Binucci *et al.*
EURASIP Journal on Advances in Signal Processing (2022) 2022:123
https://doi.org/10.1186/s13634-022-00958-0

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Adaptive resource optimization for edge inference with goal-oriented communications

Francesco Binucci[1*], Paolo Banelli[1], Paolo Di Lorenzo[2] and Sergio Barbarossa[2]

*Correspondence: francesco.binucci@studenti.unipg.it

[1] University of Perugia, Perugia, Italy
[2] Sapienza University of Rome, Rome, Italy

Abstract

Goal-oriented communications represent an emerging paradigm for efficient and reliable learning at the wireless edge, where only the information relevant for the specific learning task is transmitted to perform inference and/or training. The aim of this paper is to introduce a novel system design and algorithmic framework to enable goal-oriented communications. Specifically, inspired by the information bottleneck principle and targeting an image classification task, we dynamically change the size of the data to be transmitted by exploiting banks of convolutional encoders at the device in order to extract meaningful and parsimonious data features in a totally adaptive and goal-oriented fashion. Exploiting knowledge of the system conditions, such as the channel state and the computation load, such features are dynamically transmitted to an edge server that takes the final decision, based on a proper convolutional classifier. Hinging on Lyapunov stochastic optimization, we devise a novel algorithmic framework that dynamically and jointly optimizes communication, computation, and the convolutional encoder classifier, in order to strike a desired trade-off between energy, latency, and accuracy of the edge learning task. Several simulation results illustrate the effectiveness of the proposed strategy for edge learning with goal-oriented communications.

Keywords: Edge learning, Goal-oriented communications, Lyapunov stochastic optimization, Convolutional encoders

1 Introduction

The fifth generation (5G) of wireless networks already represents a breakthrough in wireless communication networks, as a single platform that enables a variety of services, such as enhanced mobile broadband communications, immersive experiences such as virtual and augmented reality, mission-critical communications via ultra-reliable low-latency links, and enabling also the remote control of critical infrastructures, such as Industry 4.0, autonomous driving, and massive Internet of things (IoT). Yet, even though 5G is still in its deployment phase, there is already a clear trend toward a sixth-generation (6G) system, which will come out of the fundamental merge between artificial intelligence (AI) and information and communication technologies (ICT). 6G networks are expected to be *AI-native*, meaning that AI tools will not be just services running on the communication platform, but they will rather represent the building blocks of the network itself, from the physical up to the network and application layers [1, 2]. This merge

Springer Open

© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

will give rise to more autonomous, i.e., zero-touch, networks enabling a truly pervasive deployment of intelligent services, subject to a variety of constraints, in terms of learning and inference reliability, latency, and energy consumption.

The need to tightly control latency and limit energy consumption motivates the shift toward *edge intelligence* (EI) architectures [3], where the information exchange and processing are kept as local as possible. In the EI framework, every device may have access only to a tiny fraction of the data and low-latency inference/training tasks need to be performed collectively and distributively at the wireless network edge.

An efficient design of the EI platform calls for the adoption of a *holistic approach*, where communication, computation, learning, and control are jointly orchestrated to achieve new target levels of reliability, energy efficiency, and sustainability. This trend motivates the current widespread interest in distributed, low-latency and reliable machine learning (ML) tools, calling for a major departure from cloud-based, centralized training and inference. In EI, the mobile devices, also called user equipment (UE), need to perform AI/ML tasks by partially offloading their computations to edge servers (ESs), placed at the edge of the wireless network [4]. The overall system must be then designed in order to achieve an optimal balance between accuracy of the ML tasks and usage of the network resources, by dynamically allocating transmission and computational parameters, such as transmission rates and central processing unit (CPU) clock frequencies, as well as the scheduling of transmission and computation tasks, possibly under uncertainties about the wireless channel state and task arrival rates.

The resource optimization problems formulated in this scenario are mainly focused on the trade-off between energy, latency and learning accuracy [5–9]. However, looking at the predictions about the exponential increase in traffic in next-generation networks [10], it is evident that it is time to envisage a new communication paradigm able to support EI while preventing the data rate explosion. A possible paradigm shift in this direction may come from *semantic* and *goal-oriented* communications (GOC) [11]. In this new context, the focus is not anymore on the reliable recovery of the transmitted bits, but instead on the meaning (semantics) conveyed by those bits or the goal motivating the transmission of bits.

It is clear that EI and GOCs find application whenever we have a set of devices characterized by limited capabilities that need to perform specific tasks timely and with prescribed requirements in term of reliability. In vehicular edge computing (VEC) scenarios [12], for instance, we can imagine that the UEs installed onboard the vehicles need to perform an object classification task (e.g., detection of traffic signs in the scene) or collision avoidance/pose estimation. If, due to the limited resources, the devices are not able to timely perform the task with the required quality, they may ask an edge server to perform the task and send back the outcome. It is clear that a smart communication scheme, capable to extract and transmit only the data that are *relevant* to the task, would be very attractive both energy-wise and delay-wise.

On the other hand, also IoT scenarios represent a noticeable field where EI is widely deployed [13]. As an example, we can imagine decentralized estimation tasks, possibly based on energy harvesting devices [14], where smart compression schemes are fundamental to parsimoniously offload data toward the edge cloud, in order to save as much transmission resources as possible while guaranteeing negligible estimation error

degradation. Another application of EI that would benefit of a GOC architecture is real-time automatic video surveillance, where there is a continuous flow of video data that must be processed timely by specific inference models (e.g., neural networks). Also in this case, a fully local deployment at the IoT device may be impractical or impossible, making offloading a valuable solution to guarantee the service [13].

The goal of this work is to propose a dynamic communication strategy and an optimal allocation of all the network resources, including communication, computational, and ML resources, in order to implement a dynamic goal-oriented scheme, which aims to transmit only the data that are *informative* to the fulfillment of the specified goal (e.g., image recognition), under constraints on decision accuracy, service delay, and energy consumption.

1.1 Related works

Deep neural networks (DNNs) have already been proposed to design a joint source/channel coding (JSCC), as an alternative to the conventional cascade of source and channel encoders, to achieve superior performance in the finite block-length regime for image retrieval applications [15]. Designing the JSCC encoder focusing directly on the recognition accuracy rather than performing image reconstruction and then classification separately, was investigated in [16]. In [17], the authors proposed an image retrieval scheme where, instead of sending the image, the feature vectors are first extracted and then mapped into channel input symbols, while the noisy channel output is used by the server to retrieve the most relevant images, without involving any explicit channel code. This approach has been extended in [18], where the encoder outputs are quantized prior to the mapping on the channel symbols, while in [19] a deep-JSCC with channel output feedback exploitation is proposed. In contrast to most of the works, which consider AWGN channels, the authors in [20] design a communication scheme for flat-fading channels based on an OFDM system. Other interesting work can be found in [21] and [22], where a combination of JSCC and nonlinear transform coding (NTC) [23] is proposed.

In applications such as text transmission, the semantics underlying the text has been also explicitly exploited in designing a JSCC, such as in [24] and [25], where a noise-aware JSCC system is described. The authors of [26] designed speech recognition-oriented semantic communications to directly recognize the speech signals into texts. The work in [27] exploits a hybrid automatic repeat request (HARQ) scheme in order to improve reliability in sentence semantic transmission. Semantic communications for multimodal data were considered in [28] for serving the visual question answering problem, which adopts long short-term memory for text transmission and a convolutional neural network for the image transmission. More recently, a transformer-based approach has also been investigated in [29] to support both image and text transmission. Alternative methods were also proposed in [30] and [31], to define an optimized *common-language* between a listener and a speaker, employing reinforcement learning (RL) and curriculum learning (CL). Other interesting examples can be found in [32] and [33], concerning, respectively, image classification in an unmanned aerial vehicle (UAV) scenario and visual question answering (VQA) tasks.

A more principled approach, based on the information bottleneck principle [34, 35], to limit transmission only to the information that is *relevant* for the intended goal of communication, was recently proposed in [36], for the Gaussian case, and in the more general case in [37], using a variational IB (VIB) approach. Recently, VIB has also been considered for multi-device cooperative edge inference [38]. Some rate distortion approaches were also proposed in [39] and in [40] to support goal-oriented communications.

In all the above works, except [36], the focus was on the communication system, but without optimizing the usage of the available resources, namely communication, computational, and semantic-related resources. Resource optimization has been considered in [41] and [5, 6]. Specifically, the authors in [41] propose to tune the GOC resources, e.g., bandwidths and powers, as well as the size of the goal-oriented compressed representation of the data, in order to optimize the success probability of the task under flat-fading zero-mean Gaussian channels. This optimization, which includes training of the compressive and classification (C C) architecture, and the choice of the data compression ratio, is performed once, by exploiting knowledge of the average statistics (e.g., standard deviation) of the flat-fading channel. The optimal bandwidths and transmission powers obtained this way, likewise the C&C neural network architecture and the compression ratio, are fixed for a given scenario (average SNR, etc.), and they are used over all the possible channel states that the GOC system may experience. This fixed allocation of both resources and C&C architecture, is a distinctive difference with respect to our approach, where we dynamically adapt all the energy and hardware resources, according to the system state, as we will further clarify.

Conversely, the *dynamic* analysis and optimization of the trade-offs between decision accuracy, overall (i.e., transmission and computation) energy consumption and service delay, has been considered in [5, 6], where the trade-off is achieved by dynamically adapting the source encoding rate and the *scheduling* of transmission and computation tasks. This approach was recently extended in [42], for energy-efficient edge classification with reliability guarantees, in [43] for ensemble inference at the edge, and in [36] by incorporating the information bottleneck principle to identify and transmit only the information relevant to the task. *Contributions* In this paper, we focus on the *dynamical* joint management and optimization of computation, communication, and semantic-extracting resources of a GOC system, where transmitter and receiver architectures incorporate a pair of *variable size* convolutional encoders (CE) and classifiers (CC). A finite set of CE/CC pairs, each having a variable dimension of the CE output, is pre-trained *offline*, to make possible the selection of the most suitable pair to be used *online*, depending both on how well the overall communication system is fulfilling the goal and on the constraints of the communication link. The proposed communication scheme is reported in Fig. 1, where, inspired by the IB principle, the *bottleneck* is made time-varying, by adaptively selecting in each time slot the most suitable CE/CC pair, according to a strategy resulting from the solution of two possible constrained optimization problems: i) minimum energy consumption, under average service delay and accuracy constraints strategy (MEDA); ii) maximum accuracy under average service delay and energy constraints strategy (MADE). This is

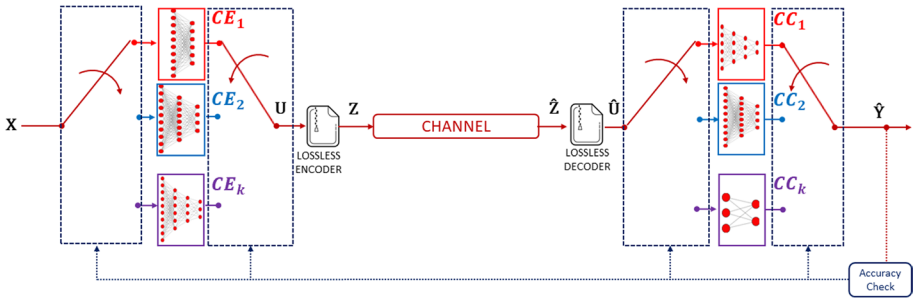

Fig. 1 Goal-oriented communication scheme. This figure shows the goal-oriented communication scheme employed in our scenario, composed of a CE, a lossless compression system (e.g., JPEG2000) from the UE perspective and a set of CCs from the ES perspective

significantly different from the static optimization proposed in [41], where scheduling (and buffering) is not considered as a fundamental ingredient to make best use of the available resources in a dynamic fashion. Furthermore, we adapt to the buffer load and channel condition the assignment of computation and transmission resources, as well as the size of the compressed data, by a dynamic choice of the proper CE/CC pair at each time slot. Note that this dynamic use of multiple low-complexity CE/CC (neural network) pairs, makes our approach quite different from [41] and the recent literature on semantic and GOC [15, 17, 19, 33, 44], where typically a fixed (very complex) DNN architecture is split among transmitter and receiver. The single DNN architecture that these GOC schemes have to train is (typically) very complex because it has to work well for a variety of state channels, noise levels, and required task performance, which the GOC may have to face. Conversely, in our case, we train a set of NNs, where each NN is much simpler because it is well matched to (and will be used with) a much more restricted variety of conditions. In particular, each NN has a different output size (the bottleneck) and we adapt the bottleneck dimension online to optimize performance.

To address the dynamic management of the overall goal-oriented architecture, we hinge on Lyapunov stochastic optimization tools [6, 45], which implement the solution in a time-slotted fashion. Specifically, in each time slot, we perform a deterministic optimization of the involved variables, valid also in the general situation where some of the involved variables, such as the channel state and the task arrival rates, are random, with unknown probability distribution. Under proper feasibility conditions, the proposed approach is shown to achieve the optimal solution, while respecting the given constraints. The simulation results confirm the effectiveness of the proposed approach to manage the system resources in an adaptive way and strike an optimal trade-off between average energy, delay, and accuracy.

Outline The paper is organized as follows. In Sect. 2, we present the scheme of our goal-oriented communication system, including the joint training procedure of the CE/CC pair, assuming as goal the classification of the images sent by the UE. In Sect. 3, we introduce the overall system model supporting the offloading of the learning task from the UE to the ES, defining all quantities of interest, e.g., latency, learning accuracy, and

energy, involved in the proposed resource optimization problems, which are then solved in Sect. 4, exploiting stochastic Lyapunov optimization. Section 5 presents the simulation results, and, finally, Sect. 6 draws some conclusions and highlights future research directions.

2 Proposed design of goal-oriented communications

We consider as an example of application of our proposed strategy, the transmission of images from a UE to an ES, where the goal is image classification. The key point of the proposed approach is to exploit knowledge of the system state (channel condition, computation load, buffer load, etc.) to dynamically compress the images to be transmitted, and then classified, using a GOC perspective, where *the goal is not to recover the image at the receiver side, but only to achieve the desired classification accuracy*.

To this end, the inspiring principle is the Information Bottleneck framework, whose purpose is to find a (probabilistic) compact representation U of the random variable X emitted by a source, in order to preserve as much information as possible about the classification output variable Y, while minimizing the complexity associated with the representation of X through U. The IB is based on the following functional optimization problem (in Lagrangian form) [34]:

$$\underset{p(h|x)}{\text{minimize}} \quad I(X;U) - \beta\, I(U;Y), \quad \beta \geq 0, \tag{1}$$

where the mutual information $I(X;U)$ represents the *complexity*, in terms of number of bits used to represent X by U; the term $I(U;Y)$ represents the *relevance* of U in conveying information about Y; β is the parameter used to control the trade-off between complexity and relevance. Since problem (1) depends on the (joint) probability density function (pdf) of X, U, and Y, the optimal solution can be found only in specific cases, e.g., when the involved random variables are either Gaussian [46] or discrete. In the latter case, the solution is known only in an iterative form [34]. However, except for the Gaussian case, (1) is quite difficult to solve in practice, especially when the dimension of the data X is very large, as it happens with images [47].

Due to the aforementioned issues, in this work we pursue a simpler approach that, while it is inspired by the IB principle in (1) and the associated GOC scheme for the Gaussian case [36], it implements a practical goal-oriented communication scheme that performs a tunable data compression at the UEs, using a convolutional encoder that is trained offline to learn how to extract the *relevant* information necessary to achieve the accuracy of the inference task, while consuming the minimum amount of resources by properly compressing the input data. Since we focus on image classification, we choose the structure of both the encoder and decoders as two convolutional neural networks, incorporating a layer-by-layer max-pooling strategy [48] to adapt the dimension of the data to be transmitted. The pictorial scheme of the proposed goal-oriented communication scheme is illustrated in Fig. 1.

The design of the CEs has been driven by two main strategies:

- *Short-CE* The compression is obtained by using a single convolutional layer, followed by a max-pooling layer, which directly implements the desired compression factor.
- *Deep-CE* The compression is obtained by cascading a set of convolutional layers, each one followed by a max-pooling step that implements a compression factor equal to 2. The number of layers n_l to be used is imposed by the total compression factor ρ that is desired at the output, e.g., $\rho = 2^{n_l}$.

It is worth to emphasize that the architecture of the CNN that we are using is not necessarily optimal. There certainly exist alternative architectures that may perform better, although, as we do in our resource management, the ultimate classification performance should always take into account complexity and energy expenditure, which may be critical for mobile and simple UEs. Thus, the reason underlying our choice is simply dictated by the request of having a few simple alternative architectures that make possible to keep the complexity and energy spent for processing at the devices as small as possible.

The training of each CE/CC pair, at the UE and ES sides, has been performed *jointly* and *offline*, as a solution of the following problem

$$\underset{\phi_\rho,\theta_\rho}{\text{minimize}} \quad \frac{1}{N_t}\sum_{n=1}^{N_t} L_{ce}(Y_n, \widehat{Y_n}; \phi_\rho, \theta_\rho), \tag{2}$$

where L_{ce} is a suitable loss function, while θ_ρ and ϕ_ρ represent the parameters of the CNNs used at the CE and CC, respectively, for a given compression (bottleneck) parameter ρ, and N_t is the size of the training set. More specifically, dealing with a multi-class classification task, we used the *categorical cross-entropy* as the loss function, so that L_{ce} reads as [49]:

$$L_{ce}(Y_n, \hat{Y}_n, \phi_\rho, \theta_\rho) = -\sum_{k=1}^{K} Y_k(X_n)\ln(\hat{Y}_k(X_n, \phi_\rho, \theta_\rho)), \tag{3}$$

where K is the number of classes, $Y_k(X_n) \in \{0, 1\}$ are the hot-coded true probabilities, i.e., those identifying the ground-truth labels, for the k-th class and n-th training sample; whereas, $\hat{Y}_k(X_n, \phi_\rho, \theta_\rho)$ are the soft probabilities estimated at the output of the classification network, i.e., those generating the predicted labels.

A key feature of the IB formulation in (1) is that the balance between complexity and relevance of the compressed representation U is tuned by acting on the trade-off parameter β. In our setup, this balance is tuned by acting on the dimension of the CE/CC pair, as depicted in Fig. 1. Hence, the architecture used in each time slot to encode the images and extract the relevant information is selected, slot-by-slot, depending on the service delay and accuracy constraints, as a function of the current values of the system parameters, such as wireless channel state and data arrivals. We remark that the training procedure is performed *offline*, while the selection of the most suitable architecture to be used in each time slot is performed in a dynamic fashion according to the criteria described in the next section.

1 ***Remark 1***

While the IB looks for a probabilistic mapping of the data source X to the compressed representation U [36, 46], in our setup, the mapping is deterministic. Nevertheless, the proposed training scheme has an important link with the IB principle, as it was proved that the $L_{ce}(Y_n, \widehat{Y_n})$ *is a good proxy for the mutual information I(U; Y) [50]. In particular, minimizing the cross-entropy loss (over the training set) leads to the maximization of the I(U; Y) of a deterministic mapping. Furthermore, IB arguments can be used to explain the performance of a deep neural network trained by a cross-entropy loss [51], which further motivates why the IB represents an information-theoretic justification of our practical procedure. In principle, we could also make our compression law probabilistic by adding noise in the encoding step as well as in the training phase, as this has been recognized as a method to improve the generalization capability of a CNN and reduce the overfitting errors [52].*

1 ***Remark 2***

Differently from works inspired by JSCC where the encoders directly map the input data to the symbols to be transmitted [15, 17, 19, 20, 37], we foresee a more traditional approach, where after compression we transmit bits over a conventional, capacity-achieving communication link, which makes use of ideal channel co-decoding, i.e., with zero bit error rate (BER). Although certainly interesting, we leave for future work the quantification, and proper handling, of the impact on classification accuracy of a residual BER in the communication link, due for instance to finite-length channel coding, where also JSCC schemes find their motivations.

Specifically, we split the encoder in a convolutional encoder (CE) followed by a lossless compression, as depicted in Fig. 1, where the compression is obtained using the lossless JPEG2000 and TIFF codecs. We follow this strategy for the sake of simplifying the overall adaptive strategy that selects, slot-by-slot, the most suitable communication architecture, and to enable an easy control at each time slot of the specific dimension of the (goal-oriented) data that have to be transmitted for every image, depending on how well the system is behaving in terms of balance between classification accuracy, service delay and energy consumption.

The relation between the (data) compression ratio ρ[1] to choose from, the dimension of the CE output and the size (number of bits) of the data to be transmitted, before and after compression, is reported in Table 1. The values for lossless compression for $\rho = 32, 64$ are not-available (N/A), since the overhead due to the zipping algorithm is higher than the file size reduction. We remark that state-of-the-art lossless compression after the CE at the UE allows us to save information bits to be transmitted, without impacting the overall accuracy granted by the offline training of the proposed CE/CC structure, under capacity-achieving ideal assumptions. Obviously, the price to be paid

[1] Note that ρ represents the compression of the image on each dimension, thus the actual data compression ratio scales with ρ^2.

Table 1 Image size at CE output with and w/o zipping

ρ	$M(\rho)$ [*px*]	Size [*kB*]	Zipped-Size [*kB*]
2	128 × 128 × 3	47.92	6.69
4	64 × 64 × 3	12.12	3.49
8	32 × 32 × 3	3.12	1.77
16	16 × 16 × 3	0.87	0.85
32	8 × 8 × 3	0.31	N/A
64	4 × 4 × 3	0.17	N/A

is a higher computational complexity of the system, which has also to perform the lossless decompression at the ES before feeding the convolutional classifier. We will take into account this computational complexity, as well as the associated delay and energy expenditure, in the resource management policies and optimization.

3 System model

The envisaged goal-oriented communication scenario includes an UE, with limited computational (or energy) capabilities, which is connected to an ES with higher computational resources and energy, through a wireless link with an access point (AP). The overall scheme is depicted in Fig. 1. We focus on image classification at the edge, assuming a pre-trained set of goal-oriented CE-CC schemes, as described in Sect. . We assume that the system state evolves in a time-slotted fashion with time-varying context parameters (i.e., wireless channels and data arrivals); each time slot t has a fixed duration τ. In our procedure, data (i.e., images) are generated/collected at the UE, with an arbitrary distribution of the arrival time, and uploaded to an ES for inference purposes. In particular, we design a procedure where data are: (i) collected and buffered locally at the device; (ii) encoded in a goal-oriented fashion, zipped, and transmitted; (iii) remotely buffered and processed by the ES for classification.

The goal of our optimization procedures is to provide inference results within a finite E2E delay considering: (i) the minimum energy consumption at the mobile device, under a prescribed inference reliability and decision delay; (ii) the maximum accuracy for a given energy consumption and delay. In this context, several resources must be optimized and adapted over time depending on dynamic system conditions, e.g., wireless channels, data arrivals, and buffered images. In particular, the UE must select its *transmission rate* $R(t)$ toward the ES, its local computational *clock frequency* $f_d(t)$, as well as the data *compression factor* $\rho(t)$, to generate the compressed latent representation U. At the same time, the ES has to allocate its computational *clock frequency* $f_c(t)$ in order to complete the specific learning task, i.e., image classification. The above quantities represent the optimization variables for the proposed resource allocation strategies. In the sequel, we illustrate the adopted model for latency, energy, and classification accuracy.

3.1 Latency model

The dynamicity of the system is modeled using queues, which are also used to control the overall delay of the service. In particular, our model involves two queues:

- A *compression/communication* queue at the UE.
- A *computation* queue at the ES.

In the sequel, we introduce some important assumptions for the resource optimization problem we are going to design:

Assumption 1 Each data unit must be compressed and transmitted by the UE in the same time slot. It is indeed impossible to choose in advance the optimal compression factor for a data unit that would have to be stored and transmitted in the future, unless we could reliably predict also the future system state (e.g., the wireless channel condition, energy status, queue lengths, computational power, etc.) at the time slot the data unit would be actually transmitted. Therefore, compression and transmission operations must be done sequentially within the same time slot.

Assumption 2 We assume that, while the UE transmits some data units, it may also simultaneously compress other data units.

The maximum number of data units that could be transmitted at the t-th time slot is expressed by

$$N_{tx}(t) = \left\lfloor \frac{\tau R(t)}{W(\rho(t))} \right\rfloor, \tag{4}$$

where $R(t)$ and $\rho(t)$ are, respectively, the transmission rate and the compression factor chosen by the device for such a time slot, and $W(\rho(t)) = M(\rho(t))N(\rho(t))$ is the average number of bits per data unit. $M(\rho(t))$ is the data unit size (in pixels) for the compression factor $\rho(t)$, and $N(\rho(t))$ is the associated number of bits that are necessary (on average) to encode a pixel in the compressed and encoded pseudo-image, that we will detail in the simulation results. On the other hand, the number of data units that is possible to compress during time slot t is given by

$$N_c(t) = \lfloor \tau f_d(t) J_d(\rho(t)) \rfloor, \tag{5}$$

where $J_d(\rho(t))$ denotes the number of data units compressed in a clock cycle (which depends on the chosen compression factor $\rho(t)$), while $f_d(t)$ denotes the clock frequency chosen by the UE during the t-th time slot. By Assumption 1 and 2, the UE cannot transmit more data units that can also (simultaneously) compress, which suggests that in (4) we have to use a rate $R(t) \leq W(\rho(t)) f_d(t) J_d(\rho(t))$ such that $N_{tx}(t) \leq N_c(t)$. Furthermore, although we are assuming parallel compression and transmission of (the previously compressed) data units, the very first data unit needs a time $1/(f_d(t)J_d(\rho(t)))$ to be compressed before transmission can start. This means that the number $N_{UE}(t)$ that the UE can actually transmit and compress in a time slot is given by

$$\begin{aligned} N_{UE}(t) &= \left\lfloor \frac{\tau - 1/(f_d J_d(\rho(t)))}{W(\rho(t))/R(t)} \right\rfloor \\ &= \left\lfloor \frac{\tau R(t)}{W(\rho(t))} - \frac{R(t)}{W(\rho(t)) f_d(t) J_d(\rho(t))} \right\rfloor. \end{aligned} \tag{6}$$

Plugging in (6) the inequality on the rate $R(t)$ that grants $N_{tx}(t) \leq N_c(t)$, we obtain the left-hand side of the following (strict) integer inequality

$$\left\lfloor \frac{\tau R(t)}{W(\rho(t))} \right\rfloor - 1 \leq N_{UE}(t) \leq \left\lfloor \frac{\tau R(t)}{W(\rho(t))} \right\rfloor, \tag{7}$$

that will be exploited later on to solve the optimization problems.

We can now write the dynamic evolution of the queue at the UE, which is fed by the arrival/acquisition of new data units (images) and is drained by the transmission of data units to the ES, thus reading as:

$$Q_{UE}(t+1) = \max(0, Q_{UE}(t) - N_{UE}(t)) + A(t), \tag{8}$$

where $A(t)$ is a data arrival process, whose statistical properties are generally unknown. Once the data units arrive at the ES, they are put into a computational queue $Q_{ES}(t)$. To make explicit the dynamic evolution of $Q_{ES}(t)$, we need to quantify the number of data units that can be processed by the ES at time slot t. To this aim, let $\frac{1}{J_s(\rho)}$ denote the number of clock cycles that are necessary to process (classify) a data unit encoded with a compression factor ρ. Then, the maximum number $N_{ES}(t)$ of data units that can be processed at time slot t by the ES is given by

$$\begin{aligned} N_{ES}(t) = & \underset{k}{\text{argmax}} \sum_{i=1}^{min(k, Q_{ES}(t))} \frac{1}{J_s(\rho_{t,i})} \\ & \text{s.t.} \sum_{i=1}^{min(k, Q_{ES}(t))} \frac{1}{J_s(\rho_{t,i})} \leq \tau f_c(t), \end{aligned} \tag{9}$$

where $\mathcal{P}_t = \{\rho_{t,i}\}_{i=1,\ldots,Q_{ser}(t)}$ is the set containing the compression factors associated with each data unit in the ES queue, during the t-th time slot and indexed from the oldest to the newest. Indeed, problem (9) maximizes the number of processed data units in the queue, which clearly must be less than or equal to the ES computational capability, i.e., $\tau f_c(t)$. Finally, the ES computation queue evolves as

$$\begin{aligned} Q_{ES}(t+1) = & \max(0, Q_{ES}(t) - N_{ES}(t)) \\ & + \min(Q_{UE}(t), N_{UE}(t)). \end{aligned} \tag{10}$$

In such a queued dynamic system, the overall latency experienced by a data unit before processing depends on the sum of the two queues in (8) and (10), i.e.,

$$Q_{tot}(t) = Q_{UE}(t) + Q_{ES}(t). \tag{11}$$

In fact, assuming an average data arrival rate $\overline{A} = \mathbb{E}\left\{\frac{A(t)}{\tau}\right\}$, the average long-term delay is defined by the Little's law as [53]:

$$D_{avg} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left\{\frac{Q_{tot}(t)}{\overline{A}}\right\}. \tag{12}$$

Thus, we can attain an average delay D_{avg} constraining the average queue length in (11) as:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{Q_{tot}(t)\} \leq Q_{avg}, \tag{13}$$

with $Q_{avg} = D_{avg}\overline{A}$. In the sequel, we introduce the model for the system energy consumption.

3.2 Energy model

The system energy consumption involves three parts:

- *Transmission energy at the UE*, needed to transmit the data units to the ES.
- *Computation energy at the UE*, needed to compress/encode the data units.
- *Computation energy at the ES*, needed to classify the data units transmitted by the UE.

Assuming a capacity-achieving transmission system in a flat-fading channel, the transmission power $p_{tx}(t)$ can be inferred by the Shannon capacity [54]:

$$R(t) = B \log_2 \left(1 + \frac{p_{tx}(t)h^2(t)}{N_0 B}\right), \tag{14}$$

where $h(t)$ is the channel gain, N_0 denotes the power spectral density at the (ES) receiver side, while B is the bandwidth allocated to the UE. The flat-fading channel assumption simplifies the analysis and the optimal resource management, which already contains several optimization variables. Conceptually, the proposed framework can be extended also to frequency-selective channels, by employing OFDM, which converts it in a set of parallel flat-fading channels. This would request to add to the optimization problems described in the following an extra vector of optimization variables to dynamically split the available transmission power among all the parallel channels, to maximize the overall system transmission rate. This solution would lead to a water-filling-like problem, which is a well-studied topic in the literature. This possible extension is, however, left for future work, which could possibly build upon the results of this manuscript.

Thus, inverting (14), the energy required for transmission during a time slot of duration τ is given by:

$$E_{tx}(t) = \tau \frac{BN_0}{h^2(t)} \left(e^{\frac{R(t)ln(2)}{B}} - 1\right). \tag{15}$$

From the computation perspective, we exploit the model in [55], which assumes a cubic dependence of the computing power with respect to the clock frequency. Thus, letting $f_d(t)$ and $f_s(t)$ be the CPU clock frequencies of the UE and ES, respectively, the corresponding energies needed for computation read as:

$$E_d(t) = \tau\kappa_d f_d^3(t), E_s(t) = \tau\kappa_s f_s^3(t) \tag{16}$$

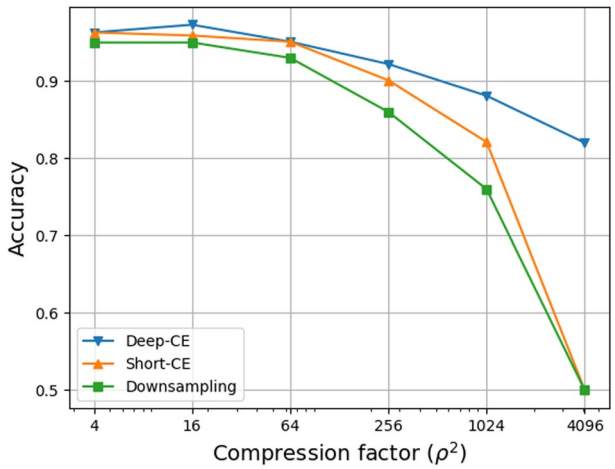

Fig. 2 Accuracy on the test set with deep/short-CE and down-sampling with anti-aliasing filter. This figure makes a comparison on the accuracy degree obtained in the test set of the GTSRB dataset considering images compressed through deep-CE, short-CE and a classical down-sampling with anti-aliasing filter

Table 2 LUT parameters for the deep-CE

ρ	$G(\rho)$ [%]	$\frac{1}{J_c}[\frac{C}{DU}]$	$\frac{1}{J_{zip}}[\frac{C}{DU}]$	$J_d[\frac{DU}{C}]$
2	96.3	4.454×10^6	1.05×10^7	6.66×10^{-8}
4	97.3	6.46×10^6	5.17×10^6	8.59×10^{-8}
8	95.1	7.752×10^6	3.38×10^6	8.98×10^{-8}
16	92.2	8.568×10^6	1.90×10^6	9.47×10^{-8}
32	88.1	9.486×10^6	N/A	1.05×10^{-7}
64	82.0	1.02×10^7	N/A	9.8×10^{-8}

Table 3 LUT parameters for the short-CE

ρ	$G(\rho)$ [%]	$\frac{1}{J_c}[\frac{C}{DU}]$	$\frac{1}{J_{zip}}[\frac{C}{DU}]$	$J_d[\frac{DU}{C}]$
2	96.3	4.454×10^6	1.05×10^7	6.66×10^{-8}
4	95.9	4.454×10^6	5.17×10^6	1.04×10^{-7}
8	95.1	4.454×10^6	3.38×10^6	1.27×10^{-7}
16	90.1	4.454×10^6	1.90×10^6	1.57×10^{-7}
32	82.1	4.454×10^6	N/A	2.24×10^{-7}
64	55.0	4.454×10^6	N/A	2.24×10^{-7}

where the constants κ_d and κ_s represent the effective switched capacitance of the UE and ES processing units, respectively. Finally, we introduce a weighted energy function $E_\alpha(t)$, which quantifies the energy consumption of the overall system during the t-th time slot:

$$E_\alpha(t) = \alpha\Big(E_d(t) + E_{tx}(t)\Big) + (1-\alpha)E_s(t) \tag{17}$$

where $\alpha \in [0,1]$ is a weighting parameter to be chosen. For instance, choosing $\alpha = 1$ leads to a pure user-centric strategy; whereas, $\alpha = 0$ determines a pure network-centric strategy. An intermediate strategy, which we term as holistic, can be obtained with

$\alpha = 0.5$. The use of this weighting parameter helps introduce more degrees of freedom and flexibility in the resource optimization, depending on the needs of the operators, users, and service providers.

3.3 Accuracy model

It is generally difficult to establish an analytic expression that relates the accuracy of the classification task over an available test set and the compression factor adopted by our goal-oriented communication scheme. Thus, in this paper we use a more practical approach, where the accuracy function $G(\rho(t))$ for the ES-based learning/classification task can be cast in the optimization problem by using a look-up table (LUT) indexed by the compression factor $\rho(t)$, whose entries have been obtained by offline testing each CE/CC associated with a specific compression factor. Examples of LUTs for the considered classification tasks will be provided in the sequel in Tables 2 and 3 and in Fig. 2. The LUT is instrumental to define constraints on the average accuracy we want to guarantee for the image classification task, as detailed in the two resource management policies described in the sequel. Note that, by the rate in (14), we are ideally assuming a capacity-achieving communication system, which also simplifies the analysis and mathematical tractability of the problem. Such a Shannon rate can be practically granted by long channel codes, which also grant (almost) zero (coded) bit error rate (BER). Thus, coherently with (14), we train the CE-CCs without taking into account possible accuracy degradation induced by a finite BER, and also the LUTs are obtained by testing the CE-CCs neural networks, with zero BER in the communication link. Although certainly interesting, the design of CE-CCs networks that are capable to handle, and possibly mitigate, communication systems with non-negligible BER is out of the scope of this manuscript and could be the subject of further studies. Anyway, the results we will obtain for the energy, accuracy, and delay trade-offs, can still be considered bounds on those obtainable for finite (coded) BER scenarios, which will be tight and achievable up to a maximum BER (that depends on the specific task).

4 Problem formulation and methodology

The latency, energy, and accuracy models defined in the previous section can be exploited in the formal definition of two dynamic resource optimization strategies, which are described in the sequel.

4.1 MEDA: minimum energy under average service delay and accuracy constraints strategy

In the first resource allocation strategy, we formulate a long-term optimization problem that aims at minimizing the average energy consumption of the system, subject to average delay and accuracy constraints. The problem can be mathematically cast as:

$$\begin{aligned}
\min_{R(t),\Phi(t)} \quad & \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\{E_\alpha(t)\} \\
\text{s.t.} \quad & (a)\ \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\{Q_{tot}(t)\} \leq Q_{avg} \\
& (b)\ \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\{G(\rho(t))\} \geq G_{avg} \\
& (c)\ 0 \leq R(t) \leq R_{max} \\
& (d)\ \rho(t) \in \mathcal{S}, f_s(t) \in \mathcal{F}_s, f_d(t) \in \mathcal{F}_d,
\end{aligned} \tag{18}$$

where $\Phi(t) = [f_s(t), f_d(t), \rho(t)]$ collects the discrete optimization variables, and $\Phi(t) \in \mathcal{J}_\Phi = \mathcal{S} \times \mathcal{F}_s \times \mathcal{F}_d$. In (18) we impose two *long-term* constraints: (a) the average queue length must be lower than Q_{avg}, i.e., we are imposing a maximum average service delay equal to $D_{avg} = Q_{avg}/\overline{A}$ (cf. 13); (b) the average classification accuracy must be greater than G_{avg}. The others are *feasibility* constraints: (c) imposes an instantaneous constraint on the transmission rate, which must be greater than zero and smaller than a maximum value R_{max}, obtained as in (14) using the maximum transmission power, say P_{max}, available at the UE; finally, (d) specifies the discrete feasible sets $\mathcal{S}$, $\mathcal{F}_s$, $\mathcal{F}_d$, for the goal-oriented compression factor and for the ES and UE computational clock frequencies, respectively. Since we do not assume any knowledge of the statistics of quantities involved in the system (e.g., data arrivals, radio channels, etc.) solving (18) is very challenging. However, resorting to stochastic Lyapunov optimization [45], we derive low-complexity dynamic solutions for the original optimization problem, as detailed in the following.

According to [45], we associate each long-term constraint, (a) and (b) in problem (18), to a specific *virtual queue*

$$\begin{aligned}
S(t+1) &= \max(0, S(t) + \lambda(E_{UE}(t) - E_{UE,avg})) \\
O(t+1) &= \max(0, O(t) + \eta(E_s(t) - E_{s,avg}))
\end{aligned} \tag{19}$$

The parameters ν and μ are step sizes, used to adjust the convergence speed of the algorithm. As detailed in [45], guaranteeing the mean-rate stability of the queues in (19) is equivalent to satisfy the constraints (a) and (b) in (18). In the sequel, we collect the virtual queues employed in the system in a vector $\Theta(t) = [Z(t), Y(t)]$. Then, to stabilize all the queues, we introduce the *Lyapunov Function* $L(t) = \frac{1}{2}[Y(t)^2 + Z(t)^2]$, and the associated *Lyapunov Drift*

$$\Delta(t) = \mathbb{E}\{L(t+1) - L(t) | \Theta(t)\}.$$

Minimizing the Lyapunov Drift $\Delta(t)$ leads to the stabilization of the virtual queues, but possibly with an unjustified and uncontrolled energy consumption. Thus, to trade-off system stability with energy consumption, the Lyapunov Drift is augmented with a term

dependent on the objective function of (18), thus obtaining the following *Lyapunov Drift plus Penalty* function [45]

$$\Delta_p(t) = \Delta(t) + V\mathbb{E}\{E_\alpha(t)|\Theta(t)\}. \tag{20}$$

In particular, the drift-plus-penalty function is the conditional expected change of $L(t)$ over successive slots, with a penalty factor that weights the objective function of (18), with a weighting parameter V. Now, if $\Delta_p(t)$ is lower than a finite constant for all t, the virtual queues are stable and the optimal solution of (18) is asymptotically reached as V increases [45, 39, Th. 4.8]. In practical scenarios with finite V values, the higher is V, the more importance is given to the energy consumption, rather than to the virtual queue backlogs, thus pushing the solution toward optimality, while still guaranteeing the stability of the system.

Following similar arguments as in [45], we proceed by minimizing an upper-bound of the drift-plus penalty function in (20) in a stochastic fashion. After some simple algebra (similar as in [6] and omitted here due to space limitations), we obtain the following per-slot problem at each time t:

$$\begin{aligned} \min_{R(t),\Phi(t)} \quad & -N_{UE}(t)Q_{tx}(t) - \nu Y(t)G(\rho(t)) \\ & -N_{ES}(t)Q_{comp}(t) + VE_\alpha(t) \\ \text{s.t.} \quad & (a) \quad 0 \le R \le R_{max} \\ & (b) \quad \Phi(t) \in \mathcal{J}_\phi \end{aligned} \tag{21}$$

where $Q_{tx}(t) = 2\mu^2(Q_{UE}(t) - Q_{ES}(t)) + \mu Z(t)$, and $Q_{comp}(t) = 2\mu^2 Q_{ES}(t) + \mu Z(t)$. In the sequel, we will show how (21) can be split into subproblems that admit low-complexity solution procedures for the optimal UE resources (i.e., rate, compression factor, local CPU clock frequency), and the computation resources at the ES (i.e., remote CPU clock frequency).

Algorithm 1 UE's Resource Allocation

let $CM \in \mathbb{R}^{(|\mathcal{S}|\times|\mathcal{F}_d|)}$ be the costs matrix
let $RM \in \mathbb{R}^{(|\mathcal{S}|\times|\mathcal{F}_d|)}$ be the rates matrix
Observe Q_{UE}, Q_{ES}, Y, Z, h
$Q_{tx} \leftarrow 2\mu^2(Q_{UE} - Q_{ES}) + \mu Z$
if $Q_{tx} > 0$ **then**
  for $i = 1$ to $|\mathcal{S}|$ **do**
    for $j = 1$ to $|\mathcal{F}_d|$ **do**
      Compute R^* by (24), with f_{dj}, ρ_i and save it in $RM(i,j)$
      Compute the cost using (31) and save it in $CM(i,j)$
    end for
  end for
  $(i^*, j^*) \leftarrow \arg\min_{i,j}(CM(i,j))$
  $R_{opt} \leftarrow RM(i^*, j^*)$
  $f_d^* \leftarrow f_{dj^*}$, $\rho^* \leftarrow \rho_{i^*}$, $R^* \leftarrow R_{opt}$
else
  $f_d^* \leftarrow 0$, $\rho^* \leftarrow NULL$, $R^* \leftarrow 0$,
end if

4.1.1 UE's resource optimization for MEDA

The resource allocation problem at the UE aims at optimizing the transmission rate $R(t)$, the compression factor $\rho(t)$, and the UE CPU frequency cycles $f_d(t)$ in (21). In the sequel, to ease the notation, the dependence from the time index t is omitted. It is clear from (21) that the UE allocation problem can be split by the optimization of the ES computation resources, thus obtaining the following subproblem at the UE:

$$\begin{aligned} \min_{R,\rho,f_d} \quad & -\left(\frac{\tau R}{W(\rho)} - 2\right) Q_{tx} + \frac{\tau V N_0 B}{h^2}\, e^{R\ln(2)/B} \\ & + \tau V \kappa_d f_d^3 - \nu Y G(\rho) \\ \text{s.t.} \quad (a) \quad & 0 \le R \le R^+_{max} \\ (b) \quad & \rho \in \mathcal{S}, f_d \in \mathcal{F}_d \end{aligned} \tag{22}$$

where

$$R^+_{max} = \min\left\{R_{max}, \frac{Q_{UE} W(\rho)}{\tau}, J_d(\rho) f_d W(\rho)\right\}. \tag{23}$$

where, exploiting (7) and $\lfloor x \rfloor \ge x - 1$, the cost function is a (tight) upper-bound of the original one, with the same optimal solution because Q_{tx} does not depend on the optimization variables. Assumptions 1 and 2, means that in practice it does not make any sense that the transmission rate could exceed the value R^+_{max} in (23), which is the minimum between three terms: (i) R_{max}, i.e., the maximum rate obtainable by the radio interface; (ii) the rate necessary to empty the UE local queue $Q_{UE}(t)$, by compressing all the data units with a specific compression factor ρ; (iii) the maximum rate that is necessary to grant transmission of all the data units that is possible to compress during the t-th time slot using a compression factor ρ and a CPU frequency f_d.

The problem in (22) is a mixed integer optimization problem since both the compression factor $\rho \in \mathcal{S}$ and the device frequency $f_d \in \mathcal{F}_d$ take values on a discrete set. However, in our case $\mathcal{S}$ and $\mathcal{F}_d$ have a limited cardinality, allowing for an exhaustive search of the optimal values in a short time. Furthermore, since the objective function in (22) is (strictly) convex with respect to R, for any fixed frequency f_d and compression factor ρ, by Lagrange theory and KKT conditions, we obtain a unique solution for the transmission rate that reads as:

$$R^*(\rho, f_d) = \left[\frac{B}{\ln(2)} \ln\left(\frac{Q_{tx} h^2}{W(\rho) V \ln(2) N_0}\right)\right]_0^{R^+_{max}} \tag{24}$$

if $Q_{tx} > 0$, and $R^* = 0$ otherwise. The overall procedure for UE resource allocation is summarized in Algorithm 1.

Algorithm 2 ES' Resource Allocation

let $CA \in \mathbb{R}^{|\mathcal{F}_s|}$ be the server costs array
let $P \in \mathbb{N}^{|\mathcal{P}|}$ be the array that stores the compression factors of the queued data units at the ES
$\forall$ time-slot observe Q_{ser}, Z, P
for $i = 1$ to $|\mathcal{F}_s|$ **do**
  $M_{cyclces} \leftarrow \tau \times f_{si}$
  $T_{cycles} \leftarrow 0$
  $N_s \leftarrow 0$
  for $j = 1$ to $|\mathcal{P}|$ **do**
    $T_{cycles} \leftarrow T_{cycles} + \frac{1}{J_s(P(j))}$
    if $T_{cycles} > M_{cycles}$ **then**
      break
    else
      $N_s \leftarrow N_s + 1$
    end if
  end for
  compute $cost$ by the objective function in (25) and save it in $CA(i)$
end for
$f_{s*} \leftarrow f_{si^*}$

4.1.2 ES' resource optimization for MEDA

The resource allocation problem at the ES aims at optimizing the CPU frequency cycles $f_c(t)$ in (21), thus leading to the following optimization:

$$\begin{aligned} \min_{f_c} \quad & -Q_{comp}N_{ES} + \tau V \kappa f_s^3 \\ \text{s.t.} \quad & f_s \in \mathcal{F}_s. \end{aligned} \tag{25}$$

Note that (25) is an integer optimization problem, where also the number $N_{ES}(t)$ of processable data units depends on $f_s(t)$ by (9). Since the number of possible CPU frequencies in $\mathcal{F}_s$ is small, we proceed using an exhaustive search procedure, which can be summarized in the following steps:

1 For each possible clock frequency $f_s(t) \in \mathcal{F}_s$, observe $Q_{comp}(t)$, evaluate $N_{ES}(t)$ by (9), and compute the value of the objective function in (25).
2 Select the frequency $f_s^*(t)$ that leads to the lowest objective value.

The main steps of the procedure are summarized in Algorithm 2.

4.1.3 Overall edge learning procedure

The two resource optimizations procedures at the UE and ES jointly contribute to the overall dynamic resource allocation procedure for edge learning, which is summarized in Algorithm 3. Lyapunov optimization theory guarantees that, as V increases, Algorithm 3 minimizes the average energy consumption, while respecting average latency and accuracy constraints.

Algorithm 3 MEDA for Goal-oriented Edge Learning

initialize $V, Z(0)$
initialize $Y(0)$, ν, μ
for all t **do**
  Observe $A(t)$ and $h(t)$
  Optimize $R(t), \rho(t), f_d(t)$ with Algorithm 1
  Optimize $f_s(t)$ with Algorithm 2
  Execute transmission and computation tasks
  Update $Q_{UE}(t), Q_{ES}(t), Z(t), Y(t)$ in (8), (9), and (19).
end for

4.2 MADE: maximum accuracy under average service delay and energy constraints strategy

In this section, we introduce an alternative strategy for optimizing edge learning with goal-oriented communications. In particular, the aim of this strategy is to maximize the average long-term accuracy, under long-term latency and energy constraints. Let $E_{UE}(t) = E_d(t) + E_{tx}(t)$ be the overall energy spent by the UE at time slot t. Then, the long-term optimization problem can be cast as:

$$\begin{aligned}
\max_{R(t),\Phi(t)} \quad & \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\{G(t)\} \\
\text{s.t.} \quad & (a)\ \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\{Q_{tot}(t)\} \le Q_{avg} \\
& (b)\ \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\{E_{UE}(t)\} \le E_{UE,avg} \\
& (c)\ \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\{E_s(t)\} \le E_{s,avg} \\
& (d)\ 0 \le R(t) \le R_{max} \\
& (e)\ \Phi(t) \in \mathcal{J}_\Phi
\end{aligned} \tag{26}$$

where $\Phi(t) = [f_s(t), f_d(t), \rho(t)]$ collects the discrete optimization variables. In this case, we have an exchange of the constraints and the objective function with respect to (18). Indeed, in (26) we have the following *long-term* constraints: (*a*) the average queue length must be lower that Q_{avg}, (as in (18)); (*b*) The average energy spent at the UE must be lower than $E_{UE,avg}$; (*c*) the average energy spent at the ES must be lower than $E_{s,avg}$; (*d*) and (*e*) impose instantaneous constraints on the optimization variables, similarly to (*d*) in (18).

To handle the long-term latency constraint (a), we use the same virtual queue $Z(t)$ we already introduced in the previous problem, and that evolves according to (19). Furthermore, we introduce the virtual queues $S(t)$ and $O(t)$, associated with the two energy constraints (*b*) and (*c*), which evolve as:

$$\begin{aligned}
S(t+1) &= \max(0, S(t) + \lambda(E_{UE}(t) - E_{UE,avg})) \\
O(t+1) &= \max(0, O(t) + \eta(E_s(t) - E_{s,avg}))
\end{aligned} \tag{27}$$

where λ and η are step sizes that control the convergence speed of the algorithm. Then, proceeding as in the previous case, we write the Lyapunov function

$$L(t) = \frac{1}{2}[Z(t)^2 + S(t)^2 + O(t)^2], \tag{28}$$

and the Lyapunov drift-plus-penalty function given by

$$\Delta_p(t) = \Delta(t) - V\mathbb{E}\{G(t)|S(t), Z(t), O(t)\}. \tag{29}$$

Exploiting the same Lyapunov framework [45], we proceed by minimizing an upper-bound of the drift-plus-penalty function in (29) in a stochastic fashion. After some simple derivations, we obtain the following per-slot problem at each time t:

$$\begin{aligned} \min_{\Phi(t)} \quad & -N_{UE}(t)Q_{tx}(t) + \lambda S(t)E_{UE}(t) \\ & + \eta O(t)E_s(t) - N_{ES}(t)Q_{comp}(t) - VG(\rho(t)) \\ \text{s.t.} \quad & (a) \quad 0 \le R \le R^+_{max} \\ & (b) \quad \Phi(t) \in \mathcal{J}_\Phi \end{aligned} \tag{30}$$

As for the MEDA strategy, it is easy to see that (30) decouples in the two separate optimization problems, as detailed in the two following subsections.

4.2.1 UE's resource optimization for MADE

The resource allocation problem at the UE aims at optimizing the transmission rate $R(t)$, the compression factor $\rho(t)$, and the UE CPU frequency cycles $f_d(t)$ in (30) at every time t. Omitting the time index t, the subproblem at the UE can be cast as:

$$\begin{aligned} \min_{R,\rho,f_d} \quad & -\left(\frac{\tau R}{W} - 2\right)Q_{tx} + \frac{\tau\lambda S N_0 B}{h^2(t)} e^{\frac{R\ln(2)}{B}} \\ & + \tau\lambda S\kappa f_d^3 - VG(\rho) \\ \text{s.t.} \quad & (a) \quad 0 \le R \le R^+_{max} \\ & (b) \quad \rho \in \mathcal{S}, \quad f_d \in \mathcal{F}_d. \end{aligned} \tag{31}$$

Problem (31) is a mixed-integer optimization program that, by the same arguments and bounds used for the MEDA problem, can be proved to be strictly convex with respect to the transmission rate R, for any fixed compression factor ρ and computational clock frequency f_d, with optimal closed form solution

$$R^*(\rho, f_d) = \left[\frac{B}{\ln(2)} \ln\left(\frac{Q_{tx}h^2}{W(\rho)\lambda S \ln(2) N_0}\right)\right]_0^{R^+_{max}}, \tag{32}$$

for $Q_{tx}(t) > 0$, and $R^* = 0$ otherwise. Thus, the overall optimal solution $R^*(t)$ can be found by an exhaustive search in the product space $\mathcal{F}_d \times \mathcal{S}$ of the UE clock frequencies and compression factors, by comparing the obtained objective values in (31) for the $|\mathcal{F}_d||\mathcal{S}|$ potential solutions $R^*(\rho, f_d)$. The procedure follows the same steps already described in Algorithm 1.

Table 4 Common parameters

ρ	$M(\rho)$ [px]	$N(\rho)[\frac{bits}{px}]$	$J_s[\frac{DU}{C}]$
2	128x128x3	1.08	9.8×10^{-9}
4	64x64x3	2.27	2.94×10^{-8}
8	32x32x3	4.72	4.9×10^{-8}
16	16x16x3	9.06	7.35×10^{-8}
32	8x8x3	8	8.12×10^{-8}
64	4x4x3	8	8.40×10^{-8}

Table 5 Channel parameters

Ch. Type	D [m]	B [kHz]	f_0 [GHz]	σ_0^2
A	50	2500	6	1.06×10^{-10}
B	500	2500	30	2.44×10^{-15}

Table 6 Maximum number for N_{UE} in channel scenario A

ρ	N_{UE}^{max} (deep)	N_{UE}^{max} (short)	N_{UE}^{max} (DS)	N_s^{max}
2	3	3	3	2
4	5	6	7	6
8	5	7	10	11
16	5	9	15	16
32	6	14	28	18
64	5	14	29	18

Table 7 Maximum number for N_{UE} in channel scenario B

ρ	N_{UE}^{max} (deep)	N_{UE}^{max} (short)	N_{UE}^{max} (DS)	N_s^{max}
2	0	0	0	2
4	1	1	1	6
8	2	2	2	11
16	4	4	5	16
32	6	14	26	18
64	5	14	29	18

4.2.2 ES' resource allocation for MADE

The resource allocation problem at the ES aims at optimizing the CPU frequency cycles $f_c(t)$ in (21), thus leading to the following optimization:

$$\begin{aligned} \min_{f_s} \quad & -Q_{comp}N_s + \tau O\eta\kappa f_s^3 \\ \text{s.t.} \quad & f_s \in \mathcal{F}_s \end{aligned} \tag{33}$$

Similarly to (25), the ES frequency $f_s(t)$ takes values in the discrete frequency set $\mathcal{F}_s$ and, consequently, the problem can be solved only by an exhaustive search, which is similar to that one proposed in subsection . The only two differences are: (i) the cost function, and (ii) the presence of the queue $O(t)$, which is used to control the energy constraint at the ES. Thus, the main steps are the same already listed in Algorithm 2. Finally, the overall resource allocation procedure following the MADE design can be described by Algorithm 3, with the aforementioned modifications for the UE's and ES's resource allocations.

5 Numerical results and discussion

In this section, we assess the performance of the proposed strategies for edge learning with goal-oriented communications. As previously mentioned in Sect. 3.3, we need to build a LUT that quantifies the behavior of the accuracy of the proposed goal-oriented learning scheme with respect the adopted compression factor ρ. To this aim, Tables 2 and 3 report the values of the accuracy $G(\rho)$, the data units $J_d(\rho)$ that the UE can at most compress (and zip by JPEG2000) in a clock cycle, the data units $J_{zip}(\rho)$ it can zip by JPEG2000 in a clock cycle, and the data units $J_c(\rho)$ that it can compress in a clock cycle, by the deep-CE and short-CE models, respectively. Also, Table 4 reports the data units $J_s(\rho)$ that the ES can at most classify in a clock cycle, as well as the image size $M(\rho)$ and the average number of bits/pixel $N(\rho)$ that are shared by both the short-CE and the deep-CE, when using JPEG-2000.

As far as the wireless channel model is concerned, we modeled the local scattering according to a Rayleigh flat-fading channel, whose statistical evolution in time obeys a *Clarke's autocorrelation function* [56], which has been used to set the time slot duration. We considered two operating scenarios, as summarized in Table 5, where σ_0^2 represents the average power path loss, which has been computed according to the *Alpha-Beta-Gamma* model [57]. Finally, the UE's and ES's CPU clock frequency sets are selected as $\mathcal{F}_d = \{0.1, 0.2, \ldots, 0.9, 1\} \times 1.4GHz$, and $\mathcal{F}_s = \{0.1, 0.2, \ldots, 0.9, 1\} \times 4.5GHz$, respectively, assuming a switched capacitance $\kappa = 1.097 \times 10^{-27} [\frac{s}{cycles}]^3$ (equal for both UE and ES). To give further insight, we report in Tables 6 and 7 the maximum number of data units the UE and ES can process with specific computation capabilities, in Channels A and B, respectively, when dealing with images from the dataset we describe in the following.

5.1 Compression–accuracy trade-off

In the experimental setup, we used the German Traffic Sign Recognition Benchmarks (GTSRB) [58] dataset, which includes 1213 pictures of German road signals divided into 43 different classes, thus representing a quite challenging classification task. The dataset has been split in an 80% training set, composed of 970 images, and 20% test set, composed of 243 images. During the data loading phase, all the images have been normalized to a size of 256×256 and then converted to a three-channel image (one channel for each RGB color), such that the initial size of each data unit, is $256 \times 256 \times 3$. We considered compression factors $\rho \in \{2, 4, 8, 16, 32, 64\}$.

In Fig. 2, we illustrate the behavior of the accuracy of the proposed scheme, versus the compression factor, for different architectures: i) deep-CE; ii) short-CE; and a

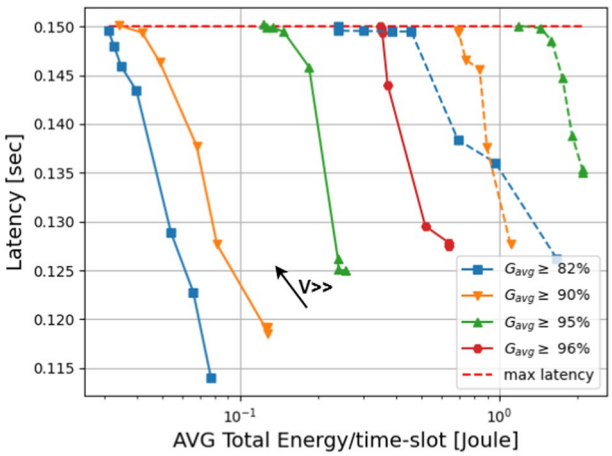

Fig. 3 Average energy versus average latency, for deep-CE (solid) and down-sampling (dashed). This figure makes a comparison of the trade-offs between the energy consumption of the overall system (UE + ES) obtained with the deep-CE and the down-sampling compression

simple image down-sampling procedure with anti-aliasing filter. As expected, and shown in Fig. 2, the accuracy $G(\rho)$ has a monotone decreasing behavior with respect to the compression factor. The deep-CE has always the best performance even if, for lower compression factors (up to 8), the difference between the three architectures is almost negligible. In contrast, at large compression factors (i.e., 16, 32, 64), there is a clear advantage in using the deep-CE architecture. For compression factor $\rho = 64$, we get output tensors with a size of $4 \times 4 \times 3 = 48$ pixels. Interestingly, although images of this size have clearly undergone a heavy transformation, the deep-CE still allows the ES CC to classify them with an 82% accuracy. For this compression factor, both image down-sampling and short-CE do not allow a meaningful classification. In the next sections, we extensively assess the trade-off between energy, latency, and performance of the proposed edge learning strategies with goal-oriented communications.

5.2 MEDA with deep-CE

In this section, we illustrate the performance of the proposed goal-oriented scheme with the MEDA strategy. We considered the wireless channel scenario A in Table 5 and the *holistic* paradigm that minimizes the energy consumption of the whole system, which corresponds to set $\alpha = 1/2$ in (17). The time slot duration τ has been set to 50 *ms*, which fits within the coherence time where the channel can be considered constant. The image arrival process, whose statistical knowledge is not exploited, has been modeled as a Poisson process with an average rate $A_{avg} = 2$ *arrival*/*slot*. This situation, for instance, is compatible with a web cam that transmits images with a rate of 40 *frame*/*sec*. The average latency constraint has been set to 150 *ms*. In the sequel, we consider only the deep-CE learning architecture, which is the one having the best performance as illustrated in Fig. 2.

In Fig. 3, we show the average system energy versus the average latency (i.e., the energy–latency trade-off), for different accuracy constraints and learning architectures. Specifically, from (20), (22), (25), the parameter V is used to explore the trade-off between energy, latency and accuracy. As the parameter V increases, we move on

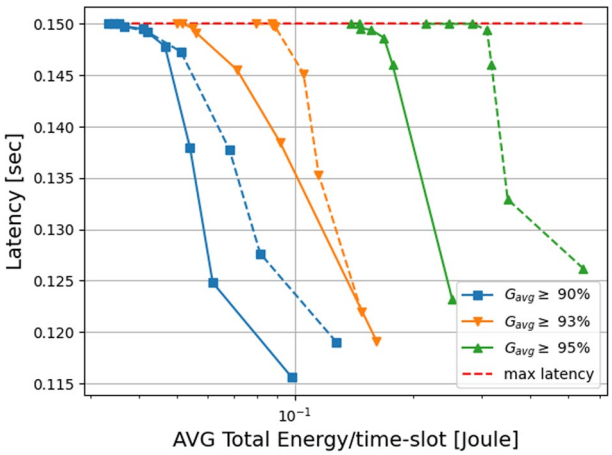

Fig. 4 Total energy/latency trade-off for ensemble learning (solid) and deep-CE only (dashed). This figure makes a comparison of the trade-offs between the energy consumption of the overall system (UE + ES) obtained with a compression system which encompasses only the deep-CE and a compression system composed of all the considered compression strategies (ensemble)

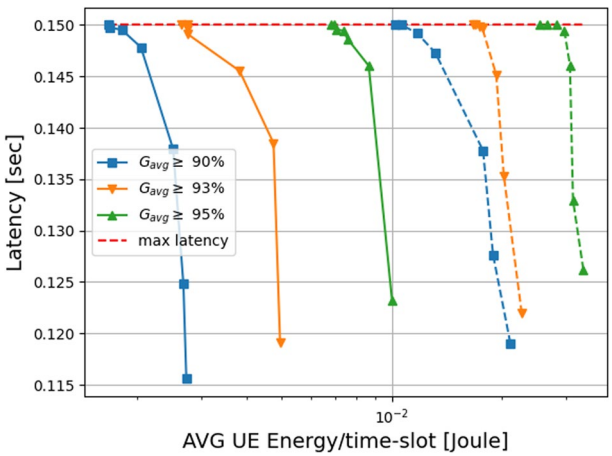

Fig. 5 UE energy/latency trade-off for ensemble learning (solid) and deep-CE only (dashed). This figure makes a comparison of the trade-offs between the energy consumption of the UE obtained with a compression system which encompasses only the deep-CE and a compression system composed of all the considered compression strategies (ensemble)

the curves in Fig. 3 from the right to the left, reducing the energy at the expense of a higher latency, up to the maximum latency constraint, which corresponds to the optimal solution of the problem. As expected, the trade-off curves reported in Fig. 3 show that a stricter accuracy constraint implies also a higher system energy consumption and latency, according to the Energy/Accuracy and Latency/Accuracy trade-offs [6]. Then, the proposed deep-CE strategy is compared with the one performing compression with down-sampling, which is depicted using dashed lines in Fig. 3. As we can notice from Fig. 3, the proposed goal-oriented compression strategy enables a considerable saving in term of energy consumption, while satisfying the same accuracy and delay constraints. This gain is obtained thanks to the proposed deep-CE learning scheme, which is capable

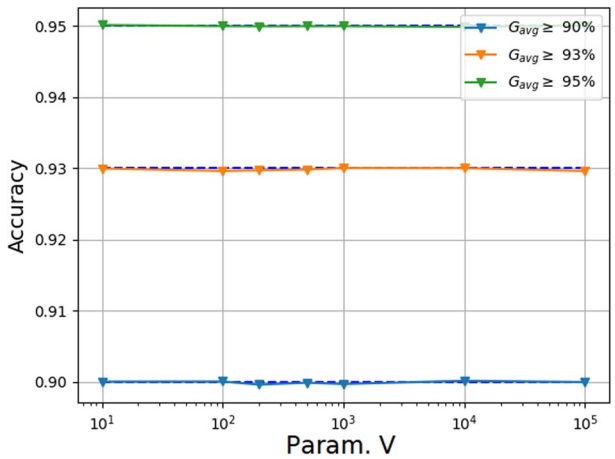

Fig. 6 Accuracy vs parameter V, for ensemble learning. This figure shows the correct classification rate as a function of the trade-off parameter V, employing the ensemble of the compression strategies

to grant quite high accuracy employing smaller data units, thus paying on average a lower energy/delay cost for transmission and classification.

5.3 Ensemble of goal-oriented compression schemes

Looking at Tables 2 and 3, we notice that also the short-CE and the classical down-sampling compression can lead to quite good accuracy results for low compression factors $\rho \in \{2, 4, 8, 16\}$, while requiring a lower computational complexity than deep-CE. Thus, it makes sense to consider an edge-based classification scheme equipped with an ensemble of all the available compression strategies, i.e., deep-CE, short-CE, and down-sampling, which might lead to enhanced performance. To this aim, in Fig. 4, we illustrate again the energy–latency trade-off curve of the system, for different accuracy constraints, comparing the ensemble of goal-oriented compression strategies (solid curves) with deep-CE (dashed curves). As we can notice from Fig. 4, there is a remarkable gain obtained by using the proposed ensemble compression scheme, since the system has more degrees of freedom (in terms of accuracy, complexity, and latency) to adapt to the instantaneous variations of the system parameters, i.e., queues, wireless channels, data arrivals, etc. The gain is even more appreciable if we consider the UE's energy consumption, whose behavior with respect to average latency is shown in Fig. 5, for the same accuracy constraints. This result shows that looking for a flexible, scalable, and finely tunable network for compression and classification is an interesting research direction.

Finally, Fig. 6 reports the actual average accuracy values obtained for the same simulations results shown in Figs. 4, 5 (i.e., for several values of the V parameter), comparing them with the accuracy constraints (dashed lines). From Fig. 6, we can notice how the system strictly respects the (minimum) accuracy prescribed by the constraints, without unnecessarily wasting energy or increasing the delay.

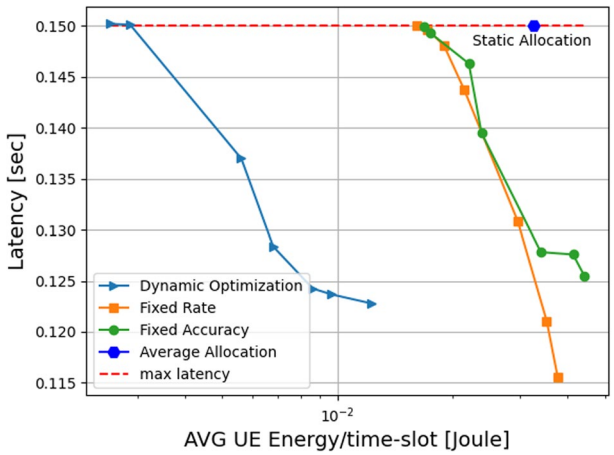

Fig. 7 Latency/energy trade-off for static and dynamic resource allocation strategies. $A_{avg} = 2DU/slot$. This figure compares the trade-off between latency and accuracy considering our fully dynamic optimization strategy, with static resources allocation strategy, where learning, transmission and computation capabilities are kept fixed

5.4 Comparison with static resource allocation

As anticipated in the Introduction, the joint dynamic adaption of the system resources and the learning models (i.e., the adaptivity of the CE-CC network), is one of the main strengths of the proposed framework with respect to most of the literature. Thus, in order to properly highlight the advantages of the framework, we did comparisons with : (i) a dynamic resource allocation strategy with a fixed CE-CC couple, which is capable to respect the average constraint imposed to our approach. This approach is quite similar to [6], where a fixed learning model is considered and the optimization of the transmission resources at the UE-side acts on the quantization bits; (ii) a completely static resource allocation strategy, which not only employs the fixed CE-CC couple, but also fixes the optimal static transmission resources (e.g., rate and power) exploiting the knowledge of the average channel statistics and the average image arrival rate; (iii) A hybrid static/dynamic optimization strategy where the transmission resources (e.g., rate and power) are fixed according to the average channel statistics, while the learning CE-CC architecture and (only) the computational resources are jointly dynamically optimized. In particular, this approach is similar in philosophy to that one in [41], where a single network is considered, whose compression degree is made adaptive by selecting only the most significant features for increasing compression ratios. However, differently from [41], we also consider the computational cost and the task scheduling. Specifically, the static resource allocation fixes the transmission power to the minimum one that guarantees a transmission rate, computed through the capacity formula for flat-fading channels [59], which makes the UE queue stable (e.g., average transmitted images per slot equal to average images arrival per slot).

For the selected learning model, we fixed the UE clock frequency to the minimum one that is capable to respect Assumptions 1 and 2 in Sect. . In this set of simulations we considered the MADE strategy, with channel scenario A in Table 5, an arrival rate $A_{avg} = 2DU/slot$, and an accuracy constraint set to $G_{avg} = 0.95$. Furthermore, we considered a UE-centric paradigm for the energy consumption, which corresponds to set

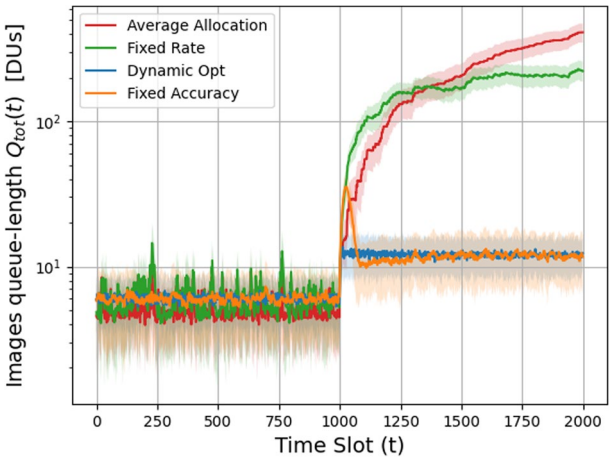

Fig. 8 Image queue length in non-stationary conditions. This figure shows the behavior of the total queue in non-stationary condition with static and dynamic optimization strategies

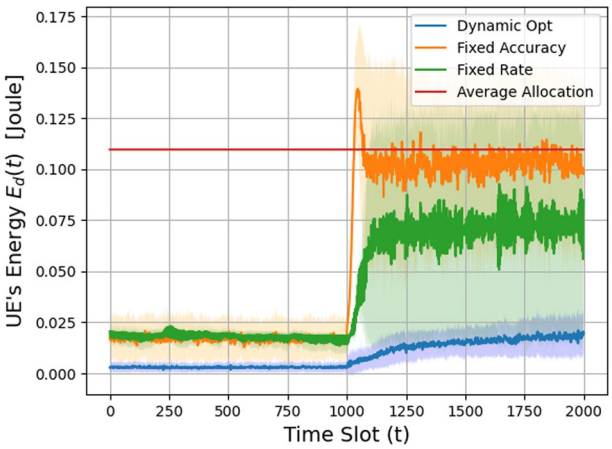

Fig. 9 Instantaneous UE energy consumption in non-stationary conditions. This figure compares the instantaneous energy consumption of the UE in non-stationary conditions with static and dynamic optimization strategies

$\alpha = 1$ in (17). In both the strategies i) and ii) we considered the deep-CE with $\rho = 8$, as the single learning model, which has a fixed accuracy equal to 0.951.

As expected and witnessed by the trade-off curves presented in Fig. 7, any dynamic resource allocation strategy that exploits instantaneous knowledge of the system status outperforms a static allocation based on the knowledge of the average system statistics. Specifically, by letting the system to jointly and adaptively choose the best compression factor (e.g., the best CE-CC network) and the system resources, as envisaged by our framework, we obtain a significantly better energy–latency trade-off, and a much lower (minimum) UE's energy consumption for the optimal solution (i.e., the maximum Vs) of the MEDA strategy.

In a second set of simulations, we tested the capability of the dynamic policies to adapt to changes in the system statistics, such as the images (average) arrival rate. To this end we considered simulation runs with a duration of 2×10^4 time slots, where the

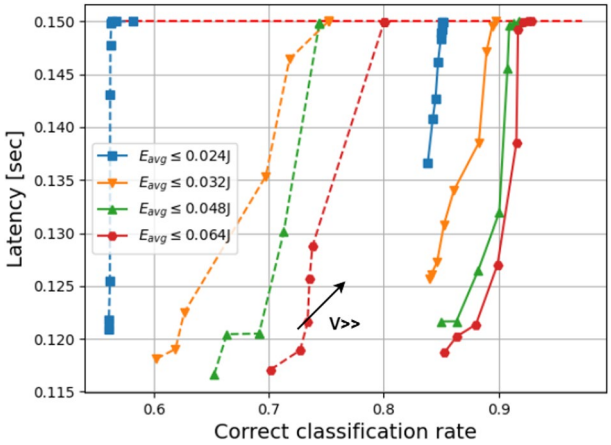

Fig. 10 Accuracy/latency trade-off. Deep-CE (solid) vs down-sampling (dashed). This figure compares the trade-off between latency and accuracy considering the deep-CE and the down-sampling compression strategy

average arrival rate suddenly doubles to $A_{avg} = 4DU/slot$ after 5×10^3 slots. In this case, according to Little's theorem, the same average delay constraint corresponds to a double average length of the images queue $Q_{tot}(t)$. Reminding that the proposed problems were targeting average performance and constraints, we performed 1.00×10^2 simulation runs. Figure 8 shows the sample mean of the UE's queue lengths $Q_{tot}(t)$ for each competitive strategy, while the shaded areas identify the associated standard deviations, computed over the 1.00×10^2 runs. From Fig. 8, it is possible to appreciate that, while our approach is capable to maintain the system stable also in a non-stationary environment by rapidly doubling the images queue length, the policies with a static allocation of the system resources experience an explosion of the latency queue Q_{tot}, as a consequence of the mismatched A_{avg} used to allocate transmission rate and power. Conversely, the mixed policy that uses a fixed CE-CCs network, even if it pays a price in energy consumption as shown in Fig. 7, is capable to adapt the queue length to the correct value, although with a longer transient and higher standard deviation with respect to our policy. Figure 9 shows the associate energy consumption for the same 1.00×10^2 simulation runs and confirms that the optimization policies that allow to adapt online the learning strategy grant the minimum UE energy consumption.

5.5 Performance of MADE

In this section, we assess the performance of the MADE goal-oriented strategy. In the sequel, we consider the channel scenario B of Table 5, while the other parameters are the same we considered for the first set of simulations in the previous subsection. Channel B is characterized by a huge attenuation, making the UE's transmission energy comparable with its computation energy. Also in this case, we start comparing the deep-CE compression with down-sampling, which are, respectively, the best and the worst strategies from the accuracy perspective (cf. Fig. 2).

Figure 10 shows the behavior of the average latency versus the accuracy of the learning task, for different UE's energy constraints, while the ES's energy constraint

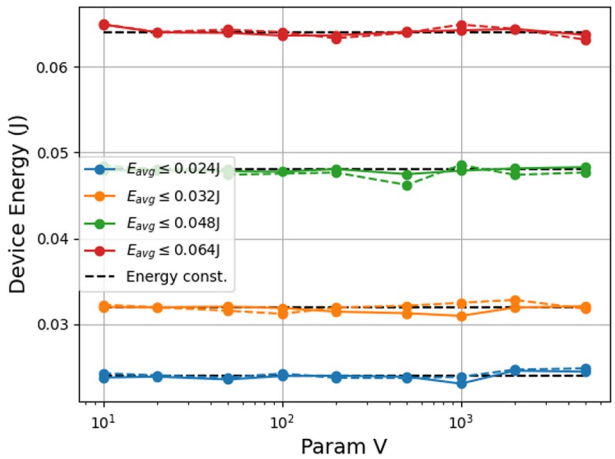

Fig. 11 UE's energy expenditure vs parameter *V*. Deep-CE (solid) vs down-sampling (dashed). This figure compares energy expenditure of the UE as a function of the trade-off parameter V considering the deep-CE and the down-sampling compression strategy

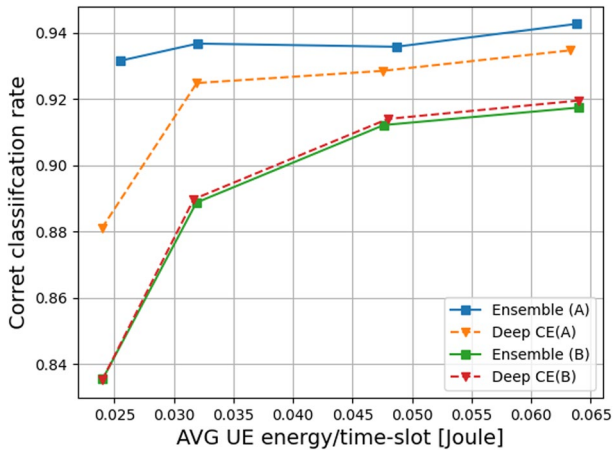

Fig. 12 Accuracy versus UE's energy, for different learning architectures and wireless channel scenarios (A, B). This figure shows the trade-off between energy and accuracy considering the deep-CE and the ensemble compression in different channel scenarios considering the maximum values of the trade-off parameter V

has been set equal to 3 Joule per time slot, i.e., a power of 60 W, which largely satisfies the task requests and pushes the algorithm toward the optimization of the UE's resources. As expected, Fig. 10 and 11 show that, while both the compression strategies satisfy the UE energy constraint, the deep-CE leads to a better latency–energy–accuracy trade-off, since it allows higher accuracy values even transmitting smaller data units, which obviously induce a lower transmission energy and latency.

Finally, we show the optimum (maximum) accuracy versus the UE's energy constraint, achieved by different CE-CCs architectures (i.e., deep-CE and ensemble) in different channel scenarios (i.e., A and B in Table 5). The values in Fig. 12 are obtained for the largest value of V (i.e., $V = 1 \times 10^5$), which push the system to tightly attain the latency constraint $1.50 \times 10^{-1} sec$. As far as the Channel B is concerned, due to the large channel attenuation (see Table 5), the transmission cost is more critical than the compression cost. Due to this reason, we do not observe for Channel B significant differences

among the CC-CEs ensemble and deep-CE-CCs, because the CE-CCs ensemble tends to employ the deep-CE model in almost every slot, since this strategy is capable to grant quite good accuracy also for large compression factors, which are the most appealing for this harsh and energy-expensive channel. On the other hand, when the channel conditions are moderately good, such as for Channel A in Table 5), the limiting factor is represented by the compression energy, which is higher for deep-CEs. In this case, for the strictest energy constraints, the UE with deep-CE tends to apply the highest compression factors, which save energy because they do not require the lossless zipping phase, but suffers of some correct classification degradation. Vice versa, the ensemble-CE-CCs tend to use also those CE-CCs with the lowest compression factors (e.g., 4/8/16), whose zipping phase is (computationally and energy-wise) less expensive with respect to the same compression factors of the deep-CE model. This fact explains the performance advantage of the ensemble-CS-CCs on the deep-CE-CCs alone.

6 Future directions

Several extensions and interesting research directions are open for investigation. For instance, the trade-off curves have shown that a careful choice of the regularization parameter V is needed to drive the system converging to the optimal solutions, i.e., those that are close to the constraints bounds. Thus, an interesting research direction for system deployment in practical scenarios is to develop algorithms that make the convergence fast, stable, and adaptive by properly controlling the regularization parameter V and the queues evolution step sizes λ and η.

We may also adapt the resource management strategy to scenarios where (low) latency or (high) accuracy constraints have to be almost always guaranteed, e.g., in *URLLC* network slices, and not just on average as we did in this manuscript. This could be done by imposing constraints on key performance indicators such as the out-of-service probability, as suggested in [60].

A key feature of our proposal is to enable the UE to dynamically select the most suitable CNN architecture to be used in every time slot, within a pool of possible architectures. Certainly, we might expand the pool by introducing other CNN architectures, with different number of nodes per layer, or different layers, or even alternative NN structures. Clearly, although we may want to expand the set of available architectures to choose from, with those capable to improve the accuracy performance, these new architectures may reasonably also require additional computational complexity and, possibly, larger power consumption at the mobile device. Then, an interesting research question is how to make a better trade-off between not only accuracy, energy consumption and service delay, but also complexity.

A further possibility would be to perform data compression to a goal-oriented latent variable with dynamically adjustable size, by exploiting a single classification network that could possibly dynamically reconfigure itself to different compression factors. The use of the variational IB principle [37, 61] is a possible step toward this direction, which deserves to be further explored. Another possibility is to consider an opportunistic offloading to the ES, for those UEs that have enough computational capabilities to perform

themselves the task, when for instance either the channel conditions are too bad, or the ES queues too long, to respect latency constraints.

Together with the extension to OFDM modulations for frequency-selective channels that we already mentioned in Sect. , the proposed scenario could also be extended to a multi-user and/or multi-server scenario, where the UEs and ESs optimization problems maybe strongly coupled.

Finally, the design of a proper online training procedure is another interesting research direction for the proposed framework.

7 Conclusions

In this paper, we have analyzed the trade-offs between energy, latency, and accuracy in an edge learning scenario equipped with goal-oriented communications, designing an adaptive classification network based on CEs and CCs. For such goal-oriented communication system, we designed two resource optimization strategies, hinging on the Lyapunov stochastic optimization framework. The proposed strategies optimize *dynamically* and *jointly* the communication parameters (i.e., rates, compression factors) and the computation resources (i.e., CPU clock cycles of UE and ES) with the aim of striking the best trade-off between energy, latency, and accuracy of the edge learning task. Even in the complex dynamic learning scenario considered in the paper, the proposed approaches require only low-complexity procedures at each time slot and enable online adaptation of the CE at the UE to dynamically control the goal-oriented communication mechanism. The presence of tunable parameters, which can be used to dynamically weight the different terms of the cost functions, makes the resource management very flexible. Finally, our experimental results have shown that using CEs to compress images at the UE leads to good performance at the ES, also with extreme compression factors, for a quite challenging classification task with 4.3×10^1-classes. Several simulations assess the good performance of the proposed strategies, illustrating the potential gain and adaptation capabilities.

Abbreviations

MEC	Mobile edge computing
EI	Edge intelligence
UE	User equipment
ES	Edge server
AP	Access point
CAE	Convolutional auto-encoder
CNN	Convolutional neural network
CE	Convolutional encoder
CC	Convolutional classifier
ML	Machine learning
AI	Artificial intelligence
GOC	Goal-oriented communications
IB	Information bottleneck
MADE	Maximum accuracy under average service delay and energy constraints strategy
MEDA	Minimum energy under average service delay and accuracy constraints strategy
DU	Data unit

Acknowledgements
Not applicable.

Author contributions
All the authors have contributed to the problem formulation, the overall case study and architecture, as well as the editing of the paper. FB is the principal contributor to the algorithmic development, under supervision of PB. FB has also contributed to practical implementation and coding. All authors read and approved the final manuscript.

Authors information

Francesco Binucci received the bachelor's and M.Sc. degrees in computer engineering in 2019 and 2021, respectively, from the University of Perugia, Perugia, Italy, where he is currently pursuing the Ph.D. degree in industrial and information engineering with the Department of Engineering. His research interests include resource management for wireless communications, edge machine learning, signal processing theory and methods, and data science.

Paolo Banelli received the Laurea degree (cum laude) in electronics engineering and the Ph.D. degree in telecommunications from the University of Perugia, Perugia, Italy, in 1993 and 1998, respectively. Since 2019, he has been a Full Professor with the Department of Engineering, University of Perugia, where he has been an Associate Professor, since 2005, and an Assistant Professor, since 1998. He was a Visiting Researcher with the University of Minnesota, Minneapolis, MN, USA, in 2001, and a Visiting Professor with Stony Brook University, Stony Brook, NY, USA, from 2019 to 2020. His current research interests include signal processing theory and methods, wireless communications and edge intelligence, goal-oriented communications, graph signal processing, and distributed learning. He was a member of the IEEE Signal Processing for Communications and Networking Technical Committee, from 2011 to 2013. In 2009, he was the General Co-Chair of the IEEE International Symposium on Signal Processing Advances for Wireless Communications. He served as an Associate Editor for the IEEE Transactions on Signal Processing (from 2013 to 2016), EURASIP Journal on Advances in Signal Processing (from 2013 to 2020), and IEEE Open Journal of Signal Processing (since 2020).

Paolo Di Lorenzo received the M.Sc. and Ph.D. degrees in electrical engineering from Sapienza University of Rome, Rome, Italy, in 2008 and 2012, respectively, where he is currently an Associate Professor with the Department of Information Engineering, Electronics, and Telecommunications. In 2010, he held a visiting research appointment with the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA, USA. From May 2015 to February 2018, he was an Assistant Professor with the Department of Engineering, University of Perugia, Perugia, Italy. He has participated in the FP7 European research projects FREEDOM, on femtocell networks; SIMTISYS, on moving target detection and imaging using a constellation of satellites; and TROPIC, on communication, computation, and storage over collaborative femtocells. He is a Principal Investigator of the research unit (CNIT-Sapienza) in the H2020 European Project RISE 6G. His research interests include signal processing theory and methods, distributed optimization, wireless edge intelligence, goal-oriented and semantic communications, and graph signal processing. He was a recipient of the three best student paper awards, respectively, at IEEE SPAWC10, EURASIP EUSIPCO11, and IEEE CAMSAP11. He was also a recipient of the 2012 GTTI (Italian National Group on Telecommunications and Information Theory) Award for the Best Ph.D. thesis. He is currently an Associate Editor for the IEEE Transactions on Signal and Information Processing Over Networks.

Sergio Barbarossa received the M.S. and Ph.D. degrees in EE from Sapienza University of Rome, where he is currently a Full Professor and a Senior Research Fellow of the Sapienza School of Advanced Studies. He has held visiting positions with the Environmental Research Institute of Michigan 1988, University of Virginia in 1995 and 1997, and University of Minnesota 1999. His current research interests are in the area of mobile edge computing and machine learning, graph signal processing, and distributed optimization. He received the IEEE Best Paper Award from the IEEE Signal Processing Society in the years 2000, 2014, and 2020. He received the Technical Achievements Award from the EURASIP Society in 2010. He coauthored the papers that received the Best Student Paper Award at ICASSP 2006, Signal Processing Advances in Wireless Communications (SPAWC) 2010, EUSIPCO 2011, and CAMSAP 2011. He has been the scientific coordinator of several EU projects on wireless sensor networks, small cell networks, distributed mobile cloud computing, and edge computing in 5G networks. He is now leading a national project on edge learning and he is involved in two H2020 European projects on 5G networks for Industry 4.0 and on reconfigurable intelligent surfaces. He served as an Associate Editor for the IEEE Transactions on Signal Processing from 1998 to 2000 and from 2004 to 2006, the IEEE Signal Processing Magazine, and the IEEE Transactions on Signal and Information Processing Over Networks. He has been the General Chairman of the IEEE Workshop on SPAWC in 2003 and the Technical Co-Chair of SPAWC in 2013. He has been the Guest Editor for Special Issues on the IEEE Journal on Selected Areas in Communications, EURASIP Journal of Applied Signal Processing, EURASIP Journal on Wireless Communications and Networking, the IEEE Signal Processing Magazine, and the IEEE Selected Topics on Signal Processing. From 1997 to 2003, he was a member of the IEEE Technical Committee for Signal Processing in Communications. He is an EURASIP Fellow and he has been an IEEE Distinguished Lecturer.

Funding

This work was supported by MIUR under the PRIN 2017 Liquid Edge project and by the H2020 EU/Taiwan Project 5G-CONNI, under contract nr. AMD-861459-3.

Availability of data and materials

All results are included in this published article. The codes used for generating the results are available from the corresponding author upon reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 27 June 2022 Accepted: 12 December 2022
Published online: 22 December 2022

References

1. E.C. Strinati, S. Barbarossa, J.L. Gonzalez-Jimenez, D. Ktenas, N. Cassiau, L. Maret, C. Dehos, 6g: The next frontier: from holographic messaging to artificial intelligence using subterahertz and visible light communication. IEEE Veh. Technol. Mag. **14**(3), 42–50 (2019)
2. W. Jiang, B. Han, M.A. Habibi, H.D. Schotten, The road towards 6g: a comprehensive survey. IEEE Open J. Commun. Soc. **2**, 334–366 (2021)
3. Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, J. Zhang, Edge intelligence: paving the last mile of artificial intelligence with edge computing. Proc. IEEE **107**(8), 1738–1762 (2019)
4. J. Park, S. Samarakoon, M. Bennis, M. Debbah, Wireless network intelligence at the edge. Proc. IEEE **107**(11), 2204–2239 (2019)
5. M. Merluzzi, P. Di Lorenzo, S. Barbarossa, Dynamic resource allocation for wireless edge machine learning with latency and accuracy guarantees, in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 9036–9040 (2020). IEEE
6. M. Merluzzi, P. Di Lorenzo, S. Barbarossa, Wireless edge machine learning: resource allocation and trade-offs. IEEE Access **9**, 45377–45398 (2021)
7. N. Skatchkovsky, O. Simeone, Optimizing pipelined computation and communication for latency-constrained edge learning. IEEE Commun. Lett. **23**(9), 1542–1546 (2019)
8. S. Wang, T. Tuor, T. Salonidis, K.K. Leung, C. Makaya, T. He, K. Chan, When edge meets learning: adaptive control for resource-constrained distributed machine learning, in, IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 63–71 (2018)
9. U. Mohammad, S. Sorour, Adaptive task allocation for mobile edge learning, in, 2019 IEEE Wireless Comm. and Networking Conf. Workshop (WCNCW), pp. 1–6 (2019)
10. M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, M. Zorzi, Toward 6g networks: use cases and technologies. IEEE Commun. Mag. **58**(3), 55–61 (2020)
11. E.C. Strinati, S. Barbarossa, 6g networks: beyond shannon towards semantic and goal-oriented communications. Comput. Netw. **190**, 107930 (2021)
12. S. Raza, S. Wang, M. Ahmed, M.R. Anwar, A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions. Wirel. Commun. Mob. Comput. **2019**, 3159762:1–3159762:19 (2019)
13. D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach, F. Giust, Mobile-edge computing architecture: the role of mec in the internet of things. IEEE Consum. Electron. Mag. **5**(4), 84–91 (2016)
14. C. Battiloro, P. Di Lorenzo, P. Banelli, S. Barbarossa, Dynamic resource optimization for decentralized estimation in energy harvesting iot networks. IEEE Internet Things J. **8**(10), 8530–8542 (2020)
15. E. Bourtsoulatze, D.B. Kurka, D. Gündüz, Deep joint source-channel coding for wireless image transmission. IEEE Trans. Cognit. Commun. Netw. **5**(3), 567–579 (2019)
16. C.-H. Lee, J.-W. Lin, P.-H. Chen, Y.-C. Chang, Deep learning-constructed joint transmission-recognition for internet of things. IEEE Access **7**, 76547–76561 (2019)
17. M. Jankowski, D. Gündüz, K. Mikolajczyk, Wireless image retrieval at the edge. IEEE J. Sel. Areas Commun. **39**(1), 89–100 (2020)
18. T.-Y. Tung, D.B. Kurka, M. Jankowski, D. Gunduz, Deepjscc-q: constellation constrained deep joint source-channel coding. arXiv preprint arXiv:2206.08100 (2022)
19. D.B. Kurka, D. Gündüz, Deepjscc-f: deep joint source-channel coding of images with feedback. IEEE J. Select. Areas Inform. Theory **1**(1), 178–193 (2020)
20. M. Yang, C. Bian, H.-S. Kim, OFDM-guided deep joint source channel coding for wireless multipath fading channels. IEEE Trans. Cognit. Commun. **8**(2), 584–599 (2022)
21. J. Dai, S. Wang, K. Tan, Z. Si, X. Qin, K. Niu, P. Zhang, Nonlinear transform source-channel coding for semantic communications. IEEE J. Select. Areas Commun. **40**(8), 2300–2316 (2022)
22. S. Wang, J. Dai, Z. Liang, K. Niu, Z. Si, C. Dong, X. Qin, P. Zhang, Wireless deep video semantic transmission. arXiv preprint arXiv:2205.13129 (2022)
23. J. Ballé, P.A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S.J. Hwang, G. Toderici, Nonlinear transform coding. IEEE J. Select. Top. Signal Process. **15**(2), 339–353 (2020)
24. H. Xie, Z. Qin, G.Y. Li, B.-H. Juang, Deep learning enabled semantic communication systems. IEEE Trans. Signal Process. **69**, 2663–2675 (2021)
25. X. Peng, Z. Qin, D. Huang, X. Tao, J. Lu, G. Liu, C. Pan, A robust deep learning enabled semantic communication system for text. arXiv preprint arXiv:2206.02596 (2022)
26. Z. Weng, Z. Qin, G.Y. Li, Semantic communications for speech recognition. arXiv preprint arXiv:2107.11190 (2021)
27. P. Jiang, C.-K. Wen, S. Jin, G.Y. Li, Deep source-channel coding for sentence semantic transmission with HARQ. IEEE Trans. Commun. **70**(8), 5225–5240 (2022)
28. H. Xie, Z. Qin, G.Y. Li, Task-oriented semantic communications for multimodal data. arXiv preprint arXiv:2108.07357 (2021)
29. H. Xie, Z. Qin, X. Tao, K.B. Letaief, Task-oriented multi-user semantic communications. arXiv preprint arXiv:2112.10255 (2021)
30. M.K. Farshbafan, W. Saad, M. Debbah, Common language for goal-oriented semantic communications: a curriculum learning framework. arXiv preprint arXiv:2111.08051 (2021)
31. M.K. Farshbafan, W. Saad, M. Debbah, Curriculum learning for goal-oriented semantic communications with a common language. arXiv preprint arXiv:2204.10429 (2022)
32. X. Kang, B. Song, J. Guo, Z. Qin, F.R. Yu, Task-oriented image transmission for scene classification in unmanned aerial systems. arXiv preprint arXiv:2112.10948 (2021)
33. H. Xie, Z. Qin, G.Y. Li, Task-oriented multi-user semantic communications for VQA. IEEE Wirel. Commun. Lett. 11(3), 553–557 (2022)
34. N. Tishby, F.C. Pereira, W. Bialek, The information bottleneck method. arXiv preprint arXiv:physics/0004057 (2000)

35. Z. Goldfeld, Y. Polyanskiy, The information bottleneck problem and its applications in machine learning. IEEE J. Select. Areas Inform. Theory **1**(1), 19–38 (2020)
36. F. Pezone, S. Barbarossa, P. Di Lorenzo, Goal-oriented communication for edge learning based on the information bottleneck. in, ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8832–8836 (2022). IEEE
37. J. Shao, Y. Mao, J. Zhang, Learning task-oriented communication for edge inference: An information bottleneck approach. arXiv preprint arXiv:2102.04170 (2021)
38. J. Shao, Y. Mao, J. Zhang, Task-oriented communication for multi-device cooperative edge inference. IEEE Trans. Wirel. Commun. (2022). https://doi.org/10.1109/TWC.2022.3191118
39. P.A. Stavrou, M. Kountouris, A rate distortion approach togoal-oriented communication. In, 2022 IEEE International Symposiumon Information Theory (ISIT), pp. 590–595 (2022)
40. F. Liu, W. Tong, Z. Sun, C. Guo, Task-oriented semantic communication systems based on extended rate-distortion theory. arXiv preprint arXiv:2201.10929 (2022)
41. C. Liu, C. Guo, Y. Yang, N. Jiang, Adaptable semantic compression and resource allocation for task-oriented communications. arXiv preprint arXiv:2204.08910 (2022)
42. M. Merluzzi, C. Battiloro, P. Di Lorenzo, E.C. Strinati, Energy-efficient classification at the wireless edge with reliability guarantees. arXiv preprint arXiv:2204.10399 (2022)
43. M. Merluzzi, A. Martino, F. Costanzo, P. Di Lorenzo, S. Barbarossa, Dynamic ensemble inference at the edge. in, 2021 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2021). IEEE
44. X. Kang, B. Song, J. Guo, Z. Qin, F.R. Yu, Task-oriented image transmission for scene classification in unmanned aerial systems. IEEE Trans. Commun. **70**(8), 5181–5192 (2022)
45. M.J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems* (Morgan and Claypool Publishers, 2010)
46. G. Chechik, A. Globerson, N. Tishby, Y. Weiss, Informationbottleneck for gaussian variables. J. Mach. Learn. Res. **6**, 165–188 (2005)
47. K.G. Larkin, Reflections on shannon information: in search of a natural information-entropy for images. arXiv preprint arXiv:1609.01117 (2016)
48. Y. Zhang, A better autoencoder for image: convolutional autoencoder. In, ICONIP17-DCEC (2018)
49. C.M. Bishop, N.M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4 (Springer, 2006)
50. M. Boudiaf, J. Rony, I.M. Ziko, E. Granger, M. Pedersoli, P. Piantanida, I.B. Ayed, A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In, European Conference on Computer Vision, pp. 548–564 (2020). Springer
51. A. Elad, D. Haviv, Y. Blau, T. Michaeli, Direct validation of the information bottleneck principle for deep nets. In, Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV) (2019)
52. H. Noh, T. You, J. Mun, B. Han, Regularizing deep neural networks by noise: its interpretation and optimization. In, Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 5115–5124 (2017)
53. J.D. Little, A proof for the queuing formula: L= λw. Oper. Res. **9**(3), 383–387 (1961)
54. C.E. Shannon, A mathematical theory of communication. The Bell Syst. Tech. J. **27**(3), 379–423 (1948)
55. T. Burd, R. Brodersen, Processor design for portable systems. J. VLSI Signal Process. Syst. Signal, Image Video Technol. **13**(2), 203–221 (1996)
56. A.F. Molisch, Statistical description of the wireless channel. In, Wireless Communications, pp. 69–99 (2011). IEEE
57. S. Sun, T.S. Rappaport, S. Rangan, T.A. Thomas, A. Ghosh, I.Z. Kovacs, I. Rodriguez, O. Koymen, A. Partyka, J. Jarvelainen, Propagation path loss models for 5g urban micro-and macro-cellular scenarios. In, 2016 IEEE 83rd Vehicular Techn. Conf. (VTC Spring), pp. 1–6 (2016)
58. J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, The german traffic sign recognition benchmark: a multi-class classification competition. In, The 2011 International Joint Conference on Neural Networks, pp. 1453–1460 (2011). IEEE
59. J. Li, A. Bose, Y.Q. Zhao, Rayleigh flat fading channels' capacity. In, 3rd Annual Communication Networks and Services Research Conference (CNSR'05), pp. 214–217 (2005). IEEE
60. M. Merluzzi, P. Di Lorenzo, S. Barbarossa, V. Frascolla, Dynamic computation offloading in multi-access edge computing via ultra-reliable and low-latency communications. IEEE Trans. Signal Inform. Process. Netw. **6**, 342–356 (2020)
61. A.A. Alemi, I. Fischer, J.V. Dillon, K. Murphy, Deep Variational Information Bottleneck. cite arxiv:1612.00410 (2016)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.