Dynamic Ensemble Inference at the Edge

Mattia Merluzzi[1], Alessio Martino[2], Francesca Costanzo[3], Paolo Di Lorenzo[3], Sergio Barbarossa[3]
[1]CEA-Leti, Université Grenoble Alpes, F-38000 Grenoble, France
[2]Italian National Research Council, ISTC-CNR, Via San Martino della Battaglia 44, 00185 Rome, Italy
[3]Sapienza University of Rome, DIET, Via Eudossiana 18, 00184 Rome, Italy
e-mail: {mattia.merluzzi}@cea.fr; {alessio.martino}@istc.cnr.it;
{francesca.costanzo, paolo.dilorenzo, sergio.barbarossa}@uniroma1.it

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685597

***Abstract*—We propose a dynamic resource allocation algorithm in the context of future wireless networks endowed with edge computing, to enable accurate energy efficient classification with end-to-end delay guarantees. In our scenario, sensor devices continuously upload data to an Edge Server (ES) for classification purposes. Merging Lyapunov stochastic optimization and ensemble inference, we propose DEsIreE, a low-complexity method that dynamically selects the data quantization level, the device transmit power, and the ES's CPU scheduling, without any prior knowledge of the statistics of wireless channels and data arrivals. Numerical simulations run on two real datasets assess the effectiveness of our algorithm in optimizing sensors' energy consumption and classification accuracy, with the ensemble yielding considerable gain.**

***Keywords*—Edge machine learning, energy efficiency, green edge computing, ensemble learning, mobile edge computing.**

I. Introduction

The future of wireless networks is to embed human, physical and digital world into the same ecosystem[1], with a holistic view of communication, computation, caching and control [1]. This complex integration will be driven by different pillars, including a performance boost over the wireless interface, and a pervasive deployment of cloud resources at the edge of the network. The latter will bring Machine Learning (ML) and Artificial Intelligence close to the end users, thus achieving new targets in terms of energy efficiency, delay, reliability and, in general, a sustainable deployment of 6G services. However, this comes with several drawbacks and challenges, among which we aim to address the following: i) Edge computing resources are limited with respect to the central cloud; ii) Reliability/dependability has to increase also from a computation point of view. Indeed, while communication reliability refers to the correct reception of packets sent through the wireless channels, computation reliability refers to the performance (e.g. in terms of accuracy) of learning/inference tasks running on the edge servers [2]. Therefore, when a learning/inference task is offloaded from end users (such as mobile phones, sensor devices, cars) to Edge Servers (ESs) through a wireless connection, the whole procedure can encounter different adversarial events. In this paper, we focus on the computation reliability, in order to design a methodology to increase the accuracy of a classification task running in the ES on data transmitted by sensor devices or, equivalently, the energy efficiency for a given classification accuracy. The basic idea, first presented in [2], is to introduce a certain distortion on the transmitted data through a coarser quantization, in order to save transmit energy. The drawback of a coarser quantization is a reduced classification performance in terms of accuracy. A possible counteracting measure, presented in this paper, is to *exploit diversity from a computation point of view*, with the decision based on the *ensemble* of more classification methods. Our work falls under the framework known as *Edge Machine Learning* (EML), whose brief literature review is presented in the following, along with the contribution of this paper.

Recently, different contributions on resource allocation for EML have been proposed [2]–[5]. In [3], a scheduling algorithm based on Lyapunov optimization is proposed, with Age of Information guarantees and predictive control of industrial actuators. In [4], the authors investigate a scenario where a single device uploads data to an edge server running a training model based on stochastic gradient descent, jointly considering communication and computation time, and optimizing the packet payload size, considering the ratio between communication and computation rates. The authors in [5] propose a communication-efficient decentralized machine learning algorithm that dynamically optimizes a stochastic quantization method, with applications to regression and image classification. Finally, in [2] we propose three different resource allocation strategies to explore the trade-off between energy, latency and accuracy in EML scenarios. None of these works consider ensemble inference to improve the aforementioned trade-off. In [6], the authors propose an ensemble method for edge inference with deep neural networks. However, they only consider delay and not the joint optimization of delay, accuracy and energy.

Instead, in this paper, we propose DEsIreE, an online method based on Lyapunov stochastic optimization [7] and ensemble inference, to optimize a weighted sum of energy consumption and inference performance, with constraints on the average End-to-End (E2E) delay. In particular, DEsIreE dynamically adjusts the data quantization, the devices' transmit power, and the CPU scheduling at the ES, to strike the best trade-off between energy consumption, delay, and accuracy of the classification task, exploiting diversity from a computation point of view to increase reliability. Our contribution is twofold: i) We design an online method able to strike a trade-off between energy, latency and inference without any knowledge of the statistics of context parameters; ii) We propose a novel ensemble method to combine different classifiers, based on a notion of *confidence* of their classification. Our method is tested on the MNIST dataset [8], as in [5], and on the Hydraulic System Monitoring (HSM) dataset [9], with different ensembles of Support

This work was partly funded by the European Commission through the H2020 project Hexa-X (Grant Agreement no. 101015956), and the H2020 EU/TW Project 5G CONNI, Nr. 861459, and by MIUR under the PRIN Liquid-Edge contract

[1]https://hexa-x.eu/

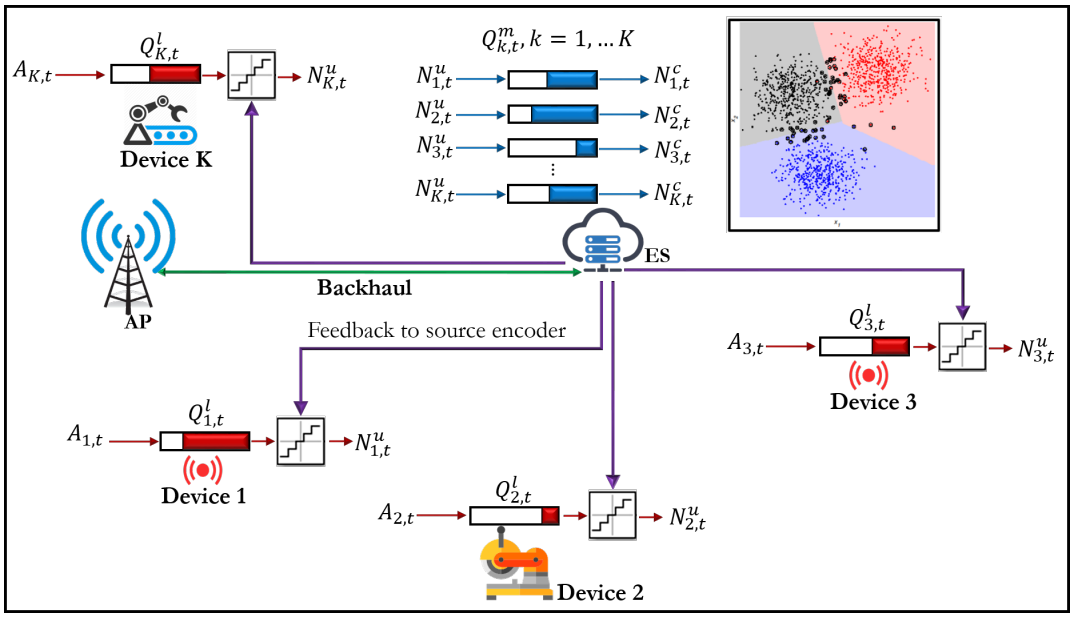

Fig. 1: Reference scenario for edge classification

Vector Machines (SVMs) [10] and Neural Networks (NNs). Numerical results show that our ensemble method is able to guarantee the same classification performance within the same E2E delay, with considerable gains in terms of sensor energy consumption. To the best of our knowledge, this is the first work that considers a dynamic joint resource allocation strategy for edge classification, with ensemble inference. As in [2], we assume a *goal-oriented* perspective, where the goal is to accomplish the classification task with acceptable accuracy, given the final application, with the lowest energy consumption, and not necessarily to perfectly convey all the original data. Classical approaches focus on energy efficiency in terms of, e.g., energy per bit. Focusing on the energy per goal, we can rely on the following notion of energy efficiency: $\text{EE} = \frac{\text{Energy}}{\text{goal}} = \frac{\text{Energy}}{\text{bit}} \times \frac{\text{bits}}{\text{goal}}$. Our proposal is not only to reduce the first term on the energy per bit, but also the *bits per goal*, thus further improving the energy efficiency.

II. System Model

As illustrated in Fig. 1, we consider sensors that continuously generate data and upload them via the wireless connection with an Access Point (AP), to an ES, which runs an inference task. Typical examples are cameras uploading video frames to the edge server ES for object detection and recognition, or industrial sensors uploading physical quantities for anomaly detection, predictive maintenance and process diagnosis. As we can notice from Fig. 1, each device buffers its data into a local communication queue (in red), to then transmit them to the ES, which stores the data into a remote computation queue (in blue) before computation. The output of the computation is a classification result (top right of the figure), coming from a single classifier, or from an ensemble strategy. Since we deal with a dynamic scenario, time is organized in time slots t of equal duration τ.

A. End-to-End delay

In this paper, we refer to a data unit as the smallest piece of information to be sent by a sensor and elaborated by the ES, as in [2]. For example, in image processing, the data unit is one image. We denote by S_k the number of samples in one data unit, for example the number of pixels in an image. Denoting by $n^q_{k,t}$ the number of bits per sample used by device k in time slot t, a transmitted data unit is represented by $S_k n^q_{k,t}$ bits. The overall delay experienced by a data unit, from its generation to its classification at the ES, is related to both the uplink queueing and transmission delays, and the remote queueing and computation delays at the ES. Then, in this paper, as in [2], we define an uplink transmission queue $Q^l_{k,t}$, and a remote queue $Q^m_{k,t}$ of data to be computed at the ES. The uplink queue is fed by the new data arrivals, and drained by the uplink data transmission. Thus, denoting by $R_{k,t}$ the uplink data rate (in bits/s) of device k, the total number of transmitted data units in time slot t is

$$N^u_{k,t} = \left\lfloor \frac{\tau R_{k,t}}{S_k n^q_{k,t}} \right\rfloor, \tag{1}$$

where $\lfloor x \rfloor$ denotes the largest integer smaller than x, and it is assumed that $n^q_{k,t}$ is the number of bits used to represent *all* data units sent by device k, during time slot t. Then, the local communication queue evolves as follows:

$$Q^l_{k,t+1} = \max\left(0, Q^l_{k,t} - N^u_{k,t}\right) + A_{k,t} \tag{2}$$

where $A_{k,t}$ are the new data arrivals, whose statistics are supposed to be unknown a priori. The importance of the quantization level will be clear later on in this section, when we introduce the accuracy of the edge inference task. At the server, the remote queue is fed by the uplink data arrivals, and is drained by the task computation. In particular, in this work, as in [2], we assume that there exist a linear relation between the number of CPU cycles, and the computed data units. This assumption builds on the fact that, given a trained classification model, a fixed number of operations (summations, non-linear operations, etc.) need to be performed on a new pattern to output the result. Then, denoting by $1/J_k$ the number of CPU cycles needed to elaborate one data unit, the number of computed data is given by

$$N^c_{k,t} = \lfloor \tau f_{k,t} J_k \rfloor, \tag{3}$$

where $f_{k,t}$ is the CPU cycle frequency assigned to device k during time slot t. As we will detail later on, J_k depends on the specific learning algorithm run by the ES (SVM, NN, etc.). Then, the remote queue evolves as follows:

$$Q^m_{k,t+1} = \max(0, Q^m_{k,t} - N^c_{k,t}) + \min\left(Q^l_{k,t}, N^u_{k,t}\right) \quad (4)$$

In particular, as we will show later on, the parameter J_k is estimated offline for each classification algorithm, with the ensemble being more computational demanding. The overall service delay is directly related to the sum of the local and the computation queues $Q^{\text{tot}}_{k,t} = Q^l_{k,t} + Q^m_{k,t}$. One of our aims is to guarantee a long-term average delay D^{avg}_k, as follows

$$\lim_{T\to\infty} \frac{1}{T} \sum\nolimits_{t=1}^{T} \mathbb{E}\left\{Q^{\text{tot}}_{k,t}\right\} \leq Q^{\text{avg}}_k := D^{\text{avg}}_k \bar{A}_k, \quad \forall k \quad (5)$$

where $\bar{A}_k = \mathbb{E}\{A_{k,t}\}/\tau$, due to the Little's law [11]. Here, the expectation is taken with respect to the random channel and data arrivals, whose statistics are unknown in advance.

B. Energy consumption

In this paper, we aim to optimize a combination of the sensors' energy consumption and inference accuracy. In particular, we consider the energy spent for the transmission of patterns from the sensor devices to the AP, thus in the uplink direction. Given a data rate $R_{k,t}$, from Shannon's formula, the energy spent for transmission during time slot t by device k is

$$e_{k,t} = \frac{\tau N_0 B_k}{h_{k,t}} \left(\exp\left(\frac{R_{k,t}\ln(2)}{B_k}\right) - 1\right),$$

where N_0 is the noise power spectral density at the receiver, B_k is the bandwidth assigned to device k, and $h_{k,t}$ is the channel power gain, which is time varying in wireless scenarios and in general includes path loss, shadowing and fading.

C. Inference Accuracy

As already mentioned, an important aspect pertaining to edge inference is the reliability of the task. In the case of a classification task, this translates into the percentage of correctly classified patterns. This parameter is strongly affected by the learning algorithm (or ensemble of algorithms), and the number of bits used to quantize the data. In particular, denoting by $G_k(n^q_k)$ the accuracy (a function of the number of quantization bits), we make the following assumption:

Assumption 1: G_k is a non decreasing function of n^q_k.

Assumption 1 is based on the fact that the quantization represents a noise over the test data set. For more specific arguments involving rate-distortion theory limits, the interested author is referred to [2]. However, it should be noted that, while a higher n^q_k leads to better accuracy, it also requires higher energy consumption to upload the data and process them within the E2E delay constraint, and viceversa. Due to Assumption 1, since one of our aims is to maximize the accuracy, we will directly consider n^q_k in the following.

III. Problem formulation and solution

Since the goal is to explore the trade-off between energy consumption, E2E delay, and inference accuracy, we consider the following weighted sum of inference accuracy (directly the number of bits due to Assumption 1) and energy consumption:

$$E^w_t = \sum\nolimits_{k=1}^{K} (-\alpha_k n^q_{k,t} + (1-\alpha_k) e_{k,t}), \quad (6)$$

where $\alpha_k \in [0,1]$ is a parameter that assigns more or less importance to the energy or the accuracy. For instance, $\alpha_k = 1$ leads to a pure energy minimization problem as in [12]. Then, the problem can be formulated as:

$$\begin{aligned} &\min_{\xi_t} \ \lim_{T\to\infty} \frac{1}{T} \sum\nolimits_{t=1}^{T} \mathbb{E}\left\{E^w_t\right\} \quad (7)\\ &\text{s.t.} \quad (a) \ \text{Eqn. (5)};\\ &(b) \quad 0 \leq R_{k,t} \leq R^{\max}_{k,t}, \ \forall k,t; \quad (c) \ n^q_{k,t} \in \mathcal{N}^q_k, \quad \forall k,t;\\ &(d) \quad f_{k,t} \geq 0 \quad \forall k,t; \quad (e) \ \sum\nolimits_{k=1}^{K} f_{k,t} \leq f^{\max}_c, \quad \forall t; \end{aligned}$$

where $\xi_t = [\{R_{k,t}\}_{k=1}^K, \{f_{k,t}\}_{k=1}^K, \{n^q_{k,t}\}_{k=1}^K]$. The constraints in (7) have the following meaning: (a) the average E2E delay does not exceed a predefined threshold $D^{\text{avg}}_k = Q^{\text{avg}}_k/\bar{A}_k$, with $\bar{A}_k$ being the data arrival rate; (b) the data rate is non negative and is lower than $R^{\max}_{k,t}$ (maximum achievable rate given the current channel state); (c) $n^q_{k,t}$ belongs to a discrete set of possible quantization levels $\mathcal{N}^q_k$; (d) the CPU clock frequency allocated to each device is non negative; (e) the sum of the CPU frequencies allocated to each device does not exceed the total computational capacity of the ES, denoted by $f^{\max}_c$. Solving problem (7) is difficult due to the lack of knowledge of wireless channel and data arrival statistics. Then, we propose a solution based on Lyapunov stochastic optimization [7].

A. Algorithmic solution

The first challenge of problem (7) is to handle the long-term constraint (a) without assuming prior knowledge on the statistics of radio channels and data arrivals. Then, to handle this, we introduce *virtual queues* $Z_{k,t}, \forall k$, which evolves as

$$Z_{k,t+1} = \max\left(0, Z_{k,t} + Q^{\text{tot}}_{k,t+1} - Q^{\text{avg}}_k\right), \ \forall k. \quad (8)$$

The virtual queue is used to capture the state of the system (in terms of constraint violations) and take suitable actions to meet the long-term constraint, as it will be clarified later in this section. The virtual queues are used to transform problem (7) into a pure stability problem through the definition of suitable functions. In particular, the mean rate stability of these virtual queues ensures that the associated constraints (i.e. (a)) are met [7], and is defined as $\lim_{T\to\infty} \frac{\mathbb{E}\{Z_{k,T}\}}{T} = 0, \forall k$. Then, we introduce the Lyapunov function $L(\mathbf{Z}_t) = \frac{1}{2}\sum_{k=1}^{K} Z^2_{k,t}$, and the *drift-plus-penalty* function as [7]:

$$\Delta_p(\mathbf{Z}_t) = \mathbb{E}\{L(\mathbf{Z}_{t+1}) - L(\mathbf{Z}_t) + V E^w_t | \mathbf{Z}_t\} \quad (9)$$

The drift-plus-penalty function is the conditional expected change of $L(\mathbf{Z}_t)$ over one slot, plus a penalty factor with a weighting parameter V, used to trade-off the queue backlogs and the objective function of (7). In particular, a higher value of V leads to a lower energy consumption but higher delay, and viceversa. Now, if (9) is upper bounded by a finite constant, the virtual queues are mean rate stable, so that (a) is satisfied. Also, as V increases, the optimal value of the objective function of (7) is asymptotically reached [7]. To

this aim, we define the following upper bound of the drift-plus-penalty function:

$$\Delta_p(\mathbf{Z}_t) \leq C + \mathbb{E}\{ \sum\nolimits_{k=1}^{K} \big[\chi_{k,t} + 2(Q^m_{k,t} - Q^l_{k,t})N^u_{k,t} + Z_{k,t}\big(\max(0, Q^l_{k,t} - N^u_{k,t}) + \max(0, Q^m_{k,t} - N^c_{k,t})\big) + VE^w_t\big] \big| \mathbf{Z}_t \} \tag{10}$$

where C is a positive constant given by

$$C = \sum_{k=1}^{K} \left[\frac{(Q^{\text{avg}}_k)^2}{2} + (A^{\max}_k)^2 + 2(N^u_{k,\max})^2 + (N^c_{k,\max})^2 \right] \tag{11}$$

and $\chi_{k,t}$ reads as

$$\chi_{k,t} = (2Q^l_{k,t} + Z_{k,t})A_{k,t} + (Q^l_{k,t})^2 + (Q^m_{k,t})^2 + Z_{k,t}\min(Q^l_{k,t}, N^u_{k,\max}) \tag{12}$$

The simple mathematical derivations of (10), (11) and (12) are omitted due to space limitations, but follow similar arguments as in [12]. Finally, hinging on the concept of opportunistically minimizing an expectation, our strategy greedily minimizes (10) in each time slot. In particular, following some simple manipulations, omitted for the lack of space, we solve the following per-slot optimization problem:

$$\min_{\xi_t} \quad \sum_{k=1}^{K} \left(-\frac{\tau \widetilde{Q}^l_{k,t}}{S_k n^q_{k,t}} R_{k,t} - \widetilde{Q}^m_{k,t} \tau f_{k,t} J_k + VE^w_t \right) \tag{13}$$

$$\text{s.t.} \quad (a)\ 0 \leq R_{k,t} \leq \widetilde{R}^{\max}_k, \quad \forall k; \quad (b)\ n^q_{k,t} \in \mathcal{N}^q_k, \quad \forall k;$$

$$(c)\ 0 \leq f_{k,t} \leq \frac{(Q^m_{k,t} + 1)}{(\tau J_k)} \quad \forall k; \quad (d)\ \sum\nolimits_{k=1}^{K} f_{k,t} \leq f^{\max}_c,$$

where $\widetilde{Q}^l_{k,t} = 2Q^l_{k,t} - 2Q^m_{k,t} + Z_{k,t}$ and $\widetilde{Q}^m_{k,t} = 2Q^m_{k,t} + Z_{k,t}$, and $\widetilde{R}^{\max}_{k,t} = \min(R^{\max}_{k,t}, (Q^l_{k,t} + 1)S_k n^q_{k,t}/\tau)$. Note that, given n^q_k, (13) is a convex optimization problem, separable among radio ($R_{k,t}$) and computation ($f_{k,t}$) resource allocation. In particular, given $n^q_{k,t}$, it is easy to show that, if $\widetilde{Q}^l_{k,t} \leq 0$, the optimal solution is $R^*_{k,t} = 0$. Otherwise, by solving the Karush-Kuhn-Tucker conditions [13], the optimal data rate is

$$R^*_{k,t} = \left[B_k \log_2 \left(\frac{h_{k,t}\widetilde{Q}^l_{k,t}}{N_0 S_k n^q_{k,t}(1 - \alpha_k)V} \right) \right]_0^{\widetilde{R}^{\max}_{k,t}} \tag{14}$$

Then, in each time slot, the optimal data rate and the number of quantization bits are optimized by computing R^*_k for each $n^q_k \in \mathcal{N}^q_k$ (which is generally a discrete set with low cardinality), plugging the result into the objective function of (13), and choosing the solution that yields the lowest possible value. With respect to f_k, (13) is a Linear Programming problem, so that it can be efficiently solved via the iterative Algorithm 1 that requires, at most, K iterations.

IV. Building the Ensemble

Naïvely speaking, an ensemble can be built by considering a finite number M of classifiers that can either be trained simultaneously or independently, provided that in the latter case a decision rule is determined in order to mark the final predictions. In our proposed scheme, given an ensemble of M classifiers and a test pattern $\mathbf{x}$ to be classified, the decision rule, hence the predicted label $l(\mathbf{x})$, is given by the classifier having the *highest confidence* on its prediction. That is, each classifier, alongside its predicted label $\hat{y}$, returns a probability vector of the form

$$\mathbf{p}_i = \Pr\{y = i|\mathbf{x}\}, \qquad i = 1, \ldots, s, \tag{15}$$

where s is the number of classes. It should be noted that returning a posteriori probabilities is possible with most of the existing classification algorithms (e.g., SVM, NNs, bagged/boosted trees, random forests, etc.) [14], [15]. Now, starting from (15), it is possible to evaluate the entropy of the prediction as $H = \sum_{i=1}^{s} \mathbf{p}_i \log(1/\mathbf{p}_i)$. Hence, the output of the ensemble has the form $\{(\hat{y}^{(1)}, H^{(1)}), \ldots, (\hat{y}^{(M)}, H^{(M)})\}$. Finally,

$$l(\mathbf{x}) = \hat{y}^{(i)}, \quad \text{where } i = \text{argmin}\{H^{(1)}, \ldots, H^{(M)}\} \tag{16}$$

It is worth addressing two caveats of the entropy: i) it does not require any ground-truth label (i.e., the true y), making it a suitable index for assessing the behaviour of the classifier in an online fashion; ii) it is a non-increasing function of the number of classifiers composing the ensemble. Obviously, exploiting an ensemble of $M > 1$ classifiers comes with the drawback of an increased computation time. This is taken into account through the parameter J_k (cf. (3)). Indeed, if device k exploits an ensemble, the parameter J_k is reduced to take into account the fact that more CPU cycles are needed to elaborate a single pattern, since we assume that the M classifiers composing the ensemble run on the same core. In general, given M classifiers with $\left\{ J^{(1)}_k, \ldots, J^{(M)}_k \right\}$ we have

$$J_k = \left(\sum\nolimits_{n=1}^{M} \frac{1}{J^{(n)}_k} \right)^{-1} \tag{17}$$

Finally, the overall procedure for ensemble classification at the edge is summarized in Algorithm 2.

Algorithm 1: Optimal CPU scheduling at the ES

Input data: $\{Z_{k,t}\}_k$, $\{Q^m_{k,t}\}_k$, $\{J_k\}_k$, $f_{\max}$, K
$f_{\text{av}} = f_{\max}$, $\mathcal{U} = \{k = 1, \ldots, K\}$
while $f_{\text{av}} > 0 \quad \& \quad \mathcal{U} \neq \emptyset$ **do**
  S1. $\widetilde{k} = \underset{k \in \mathcal{U}}{\text{argmax}} \left\{ J_k \widetilde{Q}^m_k \right\}$;
  S2. $f_{\widetilde{k}} = \min\left((Q^m_{\widetilde{k}} + 1)/(\tau J_{\widetilde{k}}), f_{\text{av}} \right)$;
  S3. $\mathcal{U} = \mathcal{U} - \left\{ \widetilde{k} \right\}$;
  S4. $f_{\text{av}} = f_{\text{av}} - f_{\widetilde{k}}$.
end

Algorithm 2: DEsIreE

Input data: $\{Z_{k,0}\}_k$, $\{Q^{l,m}_{k,0}\}_k$, $\{J_k\}_k$, K
S1. Compute J_k as in (17), $\forall k$;
for $t = 1 : N_{slot}$ **do**
  S2. Compute the optimal data rate as in (14) and transmit $N^u_{k,t}$ patterns, $\forall k$ (cf. (1));
  S3. Find the optimal CPU scheduling with Algorithm 1 and classify $N^c_{k,t}$ patterns (cf.(3)) using (16);
  S4. Update $Q^l_{k,t}$, $Q^m_{k,t}$ and $Z_{k,t}$ as in (2), (4) and (8), respectively.
end

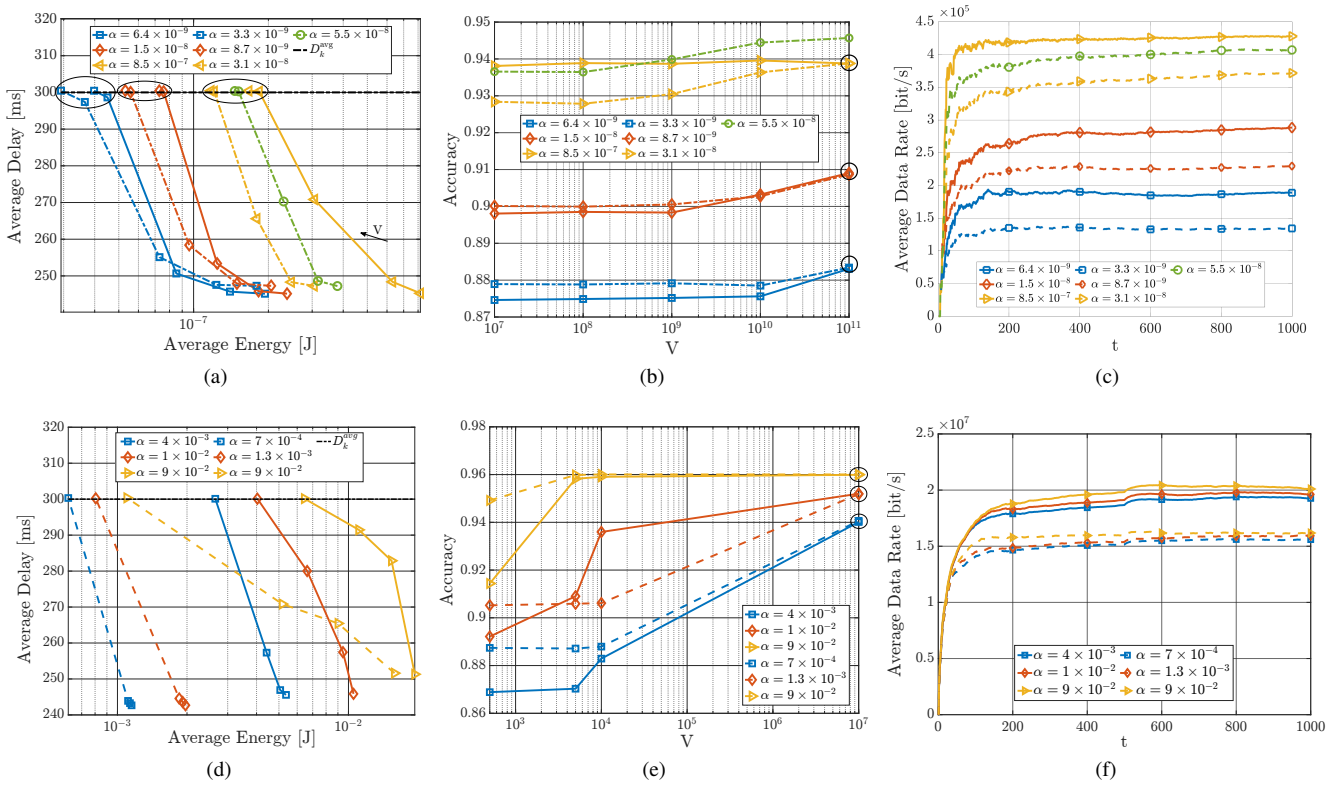

Fig. 2: Energy-delay-accuracy trade-off over the MNIST and HSM datasets

V. Numerical results

In this section, we show the performance of the proposed method. To test the proposed system on real data, the MNIST and HSM datasets are considered. The choice behind these two datasets stems from their realistic applications in edge scenarios (cf. Section II): in fact, whereas the former can emulate images sent from cameras for object (digit, in this case) recognition, the latter regards sensors that send measurements about some mechanical equipment for anomaly detection. However, let us note that our method is general and can be applied to any dataset. For the ensemble, two classifiers are considered as potential candidates: SVM with polynomial kernel (poly-SVM) and standard Multilayer Perceptron (MLP). As SVMs are concerned, Eq. (15) is evaluated thanks to a calibration phase of the classifier [16], [17], whereas MLPs, being provided with a soft-max activation function in the output layer, natively return probability estimates [18], [19]. In this work, we considered ensembles composed by $M = 2$ classifiers. All classifiers have been trained on the dataset which has been properly split in training, validation and test set. Training and validation sets have been used for hyper-parameters tuning, the test set has been left aside for assessing the final performance. For each dataset, homologous classifiers are trained with different hyper-parameters to ensure diversity within the ensemble. We consider $K = 5$ sensor devices, uniformly distributed over a square area of side 100 m, with the AP in the middle, each of them generating data with Poisson arrivals, with parameter uniformly distributed in $[3, 8]$. The channel model is the one presented in [20], with a unit variance Rayleigh fading. The noise power spectral density is -174 dBm/Hz, the total bandwidth is $B = 10$ MHz, equally shared among devices, each transmitting with 100 mW maximum power, at a carrier frequency of 6 GHz. The ES's CPU cycle frequency is set to $f_c^{\max} = 3.3$ GHz. The number J_k of CPU cycles needed to process each pattern has been estimated offline, by running the different classification algorithms: multiplying the CPU speed (in Hz) by the estimated time (in seconds) required to classify a given pattern, then one gets the number of cycles needed to perform the classification. J_k reads as the inverse of the latter, and in the case of single classifier the values are: $J_k^{\text{MNIST}} = 1.4\times10^{-7}$, $J_k^{\text{HSM}} = 3.54\times10^{-7}$, and are halved in the case of the ensemble. Our aim is to show the performance of the proposed strategy in terms of trade-off between energy and E2E delay, when achieving the same accuracy, and the associated convenience in using an ensemble instead of a single classifier. Let us first concentrate on the MNIST dataset, whose results are shown in Figs. 2 (a),(b),(c). In particular, Figs. 2a and 2b have to be read together as follows. Fig 2a shows the trade-off between average energy consumption (x-axis) and average E2E delay (y-axis), which is explored by increasing the trade-off parameter V (cf. (9)) from right to left (as shown in the plot). As we can notice, for all curves, the energy consumption decreases as V increases, while the average E2E delay increases until reaching the imposed constraint (black horizontal dashed line) with the minimum energy. The different curves represent a different value of α_k (cf. (6)), which weights energy consumption and accuracy. Also, the solid lines refer to the results obtained with a single classifier (poly-SVM), while the dashed lines represent the results obtained with the ensemble of two poly-SVMs. Furthermore, the curves (solid and dashed) with the same color and marker, refer to strategies achieving the same accuracy, which is shown in Fig. 2b. In particular, in Fig.2b, we plot the accuracy as a function of V, corresponding to the same simulation of Fig. 2a, and compare the performance of the single classifier and the ensemble in terms of energy-

delay trade-off. Indeed, by looking at Fig. 2b, we can notice how, for the highest value of V (which attains the optimum of (7)), both methods achieve the same accuracy (same color, same marker, but different line styles), also highlighted by the black circles on the right side of the figure. At the same time, looking at the trade-off in Fig. 2a, we can notice the non-negligible gain of the ensemble with respect to the single classifier. As a clarifying example, let us consider the blue curves (■) in Figs. 2a and 2b. From an accuracy point of view (Fig. 2b), they achieve the same value (around 88%), but the ensemble achieves a lower energy consumption (around 26% gain), within the same delay (Fig. 2a). This is clearly visible for all the other curves (the couples to be compared are highlighted with the black ellipses in Fig. 2a). Furthermore, the green curve (•) is not shown for the single classifier, due to the fact that the ensemble achieves an accuracy not achievable by the single classifier. In this regard, it should be noted that the green curve achieves a lower energy consumption than the yellow solid curve (▲), which represent the best accuracy case for the single classifier. This result further motivates the use of the ensemble, which is able to achieve an accuracy higher than the best case for the single classifier, with lower energy consumption, and equal E2E average delay.

Now, the question is: *Why does the ensemble achieve lower energy consumption?*

The answer is illustrated in Fig. 2c, which shows the average data rate necessary to achieve the performance in terms of E2E delay (plotted for the highest value of V of Fig. 2a). In particular, since the ensemble allows the use of a lower number of quantization bits without sacrificing the accuracy, the average data rate decreases, thus reducing the necessary transmit energy consumption. Concerning the HSM dataset, similar considerations can be done by looking at Figs. 2d, 2e, 2f, with even more gain achieved by the ensemble (83% for the yellow curves). In this case, the ensemble is composed by two MLPs. The higher gain comes from the fact that HSM presents much more features than MNIST, so that a lower n_k^q means a much lower number of bits to be transmitted. Thus, the final message of Fig. 2 is twofold: i) Our method is able to minimize the objective function of (7), thus minimizing the energy consumption and maximizing the accuracy, depending on the value of α_k, as V increases; ii) Using ensembles of algorithms, the same accuracy performance can be achieved with non negligible gains in term of average energy consumption, for the same average E2E delay. This is due to the reduction of the number of quantization bits to represent the data.

VI. Conclusions

In this paper, we proposed a novel dynamic method for latency constrained reliable classification at the edge. Starting from a long-term average optimization problem with unknown statistics of wireless channels and data arrivals, we exploited Lyapunov stochastic optimization tools to solve the problem in a per-slot fashion, with low complexity solutions. First, our method is able to explore trade-offs between energy consumption, delay and accuracy for edge inference. Second, by exploiting ensembles of classification algorithms, DEsIreE reduces the overall sensor energy consumption without any loss on the E2E delay and on the classification performance. The effectiveness of the proposed strategy has been tested on two real datasets.

References

[1] E. Calvanese Strinati *et al.*, "6G: The next frontier: From holographic messaging to artificial intelligence using subterahertz and visible light communication," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 42–50, 9 2019.

[2] M. Merluzzi, P. Di Lorenzo, and S. Barbarossa, "Wireless Edge Machine Learning: Resource Allocation and Trade-Offs," *IEEE Access*, vol. 9, pp. 45 377–45 398, 2021.

[3] A. M. Girgis, J. Park, C. F. Liu, and M. Bennis, "Predictive control and communication co-design: A gaussian process regression approach," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2020, pp. 1–5.

[4] N. Skatchkovsky and O. Simeone, "Optimizing Pipelined Computation and Communication for Latency-Constrained Edge Learning," *IEEE Communications Letters*, vol. 23, no. 9, pp. 1542–1546, 9 2019.

[5] A. Elgabli, J. Park, A. S. Bedi, M. Bennis, and V. Aggarwal, "Q-GADMM: Quantized Group ADMM for Communication Efficient Decentralized Machine Learning," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8876–8880.

[6] N. Shlezinger, E. Farhan, H. Morgenstern, and Y. C. Eldar, "Collaborative inference via ensembles on the edge," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8478–8482.

[7] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool Publishers, 2010.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] N. Helwig, E. Pignanelli, and A. Schütze, "Condition monitoring of a complex hydraulic system using multivariate statistics," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, 2015, pp. 210–215.

[10] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[11] J. D. C. Little, "A proof for the queuing formula: $l = \lambda w$," *Oper. Res.*, vol. 9, no. 3, p. 383–387, Jun. 1961.

[12] M. Merluzzi, P. Di Lorenzo, S. Barbarossa, and V. Frascolla, "Dynamic computation offloading in multi-access edge computing via ultra-reliable and low-latency communications," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 342–356, 2020.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[14] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '02. New York, NY, USA: Association for Computing Machinery, 2002, p. 694–699.

[15] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 625–632.

[16] J. Platt, "Probabilities for SV Machines," in *Advances in large margin classifiers*, A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA, USA: MIT Press, 2000, pp. 61–74.

[17] H.-T. Lin, C.-J. Lin, and R. C. Weng, "A note on platt's probabilistic outputs for support vector machines," *Machine Learning*, vol. 68, no. 3, pp. 267–276, Oct 2007.

[18] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 211–217.

[19] ——, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*, F. F. Soulié and J. Hérault, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 227–236.

[20] S. Sun *et al.*, "Propagation Path Loss Models for 5G Urban Micro- and Macro-Cellular Scenarios," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 5 2016, pp. 1–6.