. To compare the result with previous work [8, 11], we downsampled the data from 100Hz to 33.3Hz, and used a 5.12 second sliding window with 78% overlap, resulting in around 473k samples. All samples were standardized to zero mean and unit variance. As in previous work [11, 8], we use Participant 6 for test, Participant 5 for validation, and the rest of the participants for training.

The second dataset used for our experiment is the **Daphnet Gait (DG)** dataset [2]. It contains recordings of 10 Parkinson' disease (PD) patients instructed to perform activities that are likely to induce freezing of gait. Freezing of gait (FOG) is common in advanced PD, where affected patients struggle to initiate movements such as walking. The goal is to detect these freezing incidents. This is a binary classification problem. Accelerometer readings were recorded from ankle, knee, and trunk, resulting in 9 dimensions. We use Participant 9 for validation, Participant 2 for test, and the rest of the data for training. We use the same settings as used before [11], and downsampled the data to 32Hz. The sliding window size is 1 second, resulting in around 470k samples for training.

The third dataset used is the **Skoda Mini Checkpoint (Skoda)** dataset [27], which describes the activities of assembly-line workers in a car production scenario. The dataset contains a worker wearing 19 accelerometers on both arms while performing 46 activities in the factory at one of the quality control checkpoints. To recognize the right arm's gestures ("checking the boot," "opening engine bonnet," etc.) in our experiments, we focus on 10 accelerometers placed on the right arm. The recording is about 3 hours long, consisting of 70 repetitions per gesture with 98Hz sampling rate. We downsampled the data to 33Hz, and standardized to zero mean and unit variance, resulting in 60 dimensions and around 190k samples. We use 80% of the of the data in each class for training, the next 10% for validation, and the rest for test.

We use F-measure (F_1) in the evaluation. Since the traditional F_1 score is used to measure the performance of binary classification, we used mean F1 score (F_m) by weighting classes according their sample proportion

$$F_m = \frac{1}{C}\sum_{i=1}^{C}\frac{2\cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}, \tag{18}$$

where for the given class i, $\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i+\text{FP}_i}$, $\text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i+\text{FN}_i}$. Here, $i = 1, ..., C$ is the set of classes considered, TP_i,FP_i represents the number of true and false positive, respectively and FN_i is the number of false negatives.

Comparing with Traditional RNN Models for HAR

We compare our attention-based models to state-of-the-art DNN models for HAR. The baseline methods include standard LSTM [12], DeepConvLSTM [19], and LSTM-S (LSTM + sample-by-sample analysis) [11]. Although our models also perform better than the ensemble LSTM [8][1],[2] it is fair to compare with single model. In this paper, we focus on the evaluation on the single models.

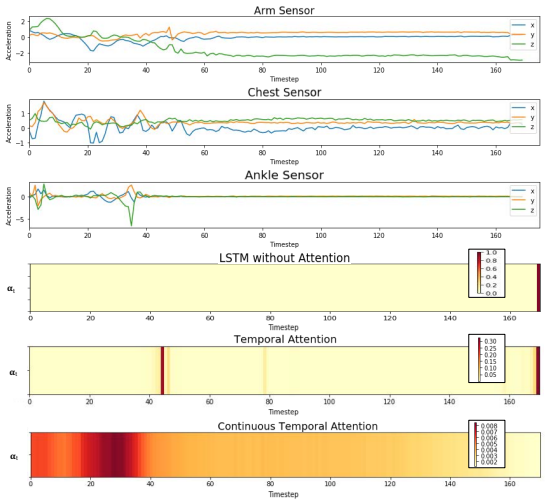

Figure 5. An example of a 5.12s window labeled as "walking" activity on PAMAP2. However, only the beginning of the acceleration signal shows it is walking. The 4th, 5th and 6th rows show the visualization of attention weights for LSTM without attention, LSTM + Temporal Attention and LSTM + Continuous Temporal Attention, respectively.

The results are in Table 1. Our LSTM Continuous Temporal Attention outperforms the state-of-the-art, LSTM-S, on DG and Skoda. Specifically, the Continuous Temporal Attention achieves around 14.50%, 17.00%, and 3.40% improvements in mean F1 scores on the three datasets, compared to the LSTM baseline [12].

Combining Continuous Sensor Attention with Continuous Temporal Attention can further improve the mean F1 scores on PAMAP2 and DG datasets with around 1.8% and 7.7%, compared to the state-of-the-art. It is interesting that the continuous sensor attention has a negative impact for Skoda. This might be because recognizing the right arm activities in car assembly lines depends on the the sensor interaction rather than a single sensor Thus constantly focusing on a single sensor, which is done in continuous sensor attention might deteriorate the classification performance on Skoda.

Visualizing Important Signal Components

This section seeks to better understand how temporal attention models help achieve high mean F1 scores in HAR. We now study which are the important signal components by looking into the temporal attention weights of the models.

We visualize the attention weights from the LSTM baseline (LSTM without attention), the temporal attention model and the continuous temporal attention model for two different walking activities(Figure 5, 6) and one running(Figure 7) on PAMAP2 dataset. Because the accelerometers are generally

[2]On PAMAP2 and Skoda, our models achieve higher mean F1 scores, compared to the results of the ensemble LSTM reported in [8].

Models	**PAMAP2**	**DG**	**Skoda**[4]
LSTM baseline ([12]) (LSTM without Attention)	0.7548	0.6675	0.9040
DeepConvLSTM ([19])	0.7480	0.7344[3]	0.9120
LSTM-S ([11])	0.8820	0.7600	0.9210
LSTM + Temporal Attention	0.8052	0.7913	0.9240
LSTM + Sensor Attention	0.7384	0.6700	0.9002
LSTM + Continuous Temporal Attention	0.8629	0.8216	**0.9381**
LSTM + Continuous Sensor Attention	0.7797	0.7817	0.8802
LSTM + Continuous Temporal + Continuous Sensor Attention[5]	**0.8996**	**0.8373**	0.8903

Table 1. Comparison of recognition results achieved using our attention-based LSTMs (bottom five result rows) versus baseline using the state-of-the-art (top three result rows) for sample-wise activity recognition (mean F1 score). Our LSTM + Continuous Temporal Attention is significantly better than the LSTM baseline with p-value < 0.01.

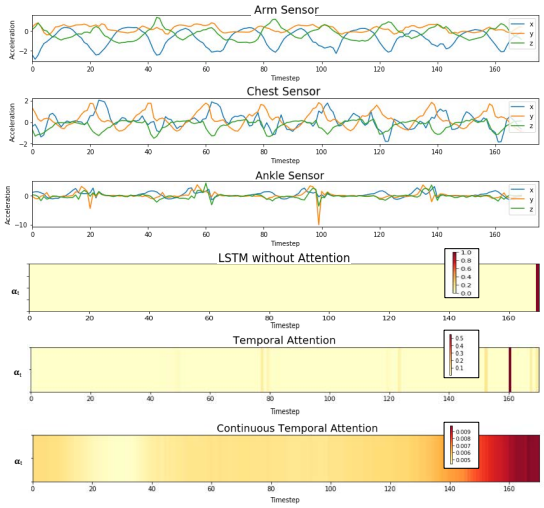

Figure 6. An example of a 5.12s window labeled as "walking" activity on PAMAP2. The entire signal shows the pattern of walking. The 4th, 5th and 6th rows show the visualization of attention weights for LSTM without Attention, LSTM + Temporal Attention and LSTM + Continuous Temporal Attention respectively.

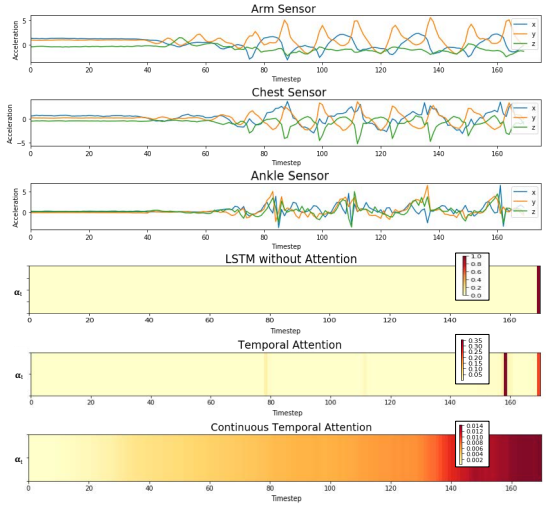

Figure 7. An example of a 5.12s window labeled as "running" on PAMAP2. Only the last half of the acceleration signal reflects it is running. The 4th, 5th and 6th rows show the visualization of attention weights for LSTM without Attention, LSTM + Temporal Attention and LSTM + Continuous Temporal Attention, respectively.

important to recognize daily-life activities [4], we only show the raw acceleration signals (the first three rows) from the arm sensor, the chest sensor, and the ankle sensor. The forth and fifth rows are the attention weights for each timestep. The darker color indicates the higher weight.

Figure 5 shows a signal window labeled as "walking", but only the beginning has the "walking" signal. The standard LSTM will use the last encoded hidden vector, $\mathbf{h}_T$, for recognition, which contains many irrelevant stationary signals. The temporal attention is able to look back at timestep 43, which is the end of the walking signal, and it encodes the previous information. Because the last hidden vector ($\mathbf{h}_T$) still contains some "walking" information, the attention model also uses the last timestep for recognition. In contrast, the continuous temporal attention model is able to force the model to attend the hidden vectors for walking signals (around timestep 0-43) and its attention weight decays to the time after walking (around timestep 70).

Figure 6 shows the attention weights for a window labeled as "walking." We observe some clear patterns in the entire raw signals, such as period and amplitude. The temporal attention model highlights the hidden vector close to the end of the signal, while the continuous temporal attention attends the hidden vector towards the end consecutively. Due to the recurrent structure, the hidden vector close to the end is more informative than its beginning Thus if the entire signal segment contains the target activity, the temporal attention model performs similar to the standard LSTM. Nevertheless, it is in-

[3]We implemented DeepConvLSTM [19], got the similar result for PAMAP2, and applied it on DG dataset. The rest mean F1 scores are from previous works.

[4]The result of Skoda is from [8].

[5]We choose the optimal value of $\lambda_1 = 0.1$ and $\lambda_2 = 0.5$.

teresting to notice that the continuous temporal attention puts less attention between timestep 20 - 40, which corresponds to the flat signal from the ankle sensor.

Figure 7 shows the attention weights a window labeled as "running", but different from Figure 6, only the last half of the window shows "running" behavior. Even though the LSTM representation encodes the irrelevant components (static signals) at the beginning, the hidden vectors close to the end accumulate more information than the previous ones. The temporal attention tends to attend the hidden vectors towards the end, but it also pay some attentions in the middle. Starting from timestep 80, the continuous temporal attention gradually increases its interest in the hidden vectors and focuses more in the end.

Visualizing Important Sensor Modalities

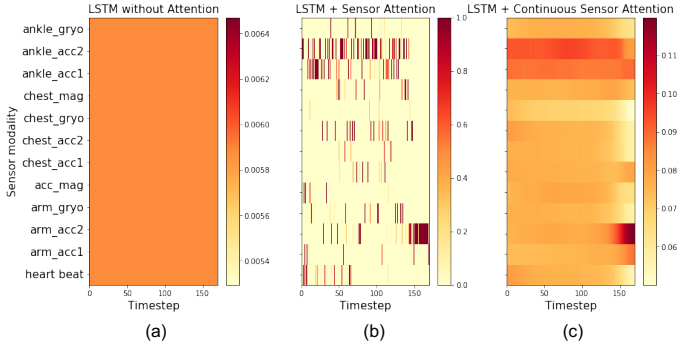

Figure 8. The visualization of LSTM (a) without sensor attention, (b) sensor attention and (c) continuous sensor attention, for a walking activity on PAMAP2 dataset. LSTM without attention is identical to the uniform attention weight.

We now study the impact of different sensor modalities placed on different parts of users body. We compare the attention weights from sensor modality attention and continuous sensor modality attention in Figure 8 (b) and (c). Both of the sensor modality attention approaches put high emphasis on the ankle sensor (Acc2, Gyro) and the arm sensor (Acc2). This is more reasonable than the LSTM without attention mechanism, which treats all the sensor modality equally. In the figure, The sensor attention is shifting-focus, acting as a feature selector in each timestep (Figure 8 (a)). The continuous sensor attention has smooth attention weights. This can prevent the model switching between sensors for recognition (Figure 8 (b)).

DISCUSSION

In this section, we discuss some issues of our attention-based approaches and the potential improvements for future work.

Dealing with sensor interaction Our sensor attention model uses the current corresponding raw signal and previous attention state to infer the current attention state. To better involve the previous attention state and previous signal, we can in future work use another LSTM to model the sensor attention state transition. In this way, the LSTM sensor attention is able to capture more complicated sensor interactions.

Better understanding for temporal attention We rely on the attention-based LSTM to interpret the recognition process. However, the signal information accumulates through time in LSTM, which makes it hard to determine how much data from each timestep contribute to the classification. Although LSTM is able to forget the unimportant signals and remember the important signals, to quantitatively evaluate the impact from the previous signals and determine signal importance, In future work, we need to use other approach, such as analyzing the gradients during the back propagation [15]. In addition, there is mismatching between attention and continuous attention in some cases (e.g. Figure 5), more analysis can be done to understand the models' behaviors.

CONCLUSION

We propose temporal attention, sensor attention, and two continuous constraints to understand and improve recurrent networks for human activity recognition. The attention-based models are able to focus on salient signal components and important sensor modalities. The experimental results demonstrate that our proposed approaches can achieve improvements in mean F1 score compared to the state-of-the-art. With visualization results, we show that our approaches improve the interpretability of the deep neural networks for HAR.

ACKNOWLEDGMENTS

We thank Anupam Datta, John P. Shen, Zhike Mao, and the anonymous reviewers for helpful comments and valuable discussion. This research is supported in part by the National Science Foundation under the award 1704845 and research fundings from Ericsson and Intel.

REFERENCES

1. Mohammad Abu Alsheikh, Ahmed Selim, Dusit Niyato, Linda Doyle, Shaowei Lin, and Hwee-Pink Tan. 2016. Deep Activity Recognition Models with Triaxial Accelerometers.. In *AAAI Workshop: Artificial Intelligence Applied to Assistive Technologies and Smart Environments*.

2. Marc Bachlin, Daniel Roggen, Gerhard Troster, Meir Plotnik, Noit Inbar, Inbal Meidan, Talia Herman, Marina Brozgol, Eliya Shaviv, Nir Giladi, and others. 2009. Potentials of enhanced context awareness in wearable assistants for Parkinson's disease patients with the freezing of gait syndrome. In *ISWC*. IEEE, 123–130.

3. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

4. Ling Bao and Stephen S Intille. 2004. Activity recognition from user-annotated acceleration data. In *PerComp*. Springer, 1–17.

5. Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM CSUR* 46, 3 (2014), 33.

6. Snehal Chennuru, Peng-Wen Chen, Jiang Zhu, and Joy Ying Zhang. 2012. Mobile Lifelogger–Recording, Indexing, and Understanding a Mobile Users Life. In *MobiCASE*. Springer, 263–281.
7. Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*. 577–585.
8. Yu Guan and Thomas Plötz. 2017. Ensembles of deep lstm learners for activity recognition using wearables. *IMWUT* 1, 2 (2017), 11.
9. G Gybenko. 1989. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems* 2, 4 (1989), 303–314.
10. Nils Yannick Hammerla, James Fisher, Peter Andras, Lynn Rochester, Richard Walker, and Thomas Plötz. 2015. PD Disease State Assessment in Naturalistic Environments Using Deep Learning.. In *AAAI*. 1742–1748.
11. Nils Y Hammerla, Shane Halloran, and Thomas Ploetz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. *IJCAI* (2016).
12. Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
13. Tâm Huynh and Bernt Schiele. 2005. Analyzing features for activity recognition. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. ACM, 159–163.
14. Suyoun Kim and Ian Lane. 2017. End-to-End Speech Recognition with Auditory Attention for Multi-Microphone Distance Speech Recognition. *Interspeech* (2017), 3867–3871.
15. Klas Leino, Linyi Li, Shayak Sen, Anupam Datta, and Matt Fredrikson. 2018. Influence-Directed Explanations for Deep Convolutional Networks. *arXiv preprint arXiv:1802.03788* (2018).
16. Francisco Javier Ordóñez Morales and Daniel Roggen. 2016. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *ISWC*. ACM, 92–99.
17. Sebastian Münzner, Philip Schmidt, Attila Reiss, Michael Hanselmann, Rainer Stiefelhagen, and Robert Dürichen. 2017. CNN-based sensor fusion techniques for multimodal human activity recognition. In *ISWC*. ACM, 158–165.
18. Graham Neubig. 2017. Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv preprint arXiv:1703.01619* (2017).
19. Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* 16, 1 (2016).
20. Thomas Plötz, Nils Y Hammerla, and Patrick Olivier. 2011. Feature learning for activity recognition in ubiquitous computing. In *IJCAI*, Vol. 22. 1729.
21. Attila Reiss and Didier Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *ISWC*. IEEE, 108–109.
22. Charissa Ann Ronao and Sung-Bae Cho. 2015. Deep Convolutional Neural Networks for Human Activity Recognition with Smartphone Sensors. In *NIP*. Springer, 46–53.
23. Xiao Wang, Tong Yu, Ming Zeng, and Patrick Tague. 2017. XRec: Behavior-Based User Recognition Across Mobile Devices. *IMWUT* 1, 3 (2017), 111.
24. Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044* (2015).
25. Jian Bo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. 2015. Deep convolutional neural networks on multichannel time series for human activity recognition. In *IJCAI*. AAAI Press, 3995–4001.
26. 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*. 351–360.
27. Piero Zappi, Clemens Lombriser, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Tröster. 2008. Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection. In *Wireless sensor networks*. Springer, 17–33.
28. Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Juyong Zhang. 2014a. Convolutional Neural Networks for human activity recognition using mobile sensors. In *MobiCASE*. IEEE, 197–205.
29. Ming Zeng, Xiao Wang, Le T Nguyen, Pang Wu, Ole J Mengshoel, and Joy Zhang. 2014b. Adaptive activity recognition with dynamic heterogeneous sensor fusion. In *MobiCASE*. IEEE, 189–196.
30. Ming Zeng, Tong Yu, Xiao Wang, Le T Nguyen, Ole J Mengshoel, and Ian Lane. 2017. Semi-supervised convolutional neural networks for human activity recognition. In *Big Data*. IEEE, 522–529.
31. Licheng Zhang, Xihong Wu, and Dingsheng Luo. 2015. Human activity recognition with HMM-DNN model. In *ICCI* CC*. IEEE, 192–197.