

Research Article

Exploiting Offloading in IoT-Based Microfog: Experiments with Face Recognition and Fall Detection

José R. Torres Neto ^{1,2}, Geraldo P. Rocha Filho,³ Leandro Y. Mano ¹,
Leandro A. Villas ⁴, and Jó Ueyama ¹

¹*Institute of Mathematical and Computer Sciences, University of São Paulo, Brazil*

²*School of Electrical Engineering and Computer Science, University of Ottawa, Canada*

³*Department of Computer Science, University of Brasília, Distrito Federal, Brazil*

⁴*Institute of Computing, University of Campinas, Brazil*

Correspondence should be addressed to José R. Torres Neto; jrtorresneto@usp.br

Received 13 November 2018; Revised 28 March 2019; Accepted 9 April 2019; Published 5 May 2019

Academic Editor: Antonio Guerrieri

Copyright © 2019 José R. Torres Neto et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The growth in many countries of the population in need of healthcare and with reduced mobility in many countries shows the demand for the development of assistive technologies to cater for this public, especially when they require home treatment after being discharged from the hospital. To this end, interactive applications on mobile devices are often integrated into intelligent environments. Such environments usually have limited resources, which are not capable of processing great volumes of data and can expend much energy due to devices being in communication to a cloud. Some approaches have tried to minimize these problems by using fog microdatacenter networks to provide high computational capabilities. However, full outsourcing of the data analysis to a microfog can generate a reduced level of accuracy and adaptability. In this work, we propose a healthcare system that uses data offloading to increase performance in an IoT-based microfog, providing resources and improving health monitoring. The main challenge of the proposed system is to provide high data processing with low latency in an environment with limited resources. Therefore, the main contribution of this work is to design an offloading algorithm to ensure resource provision in a microfog and synchronize the complexity of data processing through a healthcare environment architecture. We validated and evaluated the system using two interactive applications of individualized monitoring: (1) recognition of people using images and (2) fall detection using the combination of sensors (accelerometer and gyroscope) on a smartwatch and smartphone. Our system improves by 54% and 15% on the processing time of the user recognition and Fall Decision applications, respectively. In addition, it showed promising results, notably (a) high accuracy in identifying individuals, as well as detecting their mobility; and (b) efficiency when implemented in devices with scarce resources.

1. Introduction

In recent years, the number of people with healthcare needs, such as the elderly, disabled, and patients with reduced mobility, has increased considerably in the world [1]. This increase has created an upward trend in research involving these people, especially when they are discharged from hospitals and need to be cared for at home. Transitioning back home after hospital discharge is a vulnerable time for patients. In this context, Health Smart Homes (HSH), also known as home care environments, have emerged as a promising

option to improve the quality of life of people treated at home. Typically, treating people with reduced mobility within the context of HSH uses computational intelligence to monitor them while they are recovering in their homes [2]. Computational intelligence applied in healthcare monitoring is feasible and extremely important, especially in countries where the number of persons with reduced mobility is high [2, 3].

Smart city applications are increasingly being used in the everyday lives of the population through the Internet of Things (IoT), as well as in the health smart home context.

In the IoT, any physical or virtual sensor is considered a “thing” that can be connected to the Internet [4, 5]. Such “things” can be sensors or actuators that will activate when a critical situation is detected. Thus, data can be collected and transmitted to analyze and represent individuals’ information remotely [4, 6]. Therefore, interactive applications on mobile devices can be used for individualized sensing and monitoring using cameras and other mobile device sensors, such as specific gadgets [7]. For instance, these applications can issue warnings to teams of health professionals or family members whenever they detect some abnormality [1, 8].

In this context, studies have aggregated information from multiple sources and distributed it to consumers who do not have direct connections with the information producers, such as healthcare systems that monitor the data of a patient undergoing home treatment and share them with a hospital system [1, 9]. These systems can analyze health problems or cognitive disorders [10], monitoring older adults in their homes [1, 11], recognizing human activity [12, 13], or interacting with people [8]. Thus, HSH systems based on fog computing could help diagnose diseases and influence individuals’ social interaction, as well as intervening with some daily tasks or making decisions for the user (e.g., suggesting a film genre to relieve stress according to their current state) [3]. Fog computing is a layer of computing power between cloud and devices [14] capable of dealing with contextualized data points [15]. Fog computing enables sharing and management data at the network edge. In this way, decentralized smart processing is closer to the data source. For instance, some Raspberry Pis connected can monitor a smart home, acting as multiple Sink nodes or fog nodes. A fog node is deployed at the edge of the network and is capable of handling context awareness [16]. In this work, we designed a microfog, which is a fog node responsible exclusively for providing individualized monitoring. A microfog can be a subset of fog devices (e.g., two Raspberry Pis) enabled by fog cluster to handle data from a sensor linked to a specific monitoring application.

An infrastructure of microfog for HSH offers encouraging possibilities in edge resource provision since allowing distributed services and resources provides distinct environments for many users [17]. However, a microfog integrated into HSH has limited resources. The full outsourcing of data analysis to a microfog can, consequently, give rise to a reduced level of accuracy and adaptability [18]. However, the device’s communication is more costly than local computing, since such sensors can consume a great deal of energy if their communication is not optimized [6, 19]. Furthermore, multiple applications competing for limited resources result in higher processing latency and more transmissions to the cloud. They also reduce the performance of networking services and increase devices energy consumption [18, 19]. Managing such resources is not a trivial task. One big challenge in HSH systems, then, is how we can achieve high data processing with low latency in an environment with limited resources.

To overcome these above-mentioned limitations, this work proposes MOOSE (microfog offloading system), a

healthcare system based on data offloading in an IoT-based microfog and applications that use computational intelligence for individualized monitoring of users with reduced mobility. The proper microfog infrastructure for an autonomous healthcare system might mitigate the undesired effects of health monitoring and consequently improve resource provision efficiently. We define microfog as a subset of the fog computing environment responsible for the individualized monitoring of the user. With a microfog, it is possible to reduce the number of transmissions involving raw data; hence, only metadata are sent. Also, microfog ensures data volume and latency reduction, as well as the saturation of the wireless network [6, 17, 20]. In turn, data offloading helps mitigate the effects of the local decision and improve the architectural limitations of devices, taking advantage of a communication interface for the transmission of data to the cloud. However, some of its drawbacks are accentuated in the communication between fog layer and cloud; for instance, the processing time of health critical IoT-based applications is limited by the offloading of data to the cloud [18, 19]. Thus, the performance of remote monitoring can be severely reduced if this scenario is not carefully considered.

Additionally, MOOSE can be portable and independent of the environment infrastructure. MOOSE’s modularized architecture allows the simplified deployment of applications that use computational intelligence. In this context, this work discusses two individualized monitoring applications that were used as a case study: (1) user recognition, an application based on images, and (2) fall detection, an application based on motion sensors. Our work is mainly focused on distributing resources in a healthcare IoT-based environment and on improving the number of transmissions between smart applications and cloud. The offloading decision improved the individualized health monitoring and resource provision, increasing the performance of the infrastructure in smart environments. The results show decrease up to 54% and 15%, respectively, in the processing time of the user recognition and Fall Decision applications with five devices in MOOSE microfog, as compared to an environment with only a cloud.

The remainder of this work is organized as follows. Section 2 discusses the related works found in the state of the art. The background of smart individualized monitoring is given in Section 3. Section 4 describes the MOOSE system. Section 5 shows the performance evaluation of the proposed system. Finally, Section 6 presents final considerations and future research directions for this work.

2. Related Work

This section presents the advances achieved in HSHs, highlighting the main challenges of this research. IoT-based HSH has been extensively studied in the literature. In particular, advances in in-home healthcare in recent years are due to smart environment applications, as well as artificial intelligence [21–23]. Therefore, the combination of modern detection equipment, advanced data-processing techniques, and wireless networks results in the creation of digital environments that improve residents’ daily lives [24, 25].

For example, applications to assist people in a healthcare environment have been developed in order to improve the quality of their lives. Intel Corporation has developed an ultrasound system [26] that identifies falls and irregular movements by users as part of their IoT system. These approaches are not generic and should be used for specific purposes. Similarly, [27] provides a web-based IoT middleware platform to connect physicians and patients using coupled body sensors and actuators, called EcoHealth (Ecosystem of Healthcare Devices). The solution is divided into modules comprising device connection, data manipulation, actuation, visualization, management, storage, and other services. The focus of the authors is on monitoring people under observation; they suggest coupled breathing sensors in an oxygen mask, a temperature sensor on the patient's arm, and an electrocardiogram sensor. Thus, a physician can remotely evaluate the patient in real time. However, considering the scenario of an HSH, the above solutions are considered intrusive, and disregarding the quality of life and the day-to-day independence of the user. Furthermore, the authors do not take device competition, which can occur in a bottleneck in a wireless network, into account. Also, there has been little written about smart monitoring in smart homes using microfog architectures and data offloading on such architecture for a resource providing individualized monitoring.

Some other works are more focused on the specific IoT infrastructure, not only software but also considering networks, hardware, and middleware. For example, authors in [28] present a novel, user-centered methodology to develop intelligent environments. The authors focus on integrating technologies and create a software infrastructure to provide services. The challenge addressed by the authors is related to confidence in systems in the real world, such as performance quality, behavioral correctness, and their validation. Also, the authors focus on stakeholders involved in the building systems in intelligent environments. The authors in [29] describe an intelligent approach to the analysis of IoT applications on focused on the automation of the transitions between edge and cloud in dynamic situations. The authors describe an architecture divided into three layers, composed of different components and actors that assume roles in adaptive IoT data analytics solutions. However, any evaluation is made by the authors. In addition, none of the above authors present an evaluation to attest to how their solutions impact the infrastructure of intelligent environments.

Distributed processing in an infrastructure based on the architectural limitations of the environment devices has not been adequately addressed. The authors in [30] describe an IoT platform to ensure reliable data dissemination and analysis in the communication in rural and remote areas, where the wireless infrastructure is sparse. Thus, the authors handle with the connectivity problem by using long-range multihops between sensors and actuators. The solution proposed by the authors disseminates data both at the edge and in the cloud. It is worth highlighting that the balance of computing is essential, given that a high volume of data at the edge can overload devices, which make the system less scalable. The authors in [31] propose a self-configurable gateway capable

of performing detection and configuration of IoT devices in real time. The authors explore a three-tier architecture for IoT applications, based on different computational capabilities scenarios. Their focus is on the dynamic discovery of IoT devices and the management of their connections. However, their configurations are limited to hardware, which can make their solution dependent on specific devices or applications. Moreover, the flexibility of data volume was not addressed.

In this paper, we go further and exploit microfog in a three-layer environment to provide higher-level services. Our work provides the following additional contributions: (1) an IoT-based healthcare system that forms an intermediate layer of intelligence between applications and devices; (2) a microfog architecture capable of facing challenges in healthcare systems, such as flexibility, energy efficiency, scalability, and reliability; (3) an increase in the processing performance of individualized monitoring, distributing data through the environment; and (4) a reduction in the number of transmission and latency even when the volume of data is high, improving the energy consumption of the devices.

3. Smart Individualized Monitoring

Because of the growth in users of computer systems and access to such technologies, it is possible to capture and analyze data from individuals in different contexts, especially in the residential and personalized context. Therefore, the identification of the individual and the detection of physical behavior, for example, a fall, can help systems monitor the health of people with disabilities, those with reduced mobility, and the elderly. They can also meet the needs and assist in the daily life of each user [23, 32].

This work considered the interactive applications user recognition and fall detection to validate the proposed framework. These applications allow individualized and non-intrusive monitoring. Figure 1 shows how we performed individualized monitoring and then carries out the task in a nonintrusive manner. To this end, we followed two steps: (1) application monitoring that consists of collecting data in an individualized way, as presented in Sections 3.1 and 3.2 and (2) a learning module that consists of acquiring knowledge to detect the application tasks. This monitoring opens space to investigate methods for helping users, such as an application to call an ambulance when an older adult falls.

We relied on computational intelligence for user recognition and fall detection. In this case, we used a classifier based on the machine learning algorithm k-nearest neighbors (KNN), using Euclidean distance. KNN is a supervised algorithm, which learns to perform a task (in this research, recognition and detection) according to a specialized dataset. We chose KNN because it presented useful results in the detection and classification of falls, especially with the use of wearable sensor-based fall detection [32, 33]. In the context of face recognition, there are more robust recognition approaches presenting good results in face classification [8, 23, 34]. However, the choice to use KNN in our model is justified by self-learning based on historical examples only and by adaptability due to not requiring, at each update, changes in network topology.

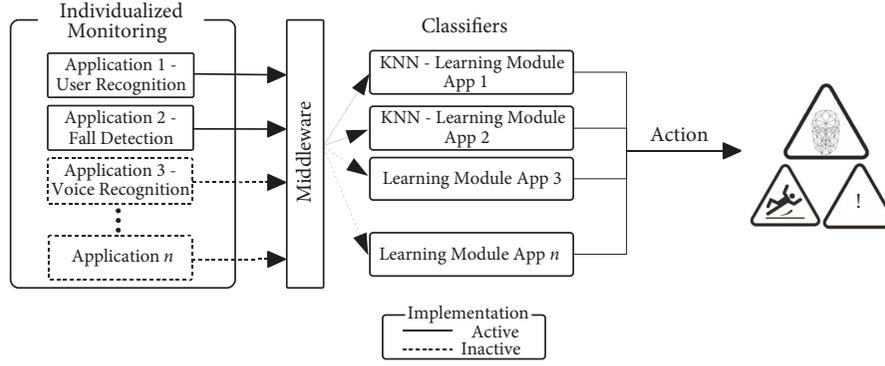


FIGURE 1: Individualized monitoring.

3.1. User Recognition Application. In HSH systems, images are transmitted by devices and used by recognition applications to identify a particular individual. The face is analyzed through the image, which determines the identity of each user. In this work, we use the face mapping proposed in [8]. This mapping is divided into stages of face acquisition, facial feature extraction, and identification based on automatic learning.

Following [8], the face mapping process is performed by considering 33 points obtained from facial feature extraction. Points are specific parts of the face that, when joined, uniquely identify an individual. The points are distributed as follows: (a) eight to map the mouth; (b) six for each eye; (c) three for each eyebrow; (d) three for the chin; (e) two for the nostrils; and (f) two to mark the lateral ends of the face near the eyes. Also, distances and angles are obtained for all possible combinations of points, to show the line connecting two distinct points with the horizontal axis. This creates a dimensionality representation of $D_1 = 2 * 33 + 8 + 2 * 528 = 1130$ attributes. After obtaining the dimensionality of the face, we used such characteristic points of each user to form a reference method, based on kNN, for the learning and identification of the user.

3.2. Fall Detection Application. A fall detection application monitors the movements of users in smart environments as a means of monitoring their routine at a distance and detecting unusual activity, such as falls. MOOSE uses fall detection based on [32, 33], but we used a smartwatch to collect accelerometer and gyroscope data. The use of a smartwatch presents an additional challenge in collecting data, namely, false positives. To remedy the nontrivial task mentioned, we used the smartwatch synchronized with a smartphone to collect the data. Thus, our system collects and correlates data from the following devices: (1) an accelerometer to determine the displacement of a moving body and (2) a gyroscope to determine the rotation and change of direction of a moving body. In other words, this application simultaneously combines two different devices for monitoring the user's movements, such as a smartphone and a smartwatch. Thus, their coordinates are crossed for better accuracy. This approach aims to reduce false positives when identifying falls, combining accelerometer and gyroscope data from

both devices. In summary, the application verifies if there is movement in both devices then checks whether a device is in contact with the user and whether it uses data to identify falls. Thus, medical care/assistance can be notified with greater precision, so as to act as soon as possible.

4. MOOSE: Microfog Offloading System

This section presents a healthcare system that uses data offloading to increase performance in an IoT-based microfog, providing resources and improving health monitoring. Our system is called MOOSE (microfog offloading system) and was based on INCA [35].

Our proposal, shown in Figure 2, is modeled for healthcare applications based on IoT systems in order to provide smart monitoring of individuals with reduced mobility, such as the elderly. The proposed system consists of resources providing increased performance for data processing and message transmission in a fog environment using data offloading. The proposed system was based on an architecture of three layers (see Figure 2): (1) Tier 1, (2) Tier 2, and (3) Tier 3. In the first tier, sensing is carried out by devices, which then collect and transmit data. In the second tier, a lightweight analysis based on images or data coordinates is conducted by a microfog, an environment where devices are used to carry out data processing, putting computing resources (e.g., processing, memory, and data) closest to the end user [6]. In the third tier, an in-depth analysis based on images or data coordinates is completed in a cloud, which has higher processing power and storage capacity.

The next subsection describes the healthcare system based on data offloading. The proposed system can be modeled in two main parts:

- (1) Data traffic management: reliable transmission of data between producers (publishers) and consumers (subscribers) of data.
- (2) Data offloading engine: satisfactory computation performance of data processing in the fog cluster or cloud.

4.1. Data Traffic Management. This section describes the transmission of data through MOOSE's architecture. The

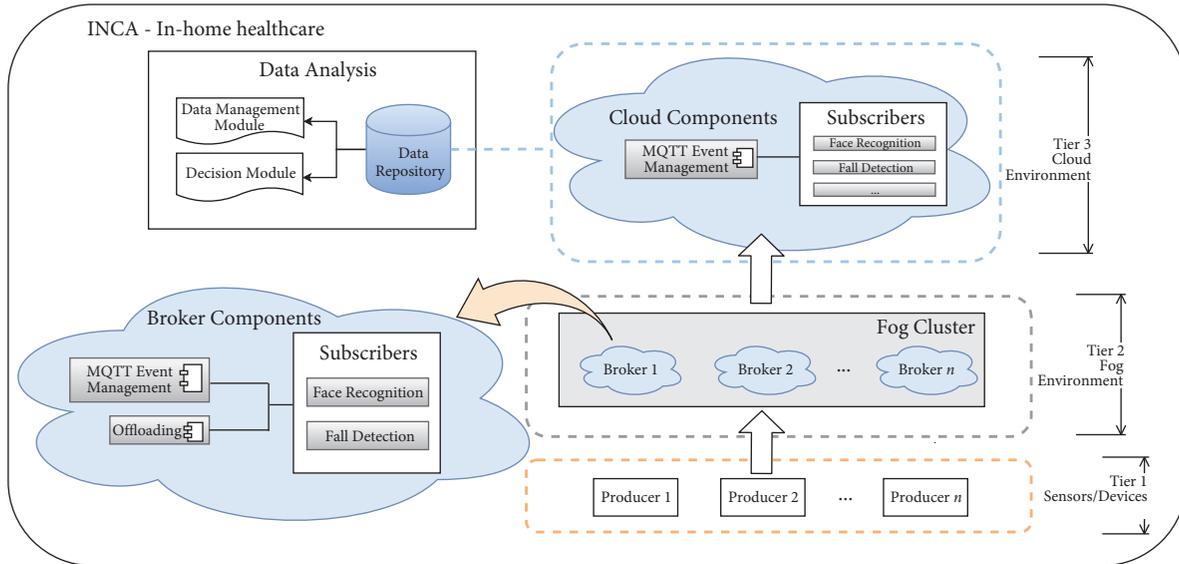


FIGURE 2: MOOSE's system architecture for individualized monitoring.

producers of information are capable of collecting and transmitting data to the fog cluster, such as an image and accelerometer and gyroscope coordinates. This study considers the Message Queue Telemetry Transport (MQTT) [36] protocol for data transmissions, as the components have loosely coupled characteristics. MQTT is one of the most common protocols used for IoT device-to-device communications and a candidate to become the standard IoT protocol because of its effectiveness and lightweight nature [37].

We used a temporal correlation protocol [38] to correlate the generated events to time. This protocol acts before the MQTT Event Management publishes the data collected by sensors, in order to prevent redundant notifications. Thus, redundant data are eliminated and new data are published in blocks. Furthermore, this increases network life, reducing the amount of data without losing quality and reducing the number of transmissions.

MQTT Event Management is responsible for managing the publisher and subscriber topics in Tiers 2 and 3 of the MOOSE's architecture. In Tier 2, the MQTT Event Management instance contains only topics related to active applications (i.e., monitoring applications currently connected to the system). In Tier 3, the MQTT Event Management instance manages all publisher and subscriber topics, since it covers all registered applications [39]. Topic management is supported by the Data Repository in Tier 3, in which data analysis is performed. Thus, the system has knowledge of the whole monitoring environment, as the cloud stores all data collected by the sensors of devices present in the users' everyday lives (e.g., smartwatch, smartphone, or cameras). The data analysis is divided into two modules:

- (i) *Data Management Module*: module responsible for managing publishers and subscribers topics using the MQTT Event Management [9].

- (ii) *Decision Module*: module that comprises Artificial Intelligence algorithms responsible for the decision process of each monitoring application.

4.2. Data Offloading Engine. This subsection describes how the design of our proposal enables the fog cluster to decide whether it transmits data to the cloud.

4.2.1. Fog Cluster. In a healthcare environment, the fog node addresses the processing time of critical IoT applications, which may be limited by the network delay in transmitting data to the cloud [19].

The proposed fog cluster consists of brokers formed by connected heterogeneous devices (e.g., embedded devices, routers, switches, set-top boxes, proxy server, and base stations). The brokers are composed of subscribers, which are the back-end of monitoring applications instantiated by the MQTT Event Management. In our proposed fog cluster, we use two monitoring applications as subscribers, namely, user recognition and fall detection, as described in Section 3.

To support our study, we created a microfog using some connected Raspberry Pi 2, fog node that form a virtualized cloud computing environment. The Raspberries were configured in a distributed manner embedded with wireless communication capabilities. Moreover, the microfog had computing capabilities able to run the monitoring applications with different scenarios, evaluated in Section 5.

4.2.2. Offloading Decision. In our fog cluster, input data processing is decided based on the impact of the application on the fog and cloud environments. In Algorithm 1, we present the pseudocode of the proposed offloading decision.

Algorithm 1 receives the block of data formed by MQTT Event Management as input. In the first execution, the applications are run in both environments, the fog cluster and the

```

Input: a new block of data  $BD$  composed with images or coordinates
1 while  $DB_i \in \{DB_1, \dots, DB_n\}$  do
2   if  $firstExecution()$  then
3      $C_c \leftarrow sendToCloud(DB_1)$ ;
4      $C_f \leftarrow processingLocal(DB_1)$ ;
5      $storeCloudCost(C_c)$ ;
6      $storeLocalCost(C_f)$ ;
7   else
8      $eC_c \leftarrow estimateCloudProcessing()$ ;
9      $eC_f \leftarrow estimateLocalProcessing()$ ;
10  if  $eC_f \leq eC_c$  then
11     $C_f \leftarrow processingLocal(DB_i)$ ;
12     $storeLocalCost(C_f)$ ;
13  else
14     $C_c \leftarrow sendToCloud(DB_i)$ ;
15     $storeCloudCost(C_c)$ ;
16  end
17 end
18 end

```

ALGORITHM 1: Algorithm for offloading decision.

cloud. The costs are obtained through energy consumption by the application's execution. Equation (1) gives the cost of the application execution in the fog cluster:

$$C_f = \int_1^x V_t I_t d_t \quad (1)$$

where V_t is the voltage and I_t is the chain read during a time interval d_t . The cost of the application execution in the cloud must consider the Round-Trip Time (RTT) in transmissions from fog to cloud. Thus, the equation used to estimate the cost of the application execution in the cloud is as follows:

$$C_c = \int_1^x V_t I_t d_t + C \quad (2)$$

where C is a constant that represents $(T_x + R_x)$, where T_x is the transmission cost and R_x is the receiving cost. Both T_x and R_x are defined by as follows [40]:

$$T_x = R_x = V_x * I_x * C_x \quad (3)$$

where C_x is the transmission rate of the network interface, V_x is the device voltage, and I_x is the electric current. It is worth pointing out that I_x has different values for transmitting and receiving data because the signal strengths differ. For instance, the Micaz architecture takes 17.4 mA to transmit a packet and 18.8 mA to receive. This means that the levels of energy consumption required to transmit and receive a packet are 0.2088 μ J/bits and 0.2256 μ J/bits respectively [40].

The previous costs are saved in a database in order to provide information for the next offloading decision. Thus, our algorithm uses a simple linear speedup to evaluate the execution time for each environment, shown in

$$S_{latency} = \frac{T}{W} \quad (4)$$

where T is the execution time of the task and W is the execution workload of the task. Thus, our algorithm chooses the lowest execution time. The workload increases according to the number of data blocks currently being processed in the environment. Thus, our proposal balances processing in both environments, as the execution time may vary depending on the workload.

Figure 3 shows a flowchart of procedures performed in our proposed framework to increase performance by data offloading in IoT healthcare systems. The offloading decision increases performance in data processing and message transmission. The procedures are divided into three stages: (1) data collection, (2) fog cluster, and (3) cCloud. The arrow indicates the direction in which the procedures flow. As soon as the data are collected, they are sent to the fog cluster. The broker receives these data and compares them to previous data. If the data are new, a block of data is created and published in the Event Manager. Our algorithm estimates the fog cluster and the cloud cost for executing the application, and decides whether to send the data to the cloud or process them locally.

5. Performance Evaluation

This section presents the methodology and results of the performance evaluation carried out using experiments with face recognition and fall detection. The validation of MOOSE was divided into two stages. In the first stage, we evaluated the applications of individualized monitoring. In the second stage, we evaluated the MOOSE's communication infrastructure, comparing it with a baseline that uses a cloud server environment.

In the baseline, processing is performed directly on the remote server, without going through an intermediary environment that could process the applications. It was possible to notice that the performance of the system improved after

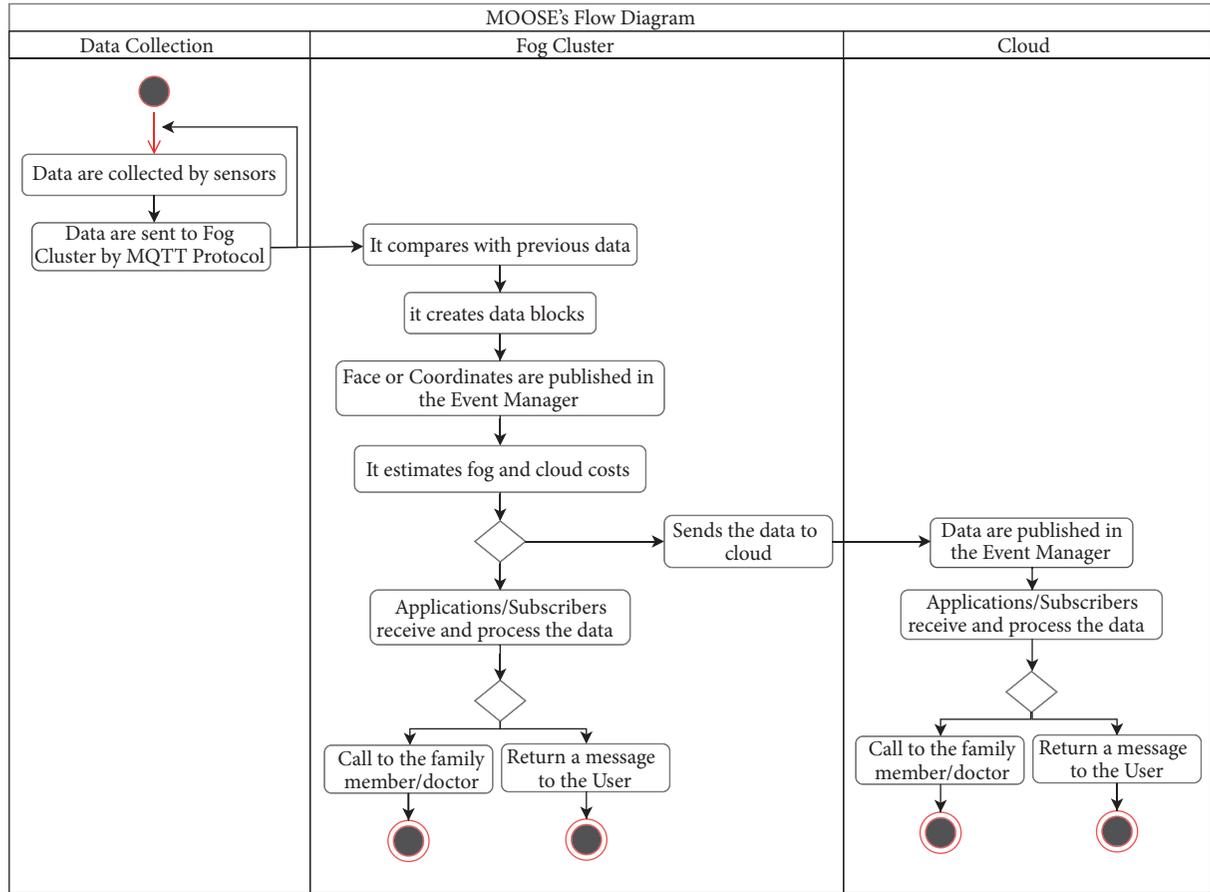


FIGURE 3: MOOSE's flow diagram into three layers.

the addition of a microfog as an intermediate layer, since the resources were distributed to the whole infrastructure. All experiments were performed in a real environment. Thus, Raspberry Pi 2 was used as microfog (i.e., a network edge node) and a desktop as a remote server. The measurement and parameters selected for each scenario are presented in the following, as well as the results obtained.

5.1. Stage 1: Impact of the Accuracy of Deployed Applications.

This work used facial expression images from the free access Cohn-Kanade (CK+) database [41], which consists of 593 facial expressions from 123 adult actors (69% female and 31% male). However, only one CK+ database subset was used for the experiments. There are many nearly identical images, due to the CK+ dataset consisting of a sequence of images obtained from videos, and the CK+ subset was thus used, removing very similar images to help create a balanced subset of relevant images that can improve the generalization performance of the investigated machine learning classifiers. This subset comprises only frontal images of 30 different individuals with 100 images each, making for a total of 3000 pictures for the experiments, both to train the algorithm and for the tests performed to identify the user.

The fall detection experiments were based on literature studies [42, 43]. Thus, accelerometer and gyroscope data were collected from three different individuals using a smartwatch

(Moto 360) and smartphone (Moto X2). This strategy of using real data was adopted due to the need to ensure that both accelerometer and gyroscope data were acquired at the same time. The collected data correspond to 15 minutes of each user performing each activity. Sensor data were collected at time intervals of twenty milliseconds. Thus, at the end of the experiment, there was a set of 135,000 data for each activity, for a total of 810,000 data. The methodology adopted for the experiments was to perform the simple average of every twenty-five data lines, equivalent to the mean of the accelerometer and gyroscope data in a time interval of one second.

For both experiments (i.e., user recognition and fall detection), the performance of the algorithms was analyzed separately using k -fold cross-validation [44]. This is probably the most popular technique for evaluating the generalization capacity of a model from a set of data. It consists of dividing the total set of data into k independent subsets of the same size, with one subset used to test and the remaining $k - 1$ to train. This process is performed k times, alternating the test subset so that all combinations are satisfied, which provides us with a more accurate estimate. In our experiments, we used $k = 10$ in the k -fold cross-validation technique.

On the other hand, the k NN algorithm calculates the Euclidian distance of all elements of the dataset to the new input data and then decides which class is the closest to the

TABLE 1: Experiments parameters values.

Experiments	Parameters for memory and processor consumption and processing time									
	Tier 2					Tier 3				
	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
Number of data blocks	2*1	2*1	2*2	2*4	2*8	2*16	2*32	2*64	2*128	22*56

input data through a vote of the k nearest elements. We used the kNN algorithm with different configurations for the user recognition model and fall detection model tests, such as $k = 1, 3, 5, 7,$ and 9 (Figure 4). Figure 4 shows the results obtained by the kNN algorithm as a function of the user recognition accuracy and fall detection models. In both applications, the classifier reaches the highest accuracy rate when $k = 1$ (i.e., when only one element votes on the classification of the input data): 99.65% for user recognition and 95.59% for fall detection.

5.2. Stage 2: Impact of Applications on Infrastructure. The interactive monitoring applications described in Section 3 were used to evaluate the MOOSE's infrastructure. The applications were run in different environments, defined as tiers in MOOSE's architecture: (1) Tier 2 (microfog, Raspberry Pi) and (2) Tier 3 (remote server). Thus, it could be inferred at what tier of the architecture the application processing is valid from the situation and the number of input data. Image data are collected using a smartphone, the accelerometer and gyroscope data using a smartphone and a smartwatch. The interfaces collected these data and sent them to the fog cluster, in which they were compared to previous data and the new blocks were created. Afterward, the blocks were published in the Event Manager to serve as input for individualized monitoring applications. Table 1 shows the parameters of the experiments, where we considered an image file and a coordinate file as input. It is noteworthy that the coordinate file provided a 1-second measurement of the accelerometer and gyroscope sensor, producing a 1KB file, whereas each image was 32KB in size.

Assuming that the embedded devices support the applications developed in terms of memory and processing power consumption, evaluations were focused on metrics of the processing time of the applications in four scenarios (see Table 2). The results were run 33 times with 95% confidence using t-student.

In order to measure the impact of the applications on the infrastructure, we evaluated both execution environments based on the performance of each application. Figures 5 and 6 show the performance evaluations of Tiers 1 and 2 in relation to the processing time of the interactive applications. Figure 5 shows the time that the user recognition application took to run at each level, according to the number of data blocks taken as input. With 1 data block, the extraction of characteristics and classification in level 1 takes around five seconds, whereas in Tier 1, the time is one-fifth of this. However, as the number of data blocks increases, level 2 processing time approaches the processing time of 1 data block in Tier 1. Therefore, it can be observed that in a large

TABLE 2: Scenarios parameters values.

Parameters for the scenarios using data offloading	
Scenario 1	1 Remote Server (Tier 3)
Scenario 2	1 Remote Server (Tier 3) e 1 Embedded Device (Tier 2)
Scenario 3	1 Remote Server (Tier 3) e 3 Embedded Devices (Tier 2)
Scenario 4	1 Remote Server (Tier 3) e 5 Embedded Devices (Tier 2)

data burst, level 1 can be used for processing, making it less idle while level 2 processes a large amount of data.

Likewise, Figure 6 shows the processing time of the fall detection application. In this scenario, the embedded device takes 2.2 seconds to detect a fall with one data block. In this scenario, the response time of level 2 is seven times less but also reaches the response time of the embedded device at a given instant. Therefore, it is possible to use the data offloading approach with fog computing to parallelize the processing of a large amount of data.

Additionally, we evaluated the impact of the microfog layer on the energy consumption caused by applications at the moment of their execution. The metrics used in this evaluation were memory and processing power consumption, as well as processing time. Figures 7(a) and 7(b) show the memory consumption of each interactive monitoring application in both performance environments. We can observe that in the execution of the user recognition app in Tier 2 requires much more memory than in Tier 3. This is because the memory resources of a Raspberry are scarce. To circumvent this problem, we diversified the microfog with more than one Raspberry and distributed the processing between them, forming a virtualized environment. On the other hand, the fall detection app did not have as much impact on Tier 1 as coordinate data is lighter than image.

Figures 8(a) and 8(b) show the consumption of processing power. Despite the high processor and memory consumption in an embedded device compared to a server, results show that embedded devices support such applications. This opens space for investigating high-performance algorithms for these devices that require fewer device resources. Thus, the maximum processing is done at the network edges, which can save device energy because of the reduction of transmissions to the cloud.

Here, it is worth emphasizing the role of the fog cluster in computing the offloading decision before transmitting the data to the cloud. It is possible to increase the infrastructure performance by distributing the processing between Tiers 2 and 3 of the MOOSE's architecture, considering the scenarios

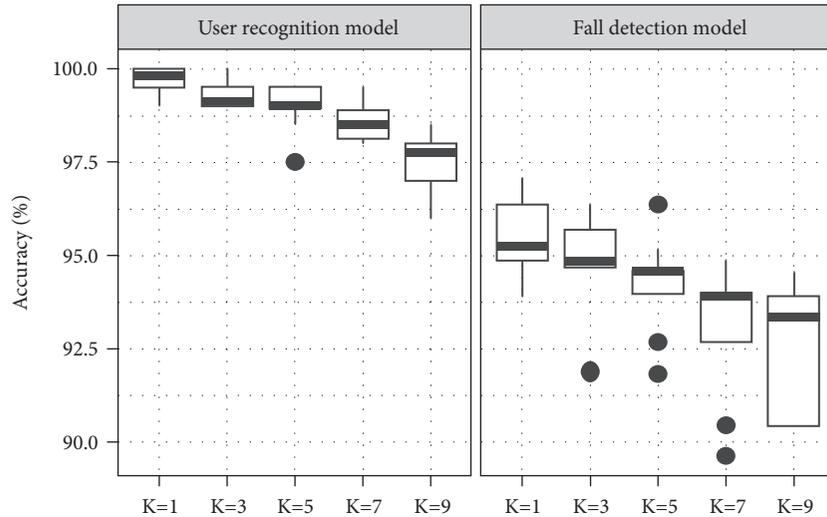


FIGURE 4: App accuracy percentage.

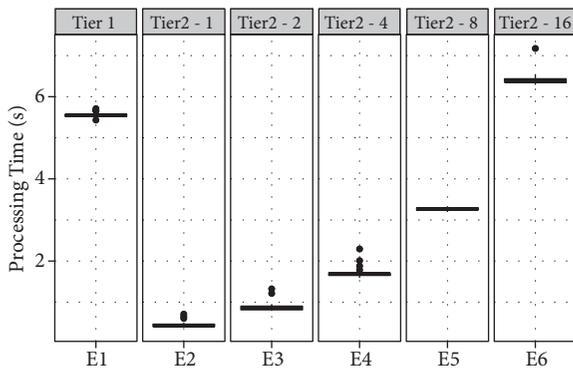


FIGURE 5: Processing time of the user recognition app.

presented in Table 2. Results show that processing time decreases for both interactive applications when we combine Tier 2 and Tier 3. When we increase Tier 2 with three devices and distribute the processing between Tier 2 and Tier 3, processing time decreases by 1.27 compared to the scenario that data are processed only in Tier 3. When we increase it with five devices, distributed processing is 1.54 shorter. In Figure 9(b), processing time using Tier 3 and Tier 2 with five embedded devices in microfog was shorter by up to 1.15 than using only the Tier 3. Therefore, microfog plays a crucial role in ensuring the efficiency of health monitoring systems in a resource-poor environment.

The behavior discussed is similar when the number of messages is evaluated as a metric, as in Figures 10(a) and 10(b). Results showed that the proposed mechanism was capable of increasing the efficiency and performance of data processing in the system. The mechanism ensured scalability, allocating data for processing in the whole infrastructure. Consequently, the number of messages sent by both interactive applications decreased when the offloading mechanism was applied. In particular, offloading has a greater benefit

for applications with higher processing power, such as user recognition (See Figure 10(a)). Therefore, MOOSE sees communication gains for applications with high (Figure 10(a)) and low (Figure 10(b)) processing power. Moreover, in more robust data input, such as images, data offloading, and fog computing had a greater impact on the performance of the proposed system.

6. Conclusion

This work presented a healthcare system, called MOOSE (microfog offloading system), to increase performance in an IoT-based microfog using data offloading. A data offloading algorithm was developed based on microfog architecture that considered the architectural limitations of its devices. The synchronization of data complexity through the environment achieved high data processing with low latency. Also, MOOSE had great advantages when the processing was distributed, since it reduced resource competition among the multiple applications, resulting in low latency and fewer transmissions in communication.

We used two interactive applications of individualized monitoring to validate and evaluate our system: (1) recognition of people using images and (2) fall detection using a combination of sensors (accelerometer and gyroscope) on a smartwatch and smartphone. Results have shown improvements of 54% and 15% in the processing time of the user recognition and Fall Decision applications, respectively. The designed system was able to provide resources and improve health monitoring, as well as being highly accurate in identifying individuals.

For future work, the evaluation of our approach will explore different devices and sensors in specific situations. Also, new techniques of data offloading will be designed to improve intelligence in the monitoring environment.

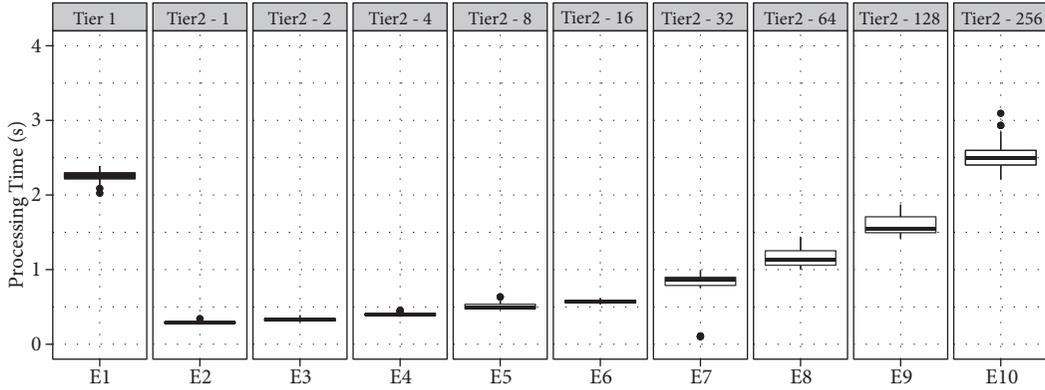


FIGURE 6: Processing time of the fall detection app.

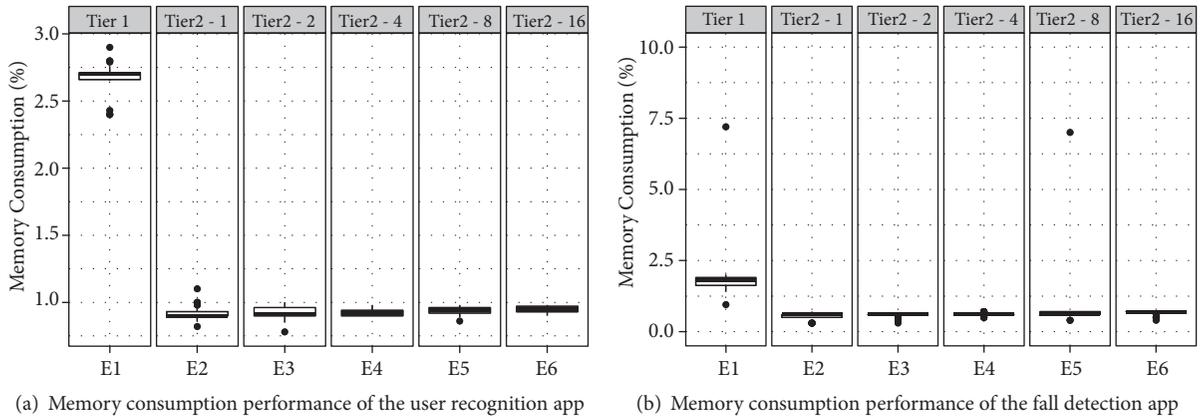


FIGURE 7: Performance evaluation of memory consumption of the applications.

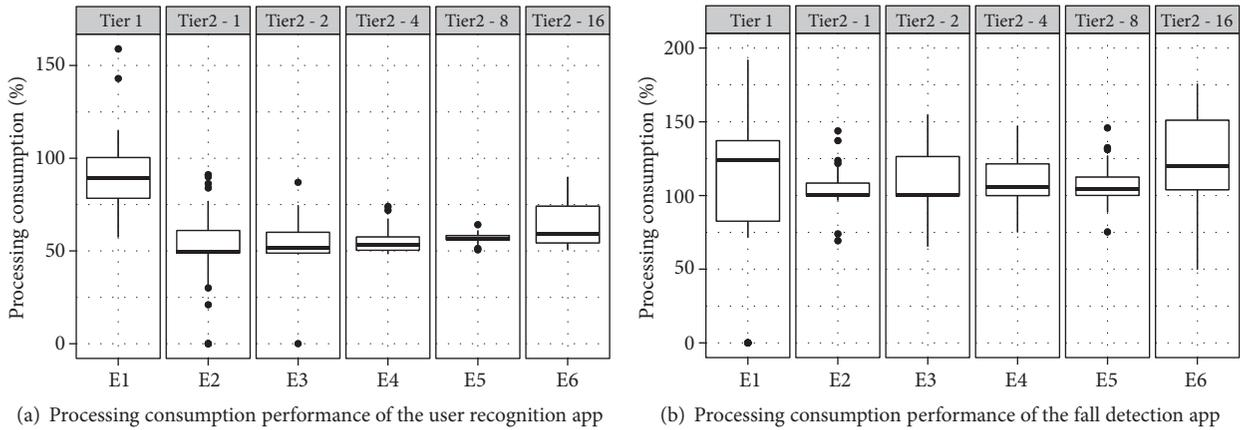


FIGURE 8: Performance evaluation of processing consumption of the applications.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the supporting organizations for funding this work: The Agency Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) and the grants #2016/25865-2 and #2017/23655-3 from Foundation for Research Support of the State of Sao Paulo-FAPESP.

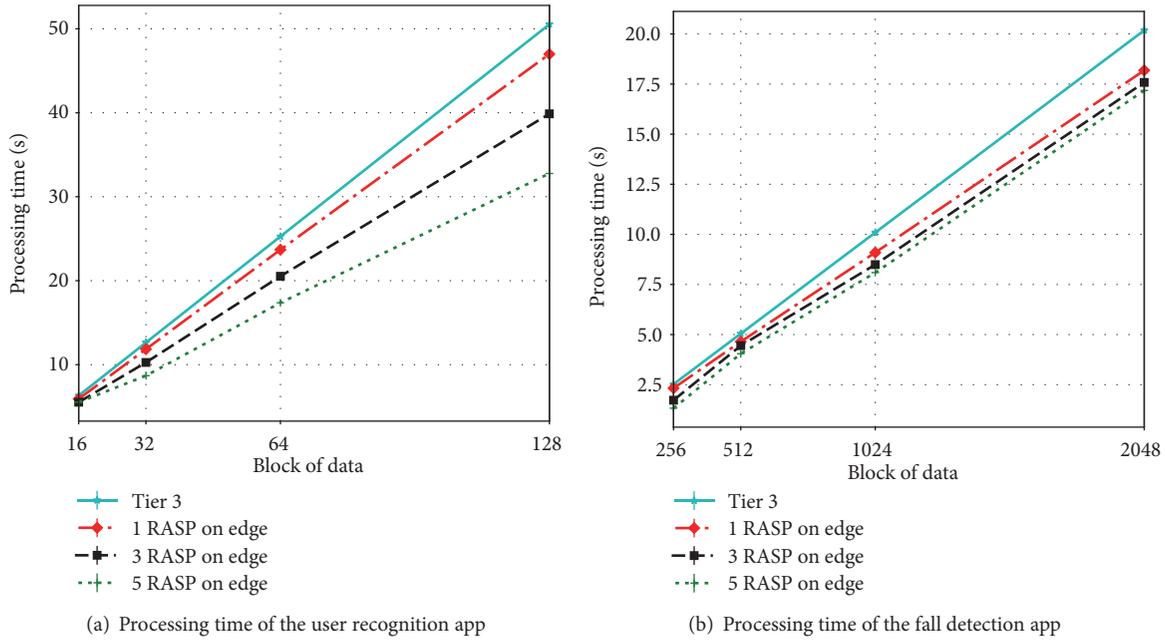


FIGURE 9: Performance evaluation of applications processing time using offloading.

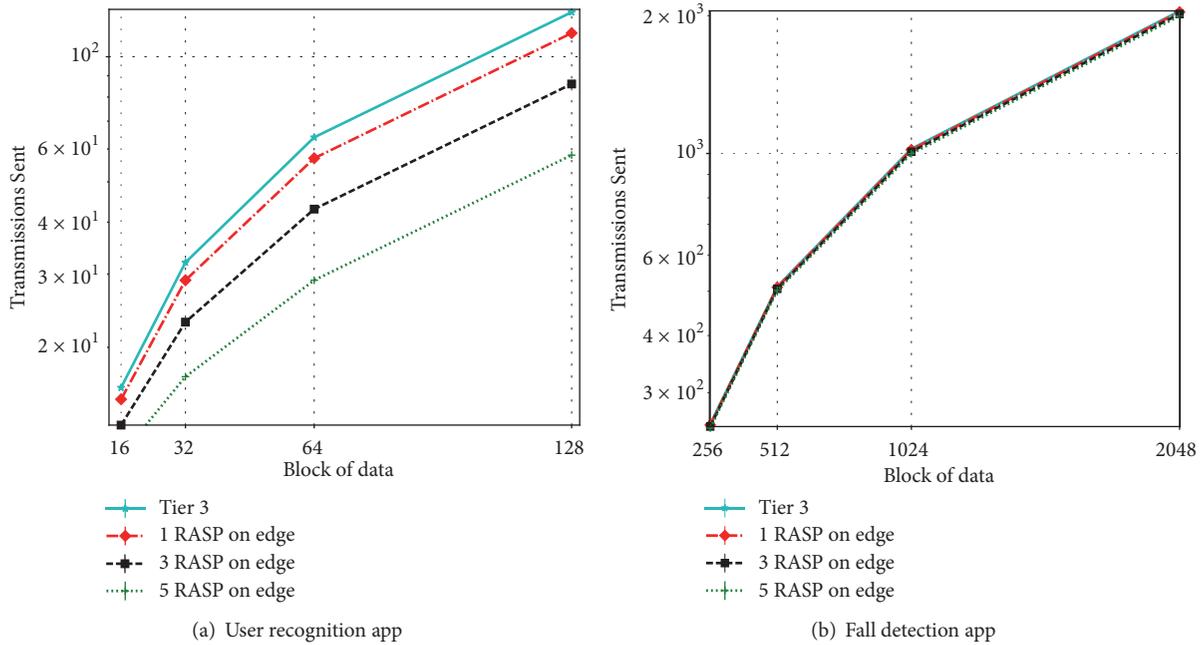


FIGURE 10: Total message transmitted.

References

- [1] V. P. Gonçalves, V. P. Neris, J. Ueyama, S. Seraphini, T. C. Dias, and G. R. Filho, "Senior citizens in interaction with mobile phones: a flexible middleware approach to support the diversity," in *Proceedings of the SERP*, vol. 1, pp. 56–62, 2013.
- [2] E. Romero, Á. Araujo, J. M. Moya et al., "Image processing based services for ambient assistant scenarios," in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, vol. 5518 of *Lecture Notes in Computer Science*, pp. 800–807, Springer, Berlin, Germany, 2009.
- [3] T. Taleb, D. Bottazzi, and N. Nasser, "A novel middleware solution to improve ubiquitous healthcare systems aided by affective information," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 335–349, 2010.
- [4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future

- directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: a survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [6] B. L. R. Stojkoska and K. V. Trivodaliev, “A review of Internet of Things for smart home: Challenges and solutions,” *Journal of Cleaner Production*, vol. 140, pp. 1454–1464, 2017.
- [7] R. Benigno, N. Pedrosa, P. Tom et al., “Annotation-based access control approach for ubiquitous systems,” in *Proceedings of the IEEE 28th International Conference on Advanced Information Networking and Applications*, pp. 218–225, 2014.
- [8] L. Y. Mano, B. S. Faical, L. H. V. Nakamura et al., “Exploiting IoT technologies for enhancing Health Smart Homes through patient identification and emotion recognition,” *Computer Communications*, vol. 89–90, pp. 178–190, 2016.
- [9] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, “The many faces of publish/subscribe,” *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, 2003.
- [10] M. Lussier, M. Lavoie, S. Giroux et al., “Early detection of mild cognitive impairment with in-home monitoring technologies using functional measures: a systematic review,” *IEEE Journal of Biomedical and Health Informatics*, vol. 23, pp. 838–847, 2019.
- [11] X. H. B. Le, M. Di Mascolo, A. Gouin, and N. Noury, “Health smart home for elders - a tool for automatic recognition of activities of daily living,” in *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pp. 3316–3319, IEEE, 2008.
- [12] T. Banerjee, J. M. Keller, M. Popescu, and M. Skubic, “Recognizing complex instrumental activities of daily living using scene information and fuzzy logic,” *Computer Vision and Image Understanding*, vol. 140, pp. 68–82, 2015.
- [13] Z. Liouane, T. Lemlouma, P. Roose, F. Weis, and H. Messaoud, “A markovian-based approach for daily living activities recognition,” in *Proceedings of the 5th International Conference on Sensor Networks (SENSORNETS)*, pp. 214–219, Rome, Italy, 2016.
- [14] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop (MCC)*, pp. 13–16, Association for Computing Machinery (ACM), New York, NY, USA, 2012.
- [15] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, “Fog computing: focusing on mobile users at the edge,” 2015, <https://arxiv.org/abs/1502.01815>.
- [16] M. T. Saqib and M. A. Hamid, “FogR: a highly reliable and intelligent computation offloading on the internet of things,” in *Proceedings of the IEEE Conference (TENCON)*, pp. 1039–1042, IEEE, 2016.
- [17] Y. Elkhatib, B. Porter, H. B. Ribeiro, M. F. Zhani, J. Qadir, and E. Riviere, “On using micro-clouds to deliver the fog,” *IEEE Internet Computing*, vol. 21, no. 2, pp. 8–15, 2017.
- [18] A. A. Mutlag, M. K. A. Ghani, N. Arunkumar, M. A. Mohammed, and O. Mohd, “Enabling technologies for fog computing in healthcare iot systems,” *Future Generation Computer Systems*, vol. 90, pp. 62–78, 2019.
- [19] N. Mohan and J. Kangasharju, “Edge-fog cloud: a distributed cloud for internet of things computations,” in *Proceedings of the Cloudification of the Internet of Things (CIoT)*, pp. 1–6, 2016.
- [20] F. Viani, F. Robol, A. Polo, P. Rocca, G. Oliveri, and A. Massa, “Wireless architectures for heterogeneous sensing in smart home applications: concepts and real implementation,” *Proceedings of the IEEE*, vol. 101, no. 11, pp. 2381–2396, 2013.
- [21] V. Rialle, F. Duchene, N. Noury, L. Bajolle, and J. Demongeot, “Health “smart” home: Information technology for patients at home,” *Telemedicine Journal and E-Health: The Official Journal of the American Telemedicine Association*, vol. 8, no. 4, pp. 395–409, 2002.
- [22] G. P. R. Filho, J. Ueyama, L. A. Villas et al., “Nodepm: a remote monitoring alert system for energy consumption using probabilistic techniques,” *Sensors*, vol. 14, no. 1, pp. 848–867, 2014.
- [23] L. Y. Mano, “Emotional condition in the Health Smart Homes environment: emotion recognition using ensemble of classifiers,” in *Proceedings of the Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–8, IEEE, Thessaloniki, Greece, 2018.
- [24] J. R. T. Neto, G. P. Filho, L. Y. Mano, and J. Ueyama, “VERBO: voice emotion recognition database in portuguese language,” *Journal of Computer Science*, vol. 14, no. 11, pp. 1420–1430, 2018.
- [25] L. O. Sawada, L. Y. Mano, J. R. T. Neto, and J. Ueyama, “A module-based framework to emotion recognition by speech: a case study in clinical simulation,” *Journal of Ambient Intelligence and Humanized Computing*, 2019.
- [26] K. A. I. S. Ram, “Internet-of-things (iot) advances home healthcare for seniors,” 2017, http://www.embeddedintel.com/special_features.php?article=2721.
- [27] P. Maia, A. Baffa, E. Cavalcante, F. C. Delicato, T. Batista, and P. F. Pires, “A middleware platform for integrating devices and developing applications in e-health,” in *Proceedings of the 33rd Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*, pp. 10–18, 2015.
- [28] J. Augusto, D. Kramer, U. Alegre, A. Covaci, and A. Santokhee, “The user-centred intelligent environments development process as a guide to co-create smart technology for people with special needs,” *Universal Access in the Information Society*, vol. 17, no. 1, pp. 115–130, 2018.
- [29] P. Patel, M. I. Ali, and A. Sheth, “On using the intelligent edge for iot analytics,” *IEEE Intelligent Systems*, vol. 32, no. 5, pp. 64–69, 2017.
- [30] G. Daneels, E. Municio, K. Spaey et al., “Real-time data dissemination and analytics platform for challenging iot environments,” in *Proceedings of the 2017 Global Information Infrastructure and Networking Symposium (GIIS)*, pp. 23–30, 2017.
- [31] B. Kang, D. Kim, and H. Choo, “Internet of everything: a large-scale autonomous iot gateway,” *IEEE Transactions on Multi-Scale Computing Systems*, vol. 3, no. 3, pp. 206–214, 2017.
- [32] L. Y. Mano, M. M. Funes, T. Volpato, and J. R. T. Neto, “Explorando tecnologias de IoT no contexto de Health Smart Home: uma abordagem para detecção de quedas em pessoas idosas,” *Journal on Advances in Theoretical and Applied Informatics*, vol. 2, no. 1, pp. 46–57, 2016.
- [33] G. Fortino and R. Gravina, “Fall-mobileguard: a smart real-time fall detection system,” in *Proceedings of the 10th EAI International Conference on Body Area Networks*, pp. 44–50,

Institute for Computer Sciences, Social-Informatics (ICST), 2015.

- [34] C. Ding and D. Tao, "A comprehensive survey on pose-invariant face recognition," *ACM Transactions on Intelligent Systems and Technology*, vol. 7, pp. 37:1–37–42, 2016.
- [35] J. R. T. Neto, G. P. R. Filho, L. Y. Mano, and J. Ueyama, "Inca: um sistema healthcare flexível baseado no paradigma fog computing e publish/subscribe," in *Proceedings of the Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, 2017.
- [36] A. Stanford-Clark and A. Nipper, *Mq Telemetry Transport*, 2014.
- [37] S. Alharbi, P. Rodriguez, R. Maharaja, P. Iyer, N. Bose, and Z. Ye, "FOCUS: a fog computing-based security system for the internet of things," in *Proceedings of the 15th IEEE Annual Consumer Communications and Networking Conference (CCNC)*, pp. 1–5, IEEE, 2018.
- [38] G. P. R. Filho, J. Ueyama, B. S. Faical, D. L. Guidoni, and L. A. Villas, "ResiDI: an intelligent decision platform for residential infrastructure using wireless sensors and actuators," in *Proceedings of the 33rd Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*, pp. 237–245, 2015.
- [39] P. G. Filho, J. R. T. Neto, A. Valejo, R. I. Meneguette, L. A. Villas, and J. Ueyama, "Um sistema de controle neuro-fog para infraestruturas residenciais via objetos inteligentes," in *Proceedings of the Simpósio Brasileiro de Redes de Computadores (SBRC)*, vol. 36, 2018.
- [40] M. Kramer and A. Gerald, "Energy measurements for micaz node," Tech. Rep. KrGe06, University of Kaiserslautern, Kaiserslautern, Germany, 2006.
- [41] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, and Z. Ambadar, "The extended cohn-kanade dataset (ck+): a complete dataset for action unit and emotion-specified expression," in *Proceedings of the IEEE Computer Society, Computer Vision and Pattern Recognition Workshops*, IEEE, 2010.
- [42] S.-L. Hsieh, C.-C. Chen, S.-H. Wu, and T.-W. Yue, "A wrist -worn fall detection system using accelerometers and gyroscopes," in *Proceedings of the 11th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 518–523, 2014.
- [43] E. Casilari, R. Luque, and M.-J. Morón, "Analysis of android device-based solutions for fall detection," *Sensors*, vol. 15, no. 8, pp. 17827–17894, 2015.
- [44] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society*, pp. 111–147, 1974.

