Low-Cost, Edge-Cloud, End-to-End System Architecture for Human Activity Data Collection

Matteo Fresta[1[0009-0000-7265-7501]], Ali Dabbous[1[0009-0004-8978-4979]], Francesco Bellotti[1[0000-0003-4109-4675]], Alessio Capello[1[0000-0003-4277-7283]], Luca Lazzaroni[1[0000-0001-8092-5473]], Alessandro Pighetti[1[0009-0001-7166-5750]], Riccardo Berta[1[0000-0003-1937-3969]]

[1] Department of Electrical, Electronic and Telecommunication Engineering (DITEN)
University of Genoa, Via Opera Pia 11a, 16145 Genova, Italy.
{matteo.fresta, ali.dabbous, alessio.capello, luca.lazzaroni, alessandro.pighetti}@edu.unige.it,
{francesco.bellotti, riccardo.berta}@unige.it

Abstract. Research in the Internet of Things (IoT) have paved the way to a new generation of applications and services that collect huge quantities of data from the field and do a significant part of the processing on the edge. This requires availability of efficient and effective methodologies and tools for a workflow spanning from the edge to the cloud. This paper presents a generic, complete workflow and relevant system architecture for field data collection and analysis with a focus on the human physical activities. The data source is given by a low-cost embedded system that can be placed on the user body to collect heterogeneous data on the performed movements. The system features a 9 DoF IMU sensor, to ensure a high level of configurability, connected to a custom board equipped with a rechargeable battery for wireless data collection. Data are transmitted via Bluetooth Low Energy (BLE) to a smartphone/tablet app, which manages the data transfer to Measurify, a cloud-based open-source framework designed for building measurement-oriented applications. Results from a preliminary functional experiment confirm the ability of the proposed end-to-end system architecture to efficiently implement the whole targeted edge-cloud workflow.

Keywords: Field data collection, human activity recognition, Internet of Things, sensor-based classification, wearable sensor, edge-cloud hardware/software architectures, embedded systems.

1 Introduction

In the Internet of Things (IoT) scenario, collecting large amounts of data from sensors and storing them within a cloud database has become a major challenge [1]. The need for data is constantly increasing with the growing development and spread of ever new machine learning (ML) technologies that rely on supervised learning techniques, thus requiring datasets.

For this reason, obtaining large amounts of data quickly and easily, while maintaining high accuracy during acquisition, is crucial for training accurate ML models. For this purpose, various runtime applications have been developed to handle the data generated by edge devices (e.g. [2]). The challenge is also to build a “smart” database that adapts as closely as possible to the resources being sent [3] and that allows data to be managed using a simple, lightweight and fast sending method, while still maintaining a high level of security during data exchange. One of the possible applications of this paradigm, based on an embedded system and requiring a flexible database due to the diversity of data to be collected, is the classification of specific actions during a human physical activity. Over the past decades, motion classification has been a constantly growing research area. Sensors are applied to the human body to accurately represent movements [4] and collect as much data as possible to best support the training of state-of-the-art ML algorithms in a smart controlled environment [5].

In this paper, a data collection workflow is implemented from the physical sensor to a database to manage the measurements. This embedded system consists of an Arduino Nano 33 BLE Sense mounted on a custom board with a battery installed for autonomous energy support. The IMU sensor on the device provides accelerometer, gyroscope and magnetometer data that will be collected and sent to the database. We were looking for a flexible and data-oriented framework that best fit our use-cases, so the choice fell on Measurify, formerly Atmosphere [6], as it is an open-source, cloud-based, measurement-oriented API Framework, which is connected to MongoDB[7] as database.

A Flutter [8] application, that can be installed on any tablet or mobile phone device, is used to store in memory a great number of data received from Arduino without stopping the data collecting phase, this maximizes the amount of data obtainable per minute and simplify the connection between the embedded system and Measurify.

The design choices aim to ensure that this workflow remains easily accessible by adopting an open-source approach, with all components available for download on Github [9]. The instrumentation is intentionally low-cost, using mainly Arduino as the only physical device, while all other components are free-to-use and can be hosted locally. This configuration allows for widespread adoption and easy replicability, making it feasible for a broad audience to participate in and benefit from the workflow.

2 Workflow

The proposed workflow can be decomposed into three main sectors as shown in Fig. 1: Edge, Fog, and Cloud. These represent the spectrum of distributed computing, from immediate data processing at the source (Edge), intermediate processing in local networks (Fog), to centralized processing and storage in remote servers (Cloud). The Edge consists of a versatile wearable embedded system designed to be attached to any part of the human body. Its primary function is to collect data while the wearer performs specific actions or activities. On the other hand, the Fog sector comprises a Flutter application and a personal device, which together facilitate the visualization of the data obtained from the wearable embedded system. Lastly, the Cloud sector consist of the Measurify Framework, which acts as a receiver for data sent from the Flutter application and saves them into a MongoDB database, enabling secure storage for future use and analysis.

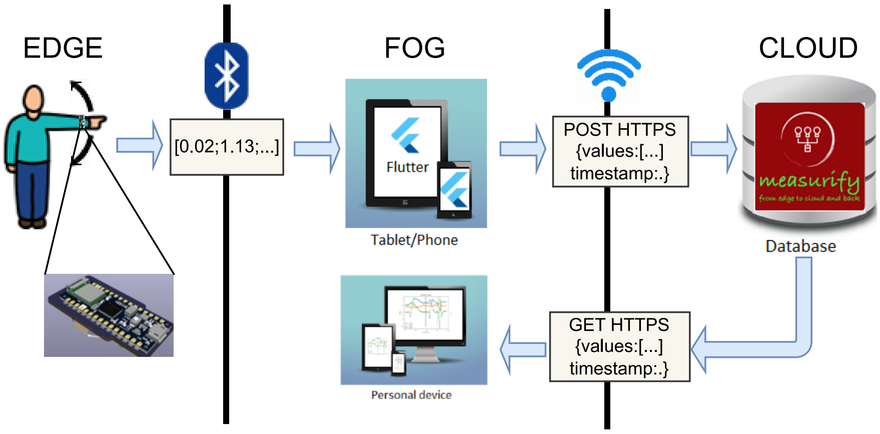

Fig. 1. Workflow enabled by the proposed system architecture.

The embedded system used consists of an Arduino NANO 33 BLE Sense soldered to a board as shown in Fig. 2, which allows a rechargeable battery to be installed in order to power the Arduino and data to be collected wirelessly.

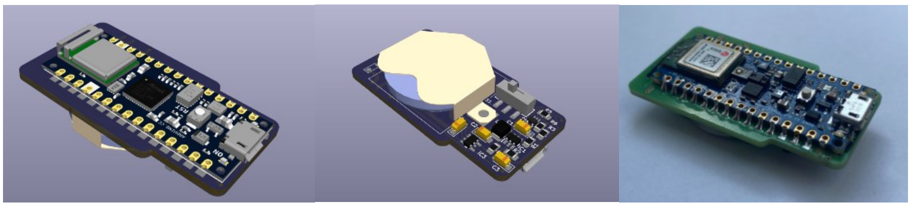

Fig. 2. Arduino board with battery: on the left, up-side prototype design; in the center, bottom-side prototype design; on the right the board used.

The board features the 9-axis IMU LSM9DS1 sensor: a 3D accelerometer with a default range [-4, +4] g -/+0.122 mg, 3D gyroscope with a default range [-2000, +2000] dps +/-70 mdps and 3D magnetometer with a default range [-4, +4] gauss +/-0.14 mgauss. With an LIR 2450 battery with a capacity of 120 mAh it is estimated that data can be collected for 6 consecutive hours.

Sensor data are collected by the Arduino via a script with a minimum sampling period of 5 ms and transmitted via Bluetooth Low Energy (BLE). The script sets up the Arduino to expose Bluetooth services with features that a device can subscribe to and be notified whenever new data are available.

To receive these data, a custom application was developed for the subscription of the characteristics exposed by the Arduino and to assign a label to the measurement, and then send it via an HTTPS POST RESTful API to the Measurify framework in the format of timeseries, a type of data that contains the numerical values and the timestamp in which it was measured.

This application can be installed on a tablet or mobile phone and was developed using the open-source Flutter framework [8] and it connects to the embedded device via BLE and stores the received numerical data in memory. Once the user stops the data recording, the stored dataset is sent to the Measurify's timeseries route through a HTTPS POST. While the process is running, values are sent in blocks of 1000 samples as they are collected. Request's body encapsulate values organized in JSON format. An example of a timeseries sample is as follows:


```
{
   "timestamp": "1684833177652",
  "values": [-0.504883,-0.401733,-0.751587,32.959,-66.284,123.474,0,0,0]
}
```


This protocol minimizes the amount of space used during the calls and speeds up the data transmission. For the data visualization, it is possible to get the values via a HTTPS GET request from a personal device. The timeseries route, secured through authentication, allows user to retrieve previously inserted values in common formats: JSON, Pandas Dataframe, and CSV [10]. User can also filter measurements to obtain only samples in a specific period, or to retrieve only values exceeding a certain threshold.

3 Results

We performed a functional test for our system with a very simple preliminary data collection experiment. For the test, the sampling period of the device was set to 250 ms and the default sensitivity values of the IMU sensors of the Arduino Nano 33 BLE Sense were used (Table 1). The dataset collected consists of the samples taken during the action of repeatedly raising and lowering an arm progressively increasing the movement speed. To perform the test, we attached the Arduino near the wrist of the hand and started recording values. After 38 samplings, we stopped collecting data and the dataset was correctly uploaded to the database. As the speed of the movement increased, we expected a reduction in the number of timesteps required to complete the movement and also an increase of the acceleration vector. Using a Python script, we plotted the values of the accelerometer, gyroscope and magnetometer (Fig. 3).

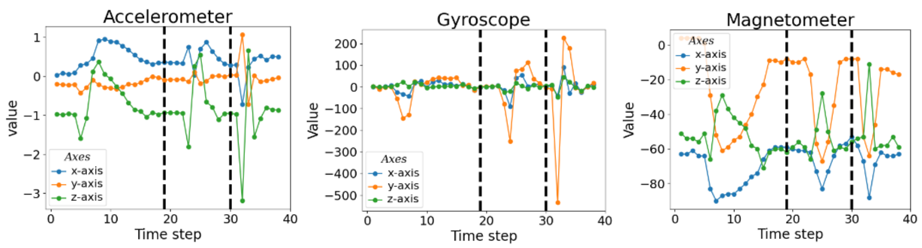

Fig. 3. Accelerometer, Gyroscope and Magnetometer plot of the movement replicated three times with incremental speed.

As expected, the plots show a movement reproduced three times at incremental speed, with a progressive decrease in the number of steps required to complete each action and an increase in the intensity of the acceleration vector.

Given the target of making the supported workflow flexible and generic for different types of tracking, we defined a set of configurable parameters (e.g., to increase the amount of data collected per time unit or to increase the sensitivity of the sensors), as summarized in Table 1.

Table 1. Configurable parameters.

Variable	Default	Range	Sensibility
Sample Period	250 ms	>5 ms	
Accelerometer	±4 g, ±0.122 mg	±2,±8,±16 g	0.061,0.244,0.732 mg
Gyroscope	±2000 dps,±70 mdps	±245,±500 dps	±8.75, ±17.50 mdps
Magnetometer	±4 gauss,0.14 mgauss	±8,±12,±16 gauss	0.29,0.43,0.58 mgauss

4 Conclusion and future works

Data collection and management have become an essential part in the development of ML models based on supervised learning. Therefore, it is crucial to obtain large amounts of data easily, quickly and accurately. One of the main challenges lies in building a workflow that starts with the edge device until it interfaces with a “smart” database that adapts to the type of data being sent, ensuring also a secure connection.

The proposed workflow offers a fully accessible, low-cost and user-friendly infrastructure for the collection and management of data from embedded systems. This approach can be applied to various applications, including classification of human physical activities, providing high quality data for training advanced machine learning models.

As future work, it is planned to integrate the study of ML techniques from the collected data, in order to generate models that can be imported into the same Arduino used previously, obtaining a complete pipeline, from edge to cloud and back. This will increase the autonomy and efficiency of the system, enabling a rapid classification of the detected actions.

References

1. Ali, I., Ahmedy, I., Gani, A., Munir, M.U., Anisi, M.H.: Data Collection in Studies on Internet of Things (IoT), Wireless Sensor Networks (WSNs), and Sensor Cloud (SC): Similarities and Differences. IEEE Access. 10, 33909–33931 (2022). https://doi.org/10.1109/ACCESS.2022.3161929.
2. Berta, R., Mazzara, A., Bellotti, F., De Gloria, A., Lazzaroni, L.: Edgine, A Runtime System for IoT Edge Applications. In: Saponara, S. and De Gloria, A. (eds.) Applications in

Electronics Pervading Industry, Environment and Society. pp. 261–266. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-66729-0_31.
3. Berta, R., Bellotti, F., De Gloria, A., Lazzaroni, L.: Assessing Versatility of a Generic End-to-End Platform for IoT Ecosystem Applications. Sensors. 22, (2022). https://doi.org/10.3390/s22030713.
4. Kos, A., Umek, A.: Wearable Sensor Devices for Prevention and Rehabilitation in Healthcare: Swimming Exercise With Real-Time Therapist Feedback. IEEE Internet of Things Journal. 6, 1331–1341 (2019). https://doi.org/10.1109/JIOT.2018.2850664.
5. Alemayoh, T.T., Lee, J.H., Okamoto, S.: A New Motion Data Structuring for Human Activity Recognition Using Convolutional Neural Network. In: 2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob). pp. 187–192 (2020). https://doi.org/10.1109/BioRob49111.2020.9224310.
6. Berta, R., Kobeissi, A., Bellotti, F., De Gloria, A.: Atmosphere, an Open Source Measurement-Oriented Data Framework for IoT. IEEE Transactions on Industrial Informatics. 17, 1927–1936 (2021). https://doi.org/10.1109/TII.2020.2994414.
7. MongoDB: The Developer Data Platform, https://www.mongodb.com, last accessed 2023/07/17.
8. Flutter - Build apps for any screen, //flutter.dev/, last accessed 2023/07/17.
9. Measurify, https://github.com/measurify, last accessed 2023/07/18.
10. Fresta, M., Bellotti, F., Capello, A., Cossu, M., Lazzaroni, L., De Gloria, A., Berta, R.: Efficient Uploading of.Csv Datasets into a Non-Relational Database Management System. In: Berta, R. and De Gloria, A. (eds.) Applications in Electronics Pervading Industry, Environment and Society. pp. 9–15. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-30333-3_2.