QualityDeepSense: Quality-Aware Deep Learning Framework for Internet of Things Applications with Sensor-Temporal Attention

Shuochao Yao
University of Illinois Urbana Champaign

Yiran Zhao
University of Illinois Urbana Champaign

Shaohan Hu
IBM

Tarek Abdelzaher
University of Illinois Urbana Champaign

ABSTRACT

Deep neural networks are becoming increasingly popular in mobile sensing and computing applications. Their capability of fusing multiple sensor inputs and extracting temporal relationships can enhance intelligence in a wide range of applications. One key problem however is the noisy on-device sensors, whose characters are heterogeneous and varying over time. The existing mobile deep learning frameworks usually treat every sensor input equally over time, lacking the ability of identifying and exploiting the heterogeneity of sensor noise. In this work, we propose QualityDeepSense, a deep learning framework that can automatically balance the contribution of sensor inputs over time by their sensing qualities. We propose a sensor-temporal attention mechanism to learn the dependencies among sensor inputs over time. These correlations are used to infer the qualities and reassign the contribution of sensor inputs. QualityDeepSense can thus focus on more informative sensor inputs for prediction. We demonstrate the effectiveness of QualityDeepSense using the noise-augmented heterogeneous human activity recognition task. QualityDeepSense outperforms the state-of-the-art methods by a clear margin. In addition, we show QualityDeepSense only impose limited resource-consumption burden on embedded devices.

EMDL'18, June 15, 2018, Munich, Germany

ACM ISBN 978-1-4503-5844-6/18/06…$15.00
https://doi.org/10.1145/3212725.3212729

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing**; • **Computing methodologies** → **Machine learning**; • **Computer systems organization** → **Embedded and cyber-physical systems**;

KEYWORDS

Deep Learning, Sensing Quality, Mobile Computing, Internet of Things

ACM Reference Format:
Shuochao Yao, Yiran Zhao, Shaohan Hu, and Tarek Abdelzaher. 2018. QualityDeepSense: Quality-Aware Deep Learning Framework for Internet of Things Applications with Sensor-Temporal Attention. In *EMDL'18: 2nd International Workshop on Embedded and Mobile Deep Learning , June 15, 2018, Munich, Germany.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3212725.3212729

1 INTRODUCTION

The proliferation of embedded and mobile devices able to perform complex sensing and recognition tasks unveils the future of intelligent Internet of things. Nowadays Internet of Things (IoT) applications cover a broad range of areas including health and well-being [8], context sensing [11, 12], crowd sensing and localization [10, 15].

At the same time, deep neural networks have advanced greatly in processing human-centric data, such as images, speech, and audio. The use of deep neural network has also gained increasing popularity in mobile sensing and computing research [18]. Great efforts have been made on designing unified structures for fusing multiple sensing inputs and extracting temporal relationships [9, 16], compressing neural network structures for reducing resource consumptions on low-end devices [4, 19], and providing well-calibrated uncertainty estimations for neural network predictions [6, 17].

To further advance such development for IoT applications, we need to address the key challenge brought by the heterogeneity of input sensor quality. On one hand, in order to

control the overall cost, IoT devices are equipped with low-cost sensors. Compared to dedicated sensors, they have insufficient calibration, accuracy, and granularity. The sensing quality of the heterogeneous on-device sensors can therefore be quite different. On the other hand, the unpredictable system workload, such as heavy multitasking and I/O load, can lead to unstable sensor sampling rates, because the OS may often fail to attach an accurate timestamp for each sensor measurement. The sensing quality can therefore be heterogeneous over time as well.

In order to tackle with the heterogeneity of sensing qualities over sensors and time, we propose QualityDeepSense that designs a deep learning framework with a sensor-temporal self attention mechanism. The proposed self attention mechanism improves a neural network's focus on different sensor inputs by inferring their sensing qualities. The key idea of QualityDeepSense is to identify the qualities of sensing inputs by calculating the dependencies of their internal representations in the deep neural network.

We assume that each sensor input is the composition of sensing quantity and noise. A sensing input with higher quality should contain a larger proportion of sensing quantity and a smaller proportion of noise. However, measuring the quality of sensing input directly is a challenging task. QualityDeepSense solves the problem by exploiting the dependencies among all input sensing quantities. For a particular IoT application, the correlated sensing quantities form complex dependencies that determine the final prediction or estimation, while the noises do not. Therefore, the extent of dependency and correlation among sensing inputs can be used to estimate the sensing quality. For example, a sensing input showing strong dependencies on other inputs is more likely a high-quality measurement.

QualityDeepSense estimates the dependencies of sensing inputs by proposing a sensor-temporal self-attention mechanism. It calculates the dependencies among different sensors and over time in a hierarchical way to reduce computation. The self-attention mechanism is a component that is inserted when the neural network merge the information from different sensors or merge over time. The self-attention component can be viewed as a weighted sum of inputs, where the weight is controlled by the degree of dependency calculated by internal representations.

We evaluate QualityDeepSense with noise-augmented heterogeneous human activity recognition task (N-HHAR). The original heterogeneous human activity recognition task performs human activity recognition with accelerometer and gyroscope measurements [12]. We add white Gaussian noise on the time domain or the frequency domain to generate noise-augmented datasets. We compare QualityDeepSense to the state-of-the-art DeepSense framework [16] to illustrate the efficacy of our sensor-temporal self-attention mechanism on exploiting heterogeneous sensing quality. We also test QualityDeepSense on Nexus 5 phones to show the low overhead of QualityDeepSense.

The rest of this paper is organized as follows. Section 2 introduces related work on dealing with heterogeneous sensing quality and attention mechanism. We describe the technical details of QualityDeepSense in Section 3. The evaluation is presented in Section 4. Finally, we discuss the results and conclude in Section 5.

2 RELATED WORK

A key problem in mobile sensing research is to handle the heterogenous sensing quality. Stisen et al. systematically investigate sensor-specific, device-specific and workload-specific heterogeneities using 36 smartphones and smartwatches, consisting of 13 different device models from four manufacturers [12]. These extensive experiments witness performance degradation due to the heterogenous sensing quality.

Recently, deep neural networks have achieved great improvement on processing human-centric data. Lane et al. proposed to use deep neural networks to solve common audio sensing tasks [9]. Yao et al. designed a unified deep neural network framework called DeepSense for mobile sensing and computing tasks. DeepSense can effectively fuse information from multiple sensing inputs and extract temporal relationships. However, none of these works has taken the heterogenous sensing quality into consideration. To the best of our knowledge, QualityDeepSense is the first deep learning framework that exploits sensing quality for IoT applications.

At the same time, attention mechanism has made great advances in traditional machine learning tasks. Bahdanau et al. propose the first attention mechanism for machine translation [3], which improves word alignment. Xu et al. design the attention mechanism for image caption with both hard and soft attentions [14]. Recently, Vaswani et al. exploit the attention mechanism by designing a neural network with only self-attention components [13]. To the best of our knowledge, we are the first to use self-attention mechanism for estimating and exploiting heterogenous sensing quality.

3 SYSTEM FRAMEWORK

In this section we introduce the QualityDeepSense framework that automatically balances the contribution of sensor inputs over time according to their sensing qualities. We separate our description into two parts. We first describe the overall structure of QualityDeepSense. Then we describe the sensor-temporal self-attention module in detail.

For the rest of this paper, all vectors are denoted by bold lower-case letters (e.g., $\mathbf{x}$ and $\mathbf{y}$), while matrices and tensors

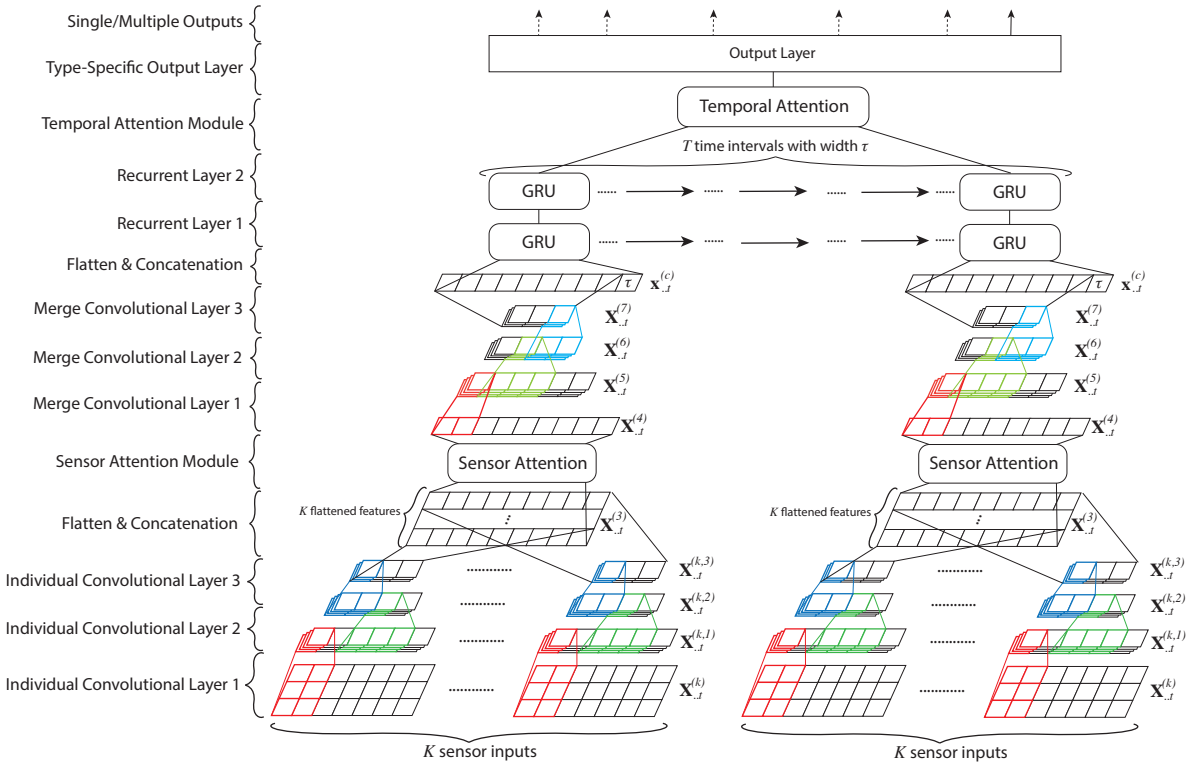

Figure 1: Main architecture of the QualityDeepSense framework.

are represented by bold upper-case letters (e.g., $\mathbf{X}$ and $\mathbf{Y}$). For a vector $\mathbf{x}$, the j^{th} element is denoted by $\mathbf{x}_{[j]}$. For a tensor $\mathbf{X}$, the t^{th} matrix along the third axis is denoted by $\mathbf{X}_{..t}$, and other slicing denotations are defined similarly. We use calligraphic letters to denote sets (e.g., $\mathcal{X}$ and $\mathcal{Y}$). For any set $\mathcal{X}$, $|\mathcal{X}|$ denotes the cardinality of $\mathcal{X}$.

3.1 QualityDeepSense Structure

For a particular application, we assume that there are K different types of input sensors $\mathcal{S} = \{S_k\}$, $k \in \{1, \cdots, K\}$. Take a sensor S_k as an example. It generates a series of measurements over time. The measurements can be represented by a $d^{(k)} \times n^{(k)}$ measured value matrix $\mathbf{V}$ and a $n^{(k)}$-dimensional timestamp vector $\mathbf{u}$, where $d^{(k)}$ is the dimension for each measurement (e.g., raw measurements along x, y, and z axes for motion sensors have dimension 3) and $n^{(k)}$ is the number of measurements. We split the input measurements $\mathbf{V}$ and $\mathbf{u}$ along time (i.e., columns for $\mathbf{V}$) to generate a series of non-overlapping time intervals with width τ, $\mathcal{W} = \{(\mathbf{V}_t^{(k)}, \mathbf{u}_t^{(k)})\}$, where $|\mathcal{W}| = T$. Note that, τ can be different for different intervals, but here we assume a fixed time interval width for succinctness. We then apply Fourier transform to each element in $\mathcal{W}$, because the frequency domain contains better local frequency patterns that are independent of how time-series data is organized in the time domain. We stack these outputs into a $d^{(k)} \times 2f \times T$ tensor $\mathbf{X}^{(k)}$, where f is the dimension of frequency domain containing f magnitude and phase pairs [16]. The set of resulting tensors for each sensor, $\mathcal{X} = \{\mathbf{X}^{(k)}\}$, is the input of QualityDeepSense.

As shown in Figure 1, QualityDeepSense inserts sensor-temporal attention modules hierarchically into DeepSense [16], which empowers the framework to estimate and utilize the input sensing qualities and boost the overall prediction performance.

The overall structure can be separated into three subnets. For each time interval t, the matrix $\mathbf{X}_{..t}^{(k)}$ will be fed into an individual convolutional subnet for extracting the relationships within the frequency domain and across the sensor measurement dimension. The individual convolutional subnet learns high-level relationships $\mathbf{X}_{..t}^{(k,1)}$, $\mathbf{X}_{..t}^{(k,2)}$, and $\mathbf{X}_{..t}^{(k,3)}$ hierarchically for each sensing input individually.

Then we flatten the matrix $\mathbf{X}_{..t}^{(k,3)}$ into $\mathbf{x}_{..t}^{(k,3)}$ and concat all K vectors $\{\mathbf{x}_{..t}^{(k,3)}\}$ into a K-row matrix $\mathbf{X}_{..t}^{(3)}$, which is the input of our sensor attention module. The sensor attention module estimate the sensing quality of K inputs by calculating their internal dependencies. Then the module generates a K-dim

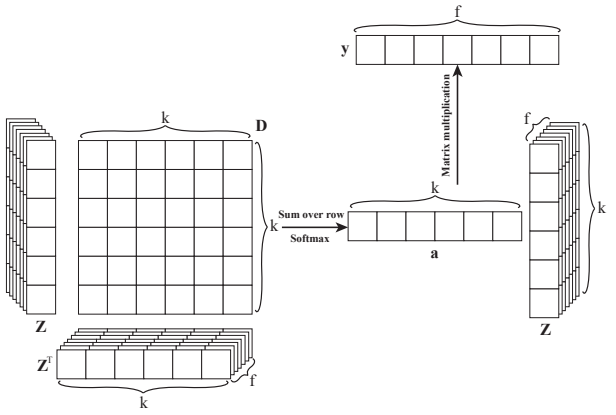

Figure 2: The structure of sensor/temporal self-attention module.

attention vector $\mathbf{a}_t$, which is used to calculate the row-wise sum $\mathbf{X}^{(4)}_{..t}$ from $\mathbf{X}^{(3)}_{..t}$. The detailed structure of sensor attention module will be described in Section 3.2.

Next, the merged convolutional subnet hierarchically learns the relationships $\mathbf{X}^{(5)}_{..t}$, $\mathbf{X}^{(6)}_{..t}$, and $\mathbf{X}^{(7)}_{..t}$ among K sensing inputs. The output of merged convolutional subnet is flatten into vector $\mathbf{x}^{(c)}_{..t}$ as the input of recurrent layers.

The recurrent layers is a two-layer Gated Recurrent Unit (GRU). The input $\{\mathbf{x}^{(c)}_{..t}\}$ for $t = 1, \cdots, T$ are fed into stacked GRU which generates outputs $\{\mathbf{x}^{(r)}_{..t}\}$ for $t = 1, \cdots, T$. We concatenate all T recurrent-layer output $\{\mathbf{x}^{(r)}_{..t}\}$ into a T-row matrix $\mathbf{X}^{(r)}$. Then we apply the temporal attention module to learn the sensing quality over time. The structure of temporal attention module is similar as the sensor attention module, which will be described in Section 3.2. The resulting vector goes through a sofmax layer for classification.

3.2 Sensor-Temporal Self-Attention Module

In this subsection, we describe the structure of sensor/temporal self-attention module. We assume that the input of self-attention module is a matrix $\mathbf{Z} \in \mathbb{R}^{k \times f}$, where k is the dimension for attention and f is the feature dimension. The structure of our self-attention module is shown in Figure 2. Notice that there is no additional parameters involves in the self-attention module.

The self-attention module can be summarized into two steps. First, we calculate the attention vector $\mathbf{a}$ based on input matrix $\mathbf{Z}$.

$$\mathbf{a} = \text{Softmax}\Big(\mathbf{1} \cdot (\mathbf{Z} \cdot \mathbf{Z}^\intercal)\Big) \tag{1}$$

where $\mathbf{1} = [1]^k$ is k-dim vector with all elements equal to 1.

Second, we calculate the weighted sum over the rows of $\mathbf{Z}$ with the attention vector $\mathbf{a}$.

$$\mathbf{y} = \mathbf{a} \cdot \mathbf{Z} \tag{2}$$

Here, we provide some explanations about our design. The purpose of the self-attention module is to estimate the dependencies among k vectors $\{\mathbf{Z}_{k\cdot}\}$. Since $\mathbf{Z}_{k\cdot}$ are internal representation in the neural network, we calculate the pair-wise dot product among $\mathbf{Z}_{k\cdot}$ to estimate their pair-wise dependency. Next, we sum the dependency matrix $\mathbf{D}$ over rows to estimate the dependency of each input vector on all others, including itself. Then, we use the Softmax function to generate the attention vector $\mathbf{a}$, which sums to 1. Finally, we calculate the weighted sum over the rows of $\mathbf{Z}$ with the attention vector $\mathbf{a}$.

4 EVALUATION

In this section, we evaluate QualityDeepSense using the task of human activity recognition with motion sensors. We first introduce our experimental settings, including the hardware, dataset, and baseline algorithm. We then evaluate our design in terms of accuracy, time, and energy consumption.

4.1 Hardware

In this evaluation, we run all experiments on the Nexus 5 phone. The Nexus 5 phone is equipped with quad-core 2.3 GHz CPU and 2 GB memory. We manually set 1.1GHz for the quad-core CPU for stable resource consumptions among different trials.

4.2 Software

In all experiments, we use TensorFlow-for-Mobile to run neural networks on Android phones [1]. For other traditional machine learning algorithms, we run with Weka for Andorid [2]. All experiments on Nexus 5 run solely with CPU. No additional runtime optimization is made.

4.3 Dataset

We use the dataset collected by Stisen et al. [12]. This dataset contains readings from two motion sensors (accelerometer and gyroscope). Readings were recorded when users executed activities scripted in no specific order, while carrying smartwatches and smartphones. The dataset contains 9 users, 6 activities (biking, sitting, standing, walking, climbStair-up, and climbStair-down), and 6 types of mobile devices. Accelerometer and gyroscope measurements are model inputs. Each sample is further divided into time intervals of length τ, as shown in Figure 1. We take $\tau = 0.25$ s. Then we calculate the frequency response of sensors for each time interval, and compose results from different time intervals into tensors as inputs. In addition, we add white Gaussian noise on either time domain or frequency domain with different different variance σ to generate our noise-augmented dataset.

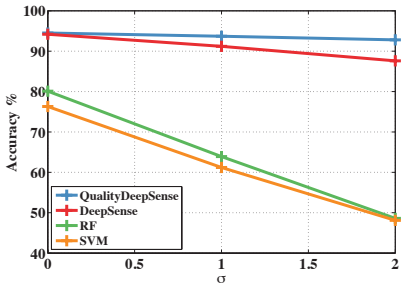

Figure 3: The accuracy of algorithms on HHAR with additive white Gaussian noise on frequency domain.

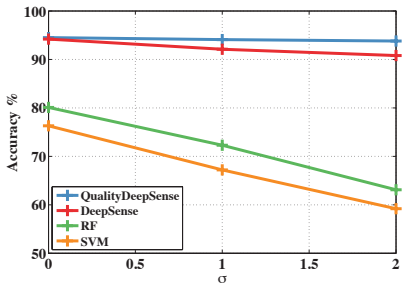

Figure 4: The accuracy of algorithms on HHAR with additive white Gaussian noise on time domain.

4.4 Baseline

We evaluate QualityDeepSense models with the following baseline algorithms:

(1) **DeepSense**: The state-of-the-art unified deep learning framework for IoT applications.
(2) **RF**: This is a random forest algorithm. It selects all popular time-domain and frequency domain features from [5] and ECDF features from [7].
(3) **SVM**: Feature selection of this model is same as the RF model. But this model uses support vector machine as the classifier.

4.5 Effectiveness

We first show the accuracy of all algorithms on the noise-augmented heterogeneous human activity recognition task, performing leave-one-user-out evaluation (*i.e.*, leaving out one user's entire data as testing data).

We add white Gaussian noise with different variance σ on frequency or time domain. The results are illustrated in Figure 3 and 4 respectively. For both cases, QualityDeepSense can reduce performance degradation by more than 50% compared to DeepSense, thanks to our sensor-temporal self-attention module that estimate and exploit the input sensing quality. Compared to the traditional machine learning algorithm, deep neural network models are better at resisting input noise. For all methods, noise on time domain is easy to deal with, because we can get rid of some high-frequency noise with pre-processing.

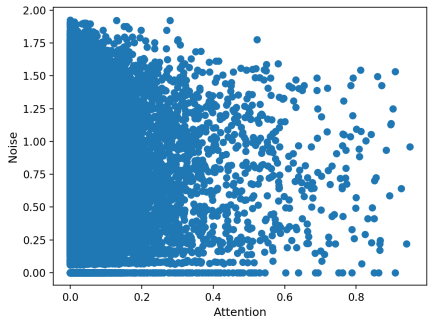

Figure 5: The correlation between attention and additive noise.

Then, we run an experiment testing the correlation between the sensing quality and attentions learnt in QualityDeepSense. Since we use the noise-augmented dataset, the quality of sensing input can be partly decided by the additive noise. Larger additive noise indicates worse the sensing quality. In QualityDeepSense, there are two types of attention, attention over sensor $\mathbf{a}_s$ and attention over time $\mathbf{a}_t$. We can easily obtain the overall attention by multiplying the corresponding elements from these two attentions. The result is shown in Figure 5. Since each sensing measurement does not contain the same amount of information, the correlation between attention and noise is not linear. However, we do witness that the attention tends to be smaller when the measurement has stronger noise.

4.6 Execution Time and Energy Consumption

Finally we measure the execution time and energy consumption of all algorithms on Nexus 5 phone. We conduct 500 experiments for each metric and take the mean value as the final measurement. The results are shown in Figure 6 and 7 respectively. Compared to DeepSense, QualityDeepSense only shows limited overhead on execution time and energy consumptions, while achieving better predictive performance. Compared to other traditional machine learning algorithms, the execution time and energy consumption of QualityDeepSense is acceptable.

5 CONCLUSION

In this paper, we introduce a deep learning framework, called QualityDeepSense, for solving the heterogeneous sensing quality problem in IoT applications. QualityDeepSense designs a novel sensor-temporal self-attention module to estimate input sensing quality by exploiting the complex dependencies among different sensing inputs over time. Experiments on noise-augmented human activity recognition show that QualityDeepSense greatly mitigates the performance

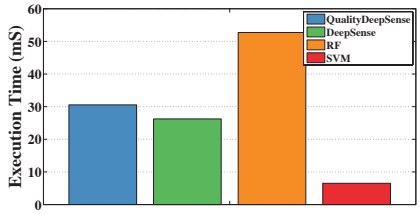

Figure 6: The execution time of algorithms on Nexus 5.

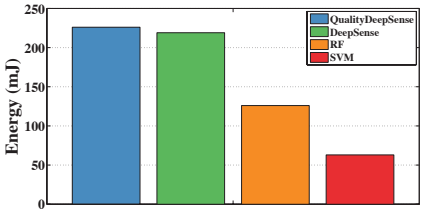

Figure 7: The energy consumption of algorithms on Nexus 5.

degradation caused by low input sensing quality with little additional computational overhead. This framework is an important step towards designing deep learning structures for handling heterogeneous sensing quality without external calibration. However, more exploration on model design and system implementation are needed. On one hand, applying attention mechanism on IoT applications can be different from its traditional usage in natural language processing and computer vision due to the nature of multiple-sensor fusing in IoT systems. On the other hand, more observations and modeling of IoT systems deployed "in the wild" are needed to design specific deep learning structures that deals with heterogeneous sensing quality.

6 ACKNOWLEDGEMENTS

Research reported in this paper was sponsored in part by NSF under grants CNS 16-18627 and CNS 13-20209 and in part by the Army Research Laboratory under Cooperative Agreements W911NF-09-2-0053 and W911NF-17-2-0196. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, NSF, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Tensorflow mobile. https://www.tensorflow.org/mobile/mobile_intro.
[2] Weka-for-android. https://github.com/rjmarsan/Weka-for-Android.
[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
[4] S. Bhattacharya and N. D. Lane. Sparsification and separation of deep learning layers for constrained resource inference on wearables. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 176–189. ACM, 2016.
[5] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.
[6] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.
[7] N. Y. Hammerla, R. Kirkham, P. Andras, and T. Ploetz. On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In *Proceedings of the 2013 International Symposium on Wearable Computers*, pages 65–68. ACM, 2013.
[8] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak. The internet of things for health care: a comprehensive survey. *IEEE Access*, 3:678–708, 2015.
[9] N. D. Lane, P. Georgiev, and L. Qendro. Deepear: robust smartphone audio sensing in unconstrained acoustic environments using deep learning. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 283–294. ACM, 2015.
[10] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 421–430. ACM, 2012.
[11] S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao. Musicalheart: A hearty way of listening to music. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 43–56. ACM, 2012.
[12] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 127–140. ACM, 2015.
[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.
[14] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.
[15] S. Yao, M. T. Amin, L. Su, S. Hu, S. Li, S. Wang, Y. Zhao, T. Abdelzaher, L. Kaplan, C. Aggarwal, et al. Recursive ground truth estimator for social data streams. In *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*, page 14. IEEE Press, 2016.
[16] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher. Deepsense: a unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017.
[17] S. Yao, Y. Zhao, H. Shao, A. Zhang, C. Zhang, S. Li, and T. Abdelzaher. Rdeepsense: Reliable deep mobile computing models with uncertainty estimations. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):173, 2018.
[18] S. Yao, Y. Zhao, A. Zhang, S. Hu, H. Shao, C. Zhang, S. Lu, and T. Abdelzaher. Deep learning for the internet of things. *Computer*, 51, 2018.
[19] S. Yao, Y. Zhao, A. Zhang, L. Su, and T. Abdelzaher. Deepiot: Compressing deep neural network structures for sensing systems with a compressor-critic framework. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. ACM, 2017.