IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 15, 2021, accepted April 11, 2021, date of publication April 22, 2021, date of current version May 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3074875

Design Space Exploration for an IoT Node: Trade-Offs in Processing and Communication

IRIDA SHALLARI[1], ISAAC SÁNCHEZ LEAL[1], SILVIA KRUG[1,2], AXEL JANTSCH[1,3], (Senior Member, IEEE), AND MATTIAS O'NILS[1], (Member, IEEE)

[1]Department of Electronics Design, Mid Sweden University, 85170 Sundsvall, Sweden
[2]IMMS Institut für Mikroelektronik - und Mechatronik-Systeme gemeinnützige GmbH (IMMS GmbH), 98693 Ilmenau, Germany
[3]Institute of Computer Technology, TU Wien, 1040 Vienna, Austria

Corresponding author: Irida Shallari (irida.shallari@miun.se)

This work was supported by the Knowledge Foundation.

ABSTRACT The proliferation of smart sensor nodes for IoT deployments comes with requirements of energy efficiency and to fulfil functional requirements, but it also demands a fast time to market. As a result, we need to facilitate the design of these IoT nodes, while providing the required performance. In this article, we introduce a design space exploration method focusing on IoT nodes that are data intensive due to the inherent complexity of their high data volume. The proposed method aims to identify areas of the design where processing optimisation would have a greater impact on the overall node energy consumption, define an energy budget for prospective additional tasks in the processing pipeline, and in conclusion evaluate the optimal node offloading configuration.

INDEX TERMS Intelligence partitioning, IoT, smart sensor, design space exploration, energy efficiency, smart camera, WSN, DSE.

I. INTRODUCTION

The Internet of Things (IoT) has attracted significant attention in recent years, resulting in the deployment of a variety of sensor devices, supported by ubiquitous computing [1]–[4]. The scenarios keep growing, aiming at large scale implementations such as smart factories, environmental monitoring, and smart cities, while this continuous expansion requires optimised utilisation of the components in a Wireless Sensor Network (WSN). Focusing on the sensor node, we need to facilitate its deployment for the different scenarios, hence batteries and energy harvesters are frequently selected as energy sources. This imposes tight constraints on the energy consumption of the sensor node, which has to be considered in the architecture of the WSN for the computational and communication configurations [5], [6].

In the analysis of IoT sensor nodes, a widely used assumption is that IoT sensor nodes are devices that sense, store and communicate scalar sensor values. This approach limits the analysis to simple sensors, such as pressure, temperature or humidity sensors, leaving out sensors that provide vectors of data such as sound or vision sensors. Thus, more complex sensors require a reevaluation of design constraints in terms of latency and energy consumption that can result from analysing the inter-dependencies of the data and processing complexity.

In a simplistic view of design space exploration (DSE) for IoT applications, the amount of data produced from the sensor node depends on both the application requirements and the sensor configuration. Similarly, the communication is defined by the protocol and technology chosen, and its characteristics such as packet size, energy consumption per packet, and communication range. In contrast, the processing requirements can be defined by many factors such as computational entity (in-sensor, fog and cloud computing), processing architecture, application requirements, and several other elements. In an IoT application, communication and processing are not independent entities, but their configurations affect one another [7]. Regardless, when analysing and optimising the node energy efficiency, efforts are mainly focused on either the communication or the processing component [8]–[10].

Optimisation is a problem that has been present in IoT design for many years, nonetheless, state-of-the-art research only focuses on the optimisation of specific scenarios rather than developing a method for DSE on IoT nodes. The following articles [11] and [12] abstract from specific applications

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

and address DSE of efficient IoT nodes by providing an analysis of platforms, design flow and complexities at different design levels. However, there is a vacuum on DSE methods that accelerate the design process and time-to-market of energy efficient IoT nodes. In this article, we introduce a DSE method, that enables generalisation of the problem by including optimisation considerations at early design stages.

To provide generalisation, this method relies on high level estimations about the IoT application to be deployed, such as the expected data flow between the computational stages, number of operations for each processing task, energy constraints, and communication considerations (communication range, technology class, etc.). Based on this data we estimate the node energy consumption for any given CPU that we can obtain an estimate of the number of clock cycles for FLOPs and other operations. The results we present in the following sections are based on the ARM M4 CPU. We begin our analysis by identifying areas of the design where processing optimisation would have higher impact on the overall energy consumption of the sensor node. This is followed by estimations of the energy budget available for any additional tasks, while the last part of this method evaluates the possibility of node offloading to other computational entities. Furthermore, we provide a MATLAB implementation of the presented method [13]. We have chosen four applications, two based on traditional image processing and two CNN-based applications, due to the contrasting nature of their data flow throughout the processing stages. To summarise, the contributions of this paper are:

- DSE method for data intensive IoT nodes relying on processing and communication interaction.
- An energy budgeting method that enables you to estimate the energy consumption available for additional processing tasks in a given application.

The remainder of this paper is organised as follows. In Section II we review state-of-the-art DSE methods for IoT applications, followed by an overview of approaches towards energy efficient IoT nodes focusing on communication technologies, processing approaches and data reduction approaches. This is followed by a description of the theory and method for DSE in data intensive IoT nodes in Section III. Section V and VI provide the results and discussion of the method, based on data from several communication technologies and IoT implementation scenarios. Section VII summarises the conclusions of this article.

II. RELATED WORK

Communication technologies with their capability to adapt to ever changing market requirements have been one of the driving forces in the IoT evolution [14]–[16]. In the last decades, a variety of technologies and protocols have been developed to meet the requirements of IoT applications regarding data volume, time criticality, accuracy in transmission, energy efficiency and other constraints. New technologies were introduced, as in the case of LPWAN (Low-Power Wide Area Network) technologies [17]–[19], providing a large communication range, while operating on low energy consumption, in contrast to existing cellular technologies such as GSM, GPRS, HSPA. However, these technologies have limitations when the data rate requirements are more than simply scalar sensor data, incurring in much higher latency compared to other communication technologies. In such cases, depending on the requirements of the coverage area, either short-range communications such as Bluetooth Low Energy 4.2 and 5 (BLE), or cellular technologies such as LTE Cat.1 and 4 have proven to perform better. In the range of IoT applications, there are also applications demanding high data rate communications in time-critical transfers with high accuracy, while operating in large areas. An example of this would be Industrial IoT (IIoT) applications. The introduction of 5G communication with promising features to meet such requirements raises the expectations that this technology might become a catalyst in the development of IIoT applications.

Looking further into the problem of energy efficient wireless sensor nodes, another approach considered relies on implementing data reduction methods in the sensor data. Image, video and sound processing are fields that work on large amounts of data, hence the problem of data reduction has been present for many decades, developing a variety of compression algorithms. Current IoT systems take inspiration from such algorithms to provide compressed sensing, based on the principle that for a sparse signal in a specific basis, it can be recovered from less random linear measurements compared to its ambient dimension [20], [21]. Compressed sensing has been used in different fields, such as wireless communication [22], medical imagery [23], cryptography [24], and many more, providing high accuracy while processing a reduced amount of raw data. Alternatively, adaptive sampling can be a helpful method to reduce redundant data relying on run-time signal variation [25]–[27] or prior knowledge of the system on expected signal behaviour [28]. For more complex systems with several IoT nodes, data reduction has been obtained by designing a hierarchical Wireless Sensor Network (WSN), where the nodes in more sensitive areas send data more frequently, while the rest sends data at more sparse intervals, resulting in reduced overall data traffic [29]. In such networks, also routing of the data in the network can have a significant impact on the energy consumption of the node itself [30]. Alternatively to the cases above, in some scenarios of scalar value sensors with redundant data, e.g. a temperature sensor sampling every second, data aggregation can be used by recording the data in intervals of several minutes, and transferring it to the server every few hours or days. However, this is not always an option, as in the case of more complex sensor data, where only parts of the information are useful to the analysis, then filtering of the information based on a set of criteria is necessary.

The ever growing attention towards autonomous systems has led to a greater interest in vision and sound systems as methods of inspection. As mentioned before, their main complexity for energy efficient deployments is the high amounts of data produced, up to hundreds of megabytes per second. An overview of state-of-the-art smart camera systems shows that in some cases, attention is paid to their processing architecture in the energy efficient implementation of image processing tasks in the smart camera node [31]. The aim is to implement a major part of the image processing pipeline in the smart camera node itself, while transmitting limited data to the server or cloud. In other cases, the method of choice relies on server-based processing of the data, with the camera node streaming every frame captured [32]. Based on our previous analysis [33], such approaches are not always the most energy efficient, due to the trade-off between the communication and processing workload in the overall energy consumption of the smart camera node. We observed that in cases where the addition of a few processing tasks resulted in a significant reduction of the data, the most efficient way of implementation was to distribute the processing between the smart camera node and the server. Instead, in cases where extensive image processing resulted in marginal data reduction, the energy efficient solution would be to implement most, or all the image processing tasks in the smart camera node. This raised the question of how to define how much energy should be used in the additional processing task to provide the necessary data reduction. Our method addresses these issues, providing IoT architects in the DSE stage with information necessary to support their decision-making process, while developing a new IoT deployment. It relies on current data volume and number of operations to estimate the node energy consumption and reduce the design space focusing only on a set of tasks. This data is used to define an energy budget for additional tasks based on energy constraint and the expected size of the output data, while in the final step we evaluate node offloading perspectives.

III. THEORY AND METHOD

In previous work [7], [33] we have introduced intelligence partitioning as a method that explores the effects of the distribution of the computational tasks between different computational entities on node energy consumption. Intelligence partitioning $\Im$ is a tuple,

$$\Im(F) = \{f_{Node}, f_{Fog}, f_{Cloud}, d_{Node\to Fog}, d_{Fog\to Cloud}, d_{Node\to Cloud}\} \quad (1)$$

where $f_{Node}, f_{Fog}, f_{Cloud}$ are the functions allocated to the sensor node, fog entity and cloud server, respectively, and $d_{S\to D}$ designates the amount of data communicated from source S to destination D.

The energy consumption in a wireless sensor node can be represented as

$$E_N = E_s + E_p(\sum_{j=1}^{x} o_j, P) + E_c(d, C) \quad (2)$$

where E_N is the node energy per sample, and its components E_s, E_p and E_c are sensing, processing and communication energy per sample, respectively. The sensing energy, E_s, is not affected by the processing and communication configuration, hence we consider it constant. In-node processing targets to reduce the data captured from the sensor node, according to an application specific configuration, thus, trading communication energy, E_c, for processing energy E_p. The processing energy E_p is dependent on the number of operations o, where x is the partition point that takes values from $1, \ldots, n$, and n is the total number of operations for the given scenario. The generalised relation between the number of operations, o, and the data sent d, is such that:

$$o \uparrow \sim d \downarrow \quad (3)$$

It represents the contrasting relationship between processing complexity and data amount; the more processing is done in the processing entity, the less data needs to be communicated to the downstream processing entity. The sum $\sum_{j=1}^{x} o_j$ represents all the operations that will be executed on the smart sensor node, while the remaining operations will be forwarded to the fog or cloud entity. Another element influencing the processing energy E_p is P, representing the processing device of the sensor node. The communication energy is dependent on the amount of data sent from the node, d, and on the communication channel used, C.

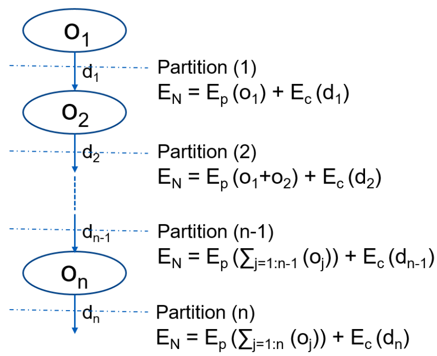

FIGURE 1. Schematic representation of intelligence partitioning in a set of tasks, for a given processing and communication technology.

Figure 1 is a schematic representation of Eq. 1, in which for a given design case, we can partition the processing flow at several sequential points. $o_1, \ldots, o_n$ are the number of operations for each of the processing tasks in the processing stream for a given scenario, while d_j is the resulting data volume at partition point j. d_1 is the initial sensor data, while d_n represents the final output of implementing all the scenarios in the sensor node. For each partition point we consider that all the operations up to the partition point will be implemented in the smart sensor node, while the remaining will be processed in another computational entity. Therefore, the processing energy consumption is the cumulative energy consumption from implementing all the processing tasks prior to the partition point in the sensor node, while the communication is defined by the volume of output data at that partition point.

Assume that there is a given energy constraint ε to achieve a certain battery lifetime, which limits the maximum energy per sample that can be consumed in the IoT node, i.e. $E_N \leq \varepsilon$. The energy constraint ε is given by

$$\varepsilon = E_{batt} \frac{T_{duty}}{T_{life}} \tag{4}$$

where the E_{bat} is the energy stored in the battery, T_{duty} is the period for processing one sample in the node and T_{life} is the battery lifetime constraint of the targeted system.

To get a reference for how different energy constraints reflect on battery lifetime for several battery types, we have applied Eq. 4 on three battery technologies: a button cell battery (0.63J), a standard AA battery (4.5J) and a larger D-cell battery (46.8J). Table 1 shows the battery lifetime for five different energy constraints expressed in the number of months for three different duty cycles, 1 transfer per minute, per hour, and per day (1/minute, 1/hour, 1/day).

TABLE 1. Battery lifetime in months versus energy per sample constraints and duty-cycle/battery size. Only values over 2 months of lifetime are included.

	Energy constraints				
$\varepsilon[J]$	0.000 05	0.000 5	0.005	0.05	0.5
Plot representation	#1	#2	#3	#4	#5
2032 battery					
1/minute					
1/hour	18				
1/day	420	42	4		
AA battery					
1/minute	2				
1/hour	125	13			
1/day	3 000	300	30	3	
D-cell battery					
1/minute	22	2			
1/hour	1 300	130	13		
1/day	31 200	3 120	312	31	3

The DSE method encompasses all the above elements into a three stage algorithm as described in algorithm 1. This method relies on a preliminary set of data such as the number of operations for each task in a given application, and the inherent data volume of each task, to support decisions about optimisation efforts and task distribution. Also, the user would have to define an energy constraint per sample which would be used in the analysis.

The first part of the algorithm focuses on processing exploration, based on an estimate of the node energy consumption for the ARM M4. After calculating the processing and communication energy consumption for different task partitioning configurations, it analyses the trade-off between the processing and communication energy consumption, identifying tasks where optimisation efforts would have a major impact on the overall node energy consumption. The following stage evaluates the possibility of introducing new tasks in the processing pipeline, and defining the processing energy consumption available for the prospective tasks.

Algorithm 1 Design Space Exploration Algorithm

User input: ε, o, d.
Result: Processing exploration
// Identify design areas where processing optimisation would have a major effect.
task_no = 2;
max_ratio = 0;
while *task_no* $\leq n$ **do**
    energy_ratio = $\frac{E_p(o_{task\_no})-E_p(o_{task\_no-1})}{E_c(o_{task\_no}-E_c(o_{task\_no-1}))}$;
    if *energy_ratio* > *max_ratio* **then**
        max_ratio = energy_ratio;
        position = task_no;
    end
    task_no++;
end
Output: position.
Result: Energy budget for adding tasks to the processing pipeline
for $i \leftarrow 1$ **to** *new_tasks_pool* **do**
    if $E_N > \varepsilon$ **then**
        $E'_p = (E_c(d_\varepsilon) - E_c(d')) - (E_N - \varepsilon) - E_s$;
    else
        $E'_p = (E_c(d') - E_c(d))$;
    end
end
Output: E'_p.
Result: Node offloading with intelligence partitioning
min_en = 100;
for $i \leftarrow 1$ **to** n **do**
    if E_N < *min_en* **then**
        min_en = E_N;
        partition_position = i;
    end
end
Output: min_en, partition_position.

This is derived from the difference in communication energy consumption due to prospective data reduction, alongside its interaction with the energy constraint. Figure 2 represents two cases where the introduction of an additional processing task, that we assume would be reducing its input data volume, would be affecting the energy efficiency of the smart sensor node. In case (1), for an expected data reduction from d_0 to d_1, we can implement the additional processing task only if the processing energy required for it is not more than the communication energy reduction due to data reduction. Instead, in case (2), we start from an energy level beyond the energy constraint, which does not allow the implementation of the overall application as it is. As a result, the additional processing task could enable its implementation in this given energy constraint, but only if the processing and sensing energy consumption is not more than the communication energy reduction due to data reduction from d_ε to d_1. In the last stage of the

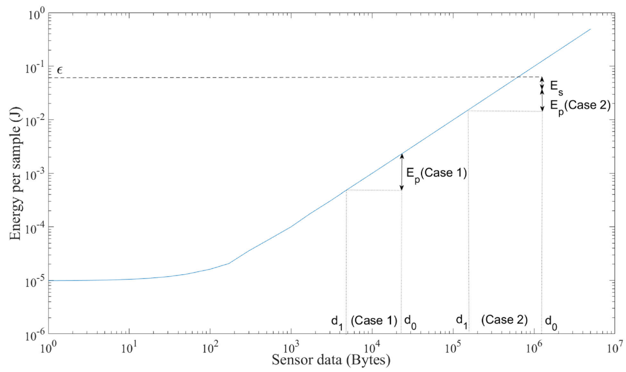

FIGURE 2. Illustration of information reduction for increased battery life (case 1) and to meet the energy constraint (case 2).

algorithm we analyse node offloading possibilities and their inherent energy consumption, to define the optimal partition for the given application.

IV. COMMUNICATION MODEL

In previous work [7], we have analysed the effects of processing and communication energy consumption on the overall energy efficiency of the sensor node, with the results underlining the importance of the choice of communication technology in the overall optimal configuration. Hence, in this DSE analysis we have included nine communication technologies, representing the main three communication categories, as shown in Table 2. To estimate the energy consumption for each technology and for the resulting data volume at each partition point, we relied on the model by Krug and O'Nils [34].

TABLE 2. Communication technologies included in the analysis of this DSE method.

Communication groups		
LAN	Cellular	IoT
BLE 5.0 [35]	GPRS [36]	802.15.4 g [37]
802.11 [38]	HSPA [39]	NB-IoT [40]
	LTE C. 4 [41]	LoRa [42]
		LTE C. 1 [43]

The relation between data rate requirements and the communication technology type, is significant for in the overall energy consumption of the sensor node. Communication technologies have been optimised for different aspects of the communication parameters, especially the data rate capacity. The plots in Fig. 3 show the variation in the communication energy consumption as we increase the data rate requirement per sample.

The communication range can be an important criteria in the selection of a communication technology for an IoT application, and the plots in Fig. 3 represent the three communication groups in Table 2. If we consider a communication range of less than 1 kilometer, it would be represented by BLE 5.0 and 802.11n from the LAN group, and by 802.15.4 from the IoT group. For data volumes below 10 000 Bytes/sample, the three communication technologies have 1 to 3 orders of magnitude difference in energy consumption. In contrast, for data volumes above 30 000 Bytes/sample, the gap in energy performance between the communication technologies is significantly smaller, and the difference between 802.15.4 and 802.11n is about 2×, and their difference with BLE 5.0 is about 10×. Overall, for short-range communication, BLE 5.0 is the most energy efficient technology. The IoT group includes all remaining communication technologies, representing a communication range of over 1 kilometer. For data volumes of 0-50 Bytes/sample, NB-IoT, HSPA, LTE C. 1, and LTE C. 4 have comparable performance, while for higher data volumes, the energy performance of NB-IoT, similarly to LoRa, deteriorates significantly, with up to two orders of magnitude higher energy consumption. For data volumes beyond 1 000 Bytes/sample, LTE C. 4 has the best energy performance in its group. GPRS and LoRa are the worst performing technologies with respect to energy efficiency for all data volumes. In a comparison of their performance, for data volumes of up to 10 Bytes/sample, there is a difference of one order of magnitude. As the data volume increases, this difference grows significantly, with beyond 4 orders of magnitude difference for data volumes above 1 000 Bytes/sample.

Analysing the same plots from the data rate requirement perspective, we can note that for the region with data volumes of up to 50 Bytes/sample the variation in energy consumption is less than one order of magnitude, with the exception of BLE 5.0. As the data volume increases, the difference in energy consumption becomes more noticeable, with LoRa, GPRS and NB-IoT providing the worst performance, while HSPA, LTE C. 1 and 802.15.4 provide similar performance. For data volumes beyond 10 000 Bytes/sample, the energy consumption gap between 802.11n and BLE 5 is reduced from 3 orders of magnitude for lower data volumes, down to less than 1 order of magnitude. Therefore, for data volumes higher than 10 000 Bytes/sample, the most energy efficient communication technologies are 802.11n, LTE Cat.4 and BLE 5. However, if we consider the communication range, the latter technology has the best performance in short range communications, while for long range communication, the best performance is achieved with LTE C. 4. In the meantime, LoRa has the worst energy performance in all sample sizes, while also providing a communication range much greater than the remaining technologies.

Another aspect of interest is to analyse the effects between processing and communication in terms of the overall node energy consumption. The following analysis is based on the plots in Fig. 4–7 resulting from the design cases and communication technologies described above. The dotted lines denoted by #1 − #5 represent energy constraints referred to in Table 1, and the remaining dotted curves represent the communication energy consumption, where the different communication technologies are colour coded. Each column of data points represents a partition in the image processing

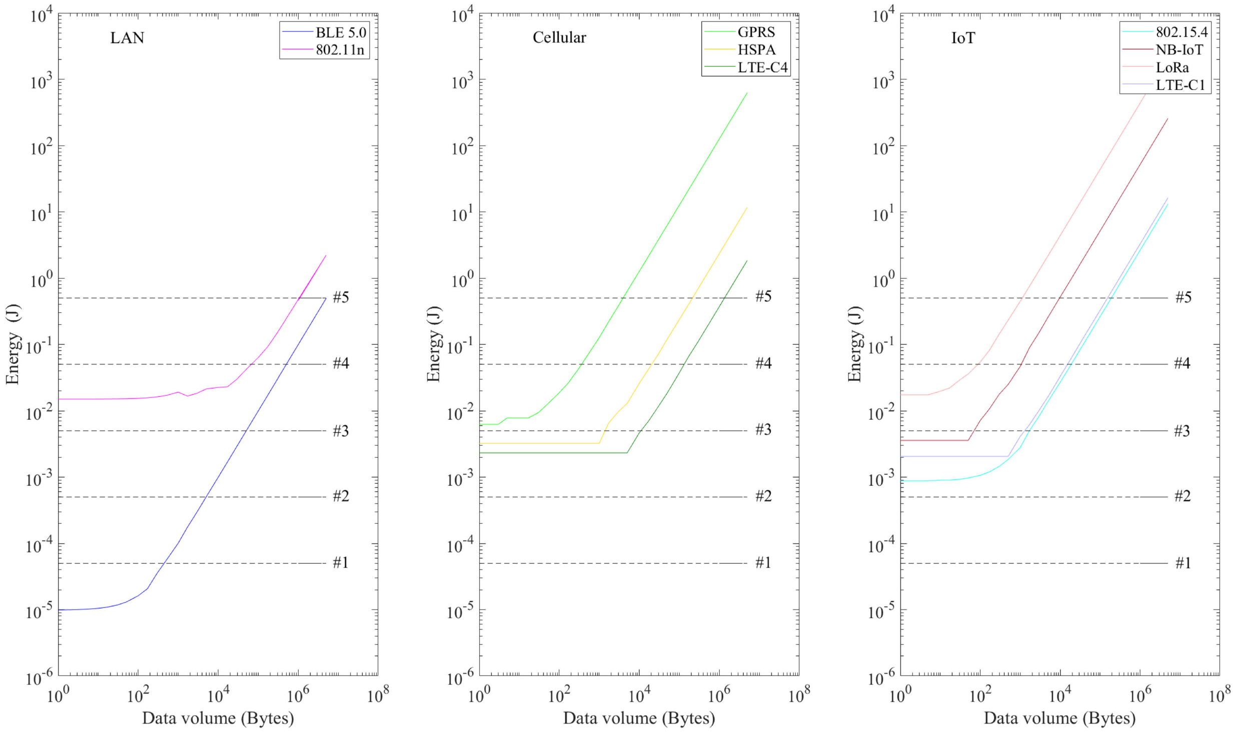

FIGURE 3. Communication energy consumption per sample, with the energy constraints according to Table 1.

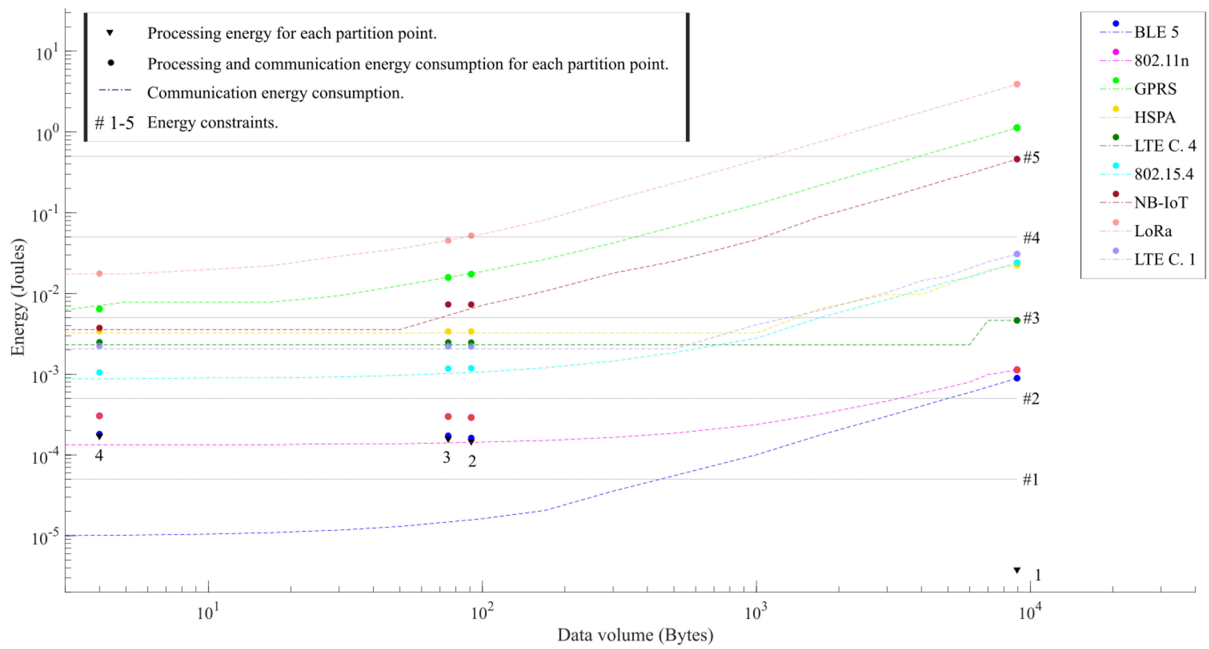

FIGURE 4. Energy consumption per operation for the data volume at each partition point in the people counting scenario.

pipelines considered in these examples, and to facilitate visualisation we have numbered each of these partition points. Partition 1 represents the case where no processing is done in the smart sensor node, a frame is captured and then transferred to the cloud server processing unit using one of the communication technologies. The final partition point represents the case where all the processing is done in the sensor node and only the final results are transferred to the cloud server processing unit. For each partition point, the data points are represented by several circle data points and a single downward triangle; the difference between them is that the former (the circles) represent the overall energy consumption at the node for that partition point as a result of processing and communication energy consumption, while the latter (the downward triangles) represent only the processing energy consumption.

V. RESULTS

The analysis of this DSE method is based on four design examples: two traditional image processing systems, and two CNN-based systems. This choice is motivated by the significant differences in data flow and processing requirements between the two classes, which subsequently affect design considerations. In our analysis we consider each image processing stage as a possible partition point, and Table 3 shows the intermediate data size at each processing stage, constituting the data volume at each partition point.

TABLE 3. Data flow in Bytes between the computational stages in design examples based on traditional image processing (1. people counting, 2. particle detection) and CNN algorithms (3. AlexNet, 4. VGGNet 16).

Tasks	Traditional systems		CNN systems	
	1 [44]	2 [45]	4 [46]	5 [47]
1	8940	256 000	154 587	150 528
2	91	680	69 984	3 211 264
3	75	500	43 264	1 605 632
4	4	259	64 896	802 816
5			9216	401 408
6			4096	100 352
7			1000	25 088
8				4096
9				1000

A. TRADITIONAL IMAGE PROCESSING SYSTEMS

The first design case we analyse is the people counting scenario in Fig. 4, relying on four partitioning points (1)-(4). Partition (1), due to highest data rate requirements of the partitions, is more sensitive in the trade-off between communication technology and energy constraint. In this partition, LoRa and GPRS do not fulfil any of the energy constraints, while none of the other communication technologies fulfils energy constraint #1 in any of the partition points. Throughout all partition points, the LAN communication technologies (BLE 5.0 and 802.11n) are the most energy efficient options.

The particle detection scenario has much higher data rate requirements compared to the people counting scenario, and this is reflected in higher limitations from the communication technologies. For the first partition point, communication technologies in the IoT group, together with GPRS and HSPA do not meet any of the energy constraints. The most energy efficient performance is provided by the LAN communication technologies, while none of the partition configurations can operate under energy constraint #1. Partition (2) with BLE 5.0 communication provides the highest energy efficiency of all configurations.

B. CNN-BASED SYSTEMS

CNN-processing systems have a different complexity in IoT deployments compared to traditional image processing systems, mainly due to more demanding processing requirements, higher data volumes, and a variation of the intermediate data volumes that is not continuously decreasing with additional processing. We begin our analysis with AlexNet in Fig. 6, where for the given energy constraints, we can note that Lora, GPRS, NB-IoT and LTE C. 1 cannot support any of the resulting data volumes from the partition points. The remaining communication technologies meet energy constraint #5, while for the first partition only BLE 5.0 and 802.11n meet energy constraint #4. Of all the partitions, at partition (1) we obtain the most energy efficient combination with BLE 5.0 communication.

The last design case is VGGNet 16 in Fig. 7, where the data flow is different compared to the other scenarios due to the significant increment in the intermediate data produced between the nine partition points, alongside the higher computational complexity. BLE 5.0 and 802.11n are the only communication technologies that can be used for both partition (1) and (2), under energy constraints #4 and #5 respectively, while LTE C. 4, HSPA, 802.15.4 and LTE C. 1 can meet energy constraint #5 for partition (1). Partition (1) with BLE 5.0 is the most energy efficient partition, meeting energy constraint #4.

VI. DISCUSSION

In this section we focus on how to use the DSE method to design an energy efficient IoT node. We provide an analysis based on the stages depicted in algorithm 1, which can also be reproduced following the Matlab example [13] implementation.

A. PROCESSING EXPLORATION

Identifying which areas of the design we should focus optimisation efforts on is an important element in DSE. In the people counting scenario, the difference in processing energy consumption between partition (1) and (2) is clearly the most prominent, suggesting that optimisation at partition (2) would be beneficial for the in-node implementation of the rest of the design. Considering that partition (2) relies on segmentation and binary image compression, the latter would be where the optimisation efforts should focus with hardware implementation or acceleration. Out of all partitions, partition (2) and (3) are the most energy efficient options. In the particle detection scenario we also have a possible optimisation area at partition (2) related to binary image compression. However, in contrast to the previous scenario, the most prominent optimisation point would be at partition (3), where morphology operations are added. This difference might be due to the significant difference in image size between the two traditional image processing scenarios.

In the case of CNN, the highest energy consumption is in the early convolutional layers, where both the computational

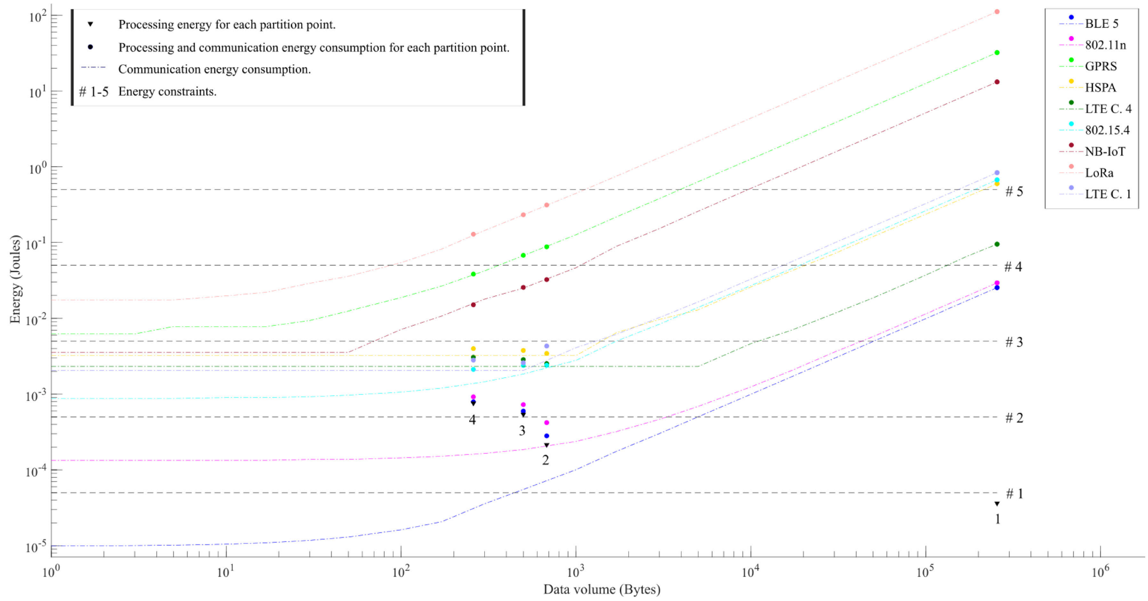

FIGURE 5. Energy consumption per operation for the data volume at each partition point in the particle detection scenario.

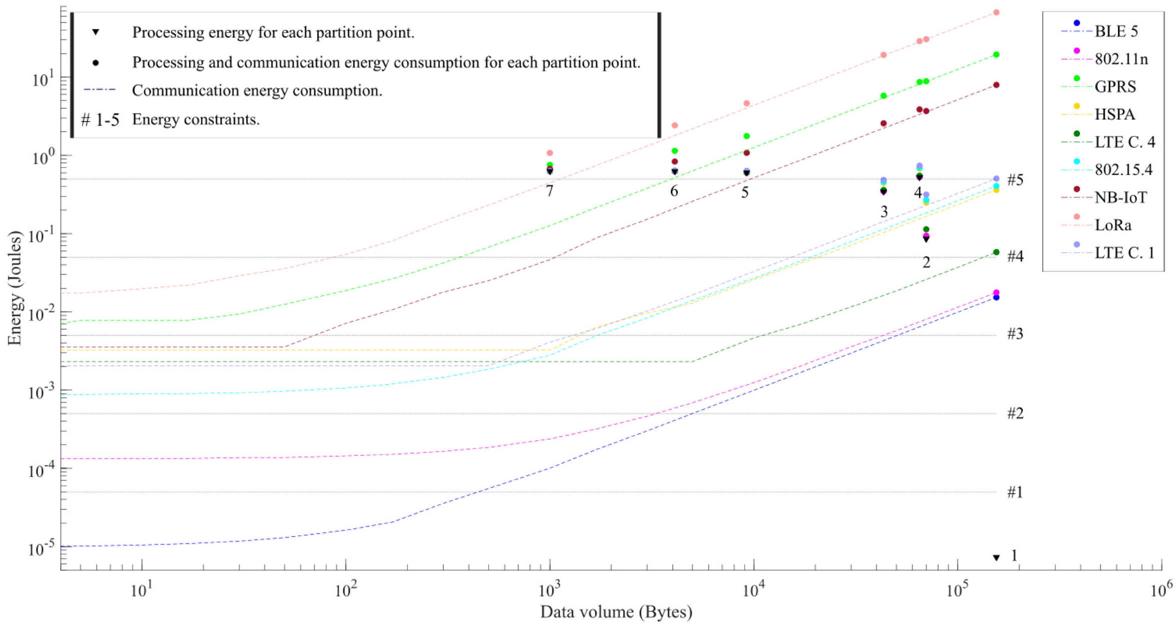

FIGURE 6. Energy consumption per operation for the data volume at each partition point in AlexNet.

requirements and data volume are at their peak. In the case of AlexNet, due to the trade-off between processing and communication energy consumption, the areas of interest for further optimisation would be between partitions (2)-(4) that represent the first three convolutional layers. Unlike the other design cases, for VGGNet 16 in partitions (2)-(5) we have altogether the highest processing energy consumption, while also the highest data volume, hence the communication energy consumption is higher than at partition (1). As a result, this area would not be optimal to partition, but processing optimisation with hardware acceleration might provide significant enhancement in the overall node energy consumption.

The energy consumption estimation for both AlexNet and VGGNet 16 is done based on an estimate of floating point operations. However, there has been ongoing research developing methods that shift from floating point to fixed point computation in CNNs, while maintaining a good trade-off in terms of energy versus accuracy. For example, Jain *et al.* [48] introduced a fixed point representation with error compensation and used it to deploy both AlexNet and VGGNet 16. Their results showed the improvement in energy efficiency could be up to 4 times compared to fixed point operations, while the accuracy loss was less than 0.5% compared to the accuracy with floating point operations. Such results suggest a possible vertical shift downwards for the data points related to partitions (2)-(9); however, for the given design examples, the energy reduction would have to be more than an order of magnitude for another partition point to be more energy efficient than partition (1).

B. ENERGY BUDGET FOR ADDITIONAL TASKS

In the people counting scenario, for the partitioning points at segmentation (2) and morphology (3), we have included binary image compression in the processing pipeline; instead, the output of background modelling (1) is an uncompressed RGB image. Therefore, we need to identify the processing energy available, alongside the gain in communication energy due to data reduction, to determine whether it would be possible to implement a compression algorithm for the given energy constraints.

The expected data volume after compression would be more than 100 Bytes, considering that it would be difficult to achieve a lower data volume than after segmentation and binary image compression altogether by using only RGB compression. Hence, in this compression range from 9 000 Bytes down to more than 100 Bytes, we note that all the communication technologies, with the exception of LTE C. 4, could shift to new energy levels. In the case of LTE C. 4, the analogy would be with case 1 in Fig. 2, where we consider the use of image compression to provide higher energy efficiency. To achieve this enhancement in the node energy consumption, it would require a data reduction down to approximately 6 000 Bytes, with the processing energy consumption being no more than the difference in communication energy consumption resulting from data reduction. This approach would provide a more energy efficient deployment by a margin of approximately 0.002 Joules, while there are a few more aspects that need to be considered. If we compare the data volume between the four partition points there is a significant difference, which is not reflected in the overall energy consumption amongst these partitions. Hence, defining the optimum partition for the case with LTE C. 4 would require an evaluation of how this change in data volume, and subsequently in channel utilisation (and its inherent costs), would affect the overall costs of the sensor node, alongside evaluating how the time-to-development for partition (1) would affect the costs.

Furthermore, the case of LoRa and GPRS would be in analogy with case 2, where to implement partition 1 with LoRa and energy constraint #5, we would need the initial image to be compressed down to at least 1 200 Bytes, or less, with the processing energy available resulting in the difference between the energy constraint, the communication energy consumption for the compressed image and the sensing energy consumption. If instead we want to use image compression in order to meet energy constraint #4, it would not be possible, considering that the compressed image would have to be smaller than the output data rate of partition (2), which is already a compressed binary image.

Similarly to the people counting scenario, in the particle detection scenario, partitions (2-4) rely on binary image compression, while partition (1) is an uncompressed RGB image. Following the same analysis, the inclusion of image compression considerations could enable deploying partition (1) with all the given communication technologies under energy constraint #5.

In the AlexNet design case, for partition (1) we consider streaming the full frame without image compression considerations. However, research by for example Dejean-Servieres *et al.* [49] shows that JPEG compression by a factor of 7 can leave the accuracy of the CNN unaffected. This generates interest in understanding how it would affect the choice of communication technology, and subsequently the allocation of the processing tasks between the sensor node and a cloud server processing unit.

If we consider a theoretical reduction of the image by a factor of 7, then it would shift the data volume of partition (1) from 154 587 Bytes down to 22 083 Bytes, which would be less than the volume of the intermediate data produced at partitions (2-4). Communication technologies such as LoRa, GPRS and NB-IoT cannot support the data volume resulting from partition (1), with or without compression. Instead, communication technologies such as LTE C. 1, 802.15.4 and HSPA, in analogy with case 1 in Fig 2, represent the case where data reduction could improve the energy efficiency of the smart sensor node, but there is no shift into lower energy levels, in contrast to the remaining communication technologies.

Considerations of image compression are also important from a processing optimisation perspective. Out of the 7 partition points, partitions (2-4) are the ones with the highest processing complexity, where the processing component prevails in the overall energy consumption, especially for communication technologies such as BLE 5 and 802.11n. This suggests that optimising an image compression algorithm could be more beneficial in terms of energy consumption and less complex compared to optimisation in the convolutional layers. Similarly to AlexNet, in the case of VGGNet 16, considering the compression by a factor of 7, the resulting data volume would be smaller than the volume of the data output at partition (7).

C. NODE OFFLOADING - INTELLIGENCE PARTITIONING

In the processing exploration stage we analysed the partition points to identify which cases would be possible to implement under the given energy constraint, and how they were interlaced with variations in communication technology. Based

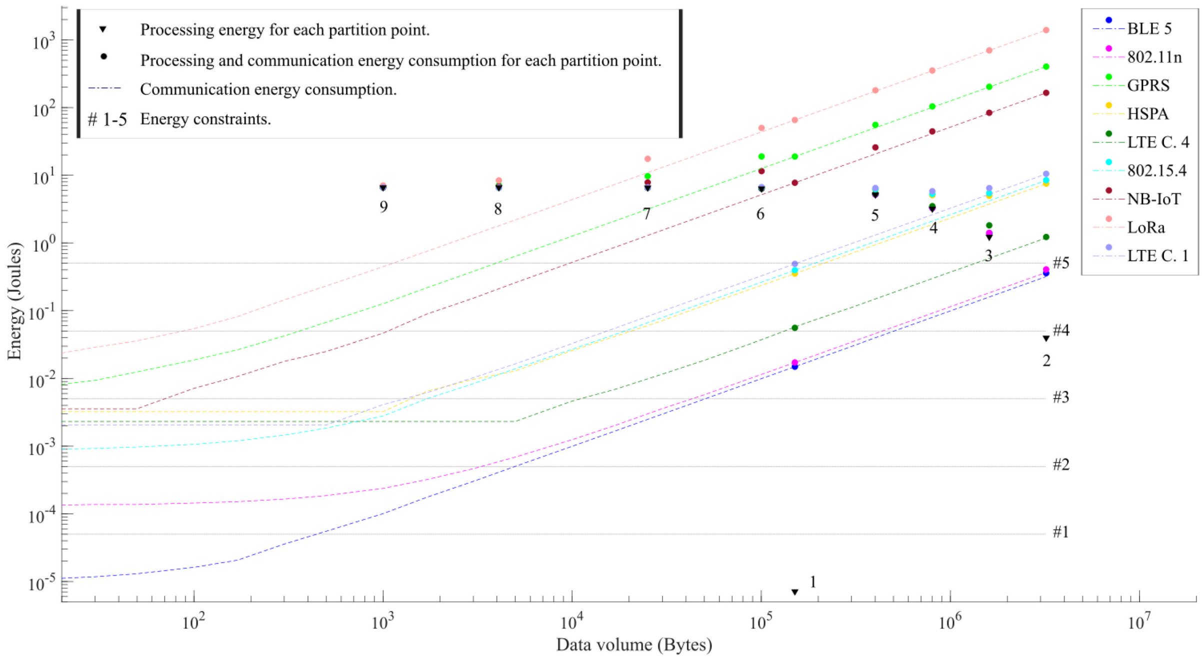

FIGURE 7. Energy consumption per operation for the data volume at each partition point in the VGGNet 16.

on these points we proceed with considerations of node offloading, which we refer to as intelligence partitioning, because it defines how we should distribute the computational workload (intelligence) between the sensor node and a cloud server processing unit, with the aim of optimising the overall energy consumption in the sensor node.

The two scenarios from traditional image processing systems share similarities in the processing tasks, which is later on reflected in the optimal partition point, which for both of them is after the segmentation stage at partition (2). Depending on the deployment requirements in terms of coverage area, for LAN communications BLE 5 would be optimal, for cellular it would be LTE C. 4 and for IoT would be 802.15.4. If we consider cellular communication and their inherent subscription costs, for the people counting scenario choosing partition (4) could be more optimal than partition (2), because at the expense of increasing the overall energy consumption in the range of 10^{-5} Joules, it would reduce the communication from 91 to 4 Bytes/sample.

The discussion about intelligence partitioning is more complex for the CNN examples, because considerations of image compression and communication technology requirements depending on the scope of the application, would define the optimum configurations. For both design examples, based on the data in Fig. 6 and Fig. 7, the overall lowest energy consumption is obtained for partition (1) with either BLE 5 or 802.11n. What distinguishes the two scenarios is the sensitivity that the optimum partition has towards the chosen communication technology. In the case of AlexNet, shifting between different communication technologies shifts the balance, where for technologies such as HSPA, LTE C. 1 and 802.15.4 the optimum is at partition (2). Therefore, the inclusion of image compression considerations might generate a new balance point, with an even better energy consumption due to the trade-off between processing and communication energy consumption. In partition (2), the close proximity between the processing energy consumption and the overall energy consumption for the case with BLE 5 or 802.11n, shows that for a new partition with compression more energy efficient than partition (2), it would need to have lower processing energy than the first convolutional layer. In contrast, for VGGNet 16 partition (1) is clearly the optimal partition point, because for partitions (2-5) the output data at each convolutional stage is higher than the initial input frame.

In all design cases we analysed, the focus was on how the variation in data volumes caused by either the tasks in the image processing pipeline, or the compression algorithms, would affect the choice of communication technology and the reference energy constraint, which thereafter defined processing considerations. However, another element that would affect these considerations would be the choice of processing device, which in these energy estimates is the ARM M4. The variation in energy consumption inherent to the choice of processing device, would cause a vertical shift in these energy data points for each partition, which in itself would not affect the selection of the optimal partition point. Regardless, this vertical shift in energy consumption would influence the effects between the energy constraint and communication energy consumption in defining the processing energy available for additional processing.

TABLE 4. Review of DSE publications.

	Publications					
	[50]	[51]	[12]	[52]	[53]	Out method
Processing optimisation	✓	✓	✓	✓	✓	✓
Communication interplay	-	-	-	-	-	✓
Application field	Cryptography	MPSoC	MPSoC	CNN	SoC	IoT node
Scope	Review of applications in cryptography and DSE reflections.	Scenario-based DSE for MPSoC	Review of DSE in embedded systems.	DSE for CNN in SoC	DSE method focused on HW/SW partitioning with Integer Programming	DSE for data intensive IoT node

D. CONTRIBUTION AND COMPARISON OF OUR METHOD TO OTHER DSE METHODS

In our literature review, we have identified a vacancy in DSE methods for IoT nodes. As summarised in Table 4, current DSE methods in embedded systems focus only on the processing component, addressing issues related to hardware/software partitioning, task allocation in multiprocessor system on chip (MPSoC), or resource allocation for CNN computing in embedded systems. Wireless communication considerations are included only in review articles, where they identify aspects of interest for further investigation. In contrast, our DSE method addresses the knowledge gap related not only to DSE for data intensive IoT nodes, but it also includes processing and communication interactions in the DSE method. In addition, it provides an energy budgeting approach that can be used when exploring optimisation or addition of tasks in the processing tasks for a given application.

VII. CONCLUSION

In this article we present a DSE method for data intensive IoT nodes, which relies on high level estimates such as data flow and number of operations. Based on considerations of processing and communication interaction, this method facilitates the analysis of which areas of the design we should focus optimisation efforts, how much energy would be available for additional processing, and whether we should implement the given tasks in the smart sensor node or in the cloud to optimise its energy consumption. We applied this method on 4 design cases from traditional and CNN image processing systems, showing how variations in the energy and communication constraints would affect design choices. In a comparison of our DSE method to other methods we underlined the importance of our method in addressing the knowledge gap in DSE for IoT nodes.

REFERENCES

[1] G. Bedi, G. K. Venayagamoorthy, R. Singh, R. R. Brooks, and K.-C. Wang, "Review of Internet of Things (IoT) in electric power and energy systems," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 847–870, Apr. 2018.

[2] J. M. Talavera, L. E. Tobón, J. A. Gómez, M. A. Culman, J. M. Aranda, D. T. Parra, L. A. Quiroz, A. Hoyos, and L. E. Garreta, "Review of IoT applications in agro-industrial and environmental fields," *Comput. Electron. Agricult.*, vol. 142, pp. 283–297, Nov. 2017.

[3] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Gener. Comput. Syst.*, vol. 87, pp. 278–289, Oct. 2018, doi: 10.1016/j.future.2018.04.057.

[4] F. Abate, M. Carratù, C. Liguori, and V. Paciello, "A low cost smart power meter for IoT," *Measurement*, vol. 136, pp. 59–66, Mar. 2019, doi: 10.1016/j.measurement.2018.12.069.

[5] M. Gusev and S. Dustdar, "Going back to the roots—The evolution of edge computing, an IoT perspective," *IEEE Internet Comput.*, vol. 22, no. 2, pp. 5–15, Mar. 2018.

[6] E. AB, "Cellular networks for massive IoT," Ericsson AB, White Paper Uen 284 23-3278, Jan. 2016, p. 13. [Online]. Available: https://www.ericsson.com/res/docs/whitepapers/wp_iot.pdf

[7] I. Shallari, S. Krug, and M. O'Nils, "Communication and computation inter-effects in people counting using intelligence partitioning," *J. Real-Time Image Process.*, vol. 17, no. 6, pp. 1869–1882, Dec. 2020, doi: 10.1007/s11554-020-00943-6.

[8] S. Hengstler, D. Prashanth, S. Fong, and H. Aghajan, "MeshEye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance," in *Proc. 6th Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2007, pp. 360–369.

[9] M. Birem and F. Berry, "DreamCam: A modular FPGA-based smart camera architecture," *J. Syst. Archit.*, vol. 60, no. 6, pp. 519–527, Jun. 2014.

[10] A. Raha and V. Raghunathan, "Towards full-system energy-accuracy tradeoffs: A case study of an approximate smart camera system," in *Proc. 54th Annu. Design Autom. Conf.*, Jun. 2017, pp. 1–6.

[11] D. Chen, J. Cong, S. Gurumani, W. Hwu, K. Rupnow, and Z. Zhang, "Platform choices and design demands for IoT platforms: Cost, power, and performance tradeoffs," *IET Cyber-Phys. Syst., Theory Appl.*, vol. 1, no. 1, pp. 70–77, Dec. 2016.

[12] A. D. Pimentel, "Exploring exploration: A tutorial introduction to embedded systems design space exploration," *IEEE Des. Test. Comput.*, vol. 34, no. 1, pp. 77–90, Feb. 2017.

[13] I. Shallari. (2021). *Example of Design Space Reduction Method Using Intelligence Partitioning*. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-41031, 2021.

[14] A. Souri, A. Hussien, M. Hoseyninezhad, and M. Norouzi, "A systematic review of IoT communication strategies for an efficient smart environment," *Trans. Emerg. Telecommun. Technol.*, pp. 1–19, Aug. 2019, doi: 10.1002/ett.3736.

[15] A. Shukla and S. Tripathi, *An Effective Relay Node Selection Technique for Energy Efficient WSN-Assisted IoT*. New York, NY, USA: Springer, 2020, Art. no. 0123456789, doi: 10.1007/s11277-020-07167-8.

[16] F. Vannieuwenborg, S. Verbrugge, and D. Colle, "Choosing IoT-connectivity? A guiding methodology based on functional characteristics and economic considerations," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 5, pp. 1–16, 2018.

[17] M. Asghari, S. Yousefi, and D. Niyato, "Pricing strategies of IoT wide area network service providers with complementary services included," *J. Netw. Comput. Appl.*, vol. 147, Dec. 2019, Art. no. 102426, doi: 10.1016/j.jnca.2019.102426.

[18] G. C. Deepak, F. Bouhafs, A. Raschellà, M. Mackay, and Q. Shi, "Radio resource management framework for energy-efficient communications in the Internet of Things," *Trans. Emerg. Telecommun. Technol.*, vol. 30, no. 12, pp. 1–17, 2019.

[19] R. Sanchez-Iborra and M.-D. Cano, "State of the art in LP-WAN solutions for industrial IoT services," *Sensors*, vol. 16, no. 5, p. 708, May 2016.

[20] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[21] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[22] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim, "Compressed sensing for wireless communications: Useful tips and tricks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1527–1550, 3rd Quart., 2017.

[23] G. Yang, S. Yu, H. Dong, G. Slabaugh, P. L. Dragotti, X. Ye, F. Liu, S. Arridge, J. Keegan, Y. Guo, and D. Firmin, "DAGAN: Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1310–1321, Jun. 2018.

[24] R. Fallahzadeh, J. P. Ortiz, and H. Ghasemzadeh, "Adaptive compressed sensing at the fingertip of Internet-of-Things sensors: An ultra-low power activity recognition," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 996–1001.

[25] C. Habib, A. Makhoul, R. Darazi, and R. Couturier, "Real-time sampling rate adaptation based on continuous risk level evaluation in wireless body sensor networks," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2017, pp. 1–8.

[26] I. Yoon and D. Noh, "Energy-aware control of data compression and sensing rate for wireless rechargeable sensor networks," *Sensors*, vol. 18, no. 8, p. 2609, Aug. 2018.

[27] D. Laiymani and A. Makhoul, "Adaptive data collection approach for periodic sensor networks," in *Proc. 9th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jul. 2013, pp. 1448–1453.

[28] G. B. Tayeh, A. Makhoul, D. Laiymani, and J. Demerjian, "A distributed real-time data prediction and adaptive sensing approach for wireless sensor networks," *Pervas. Mobile Comput.*, vol. 49, pp. 62–75, Sep. 2018, doi: 10.1016/j.pmcj.2018.06.007.

[29] M. Dong, K. Ota, and A. Liu, "RMER: Reliable and energy-efficient data collection for large-scale wireless sensor networks," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 511–519, Aug. 2016.

[30] H. Harb, A. Makhoul, and C. A. Jaoude, "A real-time massive data processing technique for densely distributed sensor networks," *IEEE Access*, vol. 6, pp. 56551–56561, 2018.

[31] K. Abas, K. Obraczka, and L. Miller, "Solar-powered, wireless smart camera network: An IoT solution for outdoor video monitoring," *Comput. Commun.*, vol. 118, pp. 217–233, Mar. 2018.

[32] J. Fernández, L. Calavia, C. Baladrón, J. Aguiar, B. Carro, A. Sánchez-Esguevillas, J. Alonso-López, and Z. Smilansky, "An intelligent surveillance platform for large metropolitan areas with dense sensor deployment," *Sensors*, vol. 13, no. 6, pp. 7414–7442, Jun. 2013.

[33] I. Shallari and M. O'Nils, "From the sensor to the cloud: Intelligence partitioning for smart camera applications," *Sensors*, vol. 19, no. 23, p. 5162, Nov. 2019.

[34] S. Krug and M. O'Nils, "Modeling and comparison of delay and energy cost of IoT data transfers," *IEEE Access*, vol. 7, pp. 58654–58675, 2019.

[35] *nRF52832 Product Specification v1.4*, Nordic Semiconductor, Trondheim, Norway, 2017.

[36] *SARA-G3 Series Dual and Quad-Band GSM/GPRS modules*, ublox, Thalwil, Switzerland, 2017.

[37] D. De Guglielmo, S. Brienza, and G. Anastasi, "IEEE 802.15.4e: A survey," *Comput. Commun.*, vol. 88, pp. 1–24, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366416301980, doi: 10.1016/j.comcom.2016.05.004.

[38] *ATWINC15x0-MR210xB IEEE 802.11 b/g/n SmartConnect IoT Module*, Microchip Technology,Chandler, AZ, USA, 2017.

[39] *SARA-U2 Series HSPA Modules With 2G Fallback*, ublox, Thalwil, Switzerland, 2018.

[40] *SARA-N2 Power-Optimized NB-IoT (LTE Cat NB1) Modules*, ublox, Thalwil, Switzerland, 2017.

[41] *TOBY-L2 Series Multi-Mode LTE Cat 4 Modules With HSPA+ and/or 2G Fallback*, ublox, Thalwil, Switzerland, 2017.

[42] *SX1276/77/78/79–137 MHz to 1020 MHz Low Power Long Range Transceiver*, Semtech, Camarillo, CA, USA, 2015.

[43] *LARA-R2 Series Size-Optimized LTE Cat 1 Modules in Single and Multi-mode Configurations*, ublox, Thalwil, Switzerland, 2017.

[44] I. Shallari, S. Krug, and M. O'Nils, "Architectural evaluation of node: Server partitioning for people counting," in *Proc. 12th Int. Conf. Distrib. Smart Cameras*, Sep. 2018, pp. 1–6.

[45] M. Imran, K. Khursheed, N. Ahmad, A. W. Malik, M. O'Nils, and N. Lawal, "Complexity analysis of vision functions for implementation of wireless smart cameras using system taxonomy," *Proc. SPIE*, vol. 8437, May 2012, Art. no. 84370C.

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1106–1114.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent., ICLR Conf. Track*, 2015, pp. 1–14.

[48] S. Jain, S. Venkataramani, V. Srinivasan, J. Choi, P. Chuang, and L. Chang, "Compensated-DNN: Energy efficient low-precision deep neural networks by compensating quantization errors," in *Proc. 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, Jun. 2018, pp. 1–6.

[49] M. Dejean-Servières, K. Desnos, K. Abdelouahab, W. Hamidouche, and L. Morin, *Study of the Impact of Standard Image Compression Techniques on Performance of Image Classification With a Convolutional Neural Network*. Rennes, France: INSA Rennes; Univ Rennes; IETR; Institut Pascal, 2017.

[50] E. Aerabi, M. Bohlouli, M. H. A. Livany, M. Fazeli, A. Papadimitriou, and D. Hely, "Design space exploration for ultra-low-energy and secure IoT MCUs," *ACM Trans. Embedded Comput. Syst.*, vol. 19, no. 3, pp. 1–34, Jul. 2020.

[51] P. van Stralen and A. Pimentel, "Fitness prediction techniques for scenario-based design space exploration," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 8, pp. 1240–1253, Aug. 2013.

[52] F. Tsimpourlas, L. Papadopoulos, A. Bartsokas, and D. Soudris, "A design space exploration framework for convolutional neural networks implemented on edge devices," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2212–2221, Jul. 2020.

[53] R. Niemann and P. Marwedel, "An algorithm for hardware/software partitioning using mixed integer linear programming," *Des. Autom. Embedded Syst.*, vol. 2, no. 2, pp. 165–193, Mar. 1997.

IRIDA SHALLARI received the B.Sc. degree from the University of Tirana, Albania, in 2014, the dual M.Sc. degree in European masters' in embedded computing systems from the University of Southampton, U.K. and Norwegian University of Science and Technology (NTNU), in 2016, and the Licentiate degree from Mid Sweden University, Sweden, in 2019, where she is currently pursuing the Ph.D. degree. Her research interest includes embedded systems design and optimisation, focusing on vision systems and smart camera node's optimisation for IoT deployments .

ISAAC SÁNCHEZ LEAL received the B.Hons. degree in aeronautics and mechanical engineering from Glyndwr University, U.K., in 2015, the B.Hons. degree in electronics and industrial automation engineering from Universidad de Alcalá, Spain, in 2016, and the M.Sc. degree in electronics design from Mid Sweden University, Sweden, in 2018. He is currently pursuing the Ph.D. degree with the Embedded Systems Design Research Group, Mid Sweden University. His research interest includes computer vision systems within the IoT technology.

SILVIA KRUG received the B.Sc. degree in computer science from the University of Applied Sciences, Darmstadt, Germany, and the M.Sc. degree in computer engineering and the Dr.Ing. degree in communication networks from Technische Universtät Ilmenau, Germany, in 2013 and 2017, respectively. She currently holds a postdoctoral position with the Embedded Systems Design Research Group, Mid Sweden University, Sweden. Her research interests include communication aspects for wireless sensor systems with a focus on energy-efficiency and design considerations for the distributed IoT systems.

AXEL JANTSCH (Senior Member, IEEE) received the Dipl.Ing. degree and the Ph.D. degree in computer science from TU Wien, Vienna, Austria, in 1987 and 1992, respectively. From 1997 to 2002, he was an Associate Professor with the KTH Royal Institute of Technology, Stockholm. From 2002 to 2014, he was a Full Professor in electronic systems design at KTH. Since 2014, he has been a Professor of systems on chips with the Institute of Computer Technology, TU Wien. He has published five books as an editor and one as an author and over 300 peer-reviewed contributions in journals, books, and conference proceedings. His current research interests include systems on chips, self-aware cyber-physical systems, and embedded machine learning. He has given over 100 invited presentations at conferences, universities, and companies.

MATTIAS O′NILS (Member, IEEE) received the B.S. degree in electrical engineering from Mid Sweden University, Sundsvall, Sweden, in 1993, and the Licentiate and Ph.D. degrees in electronic systems design from the Royal Institute of Technology, Stockholm, Sweden, in 1996 and 1999, respectively. He is currently a Professor with the Department of Electronics Design and leads the Embedded Systems Design Research Group, Mid Sweden University. His current research interests include design methods and implementation of embedded systems, especially in the implementation of real-time video processing systems.