IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 October 2024, accepted 24 November 2024, date of publication 2 December 2024, date of current version 10 December 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3509732*

RESEARCH ARTICLE

# Battery Powered Edge Computing Acceleration for Smart Agriculture Applications: A Use Case for Resonant Ultrasound Spectroscopy

**ŽILVINAS NAKUTIS[1], PAULIUS TERVYDIS[1], LINAS SVILAINIS[1], (Senior Member, IEEE), AND ALBERTO RODRÍGUEZ-MARTÍNEZ[2], (Senior Member, IEEE)**

[1]Department of Electronics Engineering, Kaunas University of Technology, 51368 Kaunas, Lithuania
[2]Communications Engineering Department, UMH, 03202 Elche, Spain

Corresponding author: Paulius Tervydis (paulius.tervydis@ktu.lt)

This work was supported by European Regional Development Fund with the Research Council of Lithuania (LMTLT) under Project 01.2.2-LMT-K-718-03-0026.

**ABSTRACT** Edge computing, using battery-powered devices, presents a viable solution for the real-time data processing for smart agriculture solutions. This paper explores the application of edge computing acceleration for smart agriculture, focusing on the use case of resonant ultrasound spectroscopy (RUS) for grape leaf analysis and monitoring. A methodology for estimating the utilization and performance of both edge and cloud data processing devices is proposed here. The effectiveness of edge and cloud data processing systems is analyzed in terms of data processing waiting time, cost, and battery life of edge devices as a function of intensity of data processing requests and load distribution in various scenarios. The analysis considers such factors as data processing capabilities, equipment cost, and energy consumption to provide insights into the optimal deployment of edge and cloud resources for smart agriculture applications, considering critical waiting and battery time criteria.

**INDEX TERMS** Smart agriculture, edge, cloud, data processing, monitoring.

## I. INTRODUCTION

Smart and precision agriculture was propelled by the rapid advancement of IoT and cloud computing technologies [1], [2], [3], [4]. Integration of IoT and cloud computing strategies can enhance agricultural practices [5]. Development of new architectures for data acquisition, transmission and processing is aimed for the improvement of quality of experience [5], [6], [7]. Main trends and challenges regarding the adoption of cloud-based IoT applications in the agricultural sector for the benefit of sustainability in climate-smart agriculture can be found in [7].

Initially, IoT nodes were dedicated for collection of data, but processing and storage were carried out by a cloud platform [6], [8]. Many of applications are sufficient with this type of architecture. However, a bottleneck occurs in case of a more intense data flow from primary sources, especially if unprocessed data is transmitted to cloud [8], [9]. Also, delays are increased in case of the large number of data processing requests for high computation demand algorithms of machine learning or artificial intelligence [9], [10].

Then a concept of edge computing was introduced. It was driven by the need for high computational performance to handle the complexity of data processing algorithms such as filtering, data aggregation, and machine learning [11], [12], [13], [14], eventually advancing to deep learning [1]. The so-called edge layer is composed of communication infrastructure equipment, which has some extra computational capacity suitable for cloud structure offloading [6], [8], [15], [16], [17], [18]. Edge devices are typically geographically closer to the data source and end user. Data networks enable to deliver the acquired data directly to the cloud [19], [20], [21], [22]. Edge devices may act as intermediate nodes for processing and forwarding data to cloud servers [23]. Processing is done in cloud and obtained results can be sent back for the inspection to the user. The balanced

The associate editor coordinating the review of this manuscript and approving it for publication was Irfan Ahmed.

relationship between edge and cloud computing is essential for achieving sustainability in smart agriculture. Edge computing handles immediate, resource-efficient tasks, like monitoring soil moisture or crop health, while cloud computing facilitates broader, more complex analyses, such as predicting weather patterns or optimizing resource allocation across the entire range of farming operations. This synergy between edge and cloud enables a more sustainable IoT ecosystem, reducing energy consumption, minimizing environmental impact, and supporting precision agriculture.

However, if processing results are required immediately, then the latency of the response becomes a critical disadvantage. Since application service providers may need to serve thousands of user requests simultaneously, ensuring the low response latency becomes a significant challenge [10], [24], [25]. The straightforward approach, increase of dedicated cloud resources (network capacity, number of server processor cores, memory), is not economically justifiable. Response time can be improved if resources available on the edge are exploited. Edge servers reduce communication delays by pre-processing sensor data locally, allowing for more efficient field monitoring and decision-making in irrigation and fertilization [2]. In large-scale operations, like subsoiling and tillage process monitoring, edge devices significantly enhance processing efficiency by collecting and analyzing tractor-mounted sensor data [26]. Smart phones, tablets, embedded systems, single board computers, portable FPGA accelerators [4], [27], [28], [29], etc. are candidates for such implementation, but these are a battery powered edge devices.

While numerous studies analyze energy efficiency [19], [30], [31], [32], [33], network latency of IoT [20], [34], [35], [36], balancing between edge and cloud computing [8], [15], [34], [37], [38], [39], [40], [41], [42], [43] or cost [9], [44], [45], [46] there is a lack of methodology that effectively considers a multiple critical factors simultaneously – such as data processing request rates, processing times in both edge and cloud environments, costs, edge battery life, and load distribution.

The particular task considered is when measurements are taken in the field [21], [22] and final result is obtained only after complex data processing [1], [3], [4], [47], [48], [49] that takes time. In such case data transmission latency is negligible compared to processing time [24], [41]. If practical implementation is considered, configuration must be optimal in a sense of response latency, cloud resources cost and edge energy capacity. Another problem is that number of parameters and their combinations considered is wide. Then, event-driven and network-driven simulation takes significant time and resources. Analytical model based on queuing theory, described in [50] was used to reduce the optimization time.

Proposed methodology fills this void by providing a systematic approach that integrates multiple parameters, enabling a more holistic view of resource allocation. Furthermore, the use of analytical models ensures that the computation is fast, allowing for real-time estimations and adjustments. It may enhance the efficiency of IoT systems and contribute to more sustainable operational practices in data-intensive environments.

An analysis of three use case scenarios is presented: i) estimation of the optimal number of edge and cloud data processing devices for a given tasks intensity, ii) analysis how the intensity of data processing requests may be increased if additional edge devices were added to the system, iii) analysis how load balancing determines the system performance parameters. The performance criteria are processing latency, battery discharge and cloud cost.

## II. EVALUATION METHODOLOGY

The proposed methodology utilizes quantitative research methods, which follows the positivist paradigm [51]. Study involves experimental measurements (e.g., processing time in IoT edge devices and cloud servers) and analytical modeling of scenarios (e.g., estimation of waiting time in queues as a function of data processing request rate). These require the collection of quantitative data, which can be statistically analyzed, fitting into quantitative research.

Quantitative research focuses on objective measurement and the analysis of numerical data. By experimenting and modeling various data processing request rates and load distribution between edge devices and cloud servers, the study aims to generalize findings based on observed data.

The use of analytical mathematical models to analyze how factors like load distribution affect performance places the research within empirical quantitative methods, which emphasize numerical and statistical evaluation [51].

### A. SYSTEM MODEL

A cloud-connected data acquisition and processing system considered in this research is shown in Fig. 1.

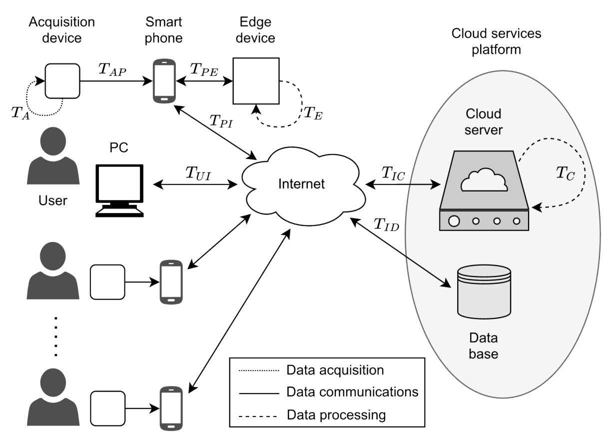


**FIGURE 1. Cloud-connected data acquisition and processing system model.**

A set of acquisition devices are operated in the field by a user aiming to collect the measurement data from plants. The local wireless link to the smart phone is implemented using

Bluetooth communication. The smart phone ensures wide area network connectivity to the centralized cloud services for data storage and processing. The concept of edge layer denotes that some computational resources are provided in a vicinity of the user and has a fast direct link to the smart phone. In addition to users operating in the field cloud services provide connection from computers connected to the internet for the users authorized to access the stored and processed data at the time of demand. Locations of data acquisition, data communication and data processing are indicated in the Fig. 1, too.

Criteria for optimization of cloud-connected systems in most of the cases include: i) latency, ii) power consumption, iii) implementation cost or profit.

The definition of latency and its acceptable limit should be considered in respect to purpose of the system. In some agriculture related applications an instant response is not demanded, and it is acceptable that information obtained from the collected data is delivered within hours or even days. In this case centralized cloud-based system suites very well. However, situations when a user is sampling data in the field and needs to verify the measurement results to decide if to collect more data or to check whether the data looks reasonable are probable as well. An example of such a system explored later in this research is the resonant ultrasound spectroscopy (RUS) for grape leaf status monitoring. In this application a user is expected to walk in the vineyard with a hand-held sensor and using a smart phone deliver raw ultrasound signal records to the edge devices or cloud servers for processing. Each data processing device may serve multiple users.

The data processing is considered as time consuming and involves an inverse solution technique [21], [47]. Because the cloud server needs to process data simultaneously acquired by many users the latency of processing and sending results back may become annoyingly long. In this case, edge devices can be used to accelerate the computation time or to decrease the load on cloud server.

Both sensors and edge devices powered by a battery need to be optimized in terms of power consumption seeking to endure the operation before replacing or recharging the battery. The higher performance is achievable at the cost of higher power consumption. In certain applications as for example grapes leaf status monitoring [21], [22], [47], at least full workday operation without battery recharging in the field is preferable.

Finally, the overall cost of the system is composed of the cost of edge device (higher performance and better energy efficiency most probably will result in higher cost), the number of edge devices in the system, and cloud server resources.

### B. SYSTEM LATENCY

The latency describes a period between the task trigger and delivery of the results to the user.

Considering the system model shown in Fig. 1 the latency can be described as

$$T_L = T_A + T_{AP} + T_{PT} T_W + T_{DT}, \tag{1}$$

where $T_A$ is sample acquisition duration, $T_{AP}$ is the latency of data transmission to the smart phone, $T_{PT}$ is the latency of data transmission to the processing unit, $T_W$ is the waiting due to processing latency and queueing in the processing unit, $T_{DT}$ is the latency of processed data transmission to the user end equipment.

$$T_W = T_P + T_Q, \tag{2}$$

where $T_P$ is processing latency and $T_Q$ is queueing time.

The sample acquisition duration $T_A$ is application dependent and includes delays of all hardware like sensors and analog-to-digital converters, excitation signals generation, recording samples to the buffering memory. $T_A$ also accounts for the number of samples (length of the signal) necessary to deliver to the processing unit to start block type processing.

The latency of data transmission to the processing unit $T_{PT}$ is dependent on the location of the processing unit: i) if processing is scheduled in the smart phone, then $T_{PT}$ can be neglected, ii) if processing is scheduled in a separate edge device, then $T_{PT}$ has to account for the data delivery to the edge device and the processed results delivery back to the smart phone, iii) of processing is performed in the cloud server, then $T_{PT}$ has to account for the latency of data delivery to the cloud server, which is responsible for data processing.

The processing latency $T_P$ is dependent on the computational load of the algorithms used and the performance of either smart phone, edge device, or cloud server.

The time of waiting in queue $T_Q$ is modeled as described in the next chapter and basically characterize the time a user task for data processing is postponed due to the non-available computational resources on unit dedicated to processing (for example edge device or cloud server).

The latency of the processing results delivery to the user $T_{DT}$ describes the time it takes to deliver processing results from processing unit (smart phone, edge device or cloud server) to the user's end device. The end device most often will be user's smart phone or computer.

### C. PROCESSING AND WAITING TIME

This chapter introduces a model used to estimate the waiting time in the system's queues of processing units, to evaluate the utilization of edge devices and cloud servers. The model will be utilized to evaluate the influence of edge devices on the system performance criteria [35], [52].

The model of requests queueing and their distribution among edge devices and cloud servers is shown in Fig. 2.

Users who perform measurements create the flow of data processing requests, which intensity is λ requests per h. The generated flow of data processing requests can be distributed among edge and cloud data processing devices by a load balancer according to a particular probability or priority. Each such data processing device can be estimated by a queueing model that evaluates waiting time in queue and processing time.

The total intensity of data processing requests is:

$$\lambda = N_A \cdot \lambda_A, \tag{3}$$

where $N_A$ is the number of all users' requests for data acquisition and processing, $\lambda_A$ is the average intensity of single user's requests per time unit:

$$\lambda_A = 1/T_A, \tag{4}$$

where $T_A$ is the mean time between requests of a single user.

The intensities of user requests forwarded to the $i$-th edge device and $j$-th cloud server are respectively:

$$\lambda_E^{(i)} = \lambda \cdot P_E \cdot p_E^{(i)}, \tag{5}$$

$$\lambda_C^{(j)} = \lambda \cdot P_C \cdot p_C^{(j)}, \tag{6}$$

where $P_E$ is probability of data processing request forwarding to an edge device, and $P_C$ is the probability of data processing request forwarding to cloud server.

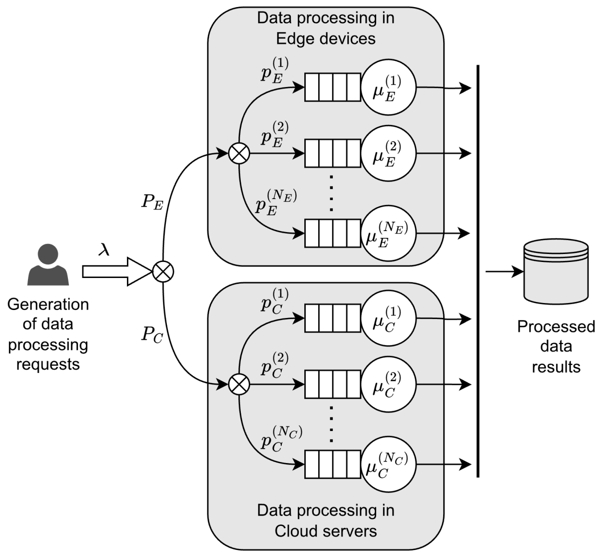


**FIGURE 2. Queueing model of data processing system.**

In some implementations the probabilities may be assumed as load distribution priorities, which are used by a load balancer. It is assumed that $P_E + P_C = 1$. Here $p_E^{(i)}$ is the probability of request forwarding to the $i$-th edge device, where $i = 1,2..\ N_E$ and $\sum_i p_E^{(i)} = 1$, and $p_C^{(j)}$ is the probability of request forwarding to the $j$- th cloud server, where $j = 1,2..\ N_C$ and $\sum_j p_C^{(j)} = 1$. $N_E$ and $N_C$ are correspondingly numbers of edge devices and cloud servers in the system.

The utilization (load) of edge devices and cloud servers are respectively:

$$\rho_E^{(i)} = \lambda_E^{(i)} / \mu_E^{(i)}, \tag{7}$$

$$\rho_C^{(j)} = \lambda_C^{(j)} / \mu_C^{(j)}, \tag{8}$$

where intensities of data processing (or average number of requests that can be processed per time unit) in the $i$-th edge device and in the $j$-th cloud server are:

$$\mu_E^{(i)} = 1/T_{PE}^{(i)}, \tag{9}$$

$$\mu_C^{(j)} = 1/T_{PC}^{(j)}, \tag{10}$$

where $T_{PE}^{(i)}$ is data processing duration in the $i$-th edge device and $T_{PC}^{(j)}$ is data processing duration in the $j$-th cloud server.

Waiting times in queues depend on distribution of time intervals between requests and the distribution of data processing times. For example, if requests of data processing arrive according to Poison process (time intervals between requests are distributed according exponential distribution), and data processing time is constant, then waiting times in queues of $i$ - th edge device and $j$-th cloud server can be expressed according to the analytical M/D/1 queueing model:

$$T_{QE}^{(i)} = \frac{\rho_E^{(i)}}{2 \cdot \mu_E^{(i)} \cdot (1 - \rho_E^{(i)})}, \tag{11}$$

$$T_{QC}^{(j)} = \frac{\rho_C^{(j)}}{2 \cdot \mu_C^{(j)} \cdot (1 - \rho_C^{(j)})}. \tag{12}$$

The average waiting time of data pin the $i$-th edge device and the $j$-th cloud server can be expressed:

$$T_E^{(i)} = T_{PE}^{(i)} + T_{QE}^{(i)}, \tag{13}$$

$$T_C^{(j)} = T_{PC}^{(j)} + T_{QC}^{(j)}. \tag{14}$$

If data processing request arrival is not Poison or data processing time is not constant, then another analytical (M/M/1, GI/G/1 or other) or simulation models should be used.

In further analysis M/D/1 queueing model is used, also it is assumed that all $i$-th edge devices are the same and all $j$-th cloud servers are the same in the System, and the load between them is evenly distributed ($p_E^{(i)} = 1/N_E$ and $p_C^{(j)} = 1/N_C$).

### D. CRITICAL ARRIVAL RATE OF DATA PROCESSING REQUESTS

Waiting time in the system depend on arrival rate of data processing requests and on the number of data processing devices in the system. In order, to ensure that the mean waiting time of data processing is less than desired critical waiting time $T_{Wcr}$, it is necessary that either the rate of data processing requests is below particular critical value, or the numbers of data processing devices are sufficient. Therefore, if the number of processing devices ($N_E$, $N_C$) and critical waiting time $T_{Wcr}$ are given, then the critical rates of data processing request

$$\lambda_{Ecr} = N_E \cdot \frac{2 \cdot \mu_E \cdot (T_{Wcr} \cdot \mu_E - 1)}{2 \cdot T_{Wcr} \cdot \mu_E - 1}, \tag{15}$$

$$\lambda_{Ccr} = N_C \cdot \frac{2 \cdot \mu_C \cdot (T_{Wcr} \cdot \mu_C - 1)}{2 \cdot T_{Wcr} \cdot \mu_C - 1}, \tag{16}$$

where $\mu_E$ and $\mu_C$ are service rates of a single edge and single cloud server, considering that all the edge devices are the same, and all the cloud servers are the same.

Using the same principle the critical (minimum) number of data processing devices can be estimated:

$$N_{Ecr} = \frac{\lambda_E \cdot (2 \cdot T_{Wcr} \cdot \mu_E - 1)}{2 \cdot \mu_E \cdot (T_{Wcr} \cdot \mu_E - 1)}, \tag{17}$$

$$N_{Ccr} = \frac{\lambda_C \cdot (2 \cdot T_{Wcr} \cdot \mu_C - 1)}{2 \cdot \mu_C \cdot (T_{Wcr} \cdot \mu_C - 1)}. \tag{18}$$

The mean waiting time in the system $(T_E, T_C) \leq T_{Wcr}$, when $\lambda_E \leq \lambda_{Ecr}$ and $\lambda_C \leq \lambda_{Ccr}$, or when $N_E \geq N_{Ecr}$ and $N_C \geq N_{Ccr}$.

### E. SYSTEM COST, REVENUE AND PROFIT

Various cost models for edge and cloud-based systems are explored in publications like [45].

To achieve the specified average system latency maintaining the minimal overall system cost the optimal number of edge devices ($N_E$) and optimal number of cloud servers ($N_C$) should be chosen. If the number of edge devices and cloud resources are too small, then the collection and processing of the required number of measurements will take longer than the desired $T_{Wcr}$. If the number of edge devices and cloud servers is larger than needed to collect and process the data per desired $T_{Wcr}$ time, then some of edge devices and cloud servers will be used inefficiently or not at all, even though they are paid for.

There are many variables that determine the exact monthly cost of a data processing system, but for the sake of simplicity, let's assume that the total cost per hour of the data processing system is $C_S = C_{ES} + C_{CS}$, where $C_{ES}$ and $C_{CS}$ are costs of data processing in edge and cloud subsystems:

$$C_{ES} = N_E \cdot C_E, \tag{19}$$

$$C_{CS} = \begin{cases} N_C \cdot C_{Cd}, & \text{dedicated cloud capacity;} \\ N_C \cdot C_{Co} \cdot \rho_C, & \text{on} - \text{demand cloud capacity;} \end{cases} \tag{20}$$

where $C_E$ – cost per hour for an edge device, $C_{Cd}$ – cost per hour for cloud server, if "dedicated capacity" plan is applied, $C_{Co}$ – pay per hour for cloud server, if the "on-demand capacity" plan is used, and $\rho_C$ – utilization of cloud server.

If such data processing system is managed by a service provider and the revenue per single data processing is $r_p$, then total revenue per hour in edge and cloud processing subsystems:

$$R_{ES} = \lambda_E \cdot P_{Ea} \cdot r_p, \tag{21}$$

$$R_{CS} = \lambda_C \cdot P_{Ca} \cdot r_p, \tag{22}$$

where $P_{Ea}$ and $P_{Ca}$ – ratio (or probability) of acceptable data processing requests. The ratios are calculated by

$$P_{Ea} = \begin{cases} \frac{\lambda_E - \lambda_{Ecr}}{\lambda_E}, & \lambda_{Ecr} \leq \lambda_E, \\ 0, & \lambda_{Ecr} > \lambda_E, \end{cases} \tag{23}$$

$$P_{Ca} = \begin{cases} \frac{\lambda_C - \lambda_{Ccr}}{\lambda_C}, & \lambda_{Ccr} \leq \lambda_C, \\ 0, & \lambda_{Ccr} > \lambda_C, \end{cases} \tag{24}$$

and the total revenue is $R_S = R_{ES} + R_{CS}$.

Then the profit per hour in edge and cloud processing subsystems are

$$P_{ES} = R_{ES} - C_{ES}, \tag{25}$$

$$P_{CS} = R_{CS} - C_{CS}, \tag{26}$$

and the total profit is $P_S = P_{ES} + P_{CS}$.

These formulas can be used to grade cost and profit criteria alongside to other system features influencing the system configuration matching user's expectations.

### F. BATTERY USAGE TIME

Battery-powered devices of the system (smart phones, edge devices) are consuming energy [46] and discharging when operated in a field. The obvious expectation of a user is that a recharging period is less than a working day. The capacity of the battery used, and energy efficiency of a device limits the number of times $N_M$ measurement samples can be acquired and number of times $N_P$ data processing is completed. A way to estimate these parameters the following expressions can be used:

$$N_M = \frac{C_{B\%}}{C_{1M\%}}, \tag{27}$$

$$N_P = \frac{C_{B\%}}{C_{1P\%}}, \tag{28}$$

where $C_{B\%}$ is percentage of battery charge level, $C_{1M\%}$ is percentage of battery discharge after completing one sample acquisition, and $C_{1P\%}$ is percentage of battery discharge after one data processing request. Higher value of the $N_P$ indicates larger capabilities of the processing unit to serve more requests until the need for recharge.

Considering probabilistic nature of edge devices load (7) the battery usage time for processing:

$$T_{EB} = \frac{N_p \cdot T_{PE}}{\rho_E}. \tag{29}$$

The $T_{EB}$ accounts for the energy efficiency from a user's perspective in terms of average edge device's battery discharge time. The more intensively a device is used for data acquisition and processing (higher $\rho_E$) the smaller is $T_{EB}$.

## III. CASE STUDY: RESONANT ULTRASOUND SPECTROSCOPY FOR GRAPE LEAF MONITORING

Further on, a use case of grape plants physiological status monitoring using RUS, which is described in [21], [47], and [48] was considered. It is based on the thickness resonance within a leaf when ultrasound is transmitted through. Two measurements were carried out: with and without the leaf in ultrasound path between two transducers. The obtained transmission function was approximated by fitting the mathematical model for layered structure [48], [53]. The result of such inverse solution allows to extract the mechanical properties of the leaf which later can be related to the physiological status of the plant.

### A. SYSTEM SPECIFICATION

The grape plants physiological status monitoring system is illustrated in Fig. 3.

Measurements were taken on the grape leaves using the RUS sensors. To obtain the results, the measured signals must be processed. In such a way the flow of data processing requests λ is created.

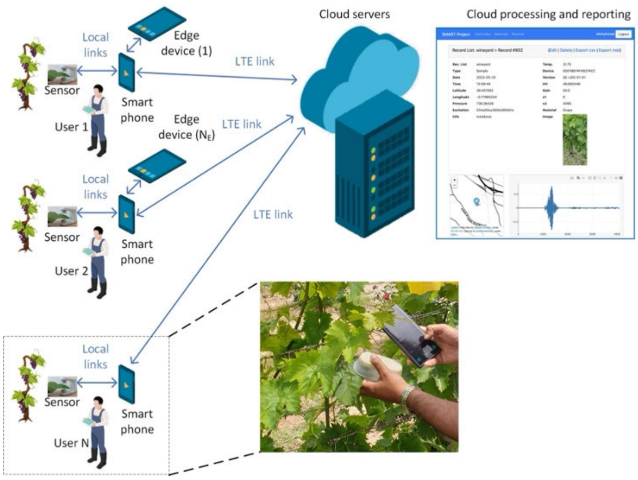


**FIGURE 3.** Grape leaf monitoring system.

In the analyzed case, the measured data can be processed either by an edge device or by a cloud server. Therefore, the measured data from the RUS sensor is transferred to a smartphone, which itself can be used as an edge device or as a load balancer that sends the data for processing to a dedicated battery powered edge device in the field or to the cloud server over LTE network.

### B. DATA PROCESSING UNITS

A comparison of real case data processing durations achieved by different computation devices is presented. As described in [21] and [47], the data collected by the acquisition device is processed using particle swarm optimization (PSO) algorithm [49]. The fit function for the PSO algorithm comprises of FFT, IFFT, complex transfer function calculation using 2K blocks of reference and sample signals [22]. The processing algorithm was implemented in Python language and exercised on the edge devices and cloud server featuring parameters shown in Table 1 (the monthly cost is calculated considering full payment in installments over 24 months).

**TABLE 1.** Specifications of data processing equipment.

| Equipment | Specifications | Monthly cost, Eur |
|---|---|---|
| Raspberry Pi 3 Model B | CPU: Quad Core 1.2GHz Broadcom BCM2837 64bit CPU<br>Memory: 1GB RAM<br>BCM43438 WLAN and Bluetooth Low Energy (BLE) on board | 3.70 |
| Samsung A13 5G | CPU: Octa core (2.3 GHz, Quad Core + 1.8 GHz, Quad core)<br>GPU: MediaTek Helio P3 56<br>Memory: 64 GB 4 GB RAM<br>Battery: 5000 mAh | 7.54 |
| Samsung GalaxyTab S6 Lite | CPU: Octa-core (4x2.3 GHz Cortex-A73 & 4x1.7 GHz Cortex-A53)<br>GPU: Mali-G72 MP3<br>Memory: 64GB 4GB RAM<br>Battery: 7040 mAh (up to 15 hours) | 12.99 |
| Lenovo Legion Y540 | Processor: 9th Generation Intel® Core™ i7-9750H<br>GPU: NVIDIA® GeForce® GTX 1650<br>Memory: 16 GB DDR4 2666 MHz<br>Battery: Up to 5 hours, 52.5 Wh | 35.63 |
| Oracle Cloud VM | Shape: VM.Standard.A1.Flex<br>OCPU count: 2<br>Network bandwidth (Gbps): 2<br>Memory (GB): 12 | 29.52 ("Dedicated capacity" pricing) |
| MacBook Air M1 | Processor: M1<br>Memory (GB): 8<br>Battery: 49.9 Wh | 41.62 |

A popular single board computer Raspberry Pi 3, state-of-the-art smart phones (can be used in the role of edge layer accelerators), and laptop computer were considered as Edge devices.

In such case, the PSO processing time is much larger if compared to data acquisition time and data transmission delay. For example, it was already determined in [22] that the mean delay between measurement triggering and data delivery to gateway (smartphone) is 5.24 s, the mean data transmission (over LTE network) latency from smartphone to the cloud server is 1.21 s, and the mean latency of processed data download is 0.05 s. Therefore, in further analysis of total waiting time estimation only the PSO processing time in cloud servers and edge devices will be considered.

Experimentally measured processing durations are summarized in Table 2 together with operation modes of corresponding devices.

### C. SCENARIO-BASED ANALYSIS OF PERFORMANCE CRITERIA

The selection of the optimal number of cloud servers and edge devices will be investigated seeking to achieve the specified waiting time and edge devices working time on battery, by applying framework described in Chapter 3. The framework was implemented in our developed Python package CloudEdgeAssetOptimizer [50], which GUI window is presented in Fig. 4. The analytical mathematic models implemented in the framework allow to estimate system parameters much faster than event-driven simulation models.

**TABLE 2.** Characteristics of data processing equipment.

| Device | Type | Battery-powered mode | Battery Saver Mode | Processing time $T_{PE}^{(i)}$ or $T_{PC}^{(i)}$, s | Percentage of battery discharge per PSO, $C_{1P\%}$, % | Energy efficiency index, $N_P$ |
|---|---|---|---|---|---|---|
| MacBook Air M1 | Edge | Yes | Off | 28.03 | 0.25 | 400 |
| MacBook Air M1 | Edge | Yes | On | 45.21 | 0.15 | 666.67 |
| Oracle Cloud VM | Cloud | No | - | 106.94 | - | - |
| Lenovo Legion Y540 | Edge | No | - | 129.32 | - | - |
| Samsung A13 5G | Edge | Yes | Off | 200.44 | 0.55 | 181.82 |
| Lenovo Legion Y540 | Edge | Yes | On | 217.63 | 2.65 | 37.74 |
| Samsung Galaxy Tab S6 Lite | Edge | Yes | Off | 259.2 | 1 | 100 |
| Samsung A13 5G | Edge | Yes | On | 274.9 | 0.5 | 200 |
| Samsung Galaxy Tab S6 Lite | Edge | Yes | On | 351.86 | 0.85 | 117.65 |
| Raspberry Pi 3 Model B | Edge | No | - | 1091.31 | - | - |

The parameter which mainly influences user's quality of experience is a waiting time between a request's trigger moment and the results delivery to the interface of user's personal device. The cost of the system is obviously a parameter which both system users and provider will seek to minimize.

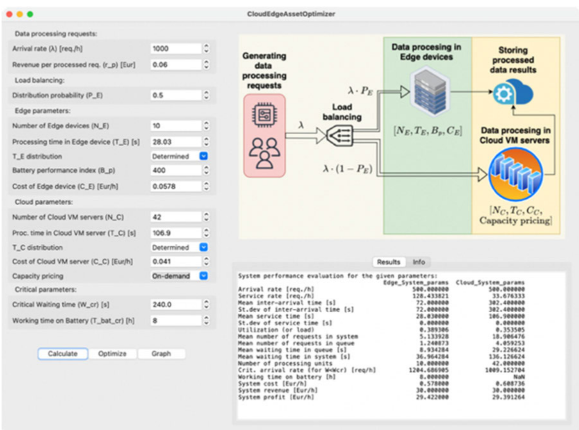


**FIGURE 4.** CloudEdgeAssetOptimizer's GUI window.

If a handheld device is discharging and demands recharging more frequently than once per working day it makes operating such a system inconvenient. The number of edge devices aimed at acceleration of acquired data processing and number if cloud servers influence waiting time and cost of the system. In overall, criteria representing the use case system performance are quantified with different units and are mutually related. Therefore, selecting implementation scenario, which meets the expectations the best and has other criteria within some boundaries, is not straightforward.

Three scenarios (A, B and C) were analyzed to estimate how a battery-powered edge devices may accelerate data processing for the given use case. To compare how the results depend on the edge device's performance parameters two types of devices were selected: i) Samsung A13 5G phones and ii) MacBook Air M1 notebooks as edge devices. The results for both edge device types are given in Table 3 and Table 4, while the cloud server parameters were the same. The font formatting in the tables is different for the parameters: in **bold** font are the parameters that were fixed, in *italic* font – the parameters that were varied, and in regular font – the estimated parameters.

The required number of edge devices and cloud servers was estimated by applying the following criteria: the mean waiting time for data processing in edge devices ($T_E$) and cloud servers ($T_C$) must be $\leq T_{Wcr} = 240$ s, and the working time on battery ($T_{EB}$) for an edge device must be $\geq 8$ h. The total profit that a potential data processing service provider may yield, also was considered. The revenue per processed request is $r_p = 0.06$ Eur, the cloud server cost $C_C = 0.041$ Eur/h ("dedicated capacity pricing"), the cost of type 1 edge device is $C_E = 0.0105$ Eur/h, and the cost of type 2 edge device is $C_E = 0.0578$ Eur/h.

### 1) SCENARIO A

This scenario was used to estimate what is the optimal (minimum) number of edge ($N_E$) and cloud ($N_C$) data processing devices as a function of $P_E$, when $\lambda = 1000$ req./h.

If $P_E = 1$, then $N_C = 0$ and all the data is processed by the edge devices. In such a case, the optimal number of edge devices of type 1 is 197 and of type 2 is 20.

On the other hand, if $P_E = 0$, then all the data is processed by the cloud servers. That's why in both cases the optimal number of cloud servers is equal to 42.

If $P_E \in (0,1)$, then a particular number of $N_E$ and $N_C$ must be used. The results reveal that the exact numbers (given in Table 3 and Table 4 ) depend on the level of load of edge devices and cloud servers, that determines that the $T_E$ and $T_C$ are $\leq T_{Wcr} = 240$ s, and $T_{EB} \geq 8$ h.

### 2) SCENARIO B

Consider case when there is a data processing system that initially has 42 cloud servers. This scenario was used to estimate how the intensity of data processing request may be increased if additional edge devices were added to the system.

The results depend on the edge device's performance. That's why, the smaller number of type 2 edge devices may process more requests, than the type 1. For example, if the data processing system has 100 type 1 edge devices and 42 cloud servers, then it may process 1500 req./h. If the

**TABLE 3. System configuration and performance parameters: edge devices are samsung A13 smart phones, $T_{PE}$ = 200.44 s, oracle VM cloud servers $T_{PC}$ = 106.9 s, $T_{Wcr}$ = 240 s, $T_{EB}$ ≥8 h.**

| Scenario | λ, req./h | $P_E$ | $N_E$ | $N_C$ | $\rho_E$ | $\rho_C$ | $T_E$, s | $T_C$, s | $T_{EB}$, h | $P_{ES}$, Eur/h | $P_{CS}$, Eur/h | $P_S$, Eur/h |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | **1000** | *1* | 197 | 0 | 0.28 | 0 | 239.92 | - | 35.82 | 47.93 | 0 | 47.93 |
| | **1000** | *0.75* | 148 | 11 | 0.28 | 0.67 | 239.83 | 217.85 | 35.88 | 35.94 | 12.05 | 47.99 |
| | **1000** | *0.5* | 99 | 21 | 0.28 | 0.70 | 239.65 | 235.88 | 36.00 | 23.96 | 24.14 | 48.10 |
| | **1000** | *0.25* | 50 | 32 | 0.28 | 0.70 | 239.10 | 229.25 | 36.36 | 11.98 | 36.19 | 48.17 |
| | **1000** | *0* | 0 | 42 | 0 | 0.70 | - | 235.88 | NaN | 0 | 48.78 | 48.78 |
| B | 1100 | 0.09 | *20* | **42** | 0.28 | 0.70 | 238.57 | 236.32 | 36.73 | 4.74 | 48.33 | 53.07 |
| | 1200 | 0.16 | *40* | **42** | 0.27 | 0.71 | 236.99 | 239.47 | 37.88 | 9.18 | 48.67 | 57.85 |
| | 1300 | 0.23 | *60* | **42** | 0.28 | 0.71 | 238.93 | 236.32 | 36.49 | 14.32 | 48.33 | 62.65 |
| | 1400 | 0.29 | *80* | **42** | 0.28 | 0.70 | 239.91 | 233.28 | 35.82 | 19.46 | 47.98 | 67.44 |
| | 1500 | 0.33 | *100* | **42** | 0.28 | 0.71 | 238.57 | 238.11 | 36.73 | 23.70 | 48.52 | 72.22 |
| C | **1000** | *0.10* | **100** | **42** | 0.06 | 0.64 | 206.35 | 200.42 | 181.82 | 3.95 | 43.28 | 47.23 |
| | **1000** | *0.20* | **100** | **42** | 0.11 | 0.56 | 212.99 | 176.50 | 90.91 | 8.95 | 38.28 | 47.23 |
| | **1000** | *0.30* | **100** | **42** | 0.17 | 0.49 | 220.54 | 159.27 | 60.60 | 13.95 | 33.28 | 47.23 |
| | **1000** | *0.40* | **100** | **42** | 0.22 | 0.42 | 229.16 | 146.28 | 45.46 | 18.95 | 28.28 | 47.23 |
| | **1000** | *0.50* | **100** | **42** | 0.28 | 0.35 | 239.10 | 136.13 | 36.36 | 23.95 | 23.28 | 47.23 |

system has 25 type 2 edge devices, then it may process 2240 req./h. In case of type 1 edge devices the λ is limited due to $T_{Wcr}$ requirements, and in case of type 2 edge devices – due to $T_{EB}$ requirements. The greater the λ that may be processed, the bigger is the potential profit, that data processing service provider could gain.

### 3) SCENARIO C

Consider case when there is a data processing system that initially has 42 cloud servers and 100 type 1 edge devices in 1 case, and 10 type 2 edge devices in 2 case. This scenario is used to estimate how load balancing ($P_E$) determines the system performance parameters. This scenario reveals how the availability of edge devices and load balancing may be used to decrease the waiting times $T_E$ and $T_C$. By increasing the $P_E$, the load of edge devices $\rho_E$ is increased, while the load of cloud servers is decreased $\rho_C$. For example, by increasing the $P_E$ from 0.1 to 0.4, it is possible to decrease $T_C$ from 200.42 s to 136.13 s. This is limited to the fact that either the waiting time $T_E \leq 240$ s criteria (in case of type 1 edge devices) or $T_{EB} \geq 8$ h criteria (in case of type 2 edge devices) are reached.

Spider diagrams (or radar diagrams) are well suited to plot a multidimensional data [54], which represents many aspects of a solution or a scenario on a two-dimensional plane. They are used to draw the combination of difficult to compare features in a condense manner to facilitate decision or choice. Spider diagram can be a useful tool when discussing customer's needs. Indeed, they represent a more professional way to elaborate on mutual relationship between system parameters, cost, and intensity of anticipated requests in contrast to explaining in a style "the more you invest the more you get".

Though spider diagrams in Fig. 5 and Fig. 6 are only a static set of possible implementation scenario a calculator enabling user to manipulate parameters could become an instrument facilitating the decision acceptance.

By examining spider diagrams, demanding for a certain waiting time $T_{Wcr}$ is related to either more computational resources on cloud or utilizing more edge devices.

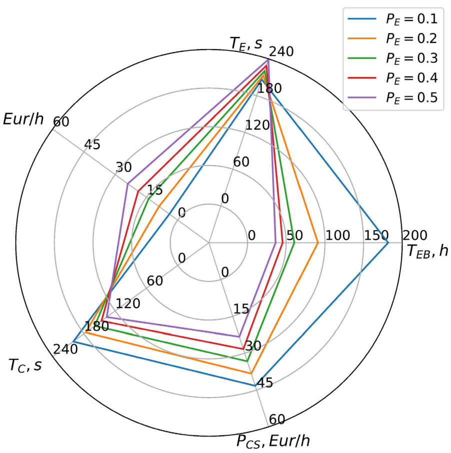


**FIGURE 5. Scenario C performance parameters as a function of $P_E$, when samsung A13 smart phones $T_{PE}$ = 200.4 s are used as edge devices oracle VM cloud servers $T_{PC}$ = 106.9 s are used, and the following criteria are applied $T_{Wcr}$ = 240 s, TEB ≤ 8 h, when λ = 1000 req./h.**

A service provider of the system may face the task of optimizing the cost, waiting time and edge device's discharge related metrics at some expected intensity of user requests. Since all these parameters are mutually related, the visualization of typical scenarios by spider diagrams can be a helpful reference point for making coarse decisions regarding the number and type of edge devices, cloud servers and the associated system costs. Fine tuning of parameters can be done afterwards by adjusting one parameter, for example the number of edge devices to meet the $T_{Wcr}$ given a certain budget.

The calculation results show how the usage of edge devices influences the performance parameters. If the monthly cost

**TABLE 4.** System configuration and performance parameters: edge devices are MacBook Air M1 notebooks, $T_{PE}$ = 28.03 s, oracle VM cloud servers $T_{PC}$ = 106.9 s, $T_{Wcr}$ = 240 s, $T_{EB}$ ≥8 h.

| Scenario | λ, req./h | $P_E$ | $N_E$ | $N_C$ | $\rho_E$ | $\rho_C$ | $T_E$, s | $T_C$, s | $T_{EB}$, h | $P_{ES}$, Eur/h | $P_{CS}$, Eur/h | $P_S$, Eur/h |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | **1000** | *1* | 20 | 0 | 0.39 | 0 | 36.96 | - | 8.00 | 48.84 | 0 | 48.84 |
| | **1000** | *0.75* | 15 | 11 | 0.39 | 0.67 | 36.96 | 217.85 | 8.00 | 36.63 | 12.05 | 48.68 |
| | **1000** | *0.50* | 10 | 21 | 0.39 | 0.71 | 36.96 | 235.88 | 8.00 | 24.42 | 24.14 | 48.56 |
| | **1000** | *0.25* | 5 | 32 | 0.39 | 0.70 | 36.96 | 229.25 | 8.00 | 12.21 | 36.19 | 48.40 |
| | **1000** | *0* | 0 | 42 | 0 | 0.70 | - | 235.88 | NaN | 0 | 48.28 | 48.28 |
| B | 1250 | 0.2 | *5* | **42** | 0.39 | 0.70 | 36.39 | 235.88 | 8.00 | 14.71 | 58.28 | 72.99 |
| | 1500 | 0.33 | *10* | **42** | 0.39 | 0.71 | 36.82 | 238.11 | 8.08 | 29.12 | 58.58 | 87.70 |
| | 1740 | 0.43 | *15* | **42** | 0.39 | 0.70 | 36.93 | 232.34 | 8.02 | 44.03 | 57.79 | 101.82 |
| | 2000 | 0.50 | *20* | **42** | 0.39 | 0.70 | 36.96 | 235.88 | 8.00 | 58.84 | 58.28 | 117.12 |
| | 2240 | 0.55 | *25* | **42** | 0.38 | 0.71 | 36.76 | 239.47 | 8.12 | 72.46 | 58.76 | 131.22 |
| C | **1000** | *0.10* | **10** | **42** | 0.08 | 0.64 | 29.21 | 200.41 | 40 | 5.42 | 52.28 | 57.70 |
| | **1000** | *0.20* | **10** | **42** | 0.16 | 0.57 | 30.61 | 176.50 | 20 | 11.42 | 46.28 | 57.70 |
| | **1000** | *0.30* | **10** | **42** | 0.23 | 0.50 | 32.30 | 159.27 | 13.33 | 17.42 | 40.28 | 57.70 |
| | **1000** | *0.40* | **10** | **42** | 0.31 | 0.42 | 34.37 | 146.28 | 10.00 | 23.42 | 34.28 | 57.70 |
| | **1000** | *0.50* | **10** | **42** | 0.39 | 0.35 | 36.96 | 136.13 | 8 | 29.42 | 28.28 | 57.70 |

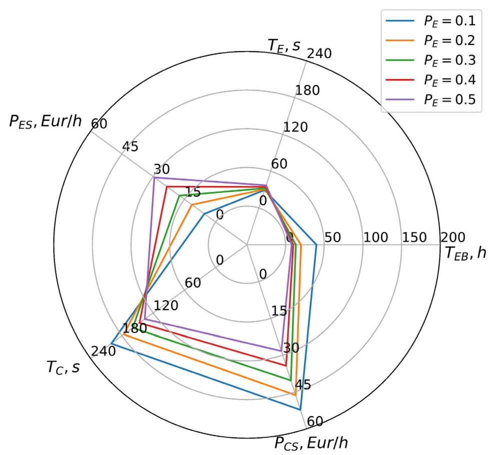


**FIGURE 6.** Scenario C performance parameters as a function of $P_E$, when MacBook Air M1 notebooks $T_{PE}$ = 28.03 s are used as edge devices. oracle VM cloud servers $T_{PC}$ = 106.9 s are used, and the following criteria are applied $T_{Wcr}$ = 240 s, TEB ≤ 8 h, when λ = 1000 req./h.

of a cloud server is greater than for an edge device, then the more the edge devices are used, the less monthly cost will be, because the load on cloud servers is reduced, but the $T_{EB}$ are increased, because the data processing time on the edge devices may be longer.

## IV. CONCLUSION

A framework based on an analytical load-balancing model was proposed to manage the broad diversity of application-dependent factors and multidimensional criteria. The framework was implemented in Python, and its application for the grape status monitoring using resonant ultrasound spectroscopy is demonstrated.

It was demonstrated that it is a valuable decision-making tool and can be effectively used for load balancing. It can optimize system cost and waiting time by considering key system implementation parameters such as the number and cost of edge devices, the expense of cloud computing resources, the energy efficiency of edge devices, the expected intensity of user requests, and the priorities of requests' distribution to either edge devices or cloud servers. The upper limit for total waiting time and minimum edge device operation from battery time were always primary priority.

At fixed requests rate (scenario A) it was found that mean load per processing device was the defining criteria for optimal number of edge devices and profit. This scenario can be used to estimate the required cloud and edge capacity.

At fixed cloud capacity (scenario B) it was found that increasing requests rate can be handled with larger profit if edge devices are more extensively used.

At fixed cloud and edge capacity and fixed requests rate (scenario C) it was found that profit does not depend on cloud-edge use ratio. The edge device operation from battery time can be increased if cloud load portion is increased. The utilization of edge devices for data processing acceleration may decrease system waiting time in comparison to scenarios where only cloud-hosted servers are used. Situations where both edge devices and cloud servers are used for data processing were investigated.

The main task was to determine the optimal number of edge devices and cloud servers concerning the total system cost, assuming a designated waiting time limit is the primary objective. Additionally, energy efficiency of edge devices has been included as a criterion, as users naturally expect edge devices not to deplete their battery until the end of a typical workday. For instance, if processing requests density 1000 req./h is equally distributed for processing to edge and cloud, then it is necessary to have 10 edge devices and 21 cloud servers, when edge processing time 28.03 s and cloud processing time 106.9 s, and if total waiting time upper limit is 240 s, and battery work before depletion minimum 8 h.

However, the current approach has limitations, such as assuming static data processing request rates and overlooking the impact of data transmission delays. Edge devices deployed in environments with highly variable workloads may experience periods of overload or underutilization. Although present analysis focuses on processing latency, the delay in data transmission over networks, especially in low-bandwidth or high-latency environments might also affect the total waiting time. In certain use cases, this transmission delay could outweigh the benefits of reduced processing latency at the edge.

Future work can focus on improving the dynamic load estimation, incorporating network latency, and refining power consumption estimation techniques. A more elaborate models for power consumption estimation can be implemented, that also consider a real-time workload variability, which would provide a more precise assessment of battery usage in edge devices. Additionally, more complex cloud pricing scenarios can be explored, allowing for better optimization of resource allocation costs in dynamic cloud environments. These enhancements would further refine the system's ability to manage computational and financial resources efficiently. Expanding the framework's applicability across another IoT use cases, like environment monitoring or healthcare, also is possible.

## REFERENCES

[1] N. Chamara, G. Bai, and Y. Ge, "AICropCAM: Deploying classification, segmentation, detection, and counting deep-learning models for crop monitoring on the edge," *Comput. Electron. Agricult.*, vol. 215, Dec. 2023, Art. no. 108420.
[2] S. Trilles, J. Torres-Sospedra, Ó. Belmonte, F. J. Zarazaga-Soria, A. González-Pérez, and J. Huerta, "Development of an open sensorized platform in a smart agriculture context: A vineyard support system for monitoring mildew disease," *Sustain. Comput., Informat. Syst.*, vol. 28, Dec. 2020, Art. no. 100309.
[3] W. Shen, Y. Sun, Y. Zhang, X. Fu, H. Hou, S. Kou, and Y. Zhang, "Automatic recognition method of cow ruminating behaviour based on edge computing," *Comput. Electron. Agricult.*, vol. 191, Dec. 2021, Art. no. 106495.
[4] Y. Zhang, J. Yu, Y. Chen, W. Yang, W. Zhang, and Y. He, "Real-time strawberry detection using deep neural networks on embedded system (rtsd-net): An edge AI application," *Comput. Electron. Agricult.*, vol. 192, Jan. 2022, Art. no. 106586.
[5] A. Poonia, D. Lakshmi, T. Garg, and G. Vishnuvarthanan, "A comprehensive study on smart farming for transforming agriculture through cloud and IoT," in *Convergence of Cloud Computing, AI, and Agricultural Science*, Hershey, PA, USA: IGI Global, 2023, pp. 67–99.
[6] X. Zhang, Z. Cao, and W. Dong, "Overview of edge computing in the agricultural Internet of Things: Key technologies, applications, challenges," *IEEE Access*, vol. 8, pp. 141748–141761, 2020.
[7] E. G. Symeonaki, K. G. Arvanitis, and D. D. Piromalis, "Cloud computing for IoT applications in climate-smart agriculture: A review on the trends and challenges toward sustainability," in *Innovative Approaches and Applications for Sustainable Rural Development*. Cham, Switzerland: Springer, 2019, pp. 147–167.
[8] M. H. Ghasemi, O. Lucía, and S. Lucia, "Computing in the blink of an eye: Current possibilities for edge computing and hardware-agnostic programming," *IEEE Access*, vol. 8, pp. 41626–41636, 2020.
[9] D. Loghin, L. Ramapantulu, and Y. M. Teo, "On understanding time, energy and cost performance of Wimpy heterogeneous systems for edge computing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 1–8.
[10] H. Hao, W. Ding, and W. Zhang, "Time-continuous computing offloading algorithm with user fairness guarantee," *J. Netw. Comput. Appl.*, vol. 223, Mar. 2024, Art. no. 103826.
[11] V. Sze, "Designing hardware for machine learning: The important role played by circuit designers," *IEEE Solid-State Circuits Mag.*, vol. 9, no. 4, pp. 46–54, Nov. 2017.
[12] A. Choudhury, M. Ghose, A. Islam, and Yogita, "Machine learning-based computation offloading in multi-access edge computing: A survey," *J. Syst. Archit.*, vol. 148, Mar. 2024, Art. no. 103090.
[13] M. Merenda, C. Porcaro, and D. Iero, "Edge machine learning for AI-enabled IoT devices: A review," *Sensors*, vol. 20, no. 9, p. 2533, Apr. 2020.
[14] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4921–4934, Jun. 2019.
[15] Y. Kalyani and R. Collier, "A systematic survey on the role of cloud, fog, and edge computing combination in smart agriculture," *Sensors*, vol. 21, no. 17, p. 5922, Sep. 2021.
[16] M. M. Alqarni, A. Cherif, and E. Alkayal, "A survey of computational offloading in cloud/edge-based architectures: Strategies, optimization models and challenges," *KSII Trans. Internet Inf. Syst.*, vol. 15, pp. 952–973, Mar. 2021.
[17] I. Ullah, H.-K. Lim, Y.-J. Seok, and Y.-H. Han, "Optimizing task offloading and resource allocation in edge-cloud networks: A DRL approach," *J. Cloud Comput.*, vol. 12, no. 1, p. 112, Jul. 2023.
[18] B. Wang, C. Wang, W. Huang, Y. Song, and X. Qin, "A survey and taxonomy on task offloading for edge-cloud computing," *IEEE Access*, vol. 8, pp. 186080–186101, 2020.
[19] M. Rizzi, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, "Synchronization uncertainty versus power efficiency in LoRaWAN networks," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 4, pp. 1101–1111, Apr. 2019.
[20] F. Moreno-Cruz, V. Toral-López, A. Escobar-Molero, V. U. Ruíz, A. Rivadeneyra, and D. P. Morales, "TreNch: Ultra-low power wireless communication protocol for IoT and energy harvesting," *Sensors*, vol. 20, no. 21, p. 6156, Oct. 2020.
[21] A. Aleksandrovas, L. Svilainis, A. Chaziachmetovas, V. Eidukynas, Z. Nakutis, P. Tervydis, and T. G. Alvarez-Arenas, "Miniature air coupled ultrasound data acquisition system for field application of resonance spectroscopy," in *Proc. IEEE Int. Ultrason. Symp. (IUS)*, Oct. 2022, pp. 1–3.
[22] L. Svilainis, Z. Nakutis, P. Tervydis, A. Chaziachmetovas, and A. Aleksandrovas, "Resource tradeoff analysis of plant physiological status sensor with cloud connectivity," in *Proc. 11th IEEE Int. Conf. Intell. Data Acquisition Adv. Comput. Syst., Technol. Appl. (IDAACS)*, Sep. 2021, pp. 241–246.
[23] M. J. O'Grady, D. Langton, and G. M. P. O'Hare, "Edge computing: A tractable model for smart agriculture?" *Artif. Intell. Agricult.*, vol. 3, pp. 42–51, Sep. 2019.
[24] P. Ferrari, E. Sisinni, A. Depari, A. Flammini, S. Rinaldi, P. Bellagente, and M. Pasetti, "On the performance of cloud services and databases for industrial IoT scalable applications," *Electronics*, vol. 9, no. 9, p. 1435, Sep. 2020.
[25] G. Plastiras, M. Terzi, C. Kyrkou, and T. Theocharidcs, "Edge intelligence: Challenges and opportunities of near-sensor machine learning applications," in *Proc. IEEE 29th Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP)*, Jul. 2018, pp. 1–7.
[26] Y. Yin, C. Zhao, Y. Zhang, J. Chen, C. Luo, P. Wang, L. Chen, and Z. Meng, "Development and application of subsoiling monitoring system based on edge computing using IoT architecture," *Comput. Electron. Agricult.*, vol. 198, Jul. 2022, Art. no. 106976.
[27] A. P. Fournaris, C. Alexakos, C. Anagnostopoulos, C. Koulamas, and A. Kalogeras, "Introducing hardware-based intelligence and reconfigurability on industrial IoT edge nodes," *IEEE Des. Test.*, vol. 36, no. 4, pp. 15–23, Aug. 2019.
[28] S. Branco, A. G. Ferreira, and J. Cabral, "Machine learning in resource-scarce embedded systems, FPGAs, and end-devices: A survey," *Electronics*, vol. 8, no. 11, p. 1289, Nov. 2019.
[29] X. Zhang, A. Ramachandran, C. Zhuge, D. He, W. Zuo, Z. Cheng, K. Rupnow, and D. Chen, "Machine learning on FPGAs to face the IoT revolution," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2017, pp. 819–826.
[30] Q. Wang, S. Guo, J. Liu, and Y. Yang, "Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing," *Sustain. Comput., Informat. Syst.*, vol. 21, pp. 154–164, Mar. 2019.
[31] R. Aldmour, S. Yousef, T. Baker, and E. Benkhelifa, "An approach for offloading in mobile cloud computing to optimize power consumption and processing time," *Sustain. Comput., Informat. Syst.*, vol. 31, Sep. 2021, Art. no. 100562.

[32] Z. Li, N. Zhu, D. Wu, H. Wang, and R. Wang, "Energy-efficient mobile edge computing under delay constraints," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 2, pp. 776–786, Jun. 2022.

[33] D. Alsadie, "Efficient task offloading strategy for energy-constrained edge computing environments: A hybrid optimization approach," *IEEE Access*, vol. 12, pp. 85089–85102, 2024.

[34] Y. Zhang, X. Chen, Y. Chen, Z. Li, and J. Huang, "Cost efficient scheduling for delay-sensitive tasks in edge computing system," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jul. 2018, pp. 73–80.

[35] W. Bai, J. Zhu, S. Huang, and H. Zhang, "A queue waiting cost-aware control model for large scale heterogeneous cloud datacenter," *IEEE Trans. Cloud Comput.*, vol. 10, no. 2, pp. 849–862, Apr. 2022.

[36] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Power-delay tradeoff in multi-user mobile-edge computing systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[37] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102781.

[38] J. Guo, C. Li, Y. Chen, and Y. Luo, "On-demand resource provision based on load estimation and service expenditure in edge cloud environment," *J. Netw. Comput. Appl.*, vol. 151, Feb. 2020, Art. no. 102506.

[39] A. Singh, G. S. Aujla, and R. S. Bali, "Container-based load balancing for energy efficiency in software-defined edge computing environment," *Sustain. Comput., Informat. Syst.*, vol. 30, Jun. 2021, Art. no. 100463.

[40] G. Tefera, K. She, M. Shelke, and A. Ahmed, "Decentralized adaptive resource-aware computation offloading & caching for multi-access edge computing networks," *Sustain. Computing: Informat. Syst.*, vol. 30, Jun. 2021, Art. no. 100555.

[41] K. Cao, S. Hu, Y. Shi, A. W. Colombo, S. Karnouskos, and X. Li, "A survey on edge and edge-cloud computing assisted cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7806–7819, Nov. 2021.

[42] S. Ding and D. Lin, "Dynamic task allocation for cost-efficient edge cloud computing," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Nov. 2020, pp. 218–225.

[43] L. Zeng, C. Yang, P. Huang, Z. Zhou, S. Yu, and X. Chen, "GNN at the edge: Cost-efficient graph neural network processing over distributed edge servers," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 3, pp. 720–739, Mar. 2023.

[44] C. Li, J. Liu, B. Lu, and Y. Luo, "Cost-aware automatic scaling and workload-aware replica management for edge-cloud environment," *J. Netw. Comput. Appl.*, vol. 180, Apr. 2021, Art. no. 103017.

[45] Z. Chen, Q. Ma, L. Gao, and X. Chen, "Edgeconomics: Price competition and selfish computation offloading in multi-server edge computing networks," in *Proc. 19th Int. Symp. Model. Optim. Mobile, Ad hoc, Wireless Netw. (WiOpt)*, Oct. 2021, pp. 1–8.

[46] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, "Estimation of energy consumption in machine learning," *J. Parallel Distrib. Comput.*, vol. 134, pp. 75–88, Dec. 2019.

[47] L. Svilainis, Ž. Nakutis, V. Eidukynas, D. Liaukonis, A. Aleksandrovas, A. Chaziachmetovas, and T. G. Alvarez-Arenas, "Air-coupled ultrasound resonant spectroscopy sensitivity study in plant leaf measurements," in *Proc. IEEE Sensors*, Oct. 2021, pp. 1–4.

[48] D. Sancho-Knapik, T. G. Alvarez-Arenas, J. J. Peguero-Pina, and E. Gil-Pelegrin, "Air-coupled broadband ultrasonic spectroscopy as a new non-invasive and non-contact method for the determination of leaf water status," *J. Experim. Botany*, vol. 61, no. 5, pp. 1385–1391, Feb. 2010.

[49] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, 1995, pp. 1–7.

[50] P. Tervydis, L. Svilainis, Ž. Nakutis, and A. Rodríguez-Martínez, "CloudEdgeAssetOptimizer: Tool to optimize the cloud-edge computing network resources at given requirements of processing delay, battery capacity and cost," *SoftwareX*, vol. 26, May 2024, Art. no. 101714.

[51] S. Rahi, "Research design and methods: A systematic review of research paradigms, sampling issues and instruments development," *Int. J. Econ. Manage. Sci.*, vol. 6, no. 2, pp. 1–5, Jun. 2017.

[52] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1405–1418, Jun. 2020.

[53] M. D. Fariñas and T. E. G. Álvarez-Arenas, "Ultrasonic assessment of the elastic functional design of component tissues of phormium tenax leaves," *J. Mech. Behav. Biomed. Mater.*, vol. 39, pp. 304–315, Nov. 2014.

[54] X. Deng, Z. Zhang, X. Hu, J. Li, S. Li, C. Su, S. Du, and L. Shi, "Estimation of photosynthetic parameters from hyperspectral images using optimal deep learning architecture," *Comput. Electron. Agricult.*, vol. 216, Jan. 2024, Art. no. 108540.


**ŽILVINAS NAKUTIS** received the Ph.D. degree from Kaunas University of Technology, Lithuania, in 2001.

Since 2015, he has been a full-time Professor with the Department of Electronics Engineering (Measurement Technologies Group), Kaunas University of Technology. His current research interests include the Internet of Things applications for precision agriculture, low power embedded systems, measurement of electrical energy and power, and fault detection in smart metering systems.

**PAULIUS TERVYDIS** received the Ph.D. degree from Kaunas University of Technology, Lithuania, in 2007.

He is currently an Associate Professor with the Department of Electronics Engineering, Kaunas University of Technology. His research interests include telecommunications engineering, data acquisition, data analysis and visualization, signal transmission, and signal processing.

**LINAS SVILAINIS** (Senior Member, IEEE) received the Ph.D. degree from Kaunas University of Technology, Lithuania, in 1996.

Since 2009, he has been a full-time Professor with the Department of Electronics Engineering, Kaunas University of Technology. Currently, he is a Principal Investigator with the Signal Technologies Group, Kaunas University of Technology. He has worked in the area of ultrasonics for more than 30 years and has authored three books, more than 150 articles, and holds two patents. His current research interests include the design and optimization of ultrasound electronics; ultrasonic signal processing; time delay (time-of-flight, ToF) estimation; spread spectrum signals; air-coupled ultrasound; large scale LED video displays; electromagnetic compatibility; and EMI protection of electronic systems.

Dr. Svilainis is an Associate Editor of IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT journal.

**ALBERTO RODRÍGUEZ-MARTÍNEZ** (Senior Member, IEEE) received the B.E. degree in telecommunications from the University of Vigo, Spain, in 1998, and the Ph.D. degree in telecommunications from the Polytechnic University of Valencia, Spain, in 2011.

He is currently an Associate Professor with the Department of Communications Engineering, University Miguel Hernandez of Elche (UMH), Elche, Spain. His research interests include time-frequency analysis, ultrasonic signals processing, nondestructive materials characterization, and biomedical signal processing.