SADeepSense: Self-Attention Deep Learning Framework for Heterogeneous On-Device Sensors in Internet of Things Applications

Shuochao Yao*, Yiran Zhao*, Huajie Shao*, Dongxin Liu*, Shengzhong Liu*, Yifan Hao*,
Ailing Piao †, Shaohan Hu ‡, Su Lu§, Tarek F. Abdelzaher*
*University of Illinois at Urbana-Champaign, †University of Washington at Seattle,
‡IBM Research, §State University of New York at Buffalo
Email: {syao9, zhao97, hshao5, dongxin3, sl29, yifanh5 }@illinois.edu,
alpiao@uw.edu, shaohan.hu@ibm.com, lusu@buffalo.edu, zaher@illinois.edu

Abstract*—Deep neural networks are becoming increasingly popular in Internet of Things (IoT) applications. Their capabilities of fusing multiple sensor inputs and extracting temporal relationships can enhance intelligence in a wide range of applications. However, one key problem is the missing of adaptation to heterogeneous on-device sensors. These low-end sensors on IoT devices possess different accuracies, granularities, and amounts of information, whose sensing qualities are heterogeneous and vary over time. The existing deep learning frameworks for IoT applications usually treat every sensor input equally over time or increase model capacity in an ad-hoc manner, lacking the ability to identify and exploit the sensor heterogeneities. In this work, we propose SADeepSense, a deep learning framework that can automatically balance the contributions of multiple sensor inputs over time by exploiting their sensing qualities. SADeepSense makes two key contributions. First, SADeepSense employs the self-attention mechanism to learn the correlations among different sensors over time with no additional supervision. The correlations are then applied to infer the sensing qualities and to reassign model concentrations in multiple sensors over time. Second, instead of directly learning the sensing qualities and contributions, SADeepSense generates the residual concentrations that are deviated from the equal contributions, which helps to stabilize the training process. We demonstrate the effectiveness of SADeepSense with two representative IoT sensing tasks: heterogeneous human activity recognition with motion sensors and gesture recognition with the wireless signal. SADeepSense consistently outperforms the state-of-the-art methods by a clear margin. In addition, we show that SADeepSense only imposes little additional resource-consumption burden on embedded devices compared to the corresponding state-of-the-art framework.

I. 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 context sensing [5], [14], [15], crowd sensing and localization [12], [18].

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 [23]. Great efforts have been made on designing unified structures for fusing multiple sensing inputs and extracting temporal relationships [19], [11], compressing neural network structures for reducing resource consumptions on low-end devices [24], [20], [4], providing well-calibrated uncertainty estimations for neural network predictions [21], [8], and reducing human labelling effort with semi-supervised learning [22].

To further advance such development for IoT applications, an increasing amount of researches have been recently made to explore deep learning structures for addressing IoT/sensing-related challenges. The heterogeneity of on-device sensor qualities is one of these key issues, waiting to be resolved [15]. On one hand, 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 different on-device sensors can therefore be heterogeneous. 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 accurate timestamps for measurements. The sensing quality can therefore be heterogeneous over time as well.

Some existing deep learning frameworks for IoT applications treat different on-device sensors equally over time [19]. Such designs overlook the characters of on-device sensors, which fails short in utilizing heterogeneous sensing qualities. Others circumvent such problem by increasing the overall neural network capability accompanied by data augmentation [13]. Therefore, the community naturally calls for a specific deep learning structure to tackle with the heterogeneity of sensing qualities over multiple sensors and time.

To this end, we propose SADeepSense, a deep learning framework with sensor-temporal self-attention mechanism for heterogeneous on-device sensor inputs. The proposed self-attention mechanism can automatically balance the contributions of multiple sensor inputs over time by inferring their sensing qualities without any additional supervision. The key idea of SADeepSense 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. SADeepSense solves the problem by exploiting the dependencies among all input sensing quantities. For an 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 to be a high-quality measurement.

SADeepSense estimates the correlations of sensing inputs through the self-attention mechanism. It is a configurable module that can be inserted into the neural network when we want to merge the information from multiple sensors or over time. The self-attention module can be viewed as a weighted sum of targeted input representations, where the weights are controlled by the degree of dependencies calculated by the scaled dot products of transformed internal representations from multiple hidden spaces. In addition, in order to stabilize the training process of self-attention modules, SADeepSense is designed to calculate residual contributions that is deviated from equal contributions from multiple sensors and over time.

We demonstrate the effectiveness of SADeepSense using the following two representative problems in IoT applications, which illustrate the potential for exploiting different sensing heterogeneous with the same self-attention framework:

1) *Heterogeneous human activity recognition (HHAR):* Human activity recognition itself is a mature problem. Stisen et al. illustrated that state-of-the-art algorithms do not generalize well across users when a new user is tested who has not appeared in the training set [15]. In this paper, we demonstrate that exploiting sensing heterogeneities can further improve the heterogeneous recognition performance.
2) *Wi-Fi signal based gesture recognition (Wisture):* Recently, radio frequency signals have been explored as another emerging resource for sensing on IoT devices. This task uses the Received Signal Strength Indicator (RSSI) of Wi-Fi signals to recognize hand gestures. In this paper, we further exploit the heterogeneity of Wi-Fi signals over time to improve the recognition accuracy.

Experiments are conducted on the above two IoT tasks under the normal and noise-augmented scenarios. For the noise-augmented case, we add additional white Gaussian noise on both sensing measurements and attached time stamps to emulate the extreme noisy sensing environments in the wild. We compare SADeepSense to the state-of-the-art DeepSense framework [19], the data-augmentation based solution for sensing heterogeneity problem [13], and other traditional machine learning algorithms. SADeepSense consistently achieve the best performance, which illustrates the efficacy of our SADeepSense framework on exploiting heterogeneous sensing quality. In addition, we test SADeepSense and other baseline algorithms on Nexus 5 phones to show the low overhead of our self-attention design.

The rest of this paper is organized as follows. Section II introduces related work on dealing with heterogeneous sensing quality and attention mechanism. We describe the technical details of SADeepSense in Section III. The evaluation is presented in Section IV. Finally, we discuss the results in Section V and conclude in Section VI.

II. Related Work

For IoT systems deployed "in the wild", the rich set of embedded sensors can confront unexpected performance variations due to device manufacturers, models, OS types, and CPU load conditions. Therefore, one key problem in mobile sensing research is to deal with heterogenous sensing qualities. 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 [15]. These extensive experiments witness performance degradation due to the heterogenous sensing quality. They also propose some clustering and interpolation pre-processing steps to mitigate the side-effects of sensing heterogeneity. However, these solutions only regain a limited part of these heterogeneity-caused accuracy losses.

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 [11]. Yao et al. designed a unified deep neural network framework called DeepSense for mobile sensing and computing tasks. Although DeepSense could effectively fuse information from multiple sensing inputs and extract temporal relationships [19], it failed to take heterogenous sensing quality into consideration Mathur et al. tried to circumvented the heterogeneous sensing quality by increasing the model complexity and applying the data augmentation technique. However, their method relied heavily on the manually generated augmented dataset, which fails to enable neural network itself to understand and utilize the heterogeneous sensing quality. Therefore, none of these works is able to infer and utilize the heterogeneous quality information adaptively by the learning model from sensing data. To the best of our knowledge, SADeepSense is the first deep learning framework that is able to exploit sensing quality for IoT applications without any additional supervision.

In addition, attention mechanism is becoming increasing popular and 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 for sequence learning. Xu et al. design the attention mechanism for image caption with both hard and soft attentions [17]. Recently, Vaswani et al. exploit the attention mechanism for machine translation by designing a neural network with only self-attention components [16]. To the best of our knowledge, we are the first to use attention mechanism

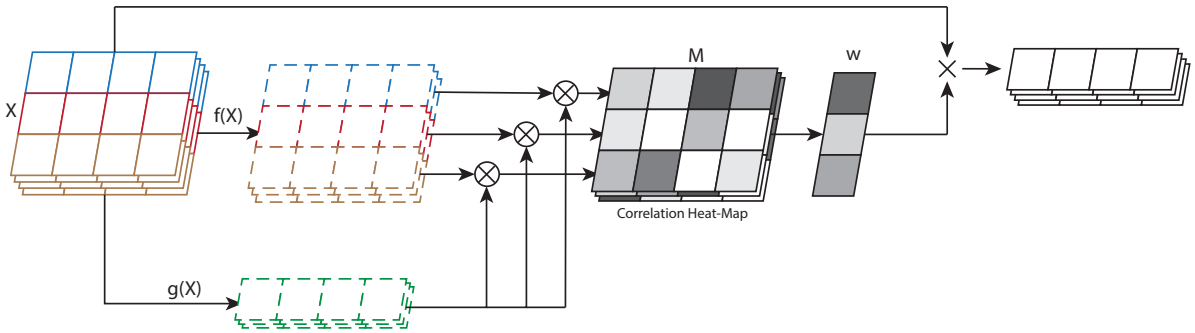

Fig. 1: The structure of Self-Attention (SA) module.

for estimating and utilizing heterogenous sensing quality for sensing-oriented IoT applications.

III. SADeepSense Framework

In this section, we introduce the technical details of our SADeepSense framework. We separate our descriptions into two parts. First, we introduce the Self-Attention (SA) module that can merge multiple inputs based on their corresponding input sensing qualities. Next, we introduce the SADeepSense framework, using the SA module to merge sensing information from multiple sensors and over the time.

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 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 $\mathbf{X}_{\cdots t}$, and other slicing denotations are defined similarly. For any tensor $\mathbf{X}$, $|\mathbf{X}|$ denotes the size of $\mathbf{X}$. 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 $\mathcal{X}$.

A. Self-Attention (SA) Module

In this subsection, we introduce the design of our Self-Attention (SA) module. We assume that the input of SA module is denoted as $\mathbf{X} \in \mathbb{R}^{n \times f}$, where n is the dimension to be merged and f is the feature dimension of each merging component. The general expression of SA module can be formulated as

$$\mathbf{w} \cdot \mathbf{X}, \quad \text{s.t.} \sum \mathbf{w} = 1, \tag{1}$$

where $\mathbf{w} \in \mathcal{R}^{1 \times n}$ is the attention weight vector that controls the contributions of n merging components in $\mathbf{X}$. Therefore, the key question is the way to generate the attention weight that can reflect the sensing quality of input components. Please notice that the SA module input, $\mathbf{X}$, does not required to be in a matrix form. $\mathbf{X} \in \mathcal{R}^{n \times \cdots \times f}$ can be a d-dim tensor whose first dimension are n components that are prepared to be merged. In this following formulation, we will still assume $\mathbf{X}$ to be a matrix if not specified otherwise.

We assume that each sensor input is the composition of targeted sensing quantity and noise. The sensing quantities are highly correlated while noises are not. The intuition behind our SA module is that the degree of correlations of hidden representations can be utilized to infer their sensing qualities.

Therefore, the inputs are first transformed into two latent spaces f and g to calculate their correlations. Two transformations have different functionalities. Function f is a local-correlation transformation. It extracts local features for each merging component, where $f(\mathbf{X}) = \mathbb{R}^{n \times \cdots \times h}$ and h is the dimension of latent space. Function g is a global-correlation transformation. It extracts global features by fusing all merging component, where $g(\mathbf{X}) = \mathbb{R}^{1 \times \cdots \times h}$. The transformation functions f and g can be implemented with simple neural network layers. For an example illustrated in Figure 1, f is a 1×1 convolutional layer and g is a 3×1 convolutional layer with no padding. In addition, we need to decide the dimension of latent space, h. In this paper, we choose h to be the number of classification categories, where each dimension of the latent space is supposed to contain category-specific correlations.

Having both the global and local correlation features, we can then calculate the correlation heat-map between each merging input and the all others,

$$\mathbf{M} = \frac{f(\mathbf{X}) \otimes g(\mathbf{X})}{\sqrt{h}}, \tag{2}$$

where $\otimes$ is the element-wise multiplication by broadcast the first dimension of $g(\mathbf{X})$. Notice that we rescale the the heat-map in (2) by $\sqrt{h}$ to prevent the magnitude of correlation heat-map from blowing up.

Then, we can utilize the correlation heat-map to calculate the attention weight $\mathbf{w}$ in (1). This is done by calculating the mean of correlation heat-map along all the dimension except for the first one and going through the softmax function,

$$\mathbf{w} = \text{softmax}\left(\frac{n}{|\mathbf{M}|} \sum_{1:d} \mathbf{M}\right). \tag{3}$$

The logit calculated by the mean of heat-map preserve the correlation information of merging components, which is used to infer their sensing qualities.

In order to further improve the stability of SA module, we enable our self-attention design to learn the residual deviation

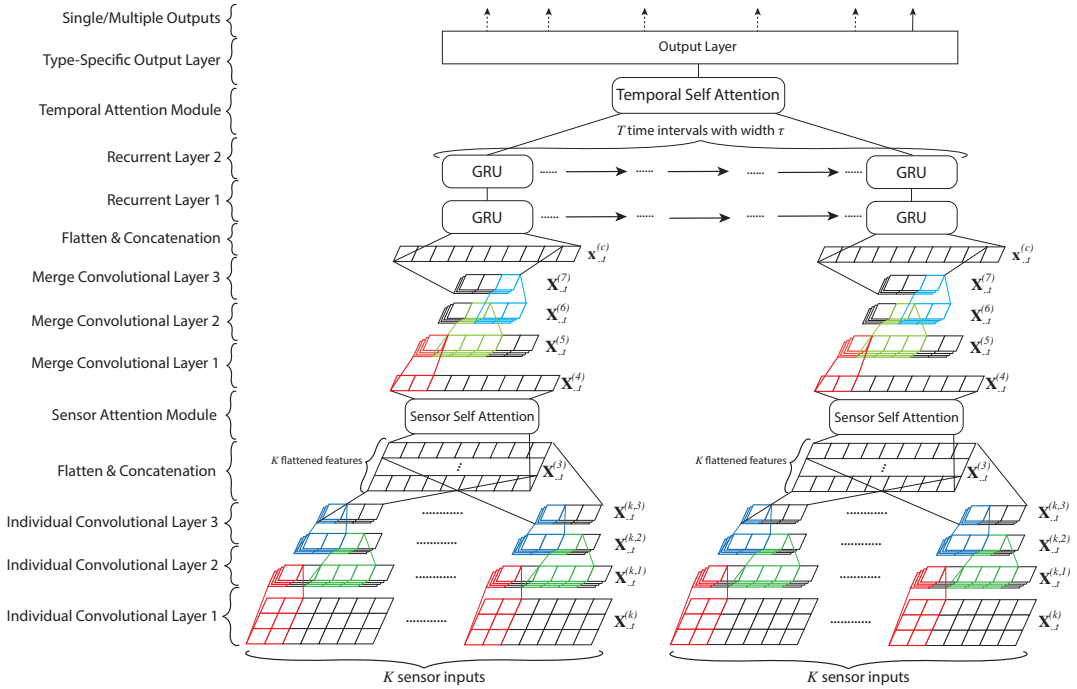

Fig. 2: Main architecture of the SADeepSense framework.

from the equal concentration of attention weight,

$$\mathbf{w} = \text{softmax}\left(\delta + (1-\gamma)\cdot\frac{n}{|\mathbf{M}|}\sum_{1:d}\mathbf{M}\right), \tag{4}$$

where δ is pre-defined constant, working as the prior of uniform attention weight. γ is a exponential decaying factor formulated as

$$\gamma = \lambda^{t/T} \tag{5}$$

where λ is the decaying rate, t is the training step, and T is the decaying steps. The exponential decaying factor γ starts from 1 and decays as the training step increases. During the evaluation, we set λ to be 0.1 and T to be the training epoch.

B. SADeepSense Structure

In this subsection, we integrate our proposed SA module with the state-of-the-art deep learning framework for IoT application, DeepSense [19]. The SA module is designed to merge multiple input representations according to the amount of carried information. In this paper, we utilize SA module to merge information from multiple sensors and over time by exploiting their heterogeneous sensing qualities. The integrated deep learning framework is called SADeepSense, whose detailed neural network structure is illustrated in Figure 2.

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 [19]. The set of resulting tensors for each sensor, $\mathcal{X} = \{\mathbf{X}^{(k)}\}$, is the input of SADeepSense.

The overall structure can be separated into three subnets: individual convolutional subnet, merged convolutional subnet, and temporal recurrent subnet.

For each time interval t, each type of sensor measurement with 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 the structure has to merge the features from multiple individual convolutional subnets from different type of sensors. The previous frameworks usually average or concatenate the representations from different sensors and feed them to the

next subnet, treating the representations from different sensor equally. The SADeepSense framework, however, exploits the heterogeneous sensing quality from multiple sensors with our proposed SA module. We called this part sensor self-attention module. 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 self attention module. The sensor self-attention module estimate the sensing quality of K inputs by calculating their internal dependencies and correlations. Then the module generates the row-wise weighted sum $\mathbf{X}_{..t}^{(4)}$ from $\mathbf{X}_{..t}^{(3)}$ whose weight is decided by their sensing qualities. The detailed structure of sensor self attention module was described in Section III-A.

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

The temporal recurrent layers is a two-layer Gated Recurrent Unit (GRU). We choose GRU instead of Long Short-Term Memory (LSTM), because GRUs show similar performance as LSTMs on various tasks, while having a more concise expression [6], which reduces network complexity for IoT applications. The input $\{\mathbf{x}_{..t}^{(c)}\}$ for $t = 1, \cdots, T$ are fed into stacked GRU which generates outputs $\{\mathbf{x}_{..t}^{(r)}\}$ for $t = 1, \cdots, T$. Then the framework has to fuse the information from T time intervals. Averaging the features from T intervals is a common choice. In this paper, we employ SA module to merge according to their sensing quality over time. We call this part temporal self-attention module. We concatenate all T recurrent-layer output $\{\mathbf{x}_{..t}^{(r)}\}$ into a T-row matrix $\mathbf{X}^{(r)}$. Then we apply the SA module to learn the sensing quality over time. The structure of SA module is again described in Section III-A. The resulting vector goes through a sofmax layer for classification.

IV. Evaluation

In this section, we evaluate SADeepSense using two representative sensing-related tasks in IoT systems: human activity recognition with motion sensors and gesture recognition with Wi-Fi signal. We first introduce our experimental settings, including the hardware, dataset, and baseline algorithms. We then evaluate our design in terms of accuracy, time, and energy consumption.

A. Hardware

In this evaluation section, 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. For fairness, all models, including deep-learning and non-deep-learning, are run solely on CPU.

B. Software

In all experiments, all neural network are trained on the workstation with GPUs. Then the deep learning models are exported and loaded into Android phones. We use TensorFlow-for-Mobile to run neural networks on 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 for any models for all experiments.

C. Datasets

We evaluate algorithms with two tasks, human activity recognition with motion sensors (HHAR) and gesture recognition with Wi-Fi signal (Wisture), under two testing scenarios, normal and noise-augmented.

For the HHAR, we perform a human activity recognition task with accelerometer and gyroscope measurements. We use the dataset collected by Stisen et al. [15]. 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 (*i.e.*, 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 2. 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. During the evaluation, we perform leave-one-user-out evaluation (i.e., leaving the whole data from one user as testing data).

For the Wisture, we perform gesture recognition (swipe, push, and pull) with Received Signal Strength Indicator (RSSI) of Wi-Fi signal. We use the dataset collected by Mohamed et al. [10]. This dataset contains labeled Wi-Fi RSSI measurements corresponding to three hand gestures made near a smartphone under different spatial and data traffic scenarios. The Wi-Fi RSSI measurements are classifier inputs, and gestures are used as labels. During the evaluation, we perform 10-fold cross validation.

Under the normal testing scenario, we directly using the above two datasets for training, validation, and testing. Since all the previous two datasets are collected under controlled experimental settings. In order to emulate the intensive heterogeneous environment for IoT systems in the wild, we add additional white Gaussian noise on both of the sensing measurements and attached time stamps. We call this noise-augmented scenario. The standard deviation of additional white Gaussian noise is controlled by the σ multiple of basic standard deviation unit, where σ is denoted as the noise intensity. For sensing measurements, the basic standard deviation unit is 10% of the standard deviation of sensing measurements collected in the dataset. For time stamps, the basic standard deviation unit is 10% of the sampling time interval.

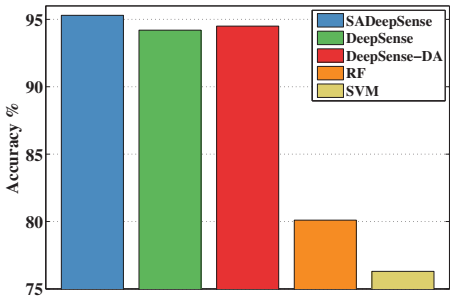

Fig. 3: The accuracy of algorithms on HHAR under the normal scenario.

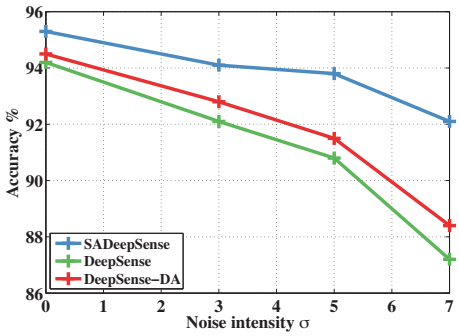

Fig. 4: The accuracy of algorithms on HHAR under the noise-augmented scenario with increasing noise intensity.

D. Baseline Algorithms

In order to show the effectiveness of our proposed SADeepSense framework, we evaluate SADeepSense with the following state-of-the-art deep learning models and other traditional machine learning models:

1) **DeepSense**: This is one of the state-of-the-art unified deep learning frameworks for IoT applications. It has the ability to fuse multiple sensor inputs and extract temporal relationships from sensing data. However, this algorithm just averages the representations when fusing information from multiple sensors or over the time. This baseline algorithm is used to illustrate the effectiveness of SA module in SADeepSense that is designed to utilize heterogeneous sensing quality.
2) **DeepSense-DA:** This is DeepSense framework with data augmentation training, which is the state-of-the-art method for solving heterogeneous sensing quality [13]. We implement their prosed Sensor Sampling Jitter noise dataset with the corresponding HHAR and Wisture dataset [15], [10]. This augmented dataset captures the variations in sensor sampling timestamps, which are likely due to system-related factors such as high CPU loads. Under the noise-augmented scenario, we further enhance the algorithm by letting it know the true noise functions we add to the sensing measurements and time stamps.
3) **RF**: This is a random forest algorithm. It selects all popular time-domain and frequency domain features from [7] and ECDF features from [9].
4) **SVM**: Feature selection of this model is same as the RF model. But this model uses support vector machine as the classifier.

E. Effectiveness

In this subsection, we will show the effectiveness of SADeepSense by comparing it with all baseline algorithms on two IoT applications under the normal and noise-augmented scenarios.

We first show the recognition accuracy of SADeepSense and other baseline algorithms on HHAR task under the normal scenario. As we mentioned before, we perform leave-one-user-out evaluation (*i.e.*, leaving out one user's entire data as testing data) during this experiment. The evaluation results are shown in Figure 3. All deep learning models (*i.e.*, SADeepSense, DeepSense, and DeepSense-DA) outperform the traditional machine learning algorithms under the experimental data collection setting. This show that the deep neural networks possess large model capacities, which are good at memorizing the sensing heterogeneity. When we compare the recognition accuracy among deep learning models, SADeepSense performs the best. It is because that heterogeneous sensing quality still exist under the experimental data collection setting. The self-attention module in SADeepSense is able to fuse multiple sensing streams with different degrees of concentrations according to their sensing qualities.

Next, we evaluate algorithms on HHAR task under the noise-augmented scenario. As we mentioned in Section IV-C, we add additional white Gaussian noise on both of the sensing measurements and attached time stamps with the standard deviation being the σ multiple of basic units. We show the recognition accuracy of algorithms with the increasing noise intensity σ in Figure 4. Since deep learning based models outperforms tradition machine learning algorithms, we only plot the accuracy curves of deep learning based models here. Although performance degradation appears as we increase the noise intensity, SADeepSense consistently performs the best with a clear margin. SADeepSense can reduce performance degradation by 50% at least compared to others. Therefore, the self-attention modules for sensors and over time work properly to identify and exploit the high-quality sensing measurement for the recognition task.

Then, we turn to show the the recognition accuracy of SADeepSense and other baseline algorithms on the Wisture task. Under the normal scenario, we perform the 10-fold cross validation for evaluation. As shown in Figure 5, deep learning models still outperform the traditional machine learning model. However, the performance gain is less than the case in the HHAR task. Since there is only one sensing modality, Wi-Fi signal, involving in Wisture, only sensing heterogeneity over time exists. Therefore, performance gain caused by blindly increasing model complexity is limited

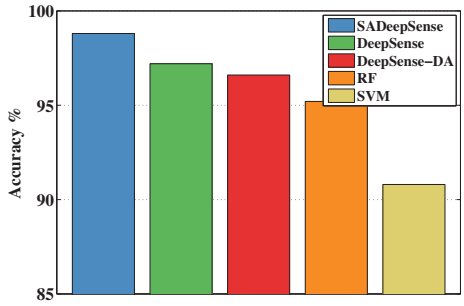

Fig. 5: The accuracy of algorithms on Wisture under the normal scenario.

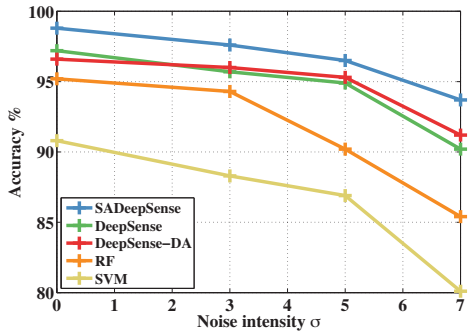

Fig. 6: The accuracy of algorithms on Wisture under the noise-augmented scenario with increasing noise intensity.

here. However, SADeepSense still performs the best, thanks to our temporal self-attention module for exploiting sensing heterogeneity over time. One interesting observation is that the accuracy of DeepSense-DA is lower than DeepSense, which means that performance degradation appears after training with data augmentation. Therefore, the sensor sampling jitter noise dataset used in DeepSense-DA fails to capture the true sensing heterogeneity in the Wisture task.

We further evaluate SADeepSense and other baseline algorithms under the noise-augmented scenario. We still add additional white Gaussian noise on both sensing measurements and attached time stamps with the standard deviation being the σ multiple of basic unit. The recognition accuracy of all algorithms with the increasing noise intensity σ is illustrated in Figure 6. Although data augmentation training causes performance degradation under the normal scenario, DeepSense-DA still performs better than DeepSense as we increase the noise intensity. This is because that we further enhance DeepSense-DA by letting the algorithm know the true noise function we add to the dataset. DeepSense-DA is then able to generate the optimal augmented data for training. Therefore, getting the exact noise function is important to the data-augmented based method. However, our proposed SADeepSense framework requires no prior knowledge about sensing heterogeneity or noise function. The framework can

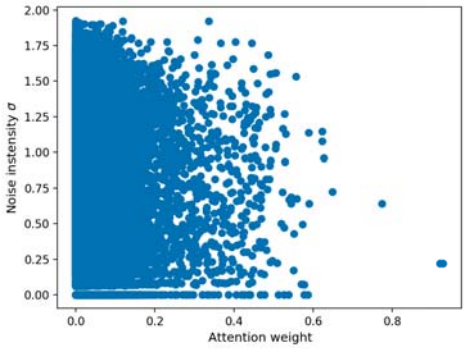

Fig. 7: The correlation between attention weights and noise intensities for HHAR.

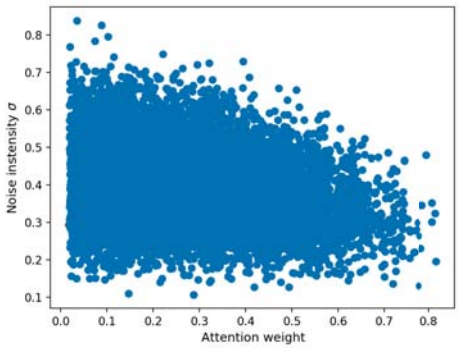

Fig. 8: The correlation between attention weights and noise intensities for Wisture.

infer and utilize the sensing measurement according to their qualities with the help of our self-attention module. In this experiment, SADeepSense still outperforms all others with a clear margin in all cases.

In order to further investigate the functionality of our prosed self attention module, we have an additional experiment testing the correlation between the sensing measurement quality and attention weights learnt in SADeepSense. During this experiment, we focus on the noise augmented scenario. It is because that, within noise-augmented dataset, the quality of sensing input can be partly decided by the additive noise. Larger additive noise indicates lower sensing quality. In SADeepSense, there are at most two types of attention weights, attention weight over sensors and attention weight over time. We can easily obtain the overall attention by multiplying the corresponding elements from these two attentions. The results for HHAR and Wisture tasks are shown in Figure 7 and 8 respectively. 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 weights tend to be smaller when the measurement has stronger noise.

Therefore this experiment indicates that the self-attention

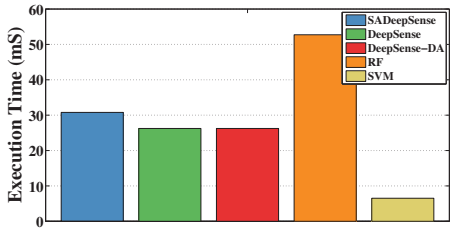

Fig. 9: The execution time of algorithms for HHAR on Nexus 5.

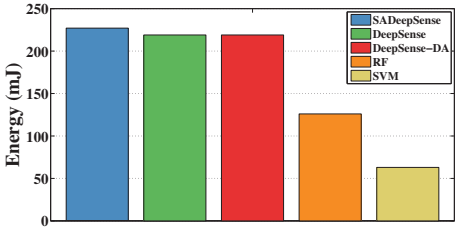

Fig. 10: The energy consumption of algorithms for HHAR on Nexus 5.

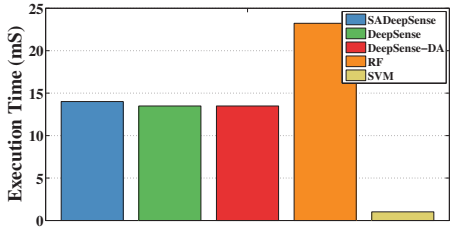

Fig. 11: The execution time of algorithms for Wisture task on Nexus 5.

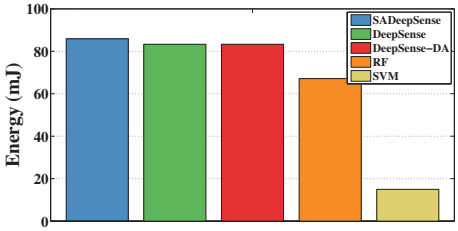

Fig. 12: The energy consumption of algorithms for Wisture task on Nexus 5.

module in SADeepSense do infer and utilize sensing quality for fusing multiple sensing streams.

F. Execution Time and Energy Consumption

Finally, we measure the execution time and energy consumption of SADeepSense on an IoT device, Nexus 5 phone. We compare SADeepIoT with all baseline algorithms introduced in Section IV-D. We conduct 500 experiments for each metric and take the mean value as the final measurement.

The evaluation results of model execution time and energy consumption for HHAR and Wisture are shown in Figure 9, 10, 11, and 12 respectively. Compared to DeepSense, SADeepSense only shows limited overhead on execution time and energy consumptions, while achieving better predictive performance. DeepSense-DA takes the same execution time and energy compared to DeepSense, because DeepSense-DA uses the same model structure as DeepSense with only addition data augmentation training. However, as shown in Section IV-E, the performance of data augmentation training relies heavily on the quality of augmented dataset, where the final prediction accuracy can even drop sometimes. Compared to other traditional machine learning algorithms, the execution time and energy consumption of SADeepSense is acceptable. In addition, random forest takes relatively long time to run. It is because random forest models consist of long decision paths that contain a large number of condition operations, which slow down the instruction pipelining in the CPU.

V. Discussion

This paper proposes a novel deep learning structure to fuse information from multiple sensing modalities and over the time according to the sensing quality. Throughout the paper, we treat other method for solving sensing heterogeneity, especially data augmentation, as the baseline algorithm to compare with. However, our proposed self-attention module is independent to the data augmentation training. The combination of two techniques are possible to further improve the ability of neural network to handle heterogeneous sensing inputs in IoT.

SADeepSense focuses on understanding and utilizing sensing quality without any additional supervision. However additional supervisions or constraints about sensing quality can be easily adopted by the SADeepSense. Supervisions or constraints can be imposed on the output of self-attention module, informing the framework to take additional information into account. Topics on designing additional supervisions and constraints, such as temporal and spatial sensing coherence, on the top of self attention module still need further studies to formally solve this problem.

In addition, this paper mainly focuses on additional overhead caused the self attention module instead of the absolute execution time and energy consumption consumed by the whole framework during the evaluation. Neural network models and structures can be compressed for improving the system efficiency. The state-of-the-art method is able to generate a compressed neural network that can reduce the system resource consumption to be less than 10% without hurting the prediction accuracy [24]. However, this is out of the scope of our paper. Therefore, during the evaluation, we take the overhead of SADeepSense compared to DeepSense as the main indicator of our system efficiency.

VI. Conclusion

In this paper, we introduce a deep learning framework, called SADeepSense, for solving the heterogeneous sensing quality problem in IoT applications. SADeepSense 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 and gesture recognition show that SADeepSense greatly mitigates the performance 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.

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

[4] Sourav Bhattacharya and Nicholas 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] Guanling Chen, David Kotz, et al. A survey of context-aware mobile computing research. Technical report, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College, 2000.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] Davide Figo, Pedro C Diniz, Diogo R Ferreira, and Joao MP Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, 2010.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[9] Nils Y Hammerla, Reuben Kirkham, Peter Andras, and Thomas 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.

[10] Mohamed Abudulaziz Ali Haseeb and Ramviyas Parasuraman. Wisture: Rnn-based learning of wireless signals for gesture recognition in unmodified smartphones. *arXiv preprint arXiv:1707.08569*, 2017.

[11] Nicholas D Lane, Petko Georgiev, and Lorena 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.

[12] Fan Li, Chunshui Zhao, Guanzhong Ding, Jian Gong, Chenxing Liu, and Feng 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.

[13] Akhil Mathur, Tianlin Zhang, Sourav Bhattacharya, Petar Veličković, Leonid Joffe, Nicholas D Lane, Fahim Kawsar, and Pietro Lió. Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 200–211. IEEE Press, 2018.

[14] Shahriar Nirjon, Robert F Dickerson, Qiang Li, Philip Asare, John A Stankovic, Dezhi Hong, Ben Zhang, Xiaofan Jiang, Guobin Shen, and Feng 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.

[15] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller 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.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[17] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[18] Shuochao Yao, Md Tanvir Amin, Lu Su, Shaohan Hu, Shen Li, Shiguang Wang, Yiran Zhao, Tarek Abdelzaher, Lance Kaplan, Charu 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.

[19] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek 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.

[20] Shuochao Yao, Yiran Zhao, Huajie Shao, ShengZhong Liu, Dongxin Liu, Lu Su, and Tarek Abdelzaher. Fastdeepiot: Towards understanding and optimizing neural network execution time on mobile and embedded devices. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 278–291. ACM, 2018.

[21] Shuochao Yao, Yiran Zhao, Huajie Shao, Aston Zhang, Chao Zhang, Shen Li, and Tarek 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.

[22] Shuochao Yao, Yiran Zhao, Huajie Shao, Chao Zhang, Aston Zhang, Shaohan Hu, Dongxin Liu, Shengzhong Liu, Lu Su, and Tarek Abdelzaher. Sensegan: Enabling deep learning for internet of things with a semi-supervised framework. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):144, 2018.

[23] Shuochao Yao, Yiran Zhao, Aston Zhang, Shaohan Hu, Huajie Shao, Chao Zhang, Su Lu, and Tarek Abdelzaher. Deep learning for the internet of things. *Computer*, 51:32–41, May 2018.

[24] Shuochao Yao, Yiran Zhao, Aston Zhang, Lu Su, and Tarek 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.