A Closer Look at Quality-Aware Runtime Assessment of Sensing Models in Multi-Device Environments

Chulhong Min
Nokia Bell Labs, UK
chulhong.min@nokia-bell-labs.com

Alessandro Montanari
Nokia Bell Labs, UK
alessandro.montanari@nokia-bell-labs.com

Akhil Mathur
Nokia Bell Labs and University College London, UK
akhil.mathur@nokia-bell-labs.com

Fahim Kawsar
Nokia Bell Labs, UK and TU Delft
fahim.kawsar@nokia-bell-labs.com

ABSTRACT

The increasing availability of multiple sensory devices on or near a human body has opened brand new opportunities to leverage redundant sensory signals for powerful sensing applications. For instance, personal-scale sensory inferences with motion and audio signals can be done individually on a smartphone, a smartwatch, and even an earbud - each offering unique sensor quality, model accuracy, and runtime behaviour. At execution time, however, it is incredibly challenging to assess these characteristics to select the best device for accurate and resource-efficient inferences. To this end, we look at a quality-aware collaborative sensing system that actively interplays across multiple devices and respective sensing models. It dynamically selects the best device as a function of model accuracy at any given context. We propose two complementary techniques for the runtime quality assessment. Borrowing principles from active learning, our first technique runs on three heuristic-based quality assessment functions that employ confidence, margin sampling, and entropy of models' output. Our second technique is built with a siamese neural network and acts on the premise that runtime sensing quality can be learned from historical data. Our evaluation across multiple motion and audio datasets shows that our techniques provide 12% increase in overall accuracy through dynamic device selection at the average expense of 13 mW power on each device as compared to traditional single-device approaches.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; **Embedded software**.

KEYWORDS

quality assessment, sensing models, multi-device environments

ACM Reference Format:
Chulhong Min, Alessandro Montanari, Akhil Mathur, and Fahim Kawsar. 2019. A Closer Look at Quality-Aware Runtime Assessment of Sensing Models in Multi-Device Environments. In *The 17th ACM Conference on Embedded Networked Sensor Systems (SenSys '19), November 10–13, 2019, New York, NY, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3356250.3360043

SenSys '19, November 10–13, 2019, New York, NY, USA

ACM ISBN 978-1-4503-6950-3/19/11…$15.00
https://doi.org/10.1145/3356250.3360043

1 INTRODUCTION

For long, mobile sensing research has focused on creating accurate, robust and multi-modal sensory models for a single device making it intelligent at understanding us and the world around us [13, 17, 20, 28]. However, with an unprecedented rise of on/near-body devices - smartphones, wearables, or IoT devices - it is common today to find ourselves surrounded by multiple sensory devices. Studies (e.g., [31]) even estimate that by the year 2025, each person will have 9.3 connected devices. Therefore, we believe that efficient and accurate sensing in a multi-device environment is a key research area for the mobile sensing community.

Such multi-device dynamics offer exciting opportunities to leverage redundant sensory signals to develop accurate and robust sensing models quantifying richer human contexts continuously and at scale. This is mainly enabled by the fact that most of these devices share a common set of core sensors such as accelerometer, gyroscope, and microphone. For example, activity tracking can be done individually on a smartphone, a smartwatch, and even an earbud. Similarly, audio sensing can be done by selectively using a microphone on one of these devices or nearby IoT devices. In both cases, different devices offer varying sensor quality, model accuracy, runtime behaviour, and usage dynamics. Moreover, these characteristics change over time due to several factors including *device variability*, e.g., hardware and software heterogeneity [15, 34], compute and energy budget, etc. and *temporal variability*, e.g., device placement [19], a user's surrounding situation, etc.

These facets uncover critical system challenges to assess and compare the characteristics of different devices and to select the best device for accurate and resource-efficient sensory inferences at any given context. Naturally, research on quality[1]-aware sensing in a multi-device setting has been recently intensified. Several works in body sensor network have looked at sensor selection, but mostly taking a static view on average accuracy with resource awareness [6, 7, 10, 45]. Multi-sensory fusion has also been studied extensively to improve model accuracy while addressing system issues, such as time synchronisation and missing data [23, 25, 37, 41, 42]. These works contributed substantially to advance our understanding of multi-device sensing research. We build on this rich body of

[1]In this paper, we define quality as expected recognition accuracy of a sensing model for a given inference task at any point of time.

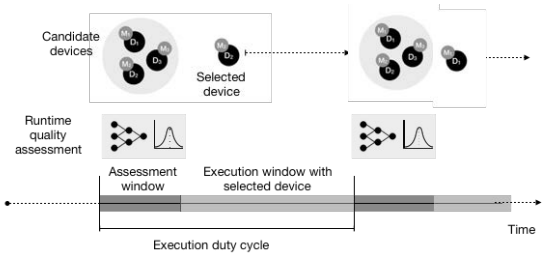


Fig. 1: An overview of the collaborative sensing system. We periodically assess the runtime sensing quality and choose the best device.

literature and investigate a specific challenge that lacked adequate attention from past research, namely:

"Given that inference accuracy of sensing models varies across devices and over time in multi-device environments, can we select a device – at runtime –, which is likely to provide the best inference accuracy in that instance?" If such a dynamic runtime selection system is developed, we can expect to have two benefits: (a) it will provide higher inference accuracy for a task than using a single device, (b) it will eliminate the redundant inference computations from multiple devices, e.g., fusion, thereby providing energy gains.

To this end, for the first time, we explore two complementary techniques for runtime assessment of sensing models and present a quality-aware collaborative sensing system. The system actively interplays between multiple devices and respective sensing models to dynamically select the best device for the recognition task at hand. Our first technique, heuristic-based quality assessment (HQA), borrows principles of certainties from active learning literature. We devise three functions operating on calibrated probabilities of models' output in deriving the quality of the recognition performance at runtime and selecting the device accordingly. Our second technique, learning-based quality assessment (LQA), is built on the premise that runtime sensing quality can be learned to predict the best device and corresponding inference path. We devise LQA by training a Siamese neural network [16]. Grounded on these techniques, we introduce a collaborative and quality-aware execution planner for model execution, as illustrated in Figure 1.

We evaluate these techniques with two motion and two audio datasets using three sensing tasks – physical activity, emotion, and keyword recognition. The results show that our techniques boost the overall recognition accuracy by 12% on average compared to single-device baselines. Energy expense of this accuracy gain is slightly higher, but less than 13 mW on average on each device. In comparison to voting and fusion baselines which require all the devices to be used continuously, our techniques show comparable accuracy, but considerably lower energy overhead – almost inversely proportional to the number of devices. We further provide an in-depth analysis to uncover the pros and cons of our methods. Finally, we present a well-being monitoring application with a smartphone, a smartwatch and a smart earbud to show the practical manifestation of our techniques in multi-device environments.

We first position our work against past research. Then, we take a data-driven view on the challenges of multi-device sensing systems. Next, we present the technical building blocks of this work, followed by systematic evaluation. Before concluding the paper, we reflect on the limitations and future avenue of this work.

2 RELATED WORK

We review past research on multi-device environments (MDE), focusing on sensor selection, sensor fusion, and system support.

2.1 Sensor Selection in MDEs

In body sensor networks (BSN), several sensing strategies have been proposed to recognise human contexts leveraging various on-body sensors. They have been studied to understand the effect of different characteristics on recognition accuracy, e.g., types, compositions, and placements of sensors. Grounded on these findings, several context-aware middlewares have been developed for dynamic BSN environments [6, 7, 10, 14, 45]. Their typical approach has been to dynamically select the best sensor based on predefined parameters, such as average accuracy, resource usage, and availability.

Although these works provided a foundation for designing execution strategies in dynamic BSN environments, their consideration of sensing quality has been limited. For instance, the selection is mostly made based on the average accuracy acquired during a training phase, which assumes that the sensing quality, i.e., expected accuracy, would be same as long as the associated sensors are available. However, we argue (and offer data-driven evidence in the next section) that quality of sensing models is different across devices and, more importantly, changes over time due to several runtime factors. This insight demands a revisit of these strategies because the best performing device dynamically changes even though the availability of sensor devices does not change. To the best of our knowledge, our work is the first to explore runtime assessment of sensing models in terms of the sensing accuracy in MDEs.

2.2 Sensor Fusion in MDEs

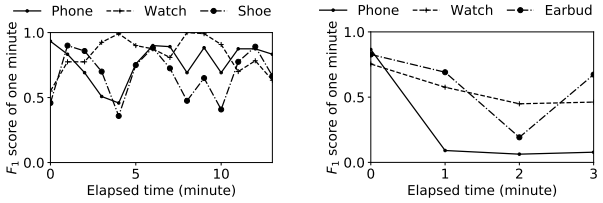
Deep learning-based fusion techniques [23, 25, 37, 41, 42] have been proposed to concatenate multiple sensor streams effectively – at an early or late stage – to achieve higher accuracy. Fusing sensory streams from different devices uncover a number of system issues, e.g., time synchronisation, missing data, different sampling rates. As such, several techniques have been studied to make fusion more robust to these factors. For instance, Yao et al. proposed a quality-aware deep learning framework that dynamically changes the contributions of sensor inputs by their sensing quality [42].

Although these works improve model accuracy, they often suffer from strong assumptions, e.g., all devices should be available and active for sensing, processing, and transmissions. Besides such practical infeasibility, these approaches also demand significant system resources, especially energy. Our work departs from these approaches and shows that, with carefully designed runtime quality assessment techniques, we can achieve comparable accuracy to the sensor fusion, however at a drastically smaller energy expense - almost inversely proportional to the number of devices.

2.3 System Support for Sensing in MDEs

Offering adequate system support for building efficient and accurate sensory applications in MDEs has also gained tremendous attention. Kang et al. proposed a novel system built on a translation function that transforms the high-level context queries into low-level sensor selection strategies, thereby hiding the details of complex resource management from applications [6, 7]. Kolamunna

et al. presented a framework for application function virtualisation of multi-wearable sensory systems isolating various system-level operations from application development [11]. Fortino et al. proposed a framework for multi-sensor data fusion supported by an underlying execution engine that offers several useful system-level capabilities, such as inter-BSN communication, proximity detection, and service discovery to simplify application development [3]. Our work contributes to this body of research by offering system-level components, i.e., an execution planner aided by runtime quality assessment, to accelerate sensory system development in MDEs.



(a) Activity recognition (IMU) **(b) Keyword detection (mic)**

Fig. 2: Variations in inference accuracy over time and across devices.

3 RUNTIME SENSING QUALITY DYNAMICS

Multi-device environments offer exciting opportunities for personal-scale sensing applications in various aspects. It is evident that **more** sensing tasks are possible as different devices have different characteristics. For instance, by virtue of different placements, recognising hand gestures from a smartwatch or head gestures from an earbud is a possibility today - which was almost impossible when personal sensory devices were limited to smartphones. Moreover, the multiplicity also means redundant capabilities, i.e., different devices can perform the same sensing task. Consider, one of the most dominant human sensing tasks - tracking physical activity. Today, this can be performed with an inertial measurement unit (IMU) on a smartphone [30], a smartwatch [2], or even an earbud [18]. Similarly, sound events or human speech can be recognised by using a microphone on one of these devices or nearby IoT devices (e.g., conversational agents and smart cars) at any given context. This redundancy means, with carefully orchestrated scheduling, sensing tasks can be supported **longer** by selectively using different devices at different times. Finally, since each device offers different runtime characteristics and sensing performances, we can achieve **better** recognition accuracy if the best device could be selected for the task at hand at every inference instance. In this work, we strive to address this **better** sensing challenge while optimising system cost. In what follows, we offer data-driven evidence that runtime recognition quality of sensing devices greatly varies across devices and over time, making it incredibly challenging to provide the promised improved sensing experience in multi-device environments.

3.1 Sensing Quality and Runtime Variability

For **better** sensing, it is essential to properly assess and compare sensing quality of available devices. Sensing quality has relation to several different, but relevant metrics such as inference accuracy, energy cost, and processing latency and can be defined as a combination of those metrics. In this work, we focus on the expected inference accuracy as a quality metric for two reasons. First, inference accuracy is the primary objective of sensing applications. Second, there are a number of tools to measure resource-related quality metrics at runtime, e.g., [24] for energy and [4] for latency, but runtime accuracy is not studied well yet, and its consideration as a system component still remains at an early stage.

A key challenge for runtime assessment of sensing quality is that it is almost infeasible to come up with a universal definition of the quality. There are several factors, spanning over hardware, software, system resources, and even users' behavioural characteristics that constitute and affect the sensing quality. More importantly, their impact on quality is different across devices and changes dynamically over time. In this work, we focus our attention on this specific characteristic and study two critical contributors to this runtime variability as described below:

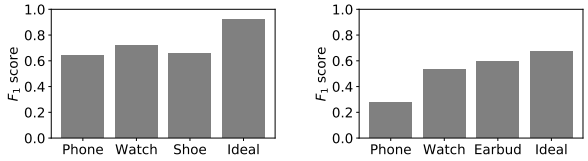
- **Device Variability**: Different devices have different sensing quality due to their heterogeneity [15, 34]. While the placement is a dominant factor, there are several other ones as well, e.g., sensor type and sampling rate. Processing and battery capacity can also be factors as they decide OS behaviour to sensory tasks.
- **Time variability**: Another important aspect, but not investigated actively yet in the research domain, is the temporal nature of sensing performance. That is, some factors affect sensing quality dynamically, thereby making the sensing quality unpredictable. For example, the activity recognition accuracy of the smartwatch could suddenly drop while users are typing with a keyboard in the office or pushing a stroller outside. The accuracy of audio sensing from nearby IoT devices could also change depending on a user's distance from the device.

3.2 Quantifying Runtime Variability

To further crystallise our arguments of runtime variability, we report a quantitative analysis on the impact of these variabilities on sensing quality with two sensing tasks, tracking physical activity and detecting keyword.

Setup: The goal of this study is to assess the degree to which sensing models' runtime accuracy varies across device over time. We focus on two widely-used sensing modalities - motion and audio for human-sensing tasks - human activity recognition (HAR) and hot keyword detection. For HAR, we use a subset of the Opportunity dataset [29] augmented with noise (see §5.1 for details) and a state-of-the-art deep HAR model [25]. For keyword detection, we collect a small dataset containing 200 keywords from 10 classes and a widely used keyword spotting model [40]. To understand the performance of these models in multi-device settings, we re-purpose some of the sources of the Opportunity dataset to different wearables based on their positions (See Table 1 for mapping. Similarly, for audio sensing task, we have collected the audio streams and trained models on three different devices, Pixel 3 phone, LG Urbane 2 watch, and a smart earbud [8]. For each sensing task, we trained and tested three different sensing models corresponding to three devices, but with the same architecture, e.g., the sensing model for the smartwatch is trained and tested with the data from the smartwatch.

Results: To quantify the impact of variability on the accuracy over time, we measure F_1 score at every minute in the test dataset. The window size for both tasks is set to 1 second. Thus, we compute



(a) Activity recognition (IMU) **(b) Keyword detection (mic)**

Fig. 3: Opportunity for quality-aware device selection.

F_1 score with 60 inference results every minute. Figure 2 shows F_1 score over time from a representative session, clearly suggesting that F_1 score varies across devices and fluctuates over time as well. More importantly, the best performing device also changes correspondingly. For example, Figure 2a shows that the shoe device outperforms the others from 1 to 2 minutes. Then, the best performing device changes to the watch at 2 minutes, the phone at 6 minutes, and the watch again at 7 minutes. Keyword detection using a microphone shows similar trend in Figure 2b.

Figure 2 implies that higher performance is achievable if we select the best device at every inference. To quantify this opportunity, we measure and compare F_1 score by assuming the *ideal* situation, i.e., at every inference, we select the device of which corresponding model provides the correct answer. Figure 3a shows the average F_1 scores of three sensing models and the ideal case for the HAR task. It suggests that, by selecting the best device, F_1 score can increase by 17% compared to the best performing single device (watch) and by 23% compared to the average score of three devices. Figure 3b shows a similar trend of an increase in F_1 score by 8% in the ideal case as compared to the best device (earbud). Such improvement uncovers brand-new opportunities for quality-aware selection.

With the Opportunity dataset, we further quantify the opportunity in the selection of *multiple* devices, e.g., two or three out of four available devices. For the study, we extend the entire sensor set to include five devices (See §5.3.2 for the selected devices) and develop the fusion models for all possible combinations of the subset of multiple devices. Then, while increasing the set size from 1 to 5, we select the best set of the devices of which corresponding fusion model gives the correct answer; in the case that the size is 1, we use the sensing model. The results show that the *one* device selection provides comparable accuracy to the selection of multiple devices. While F_1 score is 96.4% for the selection of one device, the scores are 96.9%, 97.1%, 97.2%, and 97.3% for the selection of 2, 3, 4, and 5 devices, respectively. Considering that the multiple device selection incurs significant high energy and additional training cost for the fusion model, we argue that selecting one device is reasonable.

4 QUALITY-AWARE DEVICE SELECTION

Based on the insights from the motivational studies above, we propose a collaborative sensing system that offers *quality-aware device selection* as a way to improve sensing accuracy in multi-device environments. Two concrete objectives shaped our design decisions towards the development of this system:

- **Model isolation**: The system should not require any changes to the underlying sensing models which need to be quality-assessed.
- **Application isolation**: The dynamic selection of devices should be a black-box for sensing applications, i.e., without requiring any modifications to the applications.

Governed by the design goals, we develop the collaborative sensing system composed of two critical components: *Quality Assessor* (§ 4.1) and *Execution Planner* (§ 4.2). Before detailing these components, we stress on the two assumptions that our approach makes.

- First, we assume that, in a multi-device environment, each device has a pre-trained task-specific sensing model deployed on it.
- Second, we assume that there is a host-device which can run our *quality-aware device selection* algorithm – the host could be a smartphone or a personal edge device, or can be dynamically chosen from the multiple sensing devices themselves.

4.1 Runtime Quality Assessment

The primary aim of quality-based device selection is to assess and compare the sensor data quality from each device at runtime. We re-emphasise that in this paper, quality is tied to the accuracy of the sensing task – i.e., given N sensor streams, the sensor stream which provides the highest accuracy of the inference task (e.g., HAR) for a specific time is said to have the best quality in our definition at that time. As such, the concept of quality in our work differs from traditional approaches such as signal-to-noise ratio (SNR), which are used to assess the purity of a signal but are completely agnostic of the inference task. For example, a speech audio segment with high background noise might have low SNR, but could represent a high-quality sample for a noise-detection inference task.

For the remainder of the section, we define the following:

- D: a set of available devices, $\{D^i\}$, where D^i is i^{th} device,
- M: a set of sensing models, $\{M^i\}$, where M^i is a model for D^i,
- X_t: a set of sensor data at time t, $\{X_t^i\}$, where X_t^i is a segment of sensor data from D^i at time t

Then, at time t, the quality-based device selection selects D^k at which $Q(S_t^i, M^i)$ is the highest among D, where Q is a quality function. We now present two methods to define Q.

4.1.1 Heuristic-based Quality Assessment (HQA). One way to assess sensing quality, i.e., defining Q, is to leverage confidence values reported from sensing classifiers. Typical examples are probabilities from probabilistic classifiers such as naive Bayes, probabilities from a softmax layer in neural networks, and distance to a hyperplane boundary in SVM. Since confidence values represent how confident a classifier is on the inference output from a given input data, we can interpret them as sensing quality. A probability value represents the probability of given sensor data being a member of each of the possible classes. Therefore, we can simply compute the highest probability outputted by a model M^i and use it as a measure of Q.

However, it is not straightforward to use confidence values as sensing quality. While modern neural networks provide exceptional accuracy, it is reported that their probabilities are not well calibrated, especially in multi-class classification [5]. That is, for a given sample, the relative order of probabilities among classes is accurate to select the most probable class, but the comparison of probabilities from different samples or different models could be inaccurate.

Several techniques have been proposed in the machine learning literature [21, 27, 43, 44] to calibrate the outputs of the classifiers and make them resemble the actual probability distribution of the training data. In this paper, we used Platt scaling [27] for calibrating our models because of its ease of implementation and good

performance with neural networks [22]. It performs the calibration by fitting a logistic regression model on the classifier output.

As a next step to quantify sensing quality from calibrated probabilities, we adopt *uncertainty sampling* strategies proposed in active learning literature to measure uncertainty of instances, i.e., how uncertain a given instance is to be labelled. Inspired by uncertainty sampling, we propose three heuristic methods for certainty-based quality assessment. We define calibrated probabilities of X_t^i from M^i as $P_t^i = \{P_{t,k}^i\}$ where $P_{t,k}^i$ is a probability value of X_t^i for M^i, belonging to a class C_k and C is a set of classes, $\{C_k\}$.

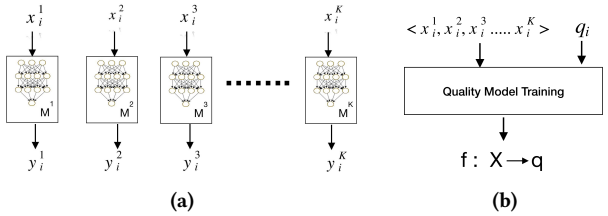
- **Highest confidence:** The simplest way is to leverage the *highest* confidence of each model. That is, the quality function, Q is defined as the confidence in the most likely label of given sensor data, i.e., $Q(S_t^i, M^i) = max(P_t^i)$.
- **Highest margin sampling:** The shortcoming of the highest confidence is that it disregards the remaining label distribution, i.e., P_t^i - $\{max(P_t^i)\}$. To handle this, a *margin sampling* has been proposed in [32]. It measures the certainty by taking the difference between the probabilities of the two most likely classes. Higher margin can be considered more confident. More specifically, Q is defined as the difference between the highest and second highest probability, i.e., $Q(S_t^i, M^i) = max(P_t^i)$ - $max_{2nd}(P_t^i)$, where max_{2nd} outputs the second maximum value.
- **Least entropy:** A more general strategy is to leverage *entropy* which refers to disorder or uncertainty [33]. Its concept has been proposed in information theory, but it is also widely used to measure uncertainty or impurity in machine learning. Here, the distribution of an instance with lower entropy can be considered more confident. Q is defined as $1 / -\sum_k P_{t,k}^i \log P_{t,k}^i$; we take a reciprocal to make the least entropy to be chosen.

In active learning, it is shown that, while all three strategies show reasonable performance compared to passive learning, the best strategy is task-specific [12]. In our datasets in §5.1, the highest margin sampling showed the best performance, thereby being used as a default quality function in the paper.

4.1.2 Learning-based Quality Assessment (LQA). In contrast to HQA, we also explore *learning-based quality assessment* wherein we adopt a data-driven approach to learn the *quality function* Q using deep neural networks (DNNs). We frame the problem of learning Q as a multi-task learning (MTL) problem [46]. In MTL, multiple learning tasks are solved simultaneously, while exploiting similarities and differences across them. In our case, the individual tasks assess the quality of each device's sensing stream with respect to the learning objective. As these tasks share significant commonalities, we can solve them simultaneously using an MTL framework.

For training a deep neural network to approximate the quality function, we seek a labelled training dataset $p(X, q)$. q is a k-dimensional vector representing the *quality* ground truth, where q^k represents the binary quality ground truth for D^k. For example, if there are three devices D^1, D^2, D^3, then $q = [1, 1, 0]$ means that D^1 and D^2 are suitable for the current task, while D^3 is not. If such a labelled dataset could be obtained, we can use it to train a neural network using supervised learning approaches.

Generating the *quality* ground truth: We denote the dataset for the quality assessment from K devices as $\mathbf{X} = \{X^1, X^2 ... X^K\}$ where



(a) (b)

Fig. 4: (a) Process to generate quality vector and (b) Overview of quality model training.

X^k $(k = 1...K)$ denotes the sensor dataset from D^k. Further, X_t^k denotes the sample at time t in X^k $(t = 1...N)$ and γ_t denotes the ground truth class for sample t. Note that as all the k sensor streams are time-synchronised, they all share the same ground truth at any time t. The training input $\mathbf{X}_t$ to the DNN is a k-tuple as follows:

$$\mathbf{X}_t =< X_t^1, X_t^2, X_t^3 \cdots X_t^K > \quad (1)$$

In addition, we need a ground truth *quality vector* $\mathbf{q}_t$ $(|\mathbf{q}_t| = k)$ which denotes the quality of each of the k sensor inputs:

$$\mathbf{q}_t =< q_t^1, q_t^2, q_t^3 \cdots q_t^K > \quad (2)$$

where q_t^k is a binary variable, $q_t^k \in \{0, 1\}$.

We explain our methodology of generating the *quality vector* $\mathbf{q}_t$. As discussed earlier, we define *quality* as the suitability of a device to a given sensing task, in that if the sensor data X_t^k from a device D^k accurately predicts the task-at-hand, then $q_t^k = 1$. We leverage pre-trained device-specific sensing models $M^k : X \rightarrow y$, which predict the output y given the sensor data X from device D^k. As shown in Figure 4a, at each step t $(t = 1....N)$, we apply the models M^k to the the data X_t^k to generate predictions y_t^k. Each prediction y_t^k is then compared with the ground truth class at step t: γ_t and a quality output q_i^k is generated as follows:

$$q_t^k = \mathbb{1}(y_t^k = \gamma_t) \quad (3)$$

In other words, if the prediction from device D^k matches the ground truth at step t, we say that the data from this device has good quality and set q_t^k as 1. Finally, as shown in Figure 4b we use $(\mathbf{X}_t, \mathbf{q}_t)$ as the labelled training samples to train the quality model.

Network architecture: We use a Siamese neural network architecture to train the model for the quality function; we call it the *quality model*. As shown in Figure 5, a Siamese neural network consists of two or more sub-networks (or towers), all of which share the same weights. Each tower receives sensor data X_t from a different device and extracts higher-order features, e.g., by using convolution and pooling layers. These feature vectors are then compared against each other by the subsequent layers of the network to assess the quality ordering across devices.

Our choice of using Siamese neural networks is motivated by two factors. a) *Input similarity*: As the input sensor data from multiple devices is of the same kind (e.g., time-aligned accelerometer data), we can use weight-sharing identical sub-networks to extract higher-order features and compare them. Similar approaches have been used in the image comparison literature [16] b) *Memory optimisation:* By sharing the weights across multiple sub-networks, we

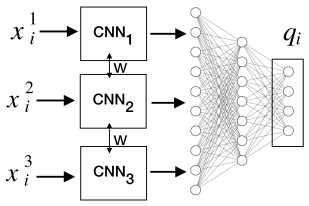


Fig. 5: Siamese neural network architecture. The feature extraction sub-networks CNN_k share the same weights w.

can significantly reduce the size of the quality model, thus making it more feasible to run on devices with limited runtime memory.

Implementation: We implement our Siamese neural network shown in Figure 5 in *Keras*. The model consists of k weight-sharing CNN towers (one for each of the k devices). The inputs to the towers (X_t) are raw or lightly-processed sensor data depending on the sensing task. For the HAR task, we input the raw IMU data to the device-specific towers. For the audio tasks, we extract log-spectrograms features and provide them as input to different towers. The input data is then processed by the CNN sub-networks to extract higher-order features. We use three 2D convolution and pooling layers for feature extraction, followed by a GlobalAveragePooling layer. The outputs of each tower are concatenated and compared using two fully-connected layers. As the individual tasks in our MTL model have binary outputs ('0' or '1'), we use *sigmoid* activation in the last layer of the network and *binary cross-entropy* as the loss function. The network is trained using the *Adam* optimiser.

4.1.3 Merits and Demerits. The two assessment methods have their merits and demerits and can be used selectively or in a collaborative way depending on the conditions. Here, we discuss their prerequisite and (de)merits. We provide a systematic evaluation of two methods across multiple motion and audio datasets in §5.1.

HQA: Compared to LQA, the biggest benefit of HQA is that it does not require additional data, e.g., for training a *quality model*. Also, it can easily adapt to changes in device availability, e.g., when a new device is added, or a device is temporarily unavailable, because the HQA function is computed individually. However, the HQA has limitations from the perspective of sensing models. First, the models need to provide confidence values as output, e.g., an output of the softmax layer in DNN, rather than a select label. Second, the confidence values need to be carefully calibrated with a large validation dataset, which is unusual for the conventional development of sensing models.

LQA: Different from HQA, LQA takes a data-driven approach, and thus its performance can enhance if a large dataset is available; note that the dataset required for training the quality model is not necessarily to be the same dataset used for training the sensing model. It is also possible to personalise the quality model using online learning. On the downside, when a new device is added to the ecosystem, the quality model needs to be learned again, thereby incurring an additional cost for computing and data. We discuss the potential solutions to address those limitations in §6.

4.2 Execution Planner

The execution planner is an essential component for the execution engine of runtime quality assessment. It uses the assessment output to dynamically orchestrate model execution across devices over time, as illustrated in Figure 1. It works as follows:

- **STEP 1:** At any given instance, it collects sensor data from all devices for a specific period (*assessment window*).
- **STEP 2:** Using either raw sensor data or processed features, it estimates the sensing quality of the available devices, applying runtime quality assessment functions (HQA or LQA) and chooses the device with the best quality for the underlying inference task. The selected device is used for the next k seconds (referred to as the *execution window*) for computing inferences. In the meantime, all the remaining devices are deactivated.
- **STEP 3:** After each execution duty cycle (assessment window + execution window), it re-runs the process of assessing the sensing quality and selecting the best device.

In § 5.3.1, we illustrate how different assessment windows and execution duty cycle values affect the performance of our system.

The execution planner runs on a smartphone in the current implementation, but can move dynamically depending on the device availability, e.g., moving from a phone to a watch if the phone becomes unavailable. We leave its implementation for future work.

5 EVALUATION

5.1 Experimental Setup

Sensing tasks: We focus our experiments on two widely-used sensing modalities, *motion* and *audio*, and representative sensing tasks for them. For motion sensing, we select *human activity recognition (HAR)*, which is performed over motion data captured from an Inertial Measurement Unit (IMU). For audio sensing, we choose *keyword spotting* and *emotion recognition*; their respective goals are to detect the presence of a keyword and the emotion of a speaking person in a given speech segment. These sensing tasks are suitable for multi-device environments because most of personal and IoT sensory devices are equipped with IMUs and microphones.

Datasets: To evaluate the performance of our approach in the HAR task, we use two public datasets which contain IMU data from multiple wearable devices: the Opportunity activity recognition dataset [29] and the RealWorld (HAR) dataset [35].

The *Opportunity dataset* consists of data collected from 4 participants performing naturalistic activities in a sensor-rich environment. While performing the activities, multiple IMUs are placed on a participant's body at different locations such as arms, back, and feet. Each IMU consists of 3-axis accelerometer and 3-axis gyroscope, sampled at 30Hz. Each participant performed 6 sessions: 5 runs of activity of daily living (ADL) where they performed daily activities with no constraints, and 1 drill session where they repeated 20 times a set of 17 activities. The ADL runs are characterised by larger variability. In our analysis, we used three devices on a hip, a left lower arm, and a right shoe by default and target to detect the mode of locomotion: *stand*, *walk*, *sit* and *lie*. In total, for all users and all recordings the dataset consists in 3653 modes of locomotion instances of variable duration (between ~0.2 and ~280 seconds).

The *RealWorld dataset* includes IMU data (3-axis accelerometer and 3-axis gyroscope) recorded from 15 participants performing 7 activities: climbing stairs down and up, jumping, lying, standing, sitting, running/jogging, and walking. The duration of each activity

Table 1: Device mapping from the datasets to energy measurements

Motion	**Pixel 3**	**LG Urbane 2**	**RPi Zero W**
Opportunity	hip	left lower arm	right shoe
RealWorld	thigh	forearm	head
Audio	**Pixel 3**	**LG Urbane 2**	**Rpi 3**
Keyword	Matrix	ReSpeaker	PlugUSB
Emotion	Matrix	ReSpeaker	PlugUSB

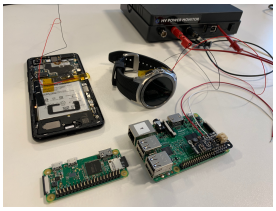


Fig. 6: Hardware setup of the power experiments.

for all participants is roughly 10 minutes except for jumping which is around 1.7 minutes due to physical exertion. Seven smartphones placed at different body positions have been used for data collection with a sampling rate to 50 Hz. In our evaluation, we use three devices on a thigh, a forearm, and the head by default. Refer to [29, 35] for additional details about the two datasets.

For audio sensing, we use the Keyword and Emotion datasets in [15]. The *Keyword* dataset consists of 2,250 1-second long speech files belonging to 10 keyword classes (yes, no, up, down, left, right, on, off, stop, go). The *Emotion* dataset is a collection of 1,440 English-language speech segments which were spoken by actors while expressing a range of emotions such as calm, happy, sad, angry, fearful, surprise, and disgust. The authors of [15] re-recorded these datasets at 16 kHz on three different embedded microphones simultaneously, thereby establishing a multi-device scenario that we aim in this paper. All datasets are split to ensure independence between the training and test set.

Noise augmentation: All four datasets were collected in a controlled setup and contain only a limited amount of real-world variabilities. As discussed in §3.1, device and time variabilities are two key factors that influence sensing accuracies – therefore we augmented real-life noise to the datasets in a principled manner to consider time variability; note that all datasets already contain device variability as they were recorded from different devices.

For the motion datasets, we used the data augmentation methods proposed in [36]. In [36], the authors introduced seven augmentation methods for IMU data on wearables: rotation, permutation, time-warping, scaling, magnitude-warping, jittering, and cropping. For example, *rotation* consists in augmenting data to reflect different sensor placement like an upside-down placement and *permutation* is to randomly perturb the temporal location of within-window events (See [36] for other methods). They enable us to consider real-life noise which cannot be observed in the data collected in a lab-controlled setting, e.g., rotating of a smartwatch. We randomly chose a period between 20 seconds and 2 minutes and applied a randomly selected augmentation method; the intervals between noise periods were randomly selected between 2 and 5 minutes. We repeated this procedure independently on each device and on each dataset to ensure time variability in a device-independent way.

For the audio, we sampled examples of real-world noise from the publicly available ESC-50 environment sound classification dataset [26] and added them to the audio datasets. The principle is that time variability of audio sensing mainly comes from various acoustic noises in the environment, e.g., vacuum cleaner noise and door creak – clearly, the amplitude of the noise received at each microphone will depend on its distance to the noise source and properties of the room. We assume that only a single type of noise is present in the environment at a given time, and each noise lasts for at least 15 seconds. We sampled a noise segment from the eight sources in [26] and augmented it for a randomly selected period between 15 and 60 seconds. More importantly, the amplitude of the noise is randomly selected for each microphone so that different microphones receive different noise powers.

Models. We use state-of-the-art DNN to build the task-specific sensing models. For HAR, we employ the architecture proposed in [25]. It consists of a CNN-based feature extractor with 4 residual blocks containing 2 convolutional layers each. They are followed by two fully-connected layers respectively with 1024 and 128 units, and then with an output layer of 4 units corresponding to our locomotion target classes. For the Keyword, we use the keyword detection architecture proposed in [40]. The input to this model is a two-dimensional tensor extracted from the 1-second-long keyword recording, consisting of time frames on one axis and MFCC on the other axis. The model consists of two convolutional layers, followed by a global average pooling layer and a fully-connected layer. Similarly, for the Emotion, we use a CNN architecture comprising of 2 convolutional layers and 1 fully-connected layer, proposed in [1].

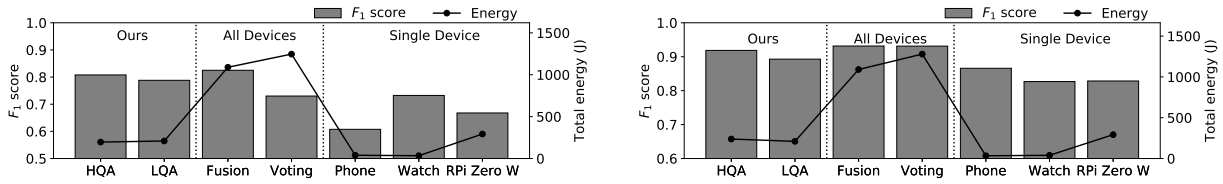
Baselines: We evaluate our approach against three baselines:

- Single: This baseline represents the traditional practice in sensing, where only a single device is used for sensing inference without any collaboration with other devices. Note that each model is trained for each device using the data from the very device.
- Voting: At any given point in time, we compute the inferences from each device individually, and select the output that has been predicted by the majority of the devices.
- Fusion: we train a fusion model that takes as input the sensor streams from all devices at the same time and outputs the predicted activity. We develop the fusion model based on [25].

For the parameters of HQA and LQA, we set 1 second to the assessment window size and 10 seconds to the duty cycle period.

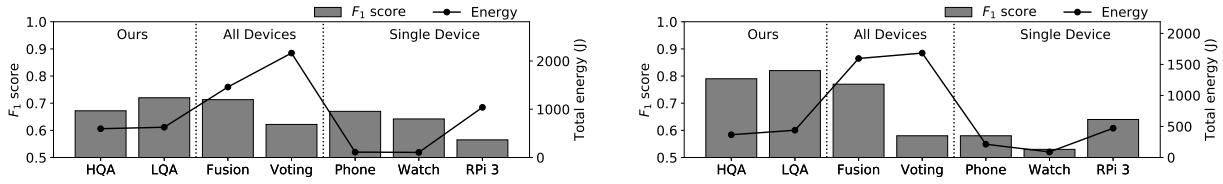
Performance metrics: We consider two metrics in the evaluation: recognition accuracy and resource utilisation. For the former, we use the micro-averaged F_1 score [38], which aggregates the contributions of all classes to compute the average metric. This metric is preferable in situations where the datasets are imbalanced (like the datasets used in this work). For the validation, we split all datasets into two parts; 70% for sensing model and 30% for a quality model for LQA. Then, we split each part again into training and test dataset. We used the leave-one-session-out validation method.

As a resource metric, we use the total energy consumed by the entire set of devices over a period of one hour (measured in Joules). To exhaustively evaluate the power characteristics of all configurations in an in-the-wild context, it would be necessary to perform



(a) Opportunity dataset (b) RealWorld dataset

Fig. 7: Performance comparison for activity monitoring tasks



(a) Keyword detection dataset (b) Emotion detection dataset

Fig. 8: Performance comparison for audio understanding tasks

multiple deployments, each one with a different set of parameter values. In fact, we need to test various combinations of execution planner parameters, quality assessment approaches, and different baseline models. To address this challenge, we devise an analytical energy model to estimate net energy increase in any given configuration, based on our power measurement of the individual operations, i.e. sensing, model execution, data transfer, but excluding the idle state. We present the details of the energy model and validate its accuracy with the Monsoon Power monitor in §5.4.

To consider realistic energy consumption, we mapped the devices in the dataset to four different off-the-shelf wearable devices, Pixel 3 smartphone, LG Urbane 2 smartwatch, Raspberry Pi Zero W (RPi Zero W), and Raspberry Pi 3 (RPi 3) as shown in Table 1. We decided the mapping based on the device placement for the motion dataset and the hardware resource for the audio dataset. RPi Zero W and RPi 3 represent wearable devices with IMU and space-embedded conversational agents with a microphone, respectively. Figure 6 shows the hardware setup of our devices and energy measurement.

5.2 Overall Performance

Figure 7 and 8 show the performance of our approaches compared with the baselines, for the motion and audio task, respectively. The bar graph represents the F_1 score and the line graph shows the total energy consumption of three devices; note that the scale of y-range for the energy consumption is all different. Overall, compared to the cases when only a single model is used, HQA and LQA have an absolute increase of 11% and 12% in the F_1 score over the average of all single models across all datasets, respectively. Even with the case when the best single model is used, HQA and LQA show 7% and 8% improvement of F_1 score. Such improvement is remarkable, considering that it is made just with the selection of devices without further optimising the sensing models nor leveraging more dataset. In terms of energy, the total energy consumption of HQA and LQA is 120 J and 131 J higher than the average consumption of the single models. Considering that three devices are used for one hour, it can be seen that HQA and LQA incur 11.0 mW and 13.1 mW of the power overhead on each device, which can be seen acceptable in daily use. The low power overhead of HQA and LQA is enabled by the duty-cycling based execution planning. The initial cost of transferring data from all devices to the one that runs the quality model or running multiple models gets amortised over time. As a result, the energy cost of the HQA and LQA with the combination of the execution planner becomes similar to the cost of having a single device producing inferences while the others are idle.

Also, HQA and LQA show the comparable F_1 score to *fusion* and outperform much *voting*. More specifically, the average F_1 score of HQA and LQA is 72%and 73%, respectively, whereas the score of fusion and voting is 73% and 63%. However, more importantly, the energy consumption of HQA and LQA is significantly lower than both fusion and voting. While HQA and LQA consume 349 J and 371 J on average across the dataset, fusion and voting consume 1,308 J and 1,598 J, respectively. Compared to single model cases, fusion and voting can be seen to consume additionally 99.9 mW and 126.2 mW on each device, respectively. Also, the power overhead of fusion and voting increases significantly as the number of available devices increases. We will discuss the detail in §5.3.

We analyse the characteristics for each dataset. Focusing first on the motion datasets, Figure 7 shows how HQA and LQA always outperform the single device baselines in terms of the inference accuracy. For example, the F_1 score of HQA and LQA is 81% and 79% on the Opportunity dataset in Figure 7a and 92% and 89% in the RealWorld dataset in Figure 7b. On the other hand, the on-average outperforming single device is the watch for Opportunity and the smartphone for RealWorld and their F_1 score is 73% and 87%, respectively. The results show that the proposed approaches are capable of selecting the most appropriate device and therefore achieve better performance over time compared to any single device.

For the total energy in the Opportunity dataset, HQA (196 J) and LQA (201 J) consume more energy than single-device models on the phone (39 J) and the watch (34 J). However, surprisingly, they consume less energy compared to the case when RPi Zero W was used continuously (293 J). For a deeper understanding, we investigate the per-device energy consumption of HQA and LQA. In HQA, the phone, the watch, and RPi Zero W consume 77 J, 46

J, and 73 J, respectively. In LQA, they consume 78 J, 41 J, and 90 J. The results show that, compared to the phone/watch-specific models, the energy increase of HQA and LQA mainly comes from two sources: one for the overhead of the operations for the device selection and the other for the high energy consumption on RPi Zero W for the sensing operations (See §5.4 for the details.) The energy saving of HQA and LQA is remarkably shown compared to the fusion (1089 J) and voting (1246 J) mechanisms. The fusion consumes 687 J, 270 J, and 133 J on the phone, watch, and RPi Zero W, respectively and the voting consumes 659 J, 267 J, and 320 J. The relatively high energy on the phone is also caused by receiving the data from all the devices and processing the fusion model or the voting decision. The per-device energy consumption is different depending on the dataset because they have different ratio of the selection of the devices. However, the other datasets also show the similar trend, i.e., energy increase of HQA and LQA compared to phone/watch baselines, but energy decrease compared to RPi Zero W and RPi 3 baselines. We omit the results in the other datasets.

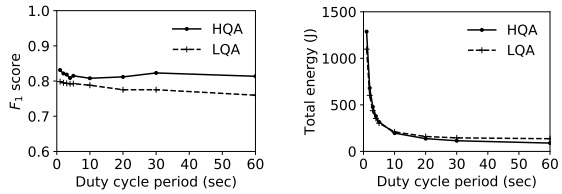
We compare the performance of our approaches to voting and fusion on the audio datasets in Figure 8. The results show that HQA and LQA show comparable F_1 score to *fusion* and outperform *voting*. More specifically, the F_1 score of HQA and LQA is 67% and 72% in Figure 8a and 79% and 82% in Figure 8b, respectively. The F_1 score of fusion is 71% and 77% and the F_1 score of voting is 62% and 58% for the Emotion and Keyword dataset, respectively. This is surprising, especially considering that HQA and LQA use one device at a time during the duty cycle period, whereas fusion and voting use all the devices continuously. We conjecture that this is because the devices with low sensing quality could degrade the performance of fusion and voting.

For the energy, fusion and voting consume significantly more energy than HQA, LQA, and single-device models. For example, the total energy consumption of fusion and voting is generally higher than 1400 J, whereas HQA and LQA remain under 700 J. While the energy consumption of HQA and LQA is higher than the energy consumed by a single model on a smartphone and a smartwatch, the increase is mainly contributed by the high power consumption of RPi 3. Also, the energy consumed by HQA and LQA is still lower than the energy of a single model on RPi 3. We discuss the details about the energy cost in §5.4.

Although fusion and voting show the high F_1 score, they are also the two methods with high energy cost. Data from all devices need to be transferred continuously to the device which runs the model in the fusion case and multiple models need to be executed for a single prediction in the voting case. On the contrary, the cost of the runtime quality assessment is incurred only during the assessment window, when selecting the best device. Therefore, the initial cost of transferring data from all devices to the one that runs the quality model or running multiple models gets amortised over time.

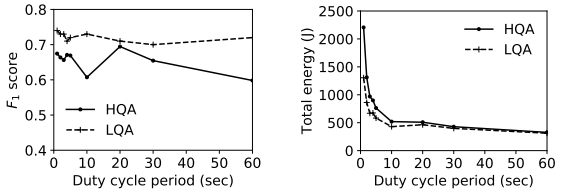
5.3 In-depth Analysis

We conduct an in-depth analysis to understand the behaviour of the quality-aware device selection more deeply. Due to space constraints, we report the analysis result on the Opportunity and Keyword dataset for the motion and audio task, respectively. The results on the RealWorld and Emotion dataset show similar trends.



(a) Effect on F_1 score **(b) Effect on total power**

Fig. 9: Effect of duty cycle (Opportunity); assessment size = 1



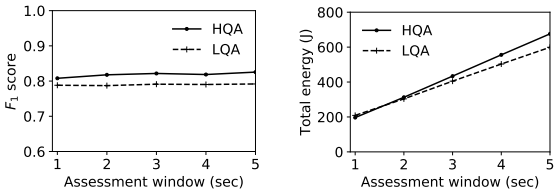
(a) Effect on F_1 score **(b) Effect on total power**

Fig. 10: Effect of duty cycle (Keyword); assessment size = 1

5.3.1 Effect of device selection parameters. The duty cycling is key to determine the performance of quality-aware device selection. We examine the effect of two parameters on the performance, execution duty cycle period and assessment window size.

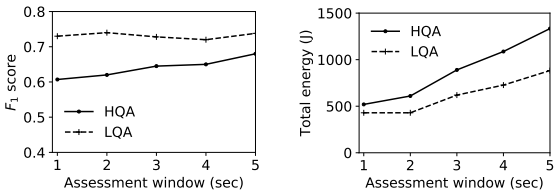
Execution duty cycle period: A key question is how far we can increase the duty cycle period to minimise accuracy loss and maximise energy saving at the same time. Figure 9 shows the F_1 score and total energy consumption for the motion task on the Opportunity dataset while varying the duty cycle period. With the increase of the period, the F_1 score tends to decrease for both HQA and LQA, but surprisingly, their decrease is marginal. The decrease of F_1 score when the duty cycle period was set to 1 second and 60 seconds is 2% and 4% for HQA and LQA, respectively. We also observe that the total energy consumption decreases exponentially as the duty cycle period increase. It is because the number of executions of quality assessment windows, in which more than one device is active, decreases accordingly. Figure 10 shows the results for the Keyword dataset. While the overall trend is similar, the F_1 score does not monotonically decrease even with the increase of the duty cycle period. We conjecture that this is because the temporal locality of the audio dataset is relatively more irregular. However, we can still observe that the F_1 score is the highest when the duty cycle period is set to the shortest, 1 second, which shows that our quality assessment methods select the most accurate device at each time interval. For a more systematical study of the selection of the duty cycle period, we plan to collect a larger dataset in real-life situations. We leave it as future work.

Assessment window size: We further investigate the effect of the assessment window size on the performance. When the assessment window size is larger than 1 second, we adopt the majority voting scheme to select the best device. Figure 11 shows the F_1 score and total energy consumption of the motion task with the Opportunity dataset. As expected, the F_1 score tends to increase both for HQA and LQA when the assessment window increases, but the increase is not significant. The increase for HQA is 2% the F_1



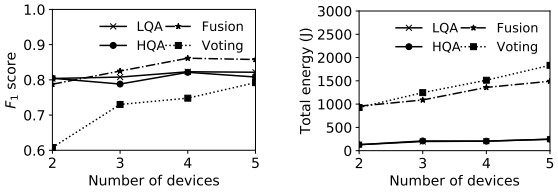
(a) Effect on F_1 score **(b) Effect on total power**

Fig. 11: Effect of assessment window (Opportunity); duty cycle = 10



(a) Effect on F_1 score **(b) Effect on total power**

Fig. 12: Effect of assessment window (Keyword); duty cycle = 10



(a) Effect on F_1 score **(b) Effect on total power**

Fig. 13: Effect of number of devices (Opportunity)

score when the assessment window is set to 1 second and 5 seconds is 81% and 83%, respectively. Similarly, the increase for LQA is 1%; the F_1 score is 78% and 79% when the window size is 1 second and 5 seconds, respectively.

On the other hand, the total energy consumption increases linearly with the assessment window size because larger data needs to be processed and transmitted. Figure 12 shows the results of the Emotion dataset. While the F_1 score of LQA saturates, the increase of HQA's score is clearly shown. The F_1 score of HQA increases from 0.61 to 0.68 when the assessment window size is set to 1 second and 5 seconds, respectively. We conjecture that it is because the voting scheme with longer window size compensates the relatively lower performance of HQA. The energy consumption shows a similar trend to that of the motion task.

5.3.2 Effect of number of devices . We examine the effect of the number of available devices. For the analysis, we use the Opportunity dataset as it contains multiple devices deployed on various locations on the participants' body. In addition to the default setting, we consider two more devices, one on a right lower arm and the other on a left shoe. For the energy estimation, we mapped two new devices to LG Urbane watch and RPI Zero 3, respectively. Figure 13 shows the results of our quality assessment methods (HQA and LQA) and two multi-device baselines (fusion and voting). Figure 13a shows their F_1 score while increasing the number of devices. Overall, the F_1 score of all schemes increases as the number of

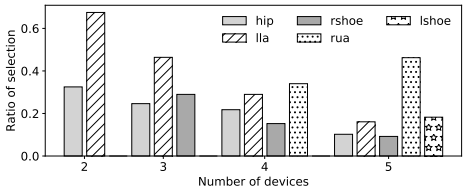


Fig. 14: Ratio of selection (LQA)

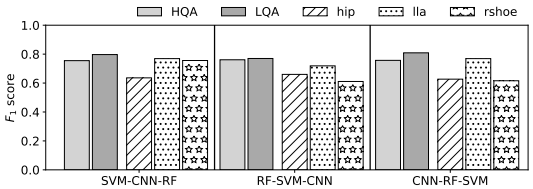


Fig. 15: Effect of heterogeneous classifiers; in each box, the two leftmost bars are HQA and LQA and the three rightmost bars are the sensing models (the type of classifiers is written in the X-axis label.)

available devices increases, as expected. However, while fusion and voting show a meaningful increase, the increase of HQA and LQA is marginal. The increase of the F_1 score when the number of devices is 2 and 5 is 0.07 and 0.19 for fusion and voting, respectively. On the contrary, the increase both of HQA and LQA is less than 2%. This is because the F_1 score of our quality assessment methods is constrained to that of single-device models, whereas fusion and voting can leverage larger and more diverse data altogether for each inference. However, regardless of the number of devices, LQA and HQA always show much higher F_1 score than voting and comparable score to fusion.

Figure 13b shows the total energy consumption while varying the number of devices. Fusion and voting consume almost linearly proportional to the number of devices because they require all the devices to process the sensing model and transmit the data all the time. However, the increase in power consumption of our quality assessment methods is marginal because the number of devices used most of the time is always 1. Also, surprisingly, the total energy consumption of LQA and HQA with 5 devices is still much lower than fusion and voting with 2 devices.

For a deeper understanding of the behaviour of our quality assessment methods, we look into how many times each device is selected. Figure 14 shows the ratio of the selection of the devices from LQA. The average F_1 score of single-device models is 61%, 78%, 67%, 78%, and 68% for the device deployed on hip, lower left arm, right shoe, right upper arm, and left shoe. Interestingly, we see that devices are mostly selected in order of their average F_1 score. In this regard, the device on a left lower arm was selected the most until the number of devices is 3 and the device on a right upper arm was selected the most whenever it was included. This shows that the sensing quality of devices is well reflected when LQA is adopted. We omit the result from HQA as it shows similar trends.

5.3.3 Effect of heterogeneous classifiers. We study the performance of our techniques when different types of classifiers are used. For the study, we further developed the sensing models with random forest (RF) and support vector machine (SVM) classifiers for the Opportunity dataset. Figure 15 shows the F_1 score with different

Table 2: Average power consumption for the motion model

Operation	Power (mW)		
	Smartphone	Smartwatch	RPi Zero W
Idle	28.08	27.81	442.58
Sense	7.24	8.59	29.42
Sensing Model	3.66	0.77	51.84
Quality Model	2.40	1.53	19.80
Bluetooth Tx/Rx	186.66	68.47	9.74

combinations of classifiers. Our techniques still provide meaningful benefits regardless of the classifier type, even though the benefit differs depending on the combinations of classifiers. For instance, in the case when RF, SVM, and CNN are used on the hip, left lower arm, and right shoe, respectively, F_1 scores of HQA and LQA increase by 10% and 11% compared to the average of the single-device accuracies. We omit the energy numbers as they are almost similar to the ones reported in Figure 7a.

5.4 System Overhead

We aim at understanding the overhead in terms of power consumption, incurred by the entire set of devices for the two runtime sensor quality assessment approaches introduced in §4. Figure 16 shows the main operations (model execution and data transfer) performed by the devices during the assessments and execution windows. We observe how the main difference is in the operations performed during the assessment window. In HQA, the assessment computation is essentially distributed across the devices since each one executes the local sensing model and reports the confidence value to the master which then uses an inexpensive heuristic to select the appropriate device. In LQA, the devices stream raw sensor data to the master, which then executes the quality model to select a device. LQA is more network intensive, because more data needs to be transferred (200Bytes/sec compared to 10Bytes/sec for HQA), but each device performs less computation locally.

Table 2 reports the power consumption measurements for each device when executing the quality assessment operations for the motion datasets. We find the smartphone and smartwatch to be energy efficient during computation but require significant energy during data transfers over Bluetooth. It implies that an execution model that requires more frequent and longer data transfers would incur a higher energy consumption over time. By contrast, the Raspberry Pi Zero consumes more energy during sensing and computation, probably due to a less power-optimised processor, which is not designed to run on batteries.

As we have seen in §5.2, there is a limited difference between the overall power consumed by the two approaches. That derives from the fact that the model execution, either the sensing model on each device or the quality model on the master, adds modest overhead to the overall power consumption, as highlighted in Table 2. Additionally, the limited amount of data transferred in the HQA does not provide a significant benefit because Bluetooth transmissions are the most expensive operations and we found only a very limited difference in power consumption when transmitting 10 or 200Bytes. This shows how, at least for typical smart devices like phones and watches, the potential for power optimisations could come from more efficient transmission media and protocols.

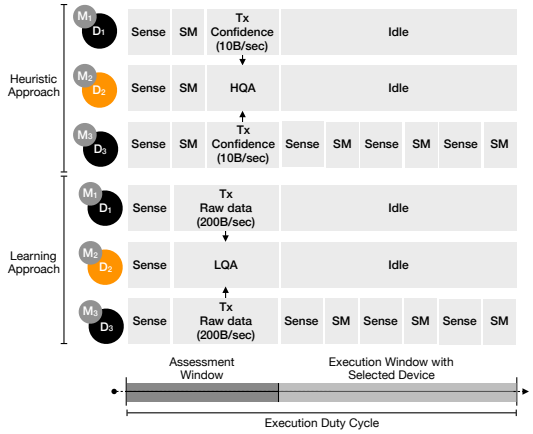


Fig. 16: Main operations involved in HQA and LQA. Device 2 (orange) is the master device responsible to perform the quality assessment and device selection. SM stands for Sensing Model.

Based on the power profiles in Table 2, we devise an energy model to estimate the total energy consumption from a sequence of operations. The energy model can be simply expressed as follows:

$$\sum_{d} \sum_{i} t_i \times p_i$$

where d is a device, t_i is the duration of ith operation and p_i is the power profile of ith operation. However, in the actual implementation, we adopt a finite state machine-based power model to consider power state of hardware components and sharing effect [24]. We verified our power model with four 30-minute-long sequences. We developed an Android application that executes a script of operations (as in Figure 16). While running the application, we measured the energy consumption with Monsoon power monitor and compared it with our estimation obtained from our power model. The results show that our proposed model achieved 91.2% of the accuracy on average across the devices.

5.5 Deployment Study

We report a small-scale deployment study that demonstrates the manifestation of different techniques discussed in this paper in a real-world artefact.

Application and devices: We developed a personal-scale well-being monitoring application that captures motion activities using an IMU (walking, standing, and sitting) and emotional states using a microphone. These attributes are then summarised to offer users with visual and conversational feedback of their physical and mental well-being (see Figure 17a). The application runs on a smartphone and can connect to multiple sensory devices providing motion and audio data. In our experiment, these devices were Google Pixel 3 Smartphone, LG Urbane 2 Watch, and an earbud in [8] - each containing an accelerometer, a gyroscope, and a microphone. The watch and earbud are connected to the smartphone over BLE for transferring data as needed. We used the same activity and emotion models as described in the earlier section. The application uses the runtime quality assessment and execution planning capabilities to selectively use one of these devices and their corresponding

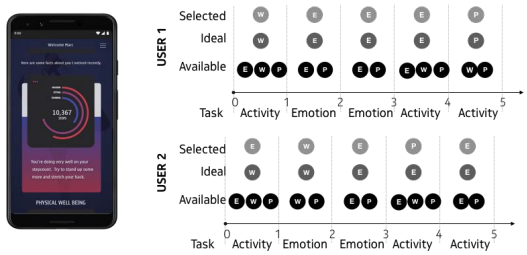


Fig. 17: Prototype well-being management application, (a) screenshot (left), and (b) timelines of two users sessions (right).

models to track the well-being attributes. Please note that we do not claim the novelty and performance of this application; instead, we evaluate our quality-aware device selection strategy through this well established sensing service.

Study design and settings: We followed the *scenario-based analysis* methodology [9] to evaluate the performance of our quality-aware device selection strategy. Essentially, we developed a 10-min experiment session with 10 one minute blocks. In each block, we exposed the users to a different condition concerning devices. These situations include varying device *1) placement, 2) distance, and 3) availability.* These conditions were randomly assigned to different blocks, however, counterbalanced across participants.

We recruited a group of 12 people using stratified sampling. Each user took part in a 10-min session with three devices performing a set of motion activities (walking, standing, sitting) or speaking with different prosody to express positive or negative emotion following predefined scripts. We set the length of the assessment window and execution duly cycle to 1 and 10 seconds, respectively. This decision means the quality assessment function was executed 60 times for each experimental session, contributing to 480 operations in total across all the sessions. Out of 120 1-min blocks across all participants, HQA (margin sampling) was used for 60 blocks, and LQA for the rest. In both cases, half of the blocks were for activity and the other half for emotion tracking. For each of this block, we assigned the ideal device that we later use for quality assessment.

Results: Given the limited scale and constrained settings of the study, we only report a subset of results that are indicative of the real-world performance of the proposed techniques. Essentially, out of 120 blocks, in 107 blocks (89.2%) a device was selected correctly. HQA selected 53 (88.3%) and LQA selected 54 (90.0%) devices correctly. Please note that we are reporting the selection performance at a 1-minute granularity with the dominant device for that minute. Figure 17b illustrates two representative timelines of the experiment with activity and emotion tracking.

6 DISCUSSION

We have explored the possibility of assessing the quality of sensing models at runtime using two different approaches: HQA which relies on the confidence values reported by the sensing models and LQA which instead adopts a data-driven approach to learn a function which determines the quality of the models. We have found that while both approaches lead to increased prediction performance with a reduction in energy cost, the difference of these two metrics between the two methods is often not significant. This has implications on the criteria used to select one method or the other. In §4.1.3, we mentioned that important drawbacks of the LQA approach are the need for labelled training data and the limited flexibility to changes in the device topology once the quality model has been trained. A possible solution to the first issue could be researched in the use of non-supervised machine learning techniques to attempt to learn the quality model without ground truth data. For the second issue, we envision the possibility of integrating a series of attention blocks [39] in the architecture of our siamese model. This should allow the model to automatically cope with the absence of sensor data in input by weighting the internal features accordingly to the data available at each moment in time. In addition, our work does not evaluate the generalisability of the LQA approach to new devices – future research should evaluate transfer learning possibilities for LQA, that is, whether the weights of a pre-trained Siamese neural network could be used to initialise the training process of another LQA model for new devices. Transfer learning and domain adaptation techniques can significantly reduce the amount of data needed to train LQA models for new scenarios.

Another aspect which is currently not included in HQA and LQA is the possibility of including resource metrics into the quality assessment computation. It could be useful to include information about the current status of the devices, such as remaining battery or CPU and memory load, as additional inputs for the device selection procedure, e.g., selecting the device in a way of balancing the battery life of the devices while achieving the reasonable accuracy.

This ties into a possible improvement of the execution planner, which can be dynamic and allow the definition of policies to prioritise accuracy or energy efficiency. As presented in §5.3, the parameters of the execution planner could be adjusted to obtain better prediction performance or more energy efficiency. However, these two aspects might require adaptation over time as the system status evolves. For example, as the energy available in the devices starts depleting, longer duty cycle windows could be adopted.

7 CONCLUSION

In this paper, we presented two complementary techniques for runtime assessment of sensing models and a quality-aware collaborative sensing system based on them for multi-device environments. Borrowing principles from active learning, our first technique operates on three heuristic-based quality assessment functions (HQA) that employ confidence, margin sampling, and entropy of models' output respectively. Our second technique, a learning-based quality assessment function (LQA) is built with a siamese neural network and acts on the premise that runtime sensing quality can be learned from historical data. Our systematic evaluation across multiple motion and audio datasets shows that these techniques can boost 12% increase in overall inference accuracy through dynamic device selection at the average expense of less than 13 mW power on each device while compared against the single-device approach. In addition, they consume considerably lower energy than the fusion approach - almost inversely proportional to the number of devices.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and shepherd for their valuable comments to improve the quality of this paper. We also thank TRAN Huy Vu for his aid in data analysis and experiments.

REFERENCES

[1] Abdul Malik Badshah, Jamil Ahmad, Nasir Rahim, and Sung Wook Baik. 2017. Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network. In *2017 International Conference on Platform Technology and Service (PlatCon)*. 1–5. https://doi.org/10.1109/PlatCon.2017.7883728

[2] Sourav Bhattacharya and Nicholas D Lane. 2016. From smart to deep: Robust activity recognition on smartwatches using deep learning. In *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. 1–6. https://doi.org/10.1109/PERCOMW.2016.7457169

[3] Giancarlo Fortino, Stefano Galzarano, Raffaele Gravina, and Wenfeng Li. 2015. A framework for collaborative computing and multi-sensor data fusion in body sensor networks. *Information Fusion* 22 (2015), 50–70.

[4] André Günther and Christian Hoene. 2005. Measuring round trip times to determine the distance between WLAN nodes. In *International conference on research in networking*. Springer, 768–779.

[5] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1321–1330.

[6] Seungwoo Kang, Jinwon Lee, Hyukjae Jang, Hyonik Lee, Youngki Lee, Souneil Park, Taiwoo Park, and Junehwa Song. 2008. SeeMon: Scalable and Energy-efficient Context Monitoring Framework for Sensor-rich Mobile Environments. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*. ACM, New York, NY, USA, 267–280. https://doi.org/10.1145/1378600.1378630

[7] Seungwoo Kang, Youngki Lee, Chulhong Min, Younghyun Ju, Taiwoo Park, Jinwon Lee, Yunseok Rhee, and Junehwa Song. 2010. Orchestrator: An active resource orchestration framework for mobile context monitoring in sensor-rich mobile environments. In *2010 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 135–144. https://doi.org/10.1109/PERCOM.2010.5466982

[8] Fahim Kawsar, Chulhong Min, Akhil Mathur, and Alesandro Montanari. 2018. Earables for Personal-Scale Behavior Analytics. *IEEE Pervasive Computing* 17, 3 (Jul 2018), 83–89. https://doi.org/10.1109/MPRV.2018.03367740

[9] Rick Kazman, Gregory Abowd, Len Bass, and Paul Clements. 1996. Scenario-based analysis of software architecture. *IEEE Software* 13, 6 (Nov 1996), 47–55. https://doi.org/10.1109/52.542294

[10] Matthew Keally, Gang Zhou, Guoliang Xing, Jianxin Wu, and Andrew Pyles. 2011. PBN: Towards Practical Activity Recognition Using Smartphone-based Body Sensor Networks. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys '11)*. ACM, New York, NY, USA, 246–259. https://doi.org/10.1145/2070942.2070968

[11] Harini Kolamunna, Yining Hu, Diego Perino, Kanchana Thilakarathna, Dwight Makaroff, Xinlong Guan, and Aruna Seneviratne. 2016. AFV: Enabling Application Function Virtualization and Scheduling in Wearable Networks. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 981–991. https://doi.org/10.1145/2971648.2971727

[12] Christine Körner and Stefan Wrobel. 2006. Multi-class ensemble-based active learning. In *European conference on machine learning*. Springer, 687–694.

[13] Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. 2010. A survey of mobile phone sensing. *IEEE Communications Magazine* 48, 9 (Sep. 2010), 140–150. https://doi.org/10.1109/MCOM.2010.5560598

[14] Youngki Lee, Chulhong Min, Younghyun Ju, Seungwoo Kang, Yunseok Rhee, and Junehwa Song. 2014. An Active Resource Orchestration Framework for PAN-Scale, Sensor-Rich Environments. *IEEE Transactions on Mobile Computing* 13, 3 (March 2014), 596–610. https://doi.org/10.1109/TMC.2013.68

[15] Akhil Mathur, Anton Isopoussu, Fahim Kawsar, Nadia Berthouze, and Nicholas D. Lane. 2019. Mic2Mic: Using Cycle-consistent Generative Adversarial Networks to Overcome Microphone Variability in Speech Systems. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks (IPSN '19)*. ACM, New York, NY, USA, 169–180. https://doi.org/10.1145/3302506.3310398

[16] Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. 2016. Siamese network features for image matching. In *2016 23rd International Conference on Pattern Recognition (ICPR)*. 378–383. https://doi.org/10.1109/ICPR.2016.7899663

[17] Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. 2008. Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*. ACM, New York, NY, USA, 337–350. https://doi.org/10.1145/1460412.1460445

[18] Chulhong Min, Akhil Mathur, and Fahim Kawsar. 2018. Exploring Audio and Kinetic Sensing on Earable Devices. In *Proceedings of the 4th ACM Workshop on Wearable Systems and Applications (WearSys '18)*. ACM, New York, NY, USA, 5–10. https://doi.org/10.1145/3211960.3211970

[19] Chulhong Min, Akhil Mathur, Alessandro Montanari, and Fahim Kawsar. 2019. An Early Characterisation of Wearing Variability on Motion Signals for Wearables. In *Proceedings of the 23rd International Symposium on Wearable Computers (ISWC '19)*. ACM, New York, NY, USA, 166–168. https://doi.org/10.1145/3341163.3347716

[20] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. 2008. Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*. ACM, New York, NY, USA, 323–336. https://doi.org/10.1145/1460412.1460444

[21] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[22] Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting Good Probabilities with Supervised Learning. In *Proceedings of the 22Nd International Conference on Machine Learning (ICML '05)*. ACM, New York, NY, USA, 625–632. https://doi.org/10.1145/1102351.1102430

[23] Francisco Javier OrdÃşÃśez and Daniel Roggen. 2016. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* 16, 1 (2016). https://doi.org/10.3390/s16010115

[24] Abhinav Pathak, Y. Charlie Hu, and Ming Zhang. 2012. Where is the Energy Spent Inside My App?: Fine Grained Energy Accounting on Smartphones with Eprof. In *Proceedings of the 7th ACM European Conference on Computer Systems (EuroSys '12)*. ACM, New York, NY, USA, 29–42. https://doi.org/10.1145/2168836.2168841

[25] Liangying Peng, Ling Chen, Zhenan Ye, and Yi Zhang. 2018. AROMA: A Deep Multi-Task Learning Based Simple and Complex Human Activity Recognition Method Using Wearable Sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2, Article 74 (July 2018), 16 pages. https://doi.org/10.1145/3214277

[26] Karol J. Piczak. 2015. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd ACM International Conference on Multimedia (MM '15)*. ACM, New York, NY, USA, 1015–1018. https://doi.org/10.1145/2733373.2806390

[27] John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers* 10, 3 (1999), 61–74.

[28] Kiran K. Rachuri, Mirco Musolesi, Cecilia Mascolo, Peter J. Rentfrow, Chris Longworth, and Andrius Aucinas. 2010. EmotionSense: A Mobile Phones Based Adaptive Platform for Experimental Social Psychology Research. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp '10)*. ACM, New York, NY, USA, 281–290. https://doi.org/10.1145/1864349.1864393

[29] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. 233–240. https://doi.org/10.1109/INSS.2010.5573462

[30] Charissa Ann Ronao and Sung-Bae Cho. 2016. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications* 59 (2016), 235–244.

[31] Bardia Safaei, Amir Mahdi Hosseini Monazzah, Milad Barzegar Bafroei, and Alireza Ejlali. 2017. Reliability side-effects in Internet of Things application layer protocols. In *2017 2nd International Conference on System Reliability and Safety (ICSRS)*. 207–212. https://doi.org/10.1109/ICSRS.2017.8272822

[32] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active Hidden Markov Models for Information Extraction. In *Advances in Intelligent Data Analysis*, Frank Hoffmann, David J. Hand, Niall Adams, Douglas Fisher, and Gabriela Guimaraes (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 309–318.

[33] Claude Elwood Shannon. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal* 27, 3 (1948), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x

[34] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart Devices Are Different: Assessing and MitigatingMobile Sensing Heterogeneities for Activity Recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys '15)*. ACM, New York, NY, USA, 127–140. https://doi.org/10.1145/2809695.2809718

[35] Timo Sztyler and Heiner Stuckenschmidt. 2016. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 1–9. https://doi.org/10.1109/PERCOM.2016.7456521

[36] Terry T. Um, Franz M. J. Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data Augmentation of Wearable Sensor Data for Parkinson's Disease Monitoring Using Convolutional Neural Networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI '17)*. ACM, New York, NY, USA, 216–220. https://doi.org/10.1145/3136755.3136817

[37] Yonatan Vaizman, Nadir Weibel, and Gert Lanckriet. 2018. Context Recognition In-the-Wild: Unified Model for Multi-Modal Sensors and Multi-Label Classification. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 168 (Jan. 2018), 22 pages. https://doi.org/10.1145/3161192

[38] Vincent Van Asch. 2013. Macro-and micro-averaged evaluation measures. *Belgium: CLiPS* (2013).
[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
[40] Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209* (2018).
[41] 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 *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 351–360. https://doi.org/10.1145/3038912.3052577
[42] 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 *Proceedings of the 2Nd International Workshop on Embedded and Mobile Deep Learning (EMDL'18)*. ACM, New York, NY, USA, 42–47. https://doi.org/10.1145/3212725.3212729
[43] Bianca Zadrozny and Charles Elkan. 2001. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Icml*, Vol. 1. Citeseer, 609–616.
[44] Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 694–699.
[45] 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*, Roberto Verdone (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 17–33.
[46] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017).