World Wide Web (2020) 23:1341–1360
https://doi.org/10.1007/s11280-019-00692-y

A lightweight and cost effective edge intelligence architecture based on containerization technology

Mabrook Al-Rakhami[1,2] • **Abdu Gumaei**[3] • **Mohammed Alsahli**[2] • **Mohammad Mehedi Hassan**[1] • **Atif Alamri**[1] • **Antonio Guerrieri**[4] • **Giancarlo Fortino**[4,5]

Received: 21 December 2018 / Revised: 27 February 2019 / Accepted: 6 May 2019
Published online: 25 May 2019

Abstract
The integration of Cloud computing and Internet of Things led to rapid growth in the edge computing field. This would not be achievable without combining the data centers' managing systems with much more restrained technologies. One significantly effective and lightweight solution to this issue is presented by the Docker technology. It is able to manage virtualization process and can therefore be used to distribute, deploy and manage cloud and edge applications assigned into the clusters. In our study, this technology was represented by the Raspberry Pi devices, which are convenient thanks to their low cost, robust applicability and lightweight nature. Our application scenario focuses on identification of the human activities. In this paper, we suggest and evaluate an architecture on the basis of the distributed edge/cloud integration paradigm. We explain all of its advantages which lie in the combination of affordability and several other benefits provided by the fact that data processing is conducted by the edge devices instead of the central server. To recognize and identify human activity, the Regularized Extreme Leaning Machine (RELM) was engaged in our architecture. Our study presents detailed information about our use case scenario and the experimental simulation we performed.

Keywords Edge intelligence · Edge computing · Human activity recognition · Docker · Containers, regularized extreme leaning machine

1 Introduction

Contemporary network architectures and computing models are mostly focusing either on the local and exclusive computing or implementation of the shared centralized resources [29].

This article is part of the Topical Collection: *Special Issue on Smart Computing and Cyber Technology for Cyberization*
Guest Editors: Xiaokang Zhou, Flavia C. Delicato, Kevin Wang, and Runhe Huang

✉ Mabrook Al-Rakhami
malrakhami@ksu.edu.sa

Extended author information available on the last page of the article

Table 1 Cloud computing and edge computing

Cloud Computing	Edge Computing
Time consuming processing of applications and data in the cloud.	More effective decentralized data processing on the edge of the network.
Since data is being sent solemnly through the cloud channels, the demands on the bandwidth are significant.	Data is collected at specific access points which decreases importance of the bandwidth.
Depending servers with remote location are causing scalability issues and slow response.	Problems with response delays and scalability can be avoided by visibility of small edge servers for the users.

Cloud computing is nowadays the most prevailing model in the field of smart devices and similar sensors adopting devices, known together as the Internet of Things (IoT) [26, 32]. This model is keen to use centralized shared resources and addresses the raising popularity of smart apps by integrating exclusive resources and the data obtained from them.

Newly emerging operational applications and information technologies are - unlike the IoT devices - very demanding when it comes to the latency and bandwidth. Their requirements for the optimized transport network and processing of data performed as close to the end devices as possible is also rapidly increasing [12, 14].

Some of the limitations that complicate the commercial use of cloud models in computing such as latency and jitter effects [28], distance to the server [1], location awareness of the application [10], data security and privacy [6], and support of mobility [22] cannot be resolved by adoption of the hyper-scale cloud computing technology [25]. This discrepancy between the architecture providing data storage and processing and the networks that would allow this architecture to access the content, creates obstructions that interfere with its potential to attract new markets and be applied in newly emerging use cases such as VR, robotics, e-health care technologies and automation in various fields [3].

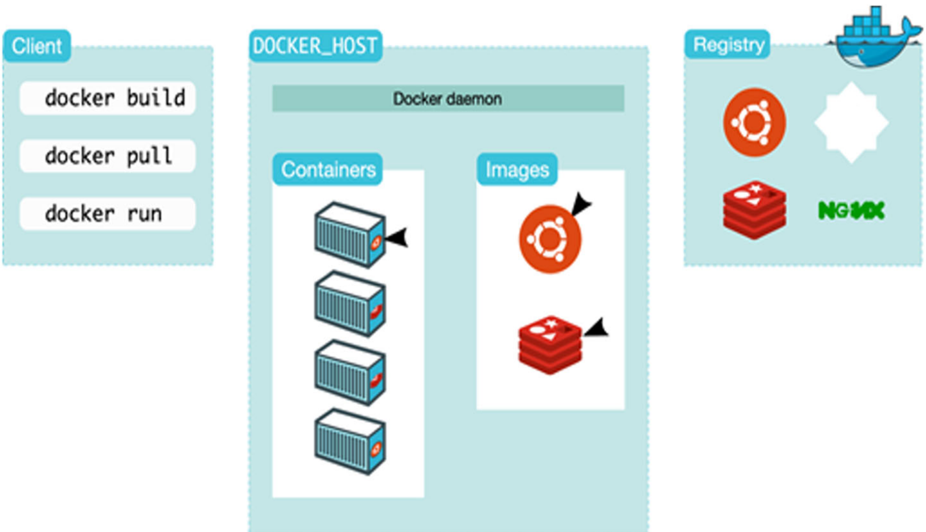

Figure 1 Docker client-server architecture

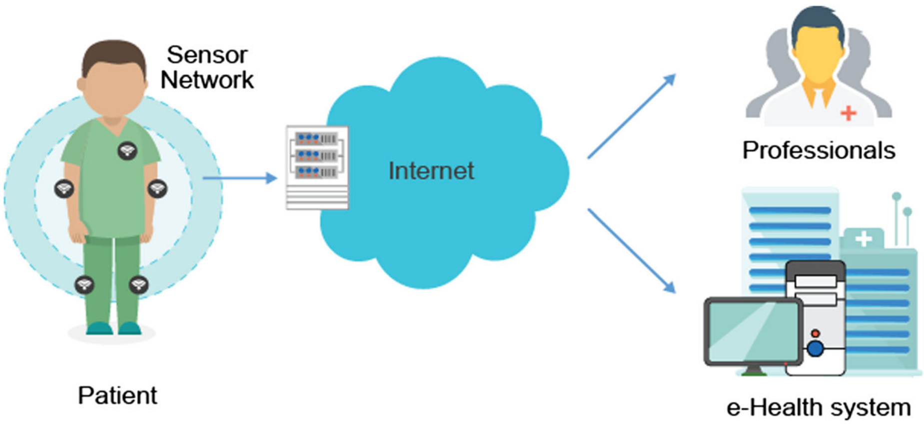

Figure 2 Basic scheme in an e-health system, adopted from [5]

All of these aforementioned use case scenarios are very demanding on the ability to gain and handle large amounts of information within a short timeframe. This subsequently places high anticipations on the architecture's capability to transfer large bundles of data from their sources in just a millisecond. It's very reasonable to predict that contemporary networks will not be able to live up to these high expectations for very long. Newly developed 5G networks are able to provide much shorter transfer delays and their bandwidths are also broadening [16]. But most of the current development in this field is related to the geographical localization of the network's optimizations [11].

In our previous work [5], we proposed the idea to build and integrate Docker containers within the edge and pinpoint the Edge Intelligence's potential in the vertical-specific scenarios of use. Our current study extends this idea further and focuses on the utilization of Docker

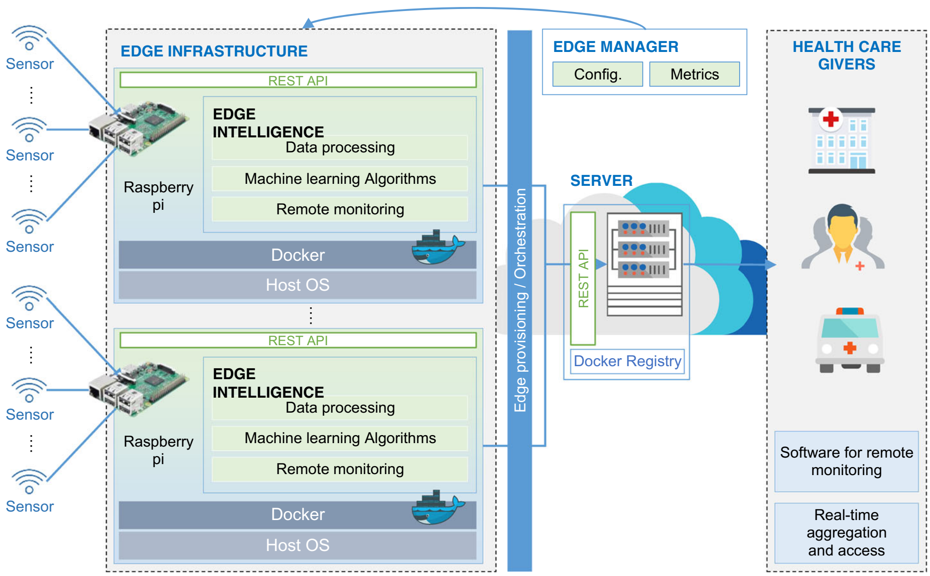

Figure 3 Proposed architcure

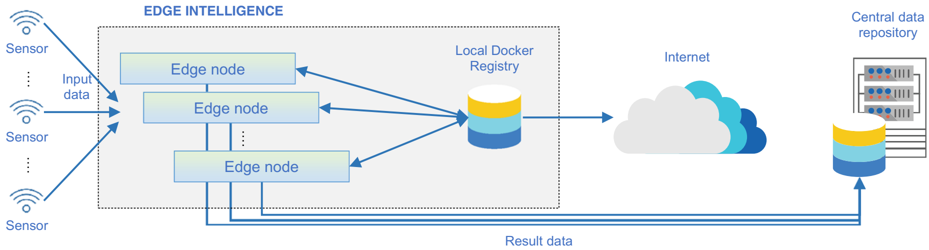

Figure 4 Model of the proposed infrastructure

technology into the recognition of human activity. We expect our architecture to provide significantly shorter delays in communication and decision-making as well as decrease the costs related to these processes thanks to adoption of lightweight and economical platform.

These are some of the abilities our edge intelligent computing model boasts:

- Because our architecture is using ML algorithms and eliminating any unnecessary communication with cloud, it is able to provide more effective and quick decision-making. Unwanted delays are decreased thanks to avoiding of unreasonable roundtrips;
- This framework is able to take advantages of the RELM application, which is known to be conveniently simple and fast in regard to the training process;
- Decreased use of public wide area networks and utilized local algorithms and caching result in less costly communication. None of the data that is not crucial will be transferred over to cloud;
- Application's, network's and user's requests can be balanced better when the edge or core infrastructure can be flexibly amended in order to address interim issues and maintenance;
- Our computing model can utilize decisions related to the pre-processed data and adopt alarms traded between several edge devices.

As far as we know, no other economic-aware computing model has attempted to implement the Docker technology (which is nowadays rapidly gaining popularity) and the edge intelligence before. The system we present builds, analyzes and empirically proves the pioneering concept of Human Activity Recognition. We thoroughly examine this use case on various levels and with various approaches in mind.

The remaining parts of the paper are organized as follows: Section II - Theory and background of concepts related to our paper. This information is important in order to understand the Docker technology and the concept of edge intelligence. Section III - presentation of our own framework and description of all of its important components. Section IV -

Table 2 Characteristics of the participants

Subject	Genre	Age	Height	Weight	#Instances
A	Female	46 y.o.	1.62 m	67 kg	51,577
B	Female	28 y.o.	1.58 m	53 kg	49,797
C	Male	31 y.o.	1.71 m	83 kg	51,098
D	Male	75 y.o.	1.67 m	67 kg	13,161

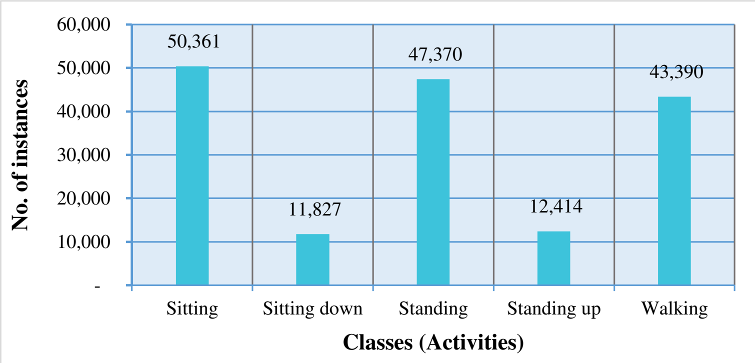

Figure 5 Distribution of dataset classes

Description of our empirical experiments and evaluation of our framework's performance in the use case of human activity recognition scenario. Section V - Conclusion and introduction of the further extension of the research.

2 Theoretical background

2.1 Edge computing

Edge computing is in many regards opposite to cloud computing. The data is processed here in the decentralized manner at the network's edge [2, 15]. This computing model is able to connect storage services, intelligent devices and cloud. To illustrate differences between the aforementioned computing models, Table 1 provide a brief comparison. [27].

Edge computing allows processing of large amounts of data close to the internet of things devices such as sensors in the network. This provides increased quality of the services and helps to prevent issues with delays and latency. The overall consumption of the networks can also be decreased and another advantage lies in upgraded security. These are the most important demands placed on the edge computing:

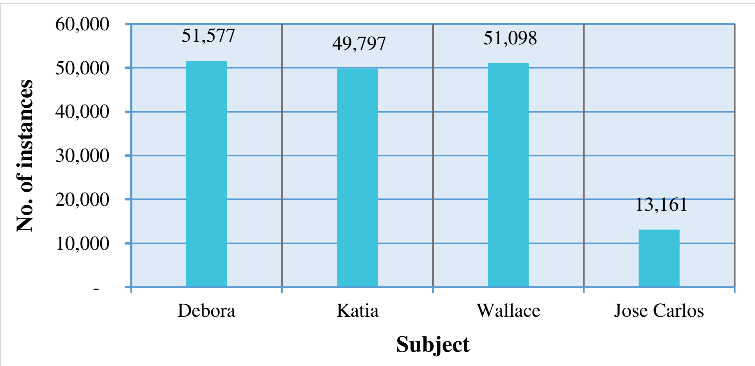

Figure 6 Distribution of instances per subject

Table 3 Dataset subjects training and testing

Subject	No. of training instances	No. of testing instances
Debora	4000	1000
Katia	4000	1000
Wallace	4000	1000
Jose Carlos	3200	800

- In some of the processes, literally every millisecond counts. **Elimination of latency** is therefore more than welcomed. Latency can be avoided when data is processed by the device as close to the point of their aggregation as possible.
- Huge amounts of generated data in the network of sensors require **effective management of the available bandwidth.**
- **Security** remains one of the largest concerns. Data must be safe and secured both during the transfer and in the static state.
- Data must be always available and integrated, because it is often crucial for the safe operation of systems related to the security and general infrastructure of the human society. The network has to be **reliable** in every situation.
- Internet of things devices are highly **versatile** and can be deployed for data collection in many different environments and ggeographical conditions.
- **Good localization of data processing** is important for improvement of the response time. This is especially critical in cases when extremely fast responses are required. However, the past data is stored and processed in the cloud.

There are many scenarios that can take advantage of the edge computing. Mostly, it is being used to collect real-time data related to the IoT devices integrated into the network that are depending on the provided data in their decision-making. Aside of its effectiveness and versatility, edge computing also has its weaker points that need to be attended, secure public use and offloading / partitioning of the tasks being just two of them [31]. Another concern lies in the probable lack of insight on the complete data when some parts of it are processed on the edge.

2.2 Edge intelligence

Edge intelligence (EI) comprises edge computing that is supported with machine learning algorithms and advanced networking abilities. The implication is that a number of information

	Sitting	Sitting down	Standing	Standing up	Walking
Sitting	209	0	0	0	0
Sitting down	0	196	0	0	0
Standing	0	0	193	0	0
Standing up	0	1	1	207	0
Walking	0	9	19	1	164

Figure 7 Confusion Matrix of RELM for Debora

	Sitting	Sitting down	Standing	Standing up	Walking
Sitting	209	0	0	0	209
Sitting down	7	186	1	2	7
Standing	0	0	193	0	0
Standing up	0	1	7	201	0
Walking	10	40	31	6	10

Figure 8 Confusion Matrix of ELM for Debora

technology and operational technology industries are gravitating to the edge of the network. The result is that issues such as cybersecurity, self-learning, real-time networks and tailored connectivity can all be adequately dealt with. There is no doubt that the leading technological innovations to the needs of startups and new markets are container technology and machine learning. Deploying these technologies to the edge ensures the organizations can leverage the power of EI. Following a thorough analysis of such developments, we can come to the conclusion that:

- Although containerization is crucial for EI, and there are currently no exact Standards in the area, various related open-source programs are developed, such as the Open container initiative and the aforementioned Docker.
- The EI's success, in general, depends on Common Data Models for Edge Computing Node (ECN) communication.
- Micro Data Centers (MDS) are about to gain in importance. Among many other reasons, this will be caused by their capability to achieve low latency, process huge amounts of data, and avoid their forwarding to the cloud servers.
- Upgraded wireless networking technology will be able to provide the edge data centers and allow industry-specific services to be implemented by software-defined principles of networking and virtualization.
- The best user interface is no user interface. This can be now achieved, as the manual input of data becomes unnecessary with IoT and the decisions are being made by machine learning and artificial intelligence.

2.3 Docker technology

The Docker platform allows deployment of application and all of its dependent parts automatically within the containers - characteristic autonomous environment typical for this technology [9]. Docker is lightweight and enables quick and smooth arrangement of the containers, which are all completely independent, divided from one another and provided with separate network interfaces. The aforementioned division of containers creates environment not very different from the virtual machines. But it also spares user from the use of overhead guest OS (operating system). The client/server architecture[1] built on the Docker where the Docker client connects to the Docker daemon can be seen on the Figure 1. Both

[1] https://docs.docker.com/engine/docker-overview/

	Sitting	Sitting down	Standing	Standing up	Walking
Sitting	209	0	0	0	209
Sitting down	0	189	0	0	0
Standing	0	0	193	0	0
Standing up	0	1	4	204	0
Walking	1	28	35	5	1

Figure 9 Confusion Matrix of SVM for Debora

parties of this connection (which is provided through RESTful API) can function in remote and local systems alike.

The minimal overhead for Docker is created by sharing of the kernel with the host operating system. This distinguishes the system from virtual machines, which use hypervisors to run the kernel of the guest OS. [24]. But there are even more features which are making Docker technology more convenient than virtual machines for the purpose of running various applications. Its containers are, for example, great for development of dynamic cloud-based applications which are scaling with load or adding/removing various features on the user's demand. The containers are managed via the Linux kernel namespaces and control groups. This allows isolation of separate processes, which are led to believe that they are all running on separate operating systems with memory and processor dedicated to them only. This limitation of resources results in achieving only insignificant overhead.

2.4 Human activity recognition

The technology of human activity recognition is usable and desirable in many different fields and environments ranging from security to e-health care. It's ability to identify activities of different individuals by evaluating this specific person in the given situation and surroundings [20] can be, for example, invaluable for surveillance and rehabilitation on a remote basis [21].

By observing the human body in the relation to the gravity, this technology can identify any movement of the body and recognize the activity it represents. Both full body and partial (limbs/head only) movements and positions can be observed and analyzed. Observed activity is consequently evaluated as static (lying, standing, sitting) or dynamic (walking, jumping, running). Initially, the technology recognized activities through various image and video processing technologies. However, these are all quite limited in cases of imperfect light conditions. Another concern relates to the privacy of users, who may feel uneasy about constant presence of cameras. Intelligent body sensors such as accelerometers can resolve this

	Sitting	Sitting down	Standing	Standing up	Walking
Sitting	196	0	0	0	0
Sitting down	0	194	2	0	3
Standing	0	0	204	0	0
Standing up	0	0	3	202	0
Walking	0	5	2	0	189

Figure 10 Confusion Matrix of RELM for Katia

	Sitting	Sitting down	Standing	Standing up	Walking
Sitting	196	0	0	0	0
Sitting down	0	190	4	1	4
Standing	0	0	204	0	0
Standing up	0	1	3	201	0
Walking	1	7	6	10	172

Figure 11 Confusion Matrix of ELM for Katia

easily [7]. And that is also reflected by the rapid development that currently can be seen in this field.

Wearable and comfortably lightweight body sensors are being upgraded and evolved in recent years as never before [23]. The most recent inertial sensors can collect data related to human activity very effectively and conveniently. Our paper focuses on their use in the field of health care - particularly, we analyze its potential in care for non-critical patients who do, however, need permanent monitoring.

In order to evaluate this, we first need to understand and discuss the contemporary services in this field. Generally, services provided to this type of outpatients (Figure 2) work on the basis of data collecting provided by various devices and sensors. These technologies aggregate important measurements and forward them to the dedicated caregiver located somewhere else. This data transfer usually happens through Internet, if no constraints related to its availability are present. After reaching its final destination, the data can be rigorously inspected and acted upon. This systematic scheme can provide patients with constant monitoring that can be very helpful as a prevention of overseen issues and changes in the patient's state. This system also collects large amounts of data that can be useful for achieving more integral and objective diagnostics. Generally, this modification will also allow health care to be less costly, more efficient and provide faster response [4].

2.5 Regularized extreme leaning machine (RELM)

Extreme Learning Machine (ELM) are neural networks of feedforward type. This architecture is similar to the Multilayered Perceptron Network (MLP), but presents a much faster learning phase [17, 19]. More specifically, the ELM network consists of an input with p characteristics, a hidden layer with q neurons and an output layer with c neurons, all neurons with sigmoidal activation functions. The output of the i-th neuron, $y_i\ (t)$, $i = 1, \ldots, c$, is given by:

	Sitting	Sitting down	Standing	Standing up	Walking
Sitting	196	0	0	0	0
Sitting down	0	192	2	1	4
Standing	0	0	204	0	0
Standing up	0	0	2	203	0
Walking	0	6	2	0	188

Figure 12 Confusion Matrix of SVM for Katia

	Sitting	Sitting down	Standing	Standing up	Walking
Sitting	210	0	0	0	0
Sitting down	1	189	14	9	5
Standing	0	0	192	0	0
Standing up	0	6	1	171	0
Walking	0	0	9	0	193

Figure 13 Confusion Matrix of RELM for Wallace

$$y_i(t) = \tanh\left[\sum_{k=1}^{q} m_{ik}(t)\tanh\left(\sum_{j=0}^{p} w_{kj}(t)x_j(t)\right)-\theta_k(t)\right] \tag{1}$$

where $x_j(t)$ is the j-th component of the current input vector, w_{kj} is the weight that connects the j^{th} entry to the k^{th} hidden neuron, and m_{ik} is the weight that connects the k^{th} hidden neuron to the i-th output neuron. We still define $x_0 = +1$ and the weights w_{k0} correspond to the thresholds of the hidden neurons. The parameter θ_k is the threshold of the i^{th} output neuron. The ELM network training is performed through three steps: 1) random initialization of the weights of the hidden neurons, 2) accumulation of the outputs of the hidden neurons and 3) calculation of the weights of the exit neurons.

In ELM, one of the parameters that needs to be well chosen is the number of neurons in the hidden layer to obtain a good inter-compromise / overfitting compromise. To circumvent this, Deng et al., [13] proposed a regularized version of ELM. Although this approach results in good generalization, the obtained network is dense and generally requires more storage space and processing time for new samples for applications with large volumes of data [18]. Moreover, in these applications, the ELM in its regularized version may suffer from memory limitation and an intense cost-computation for the inversion of large matrices. For binary classification, the extension of the R-ELM is straight. For multi-class classification problems, approaches using classifier-vs-rest can be adopted. However, a large number of classes usually results in a higher cost of ownership in the training phase. Although the use of RELM generally results in a more compact network, it suffers from the same problems as ELM with regularization for learning tasks with larger data volumes.

3 Proposed framework

In the following section, we explain our proposed framework in full detail. This framework is consisting of network of sensors and central server that controls them, as can be seen in

	Sitting	down	Standing	Standing up	Walking
Sitting	204	2	0	0	4
Sitting down	1	164	26	23	4
Standing	0	0	192	0	0
Standing up	0	12	2	162	2
Walking	0	24	34	32	112

Figure 14 Confusion Matrix of ELM for Wallace

	Sitting	Sitting down	Standing	Standing up	Walking
Sitting	207	0	0	0	3
Sitting down	1	184	2	28	3
Standing	0	0	190	1	1
Standing up	1	12	1	160	4
Walking	0	2	10	3	187

Figure 15 Confusion Matrix of SVM for Wallace

Figure 3. The data load in the framework is classified and distributed towards the edge micro-computers to lower the server's demand on resources providing its functionality. A model for this framework can be seen in Figure 4, where huge amounts of cached data is stored on the edge where it will be processed by edge nodes. Results will be forwarded to the remote side directly, where they will be notified and stored. Each group of sensors is paired with its dedicated Raspberry Pi which serves as a bridge between the sensors and central server. This server is equipped with repository for these micro-computers and it can pull the latest image of the trained classifier.

In our framework, all components use REST API for communication. This allows them to further abstract every layer and achieve better implementation and extension. This system can be used with any kind of sensors with classifier images being pushed through the network solemnly. The edge manager allows API specification to be amended anytime through the menial description of the network. Edge provisioning and orchestration allows to manage several automation services with the same kinds to ensure effectiveness. It can be considered as a middle-layer between edge infrastructure and the server to fetches service and resources management capabilities to containers. Moreover, it enables the management of containers to ensuring each has enough resources for based on its needs.

Our design is run by Docker and there are Docker images built on the central server and consequently deployed on the edge in order to classify the given data. This platform allows every image to be sent with all demanded dependency and requirements of the environment. The deployment process is automatized through the Docker registry service located on the server. This registry contains Docker images which represent micro-services provided by the edge computers and the central server. Any computer which is connected to this network can, therefore, pull and use any of the available micro-services. One possible limitation is linked to the different computing power requirements of each of these services. This registry also allows the computers on the edge to validate currently running image instance and check if it is not corrupted or amended.

	Sitting	Sitting down	Standing	Standing up	Walking
Sitting	184	0	0	1	1
Sitting down	0	81	18	1	8
Standing	0	0	209	1	2
Standing up	0	2	2	103	0
Walking	0	0	36	3	148

Figure 16 Confusion Matrix of RELM for Jose Carlos

	Sitting	Sitting down	Standing	Standing up	Walking
Sitting	178	1	0	7	0
Sitting down	0	81	15	0	12
Standing	0	0	210	0	2
Standing up	1	43	3	16	44
Walking	1	2	76	2	106

Figure 17 Confusion Matrix of ELM for Jose Carlos

Our focus lies on the scenario of human activity recognition. This requires the system to be capable of measuring highly variable physiological activities and measurements of different patients. The component providing human activity recognition must be able to function upon the data gathered by the wearable sensors. By implementing algorithms used in machine learning, our framework is also expected to improve processing of data and decision-making. According to specific abilities and statuses of patients, smart alarms can be set to perfectly suit their needs. In our experimental test, the Docker image contains the RELM deployed to classify activities by processing the data forwarded by the body sensors.

4 Experments and results

In this section we briefly describe the used dataset and results of our architecture's experimental testing.

4.1 Dataset

For experimental testing of our framework, we used the publicly available dataset called Ugulino [30]. Our experimental team consisted of four persons wearing accelometers placed on their abdomens, right arms, left tights, and right ankles. Five types of activities were observed in our experiment: sitting, standing, the movements of standing up from the sitting position/ sitting down from the standing position, and walking. All four members of our team engaged into these activities separately in the timeframe of 2 h. Each of the subjects contributed with twelve values, as sensors measured their activities in the x, y, and z directions. The continuous flow of data was consequently separated into the one-second-long windows with 150 ms overlaps. To provide as accurate measurements as possible, we calibrated the accelerometers before and also after the performing of experiments. We achieved 165,633 instances of the gathered data in total. Here are some samples of the given instances:

	Sitting	down	Standing	Standing up	Walking
Sitting	185	0	0	1	0
Sitting down	0	86	11	3	8
Standing	0	0	210	1	1
Standing up	4	10	3	90	0
Walking	0	5	59	14	109

Figure 18 Confusion Matrix of SVM for Jose Carlos

Table 4 Parameters setting of SVM

Parameter	Value
Kernel Function	Linear
Box Constraint Level	5
Multiclass Method	One-vs-All

Users; Gender; Age; HowTallInMeters; Weight; BodyMassIndex; X1; Y1; Z1; X2; Y2; Z2; X3; Y3; Z3; X4; Y4; Z4; activity.

Class.jose_carlos;Man;75;1,67;67;24,0;8;114;-174;-36;86;-112;13;170;-130;-172;-106;-123;walking.

For better convenience, we changed activities labels for the basic integrals as follows: 1 (sitting), 2 (sitting down), 3 (standing), 4 (standing up), 5 (walking). The indexes were later extracted in order to separate sets of testing and training data. Table 2 shows some dataset details while Figures 5 and 6 depict the data frequency belonging to each category and each subject based on activity class (standing up, sitting down, walking, sitting, standing).

Thousand of each activity instances were randomly chosen gather data subset for each of the engaged subjects. Since the total count of standing up / sitting down activities of one of the members was rather low, we chose only 500 instances for this subject. That means that each of the other subjects' datasets consist of 5000 instances, while this specific subject's dataset is containing only 4000 instances. One-fifth of these datasets was used for testing. Remaining data was applied in the training. Number of instances used for testing / training per each subject can be seen in Table 3.

4.2 Experimental results

All experiments were performed on Raspberry pi 3 and with 2 x 16GB and 8 x virtual central processing units on Digitalocean[2] . After connecting the Raspberry pi, we simulated collection of the patients' data through sensors monitoring physiological activities. We rented two Digitalocean servers which were used in order to simulate huge amount of sensors connected into the micro-computer. We left the Docker status on default setting during all of our experiments to collect metrics of all of the working containers (running instance of the Docker image). To assess our framework, three main measures were used: testing / training time, accuracy, bandwidth utilization / network scalability. Below, we present results of these experimental measurements.

4.2.1 Accuracy

In this subsection, we measure the activity classification accuracy of RELM classifier proposed in our architecture compared to SVM [8] and traditional extreme learning machine (ELM) classifiers. RELM algorithm has enough flexibility to transform the data from low dimensions to high dimensions by adding any number of hidden neurons, while the SVM is limited in this regards. This means that RELM classifier has better generalization than the limited regularization constraints of SVM.

[2] https://www.digitalocean.com

Table 5 Parameters setting of ELM and RELM

Parameter	Value
Number of Hidden Nodes	400
Activation Function	Sigmoid
Regularization Set of RELM	{−3,-2.99,-2.98,…,3}

To calculate the accuracy, the confusion matrix of testing dataset after training each classifier model is first computed. Figures 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 and 18 show the confusion matrices of each subject for the RELM, ELM and RELM classifiers. Before applying the classification task, the parameters of these classifiers are initialized as mentioned in Tables 4 and 5.

In all confusion matrices, the number of instances in dark blue cells represent the instances of each activity which are correctly classified. Thus, the accuracy is computed as:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{2}$$

Where TP and TN are the true positive and negative rates; whereas FP and FN are the false positive and negative rates.

Table 6 demonstrates the accuracy of proposed RELM compared to other classifiers.

We can see that RELM is more accurate for human activity recognition than One-versus-All linear SVM and traditional ELM classifiers.

4.3 Training and testing time

Training and testing time is very critical in the case of big data coming from huge number of sensors. Therefore, in our study, the time of training and testing for each classifier is also evaluated for showing the efficiency of proposed RELM classifier against SVM proposed in [18]. An initial experiment using MATLAB R2015a on a laptop Core (TM) i7 with Windows 10 (×64), Intel(R) CPU of 2.0-GHz, and 8-GB RAM is performed. The training and testing time in second of each classifier model is reported in Table 7.

From Tables 7, we notice that the training time of RELM is almost small and acceptable compared to the SVM. Moreover, the testing time of ELM and RELM is almost equal and lower than the testing time of SVM.

Table 6 Accuracy for SVM, ELM and RELM

Classifier Subject (No. of instances)	SVM	ELM	RELM
Debora (4000 instances for training and 1000 instances for testing)	91.9%	89.5%	96.9%
Katia (4000 instances for training and 1000 instances for testing)	98.3%	96.3%	98.5%
Wallace (4000 instances for training and 1000 instances for testing)	92.8%	83.4%	95.5%
Jose_carlos (3200 instances for training and 800 instances for testing)	85.0%	73.9%	90.625%
Avg.	92.0%	85.8%	95.38%

4.4 Network scalability and bandwidth utilization

In every experiment, the classification request to micro-services was created by our simulated sensors and sent to the Raspberry pi or simulated micro-computer. The tool that was used for creating these requests was built on the Docker platform and scaled according to the demand. The experiments were performed by stressing the connected computers, as can be seen in Tables 8 and 9 and Figure 19.

Table 8 pinpoints the successful implementations on a Raspberry pi and also shows the systems restraint. The green fields represent Raspberry pi's successful experiments and the yellow ones depict server experiments. The experiments were not marked at all in cases when the classifier wasn't able to conduct complete classification in less than one second. In cases when the classifier couldn't classify even single request, these experiments were considered unsuccessful. Requests per second figure represents the amount of requests each sensor sends to the service per one second (for example, if the sensor sends data every 200 ms, it means that per single second, 5 requests are sent). The average size of request data is 0.82 KB without the TCP/IP headers on every sensor request.

In regard to the bandwidth utilization, our experiments reveal a maximal network's bandwidth utilization with minimal size of the data traded between both sensors and micro-computers and micro-computers and server. Figure 9 shows comparison of the network bandwidth under distinct configurations for micro-computers and sensors. The average size of the sensor to backend micro-computer requests is 0.83 KB. The number of update requests coming from the sensor is 5.

It should be noted here that in case that the number of connected sensors for one backend is 50 there are two micro-computers, it would result in 100 sensors connected to two separate computers.

Table 7 provides comparison of the server's utilization for bandwidth with the anticipated network bandwidth in scenario including the micro-computer connected to one server. The average request size sent from micro-computer to server is 0.62 KB. Average size of request sent from sensor to backend micro-computer is 0.83 KB. Number of update requests from sensor to backend is 5 per second. Number of update requests from micro-computer to server is 5 per second. The single server with amount of connected micro-computers is marked in yellow. The red columns illustrate the inability of micro-computers to cope with the update requests from 1000 / 2000 sensors when the experimental amount of micro-computers.

In our experiment, a single micro-computer was able to manage up to 470 sensors and every sensor could update its value every 200 ms (5 times per second). This creates 1.95MBps bandwidth. In case that the sensors would be connected with a server, the internet would be idle with their updates, aside from the lost packets and the server's remote connection. However, when the sensors are connected with a single micro-computer with ability to classify their updates and forward the data to the server (and that would take 3.1KBps), it would make a huge difference in the number of required updates. Furthermore, the micro-computer could

Table 7 Training and testing time for SVM, ELM and RELM

Classifier No. of instances	SVM	ELM	RELM
Training on 4000 instances	48.23301	0.131870	9.747786
Testing on 1 instance	0.0066	0.0000871	0.0000869

Table 8 Updates requests per second vs. number of sensors

Number of sensors	Value Update per Second											
	1	2	3	4	5	6	20	25	40	45	60	1000
5	5.0	10.0	15.0	20.0	25.0	30.0	100.0	125.0	200.0	225.0	300.0	5000.0
10	10.0	20.0	30.0	40.0	50.0	60.0	200.0	250.0	400.0	450.0	600.0	10,000.0
15	15.0	30.0	45.0	60.0	75.0	90.0	300.0	375.0	600.0	675.0	900.0	15,000.0
20	20.0	40.0	60.0	80.0	100.0	120.0	400.0	500.0	800.0	900.0	1200.0	20,000.0
25	25.0	50.0	75.0	100.0	125.0	150.0	500.0	625.0	1000.0	1125.0	1500.0	25,000.0
30	30.0	60.0	90.0	120.0	150.0	180.0	600.0	750.0	1200.0	1350.0	1800.0	30,000.0
35	35.0	70.0	105.0	140.0	175.0	210.0	700.0	875.0	1400.0	1575.0	2100.0	35,000.0
40	40.0	80.0	120.0	160.0	200.0	240.0	800.0	1000.0	1600.0	1800.0	2400.0	40,000.0
45	45.0	90.0	135.0	180.0	225.0	270.0	900.0	1125.0	1800.0	2025.0	2700.0	45,000.0
50	50.0	100.0	150.0	200.0	250.0	300.0	1000.0	1250.0	2000.0	2250.0	3000.0	50,000.0
100	100.0	200.0	300.0	400.0	500.0	600.0	2000.0	2500.0	4000.0	4500.0	6000.0	100,000.0
200	200.0	400.0	600.0	800.0	1000.0	1200.0	4000.0	5000.0	8000.0	9000.0	12,000.0	200,000.0
400	400.0	800.0	1200.0	1600.0	2000.0	2400.0	8000.0	10,000.0	16,000.0	18,000.0	24,000.0	400,000.0
800	800.0	1600.0	2400.0	3200.0	4000.0	4800.0	16,000.0	20,000.0	32,000.0	36,000.0	48,000.0	800,000.0
1000	1000.0	2000.0	3000.0	4000.0	5000.0	6000.0	20,000.0	25,000.0	40,000.0	45,000.0	60,000.0	1,000,000.0
10,000	10,000.0	20,000.0	30,000.0	40,000.0	50,000.0	60,000.0	200,000.0	250,000.0	400,000.0	450,000.0	600,000.0	10,000,000.0
100,000	100,000.0	200,000.0	300,000.0	400,000.0	500,000.0	600,000.0	,000,000	2,500,000.0	4,000,000.0	4,500,000.0	6,000,000.0	100,000,000.0

Table 9 Comparison of network bandwidth between the server and different configuration for sensor and microcomputers

#Micro-computers	Server	#Sensors														
		1	2	3	4	5	6	7	8	9	10	100	200	470	1000	2000
1	3.10	4.20	8.30	12.50	16.60	20.80	24.90	29.10	33.20	37.40	42.00	415.00	830.00	1951.00	4150.00	8300.00
5	15.50	20.80	41.50	62.30	83.00	103.80	124.50	145.30	166.00	186.80	208.00	2075.00	4150.00	9753.00	20,750.00	41,500.00
10	31.00	41.50	83.00	124.50	166.00	207.50	249.00	290.50	332.00	373.50	415.00	4150.00	8300.00	19,505.00	41,500.00	83,000.00
15	46.50	62.30	124.50	186.80	249.00	311.30	373.50	435.80	498.00	560.30	623.00	6225.00	12,450.00	29,258.00	62,250.00	124,500.00
20	62.00	83.00	166.00	249.00	332.00	415.00	498.00	581.00	664.00	747.00	830.00	8300.00	16,600.00	39,010.00	83,000.00	166,000.00
25	77.50	103.80	207.50	311.30	415.00	518.80	622.50	726.30	830.00	933.80	1038.00	10,375.00	20,750.00	48,763.00	103,750.00	207,500.00
30	93.00	124.50	249.00	373.50	498.00	622.50	747.00	871.50	996.00	1120.50	1245.00	12,450.00	24,900.00	58,515.00	124,500.00	249,000.00
35	108.50	145.30	290.50	435.80	581.00	726.30	871.50	1016.80	1162.00	1307.30	1453.00	14,525.00	29,050.00	68,268.00	145,250.00	290,500.00
40	124.00	166.00	332.00	498.00	664.00	830.00	996.00	1162.00	1328.00	1494.00	1660.00	16,600.00	33,200.00	78,020.00	166,000.00	332,000.00
45	139.50	186.80	373.50	560.30	747.00	933.80	1120.50	1307.30	1494.00	1680.80	1868.00	18,675.00	37,350.00	87,773.00	186,750.00	373,500.00
50	155.00	207.50	415.00	622.50	830.00	1037.50	1245.00	1452.50	1660.00	1867.50	2075.00	20,750.00	41,500.00	97,525.00	207,500.00	415,000.00
55	170.50	228.30	456.50	684.80	913.00	1141.30	1369.50	1597.80	1826.00	2054.30	2283.00	22,825.00	45,650.00	107,278.00	228,250.00	456,500.00
60	186.00	249.00	498.00	747.00	996.00	1245.00	1494.00	1743.00	1992.00	2241.00	2490.00	24,900.00	49,800.00	117,030.00	249,000.00	498,000.00
65	201.50	269.80	539.50	809.30	1079.00	1348.80	1618.50	1888.30	2158.00	2427.80	2698.00	26,975.00	53,950.00	126,783.00	269,750.00	539,500.00
70	217.00	290.50	581.00	871.50	1162.00	1452.50	1743.00	2033.50	2324.00	2614.50	2905.00	29,050.00	58,100.00	136,535.00	290,500.00	581,000.00
75	232.50	311.30	622.50	933.80	1245.00	1556.30	1867.50	2178.80	2490.00	2801.30	3113.00	31,125.00	62,250.00	146,288.00	311,250.00	622,500.00
80	248.00	332.00	664.00	996.00	1328.00	1660.00	1992.00	2324.00	2656.00	2988.00	3320.00	33,200.00	66,400.00	156,040.00	332,000.00	664,000.00
85	263.50	352.80	705.50	1058.30	1411.00	1763.80	2116.50	2469.30	2822.00	3174.80	3528.00	35,275.00	70,550.00	165,793.00	352,750.00	705,500.00
90	279.00	373.50	747.00	1120.50	1494.00	1867.50	2241.00	2614.50	2988.00	3361.50	3735.00	37,350.00	74,700.00	175,545.00	373,500.00	747,000.00
95	294.50	394.30	788.50	1182.80	1577.00	1971.30	2365.50	2759.80	3154.00	3548.30	3943.00	39,425.00	78,850.00	185,298.00	394,250.00	788,500.00
100	310.00	415.00	830.00	1245.00	1660.00	2075.00	2490.00	2905.00	3320.00	3735.00	4150.00	41,500.00	83,000.00	195,050.00	415,000.00	830,000.00
1000	3.10	4.20	8.30	12.50	16.60	20.80	24.90	29.10	33.20	37.40	42.00	415.00	830.00	1951.00	4150.00	8300.00
2000	15.50	20.80	41.50	62.30	83.00	103.80	124.50	145.30	166.00	186.80	208.00	2075.00	4150.00	9753.00	20,750.00	41,500.00

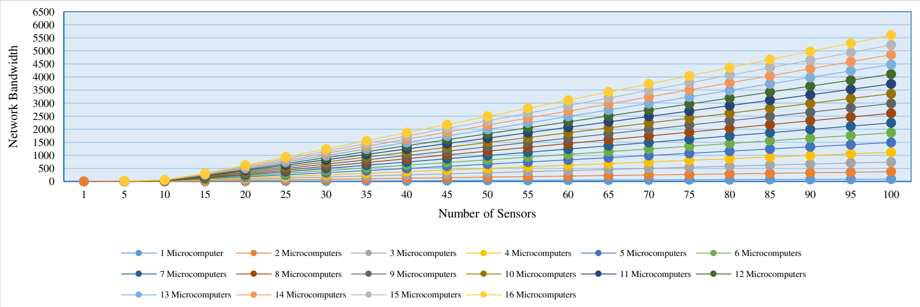

Figure 19 Comparison of network bandwidth of different configuration for sensor and microcomputers

also classify the data altogether and report the status for all engaged sensors in one bundle of data.

5 Conclusion

Its beyond doubt that the way we use omnipresent IoT computing devices and gather and store our data will have to change in the near future. The innovation can come from the field of the edge computing - the decentralized model which looks away from the centralized cloud and focuses on the edge of the network instead. This computing model operates with the resources and storage at the edge of the network and it deploys services of IoT on the edge devices to lower processing costs and overcome the issues with latency.

Our paper aims to identify potential and advantages of the Edge Intelligence by testing it on the vertical-specific use scenarios. We implemented a scenario of Human Activity Recognition and utilized the Docker platform on Raspberry Pi. Our main motivation was to propose a new architecture that would allow use of the intelligent edge application built upon inexpensive, effective and lightweight container technology. The benefits would include lower delays, increased effectiveness in communication and faster decision-making. Our future work will focus on the practical implementation of this simulated framework. Furthermore, we will introduce more possible use cases and perform advanced measurements to provide data for comparison with other similar systems. Hyper parameter optimization is also an open direction to be investigated, is it differ from application to application, accordingly, regulated extreme learning parameter optimization should be tested. Proposed framework should give researchers an opportunity to shape, extend and upgrade this solution to amend it for use in different scenarios and to suit individual needs.

References

1. Abdou, A., Van Oorschot, P.C.: Server location verification (SLV) and server location pinning: augmenting TLS authentication. ACM Trans. Privacy Sec. (TOPS). **21**(1), 1 (2018)
2. Ali, Z., Hossain, M. S., Muhammad, G., Ullah, I., Abachi, H., Alamri, A.: Edge-centric multimodal authentication system using encrypted biometric templates. Futur. Gener. Comput. Syst. (2018)
3. Al-Qurishi, M., Al-Rakhami, M., Al-Qershi, F., Hassan, M.M., Alamri, A., Khan, H.U., Xiang, Y.: A framework for cloud-based healthcare services to monitor noncommunicable diseases patient. Int. J. Distrib. Sensor Netw. **11**(3), 985629 (2015)
4. Al-Rakhami, M., Alhamed, A.: Cloud-based graphical simulation tool of ECG for educational purpose. 25

5. Al-Rakhami, M., Alsahli, M., Hassan, M. M., Alamri, A., Guerrieri, A., Fortino, G.: Cost Efficient Edge Intelligence Framework Using Docker Containers. 800-807
6. Au, M.H., Liang, K., Liu, J.K., Lu, R., Ning, J.: Privacy-preserving personal data operation on mobile cloud—chances and challenges over advanced persistent threat. Futur. Gener. Comput. Syst. **79**, 337–349 (2018)
7. Bellifemine, F., Fortino, G., Giannantonio, R., Gravina, R., Guerrieri, A., Sgroi, M.: SPINE: a domain-specific framework for rapid prototyping of WBSN applications. Softw.: Pract. Exper. **41**(3), 237–265 (2011)
8. Ben-Hur, A., Ong, C.S., Sonnenburg, S., Schölkopf, B., Rätsch, G.: Support vector machines and kernels for computational biology. PLoS Comput. Biol. **4**(10), e1000173 (2008)
9. Boettiger, C.: An introduction to Docker for reproducible research. ACM SIGOPS Operat. Syst. Rev. **49**(1), 71–79 (2015)
10. Chang, D., Patra, A., Bagepalli, N., Anantha, M.: Location-Aware Virtual Service Provisioning in a Hybrid Cloud Environment, Google Patents (2017)
11. Cicirelli, F., Fortino, G., Guerrieri, A., Spezzano, G., and Vinci, A.: Edge enabled development of smart cyber-physical environments. 003463-003468
12. Delsing, J., Eliasson, J., van Deventer, J., Derhamy, H., Varga, P.: Enabling IoT automation using local clouds. 502-507
13. Deng, W., Zheng, Q., Chen, L.: Regularized extreme learning machine. Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium. 389-395, (2009)
14. Derhamy, H., Eliasson, J., Delsing, J.: IoT interoperability—on-demand and low latency transparent multiprotocol translator. IEEE Internet Things J. **4**(5), 1754–1763 (2017)
15. Garcia Lopez, P., Montresor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., Barcellos, M., Felber, P., Riviere, E.: Edge-centric computing: vision and challenges. ACM SIGCOMM Comput. Commun. Rev. **45**(5), 37–42 (2015)
16. Goertz, C., Vits, T., Eßmann, C.: Edge Computing: the Third Major Step in the Evolution of Telco Networks. Future Telco, Pp. 131-142: Springer, 2019
17. Gumaei, A., Sammouda, R., Al-Salman, A.M.S., Alsanad, A.: An improved multispectral Palmprint recognition system using autoencoder with regularized extreme learning machine. Comput. Intell. Neurosci. **2018**, 13 (2018)
18. Gumaei, A., Sammouda, R., Al-Salman, A.M., Alsanad, A.: An effective Palmprint recognition approach for visible and multispectral sensor images. Sensors. **18**(5), 1575 (2018)
19. Gumaei, A., Sammouda, R., Al-Salman, A.M.S., Alsanad, A.: Anti-spoofing cloud-based multi-spectral biometric identification system for enterprise security and privacy-preservation. J. Parall. Distrib. Comput. **124**, 27–40 (2019)
20. Hassan, M.M., Uddin, M.Z., Mohamed, A., Almogren, A.: A robust human activity recognition system using smartphone sensors and deep learning. Futur. Gener. Comput. Syst. **81**, 307–313 (2018)
21. Hossain, M. S., Hoda, M., Muhammad, G., Almogren, A., Alamri, A.: Cloud-supported framework for patients in post-stroke disability rehabilitation. Telematics Inform., (2017)
22. Li, W., Zhao, Y., Lu, S., Chen, D.: Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing. IEEE Commun. Mag. **53**(3), 89–97 (2015)
23. Lin, K., Wang, W., Bi, Y., Qiu, M., Hassan, M.M.: Human localization based on inertial sensors and fingerprints in the industrial internet of things. Comput. Netw. **101**, 113–126 (2016)
24. Morabito, R., Kjällman, J., Komu, M.: Hypervisors vs. lightweight virtualization: a performance comparison. 386-393
25. Rad, P., Boppana, R. V., Lama, P., Berman, G., Jamshidi, M.: Low-Latency Software Defined Network for High Performance Clouds. 486-491
26. Rahmani, A.M., Gia, T.N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., Liljeberg, P.: Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach. Futur. Gener. Comput. Syst. **78**, 641–658 (2018)
27. Raj, P., Raman, A.: Handbook of research on cloud and fog computing infrastructures for data science: IGI Global, (2018)
28. Sadooghi, I., Martin, J.H., Li, T., Brandstatter, K., Maheshwari, K., de Lacerda Ruivo, T.P.P., Garzoglio, G., Timm, S., Zhao, Y., Raicu, I.: Understanding the performance and potential of cloud computing for scientific applications. IEEE Trans. Cloud Comput. **5**(2), 358–371 (2017)
29. Satyanarayanan, M.: The emergence of edge computing. Computer. **50**(1), 30–39 (2017)
30. Ugulino, W., Cardador, D., Vega, K., Velloso, E., Milidiú, R., Fuks, H.: Wearable computing: accelerometers' data classification of body postures and movements. Advances in Artificial Intelligence-SBIA 2012, 52-61: Springer, (2012)

31. Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., Nikolopoulos, D. S., Challenges and Opportunities in Edge Computing. 20-26
32. Zamora-Izquierdo, M.A., Santa, J., Martínez, J.A., Martínez, V., Skarmeta, A.F.: Smart farming IoT platform based on edge and cloud computing. Biosyst. Eng. **177**, 4–17 (2019)

Affiliations

Mabrook Al-Rakhami [1,2] • Abdu Gumaei [3] • Mohammed Alsahli [2] • Mohammad Mehedi Hassan [1] • Atif Alamri [1] • Antonio Guerrieri [4] • Giancarlo Fortino [4,5]

Abdu Gumaei
abdugumaei@gmail.com

Mohammed Alsahli
mohmmad1024@gmail.com

Mohammad Mehedi Hassan
mmhassan@ksu.edu.sa

Atif Alamri
atif@ksu.edu.sa

Antonio Guerrieri
guerrieri@icar.cnr.it

Giancarlo Fortino
g.fortino@unical.it

1 Research Chair of Pervasive and Mobile Computing, Riyadh, Saudi Arabia

2 Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

3 Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia

4 National Research Council of Italy, Institute for High Performance Computing and Networking, Calabria, Italy

5 Department of Informatics, Modeling, Electronics and Systems (DIMES), University of Calabria, Calabria, Italy