



A Container-Based Edge Computing System for Smart Healthcare Applications

Tuan Le-Anh^(✉), Quan Ngo-Van, Phuong Vo-Huy, Dang Huynh-Van,
and Quan Le-Trung

UiTiOt Research Group, Department of Computer Networks, University of Information
Technology, Vietnam National University – Ho Chi Minh City, Ho Chi Minh City, Vietnam
tuanla.14@grad.uit.edu.vn, {16520981,16520975}@gm.uit.edu.vn,
{danghv,quanlt}@uit.edu.vn

Abstract. Edge computing is evolving how data are processed and analyzed from a large figure of various Internet of Things (IoT) devices globally. The rapid development of IoT, 5G, artificial intelligence (AI), and applications that require real-time computing capability steadily propel edge computing systems. In this paper, we propose a container-based edge computing system for smart healthcare applications. The smart care mobile and web-based applications aim to assist doctors or nurses with intelligent monitoring and caring for patients in the recovery phase in real-time. The proposed system's design takes advantage of edge computing's capabilities to timely deal with the patient's facial emotion detection and heart disease diagnosis AI applications and a robust cloud computing infrastructure to centralize the patient's data in the secure, scalable, and fault-tolerance database. Moving these AI applications to the edge outperforms cloud computing in processing time, energy efficiency, and bandwidth saving. Finally, implementing the AI applications on a lightweight container orchestration platform for management efficiency with high availability, scalability, and deployment automation.

Keywords: Edge computing · Cloud computing · IoT · Artificial intelligence · Container-based virtualization · Container orchestration platform

1 Introduction

In contemporary and prospective trends, IoT equipment has produced enormous data in our daily lives over a large-scale range of sectors, such as agriculture, transportation, energy, industrial, smart cities, and manufacturing. In 2016, the generated data was about 220 zettabytes and an estimate to strike over 250 zettabytes by 2021 [1]. The figures for appliances connected to the IoT application network can reach 34.2 billion devices worldwide by 2025 [2]. In hindsight, cloud computing has been a widespread technology trend and an indispensable component in the era of IoT with data analytics, AI, and big data [3]. Despite its overall prosperity, cloud computing is not a good fit for smart applications that require real-time data and low latency [4]. The advent of edge

computing aims to overcome these challenges. It does not intend to replace, however, cloud computing rather than its complementation. Edge computing expects to place computing close to where the data originates and the fact that it yields some benefits of energy efficiency, latency, and bandwidth reduction. The advent of edge computing aims to overcome these challenges [25]

According to the IEEE Computer Society [5], AI at the edge is one of the leading study tendencies in 2020. With the innovation of smart IoT hardware, modern AI algorithms, and the proliferation of 5G, machine learning (ML) applications have significantly transformed to the edge for more efficiency at a more economical cost. In current years, the confluence of edge computing and AI [6] continually drive smart IoT applications to adapt our modern life with a broad range of use cases, such as landslides monitoring [7], poultry real-time monitoring [8], patients and elderly tracking [9], cattle behavior analysis [10], climatic enclosure [11] and pivot irrigation [12]. Concerning the edge virtualization trends [13, 14], they have innovated applications at the edge with flexibility and efficiency in provisioning services. Compared to hypervisors, the advent of containerization reduces a significant overhead by implementing process isolation at the operating system level.

This research proposes a container-based edge computing system for smart healthcare applications with cloud-edge-based computing solutions, modern deep learning algorithms, docker-based, and lightweight cluster technologies.

This paper is structured as follows. Section 1 presents an overview of IoT data trends and the edge computing's benefits, the AI deployment at the edge for smart IoT applications, and virtualization for the edge. Afterward, Sect. 2 describes related works on IoTs smart healthcare applications, AI deployment at the edge, and the container-based virtualization. Next, Sect. 3 provides insights into the design and implementation of the proposed system. Then, Sect. 4 demonstrates the experimental results to evaluate the system performance, the hardware usage, and the accuracy of the AI applications. Finally, Sect. 5 ends this paper with conclusions and future work.

2 Related Work

This section presents a literature review of smart healthcare, the AI deployment at the edge, and the virtualization for edge computing.

2.1 Smart Healthcare

These days, patients suffering from chronic diseases [15] continue to grow, while nurses and doctors are limited. Therefore, the efficient use of these resources is a step towards smart healthcare systems. The rapid growth of next-generation technologies, including cloud computing, edge computing, artificial intelligence, and the proliferation of 5G technologies, have transformed smart healthcare platforms. In some recent research, notably [16], the authors introduced an edge computing-based smart healthcare framework for resource management. Besides, [27] proposed a Cloud of Things (CoT) for smart healthcare, and [17] presented an effective training scheme for the deep neural network in edge computing that enabled the Internet of Medical Things (IoMT) systems. Finally, [26] introduced next-generation technologies: challenges, vision, model, and future directions for smart healthcare applications.

2.2 Artificial Intelligence at Edge Computing

Nowadays, developing AI applications at the edge has been significantly growing around the world. The AI on edge contributes to shining intelligent applications with low latency, bandwidth saving, and low energy usage. However, this approach faces a variety of challenges, especially handling and power consumption. In the cloud, energy-intensive training occurs, and then deploy trained applications at the edge for prediction tasks. In edge computing, training takes place at the edge, leads to a higher demand for edge hardware computing capabilities. Besides, data storage and security pose a challenge as edge devices will hold most of the data and transfer only a small portion to the cloud, and security standard for IoT devices. As a result, leading technology companies have continually researched and developed to create more powerful computing hardware with lower energy consumption, also focus on optimizing AI algorithms. Besides, organizations concentrate on smart IoT hardware consist of BrainChip (Akida Neuro-morphic System-on-Chip), CEVA (NeuPro series), Google (Edge TPU), GreenWave (AI GAP8 Processor), Huawei (Ascend Chips), Intel (Xeon), Nvidia (Jetson TX2), Qualcomm (Vision Intelligence Platforms) and STMicroelectronics (STM32 microcontrollers). Besides, leading technology corporations building ecosystems create solutions for each industry and each specific scenario. These companies include Google (AI platform), Huawei (MindSpore), IBM (Watson), Intel (AI Developer Program), Microsoft (Azure), and enterprise IoT building blocks such as IoT Hub, Azure Databricks, ML Studio, and Power BI [18].

Some significant studies following this approach: Landslides monitoring [7], the author introduced AI with edge computing architecture on Kubernetes and Docker Platform, and others, which are monitoring poultry in real-time [8], cattle behavior analysis [10], and pivot irrigation, plant diseases, and pests identification [12]. Interestingly, the patients and elderly tracking [9], the authors presented a cloud-fog-IoT-based system that observes the health of the elderly and patients during the recovery phase that belongs to the healthcare sector that our research will introduce in this paper.

2.3 Edge Computing Virtualization

In recent years, applying virtualization technologies for applications at the edge has significantly increased [13, 14]. These virtualization technologies provide a platform for running various physical machine services, using either hypervisor or container techniques. However, container-based solutions and other emerging ones are challenging traditional hypervisor-based virtual machines. New technology trends are lightweight, more flexible, efficient, and efficient in the deployment of services. Recent studies focus on optimizing virtualization to increase its performance, like a bare-metal environment, also a comparison between virtualization technologies. According to [28], the authors found increasing container solutions on Platform as a Service (PaaS). In [19–23], the authors measured and compared the performance of KVM, Xen, and Linux Container (LXC), Docker, OSv, and compared the runtime of each environment with the bare metal server. In the article [24], the authors also explained comparing the performance of container-based technologies for the cloud with a diverse set of parameters. The results clearly showed that the performance of hypervisors had improved significantly

over the past few years. Container solutions outperformed, however, and their overhead was almost negligible. Despite the container solution’s flexibility and manageability, we may consider trading-off them with security matters.

3 Proposed System

This part will introduce the proposed system’s design and implementation. The paragraph structure is as follows: Sect. 3.1 presents the system architecture and design, and Sect. 3.2 describes performing the proposed paradigm.

3.1 System Architecture and Design

The proposed system bases on the cloud-edge-based computing paradigm. Figure 1 illustrates the high-level architecture of the proposed system model. The design incorporates three components: cloud data center, edge, and end-device.

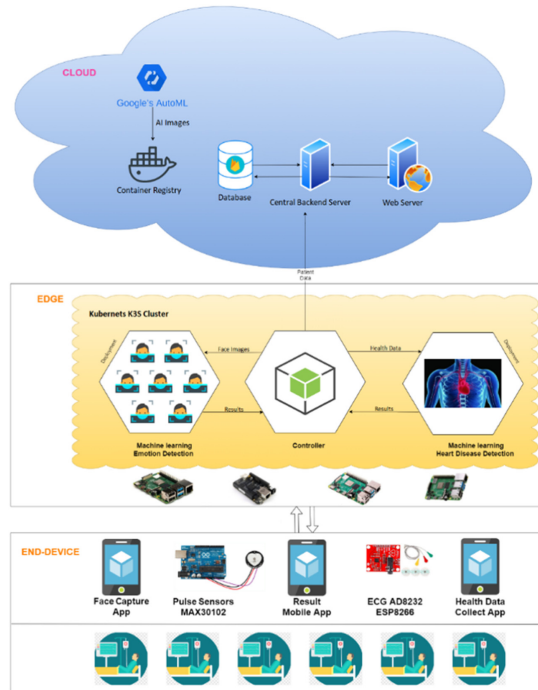


Fig. 1. The high-level design of the proposed system model.

- **Cloud data center:** The design embraces a private container registry to retain facial emotion detection and cardiovascular disease diagnosis AI images, databases, back-end, and web servers for the patient’s information and health data supervision. The

cloud backend server processes data and stores the results in the database. Then, it refreshes data from the database to the webserver. Concerning the database, it is a NoSQL, reserves the patient's information and health data, including name, gender, age, heart rate, electrocardiogram (ECG), facial emotion, heart disease diagnosis, and other reports. Besides, the web application assists doctors in observing the patient's health statistics.

- **Edge:** The edge structure incorporates modern edge devices such as Raspberry PI's or other vendor hardware to form a lightweight container orchestration platform to coordinate the edge backend server, facial emotion detection, and cardiovascular disease diagnosis AI applications. Depending on hardware constraints and requirements, the cluster could involve more masters and multiple worker nodes for scalability, fault-tolerant, and high availability. The backend server listens and deals with data from end devices or dispatch them to the corresponding AI applications regarding the workflow. It then returns the results to the mobile application and the cloud backend server that updates the database and refreshes the web application. The patient's facial emotion detection AI application interprets the base64-encoded images to predict the current patient's mood (e.g., happy, sad, neutral, fearful, disgusted, angry, or surprised). Likewise, the heart disease diagnosis AI function analyzes and predicts the patient's percentage at risk based on age, gender, height, weight, systolic and diastolic blood pressure, cholesterol, smoking, glucose, alcohol, and activity.
- **End-device:** IoT appliances and smart sensors collect and send data to the edge backend server for data processing and analysis. The mobile application reflects the results in real-time. Next, Fig. 2 exhibits the context design of observing the patient's health data in hospitals. In this context, doctors or nurses can track the patient's information and data health status via mobile and web applications.

3.2 System Implementation

This paragraph summarizes implementing state-of-art technologies for the proposed system:

- **Cloud data center:** We deployed a private Docker registry to hold the AI application's images based on Google's AutoML with deep learning algorithms for training datasets and creating AI models. When it comes to the web application, it consists of a Node.js framework-based backend server, a NoSQL database using Google Firebase, and Flutter framework-based web server written in Dart language.
- **Edge:** We performed a lightweight Kubernetes cluster (K3S) rather than Kubernetes (K8S) to coordinate Docker-based containers for the backend server and AI applications.
- **End-device:** We implemented IoT functions written in C/C++ language with Arduino ESP8266 hardware, MAX30102 pulse, and ECG AD8232 ECG sensors to collect and send the patient's health data to the edge backend server. Figure 3 shows the mobile and web user interface. These applications help doctors and nurses to observe the patient's health status in real-time.

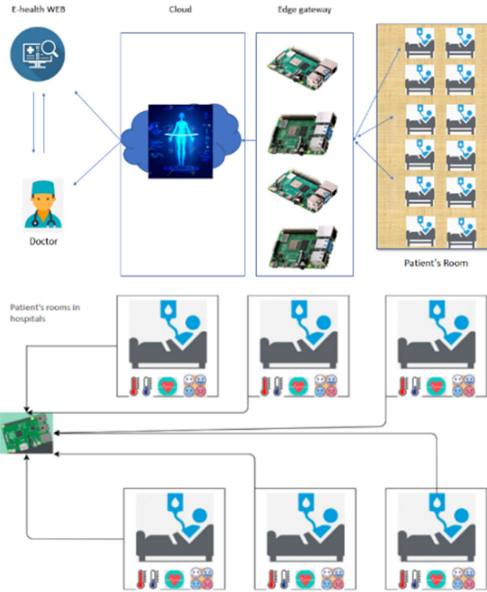


Fig. 2. The context design.

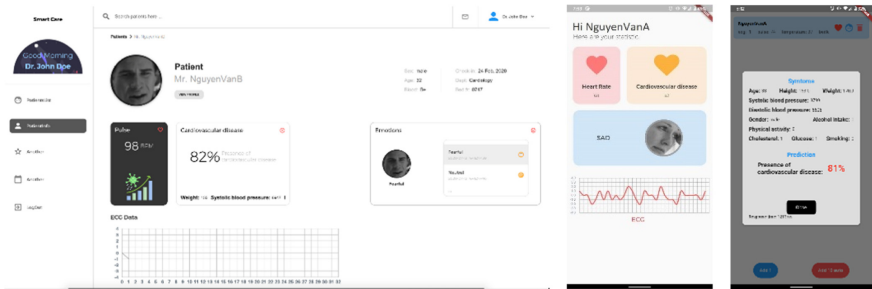


Fig. 3. The mobile and web application user interface.

4 Experimental Results

In this section, we demonstrate the experimental results. Section 4.1 measures and evaluates the proposed system’s performance compared to the traditional cloud-based computing paradigm. Next, Sect. 4.2 assesses the AI training model’s accuracy. Finally, Sect. 4.3 analyzes the hardware resource usage of the proposed system during the experiment.

4.1 Performance Comparison

This part compares the proposed system’s processing time (see Fig. 1) with the traditional cloud-based computing paradigm (see Fig. 4).

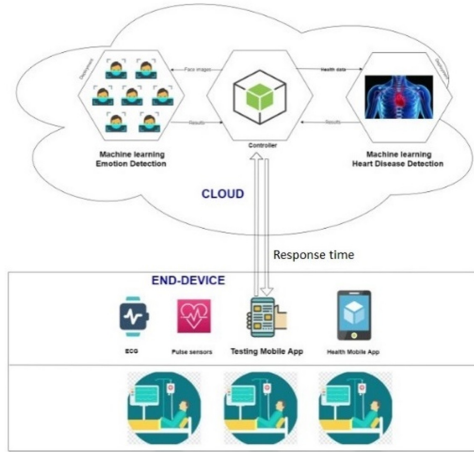


Fig. 4. The traditional cloud-based computing paradigm.

In the traditional cloud-based model, we deployed AI applications in the cloud. On the contrary, these AI functions we deployed at the edge of the proposed system. The specifications for the experiment were as follows (Table 1).

Table 1. Technical specifications

No.	Component	Specification
1	End devices	Mobile application (Dart, Android 11); Request: Facial emotion detection and heart disease diagnosis; Function: Send requests and measure response times
2	Edge server of the proposed system model (see Fig. 1)	Container-based nodes: Facial emotion detection, heart disease diagnosis, and backend server; Edge device: Raspberry Pi; Operating system: Raspbian; CPU: ARMv8 64-bit SoC @ 1.4 GHz; RAM: 1 GB; SD card: 32 GB
3	Cloud server of the cloud-based computing model (see Fig. 4)	Container-based nodes: Facial emotion detection, heart disease diagnosis, and backend server; Cloud provider: Google Cloud Platform; Operating system: Debian Buster; CPU: Intel Xeon; RAM: 2GB; Hard drive: 60 GB

This test was conducted on a network bandwidth of 96.64 Mbps download, and 92.98 Mbps upload speed. As a result, Fig. 5 illustrates the processing time of the AI applications of the proposed system was remarkably better.

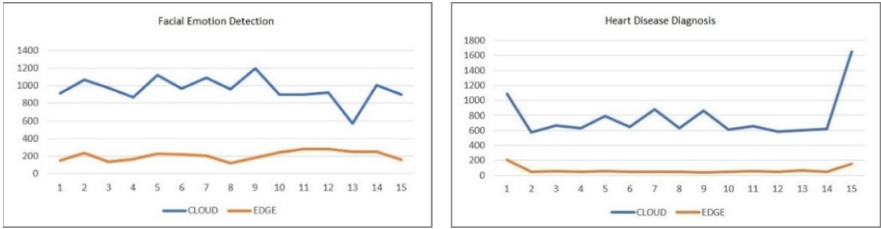


Fig. 5. Performance comparison: the proposed system vs. the traditional cloud-based paradigm

Table 2. Facial emotion detection’s response time

No	Cloud-based paradigm	Proposed system model	Relative difference
1	913 ms	153 ms	760 ms
2	1062 ms	235 ms	827 ms
3	971 ms	134 ms	837 ms
4	869 ms	169 ms	700 ms
5	1118 ms	226 ms	892 ms
6	968 ms	218 ms	750 ms
7	1090 ms	206 ms	884 ms
8	960 ms	123 ms	837 ms
9	1197 ms	178 ms	1019 ms
10	898 ms	245 ms	653 ms
11	896 ms	283 ms	613 ms
12	921 ms	278 ms	643 ms
13	567 ms	251 ms	316 ms
14	1002 ms	248 ms	754 ms
15	900 ms	155 ms	745 ms

Table 2 shows the results of the facial emotion detection AI application over 15 runs. We can find a noticeable difference in terms of processing time between the two models. The proposed system is typically 4–5 times faster on average than the traditional cloud-based paradigm.

Table 3. Heart disease diagnosis’s response time

No	Cloud-based paradigm	Proposed system model	Relative difference
1	1088 ms	205 ms	883 ms
2	577 ms	43 ms	534 ms
3	661 ms	51 ms	610 ms
4	625 ms	43 ms	582 ms
5	792 ms	53 ms	739 ms
6	647 ms	45 ms	602 ms
7	885 ms	47 ms	838 ms
8	629 ms	44 ms	585 ms
9	867 ms	33 ms	834 ms
10	614 ms	45 ms	569 ms
11	652 ms	51 ms	601 ms
12	586 ms	46 ms	540 ms
13	603 ms	63 ms	540 ms
14	624 ms	43 ms	581 ms
15	1648 ms	153 ms	1495 ms

Same as above, Table 3 displays the outcomes of the heart disease diagnosis AI application. As a result, the proposed system model performed ten times faster on average than the cloud-based computing model.

4.2 Accuracy Evaluation of Artificial Intelligence Applications

This section evaluates the AI model’s accuracy for the facial emotion detection and cardiovascular disease diagnosis applications we deployed at the edge. We conducted the experiment with the datasets “fer2013.csv”, “cardio_train.csv” provided by Kaggle with 80% of the datasets for training, 10% for PublicTest, and 10% for other purposes of AutoML. As a result, it shows in Fig. 6 below.

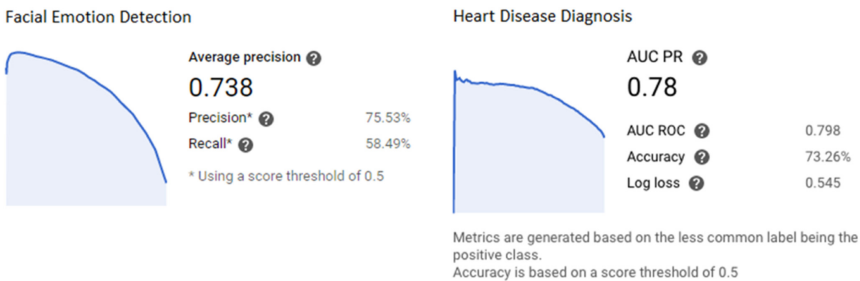


Fig. 6. The AI model’s evaluation

About the facial emotion detection, the AI model's precision was about 0.738 on average. More details, precision, and recall were approximately 75.53 and 58.49%. Regarding the cardiovascular disease diagnosis, the area under the precision-recall (PR) curve and receiver operating characteristic (ROC) curve were around 0.78 and 0.798. More, its accuracy was proximately 73.26%.

4.3 Hardware Resource Usage and Evaluation

This paragraph analyzes and evaluates the hardware resource consumption corresponding to the proposed system model during the experimental test, with the number of connections increased from 1 to 200. The requests included the patient's facial emotion detection and health data diagnosis. Concerning the testing methodology, we employed Postman to create a series of HTTP POST requests with base64-encoded images and the mobile application to simulate patients' figures corresponding to data from IoT sensors. Table 4 manifests the hardware specification for the test.

Table 4. Hardware specifications

No.	Role	Description
1	Master	Debian Buster, Core i5-8250U 1.80 GHz, Memory 2 GB, Hard drive 15 GB
2	Worker 1	Debian Buster, Core i5-8250U 1.80 GHz, Memory 2 GB, Hard drive 15 GB
3	Worker 2	Raspbian, ARM Cortex-A53 1.4 GHz, Memory 2 GB, SD card 32 GB

The following outcomes (see Fig. 7) indicate that increasing the number of requests demands more CPU resources, and memory gains correspond to the number of connections. For every 40 connections, memory usage requires about 200 MB. Besides, the bandwidth significantly increased, corresponding to the number of sockets or requests.

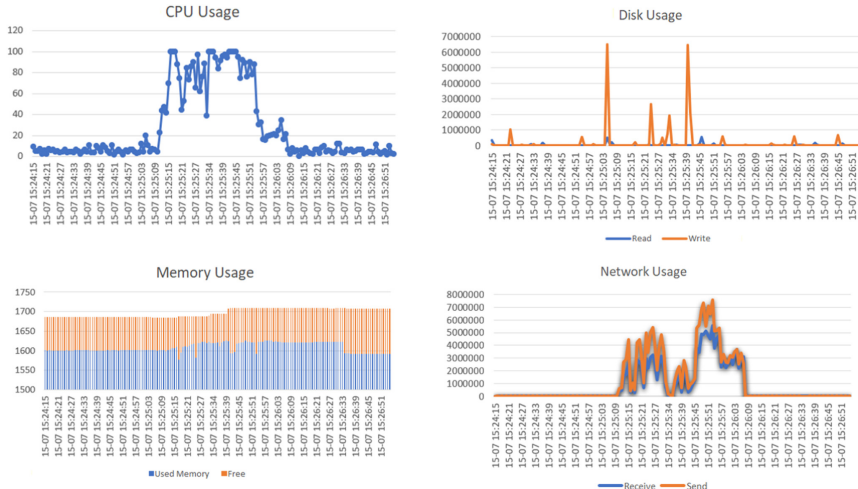


Fig. 7. Hardware resource statistics

5 Conclusion and Future Work

In this paper, we proposed the container-based edge computing system for smart healthcare applications. In short, we introduced related works in terms of this research and exhibited insights into system design and architecture. Next, we demonstrated the implementation of the proposed system. Finally, we analyzed the experimental results to evaluate the proposed model. Forthcoming work is open to researching new smart healthcare services to enhance the design and innovative technologies to secure data compression in transit.

References

1. Cisco Global Cloud Index: Forecast and Methodology 2016–2021 (white paper) Cisco, pp. 2016–2021 (2016)
2. State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating. <https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>. Accessed 09 Nov 2020
3. Verma, A.: The Relationship between IoT, Big Data, and Cloud Computing. Whizlabs Blog, 28 Nov 2018. <https://www.whizlabs.com/blog/relationship-between-iot-big-data-cloud-computing/>. Accessed 09 Nov 2020
4. Caprolu, M., Pietro, R.D., Lombardi, F., Raponi, S.: Edge computing perspectives: architectures, technologies, and open security issues. In: 2019 IEEE International Conference on Edge Computing (EDGE), pp. 116–123, July 2019. <https://doi.org/10.1109/EDGE.2019.00035>
5. Exclusive Content – IEEE Computer Society’s Tech Trends for 2020—IEEE Computer Society. <https://www.computer.org/publications/tech-news/trends/exclusive-content-ieee-cs-tech-trends-for-2020/>. Accessed 10 Nov 2020
6. Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., Zomaya, A.Y.: Edge intelligence: the confluence of edge computing and artificial intelligence. *IEEE Internet Things J.* 7(8), 7457–7469 (2020). <https://doi.org/10.1109/JIOT.2020.2984887>

7. Elmoulat, M., Debauche, O., Saïd, M., Mahmoudi, S., Manneback, P., Lebeau, F.: Edge computing and artificial intelligence for landslides monitoring. *Proc. Comput. Sci.* **177**, 480–487 (2020). <https://doi.org/10.1016/j.procs.2020.10.066>
8. Debauche, O., Saïd, M., Mahmoudi, S., Manneback, P., Bindelle, J., Lebeau, F.: Edge computing and artificial intelligence for real-time poultry monitoring. *Proc. Comput. Sci.* **175**, 534–541 (2020). <https://doi.org/10.1016/j.procs.2020.07.076>
9. Debauche, O., Saïd, M., Manneback, P., Assila, A.: Fog IoT for health: a new architecture for patients and elderly monitoring. *Proc. Comput. Sci.* **160**, 289–297 (2019). <https://doi.org/10.1016/j.procs.2019.11.087>
10. Debauche, O., Saïd, M., Mahmoudi, S., Manneback, P., Bindelle, J., Lebeau, F.: Edge Computing for Cattle Behavior Analysis, April 2020
11. Debauche, O., Mahmoudi, S., Mahmoudi, S.A., Manneback, P., Lebeau, F.: Edge computing and artificial intelligence semantically driven. Application to a climatic enclosure. *Proc. Comput. Sci.* **175**, 542–547 (2020). <https://doi.org/10.1016/j.procs.2020.07.077>
12. Debauche, O., Saïd, M., Elmoulat, M., Mahmoudi, S., Manneback, P., Lebeau, F.: Edge AI-IoT pivot irrigation, plant diseases, and pests identification. *Proc. Comput. Sci.* **177**, 40–48 (2020). <https://doi.org/10.1016/j.procs.2020.10.009>
13. Morabito, R.: Virtualization on internet of things edge devices with container technologies: a performance evaluation. *IEEE Access* **5**, 8835–8850 (2017). <https://doi.org/10.1109/ACCESS.2017.2704444>
14. Jaiswal, K., Sobhanayak, S., Turuk, A.K., Bibhudatta, S.L., Mohanta, B.K., Jena, D.: An IoT-cloud based smart healthcare monitoring system using container-based virtual environment in edge device. In: 2018 International Conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR), pp. 1–7, July 2018. <https://doi.org/10.1109/ICETIETR.2018.8529141>
15. Kim, J.: The effect of patient participation through physician’s resources on experience and wellbeing. *Sustainability* **10**, 2102 (2018). <https://doi.org/10.3390/su10062102>
16. Oueida, S., Kotb, Y., Aloqaily, M., Jararweh, Y., Baker, T.: An edge computing based smart healthcare framework for resource management. *Sensors* **18**, 4307 (2018). <https://doi.org/10.3390/s18124307>
17. Pustokhina, I.V., Pustokhin, D.A., Gupta, D., Khanna, A., Shankar, K., Nguyen, G.N.: An effective training scheme for deep neural network in edge computing enabled Internet of Medical Things (IoMT) systems. *IEEE Access* **8**, 107112–107123 (2020). <https://doi.org/10.1109/ACCESS.2020.3000322>
18. Koon, J.: How AI Changes the Future of Edge Computing. *EE Times Europe*, 24 June 2019. <https://www.eetimes.eu/how-ai-changes-the-future-of-edge-computing/>. Accessed 14 Nov 2020
19. Morabito, R., Kjällman, J., Komu, M.: Hypervisors vs. lightweight virtualization: a performance comparison. In: 2015 IEEE International Conference on Cloud Engineering, pp. 386–393, March 2015. <https://doi.org/10.1109/IC2E.2015.74>
20. Estrada, Z., Stephens, Z., Pham, C., Iyer, R.: A Performance Evaluation of Sequence Alignment Software in Virtualized Environments, pp. 730–737, May 2014. <https://doi.org/10.1109/CCGrid.2014.125>
21. Felter, W., Ferreira, A., Rajamony, R., Rubio, J.: An updated performance comparison of virtual machines and Linux containers, pp. 171–172, March 2015. <https://doi.org/10.1109/ISPASS.2015.7095802>
22. Kominos, C., Seyvet, N., Vandikas, K.: Bare-metal, virtual machines and containers in OpenStack, pp. 36–43, March 2017. <https://doi.org/10.1109/ICIN.2017.7899247>
23. Potdar, A.M., Dg, N., Kengond, S., Mulla, M.M.: Performance evaluation of Docker container and virtual machine. *Proc. Comput. Sci.* **171**, 1419–1428 (2020). <https://doi.org/10.1016/j.procs.2020.04.152>

24. Kozhirbayev, Z., Sinnott, R.O.: A performance comparison of container-based technologies for the cloud. *Future Gener. Comput. Syst.* **68**, 175–182 (2017). <https://doi.org/10.1016/j.future.2016.08.025>
25. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: vision and challenges. *IEEE Internet Things J.* **3**(5), 637–646 (2016). <https://doi.org/10.1109/JIOT.2016.2579198>
26. Tuli, S., et al.: Next-generation technologies for smart healthcare: challenges, vision, model, trends and future directions. *Internet Technol. Lett.* **3**(2), e145 (2020). <https://doi.org/10.1002/itl2.145>
27. Mahmoud, M.M.E., et al.: Enabling technologies on cloud of things for smart healthcare. *IEEE Access* **6**, 31950–31967 (2018). <https://doi.org/10.1109/ACCESS.2018.2845399>
28. Pahl, C.: Containerisation and the PaaS cloud. *IEEE Cloud Comput.* **2**, 24–31 (2015). <https://doi.org/10.1109/MCC.2015.51>