Federated Feature Selection for Cyber-Physical Systems of Systems

Pietro Cassará, Alberto Gotta, *Member, IEEE*, and Lorenzo Valerio

***Abstract*—Autonomous vehicles (AVs) generate a massive amount of multi-modal data that once collected and processed through Machine Learning algorithms, enable AI-based services at the Edge. In fact, only a subset of the collected data present informative attributes to be exploited at the Edge. Therefore, extracting such a subset is of utmost importance to limit computation and communication workloads. Doing that in a distributed manner imposes the AVs to cooperate in finding an agreement on which attributes should be sent to the Edge. In this work, we address such a problem by proposing a federated feature selection (FFS) algorithm where the AVs collaborate to filter out, iteratively, the less relevant attributes in a distributed manner, without any exchange of raw data, thought two different components: a Mutual-Information-based feature selection algorithm run by the AVs and a novel aggregation function based on the Bayes theorem executed on the Edge. The FFS algorithm has been tested on two reference datasets: MAV with images and inertial measurements of a monitored vehicle, WESAD with a collection of samples from biophysical sensors to monitor a relative passenger. The numerical results show that the AVs converge to a minimum achievable subset of features with both the datasets, i.e., 24 out of 2166 (99%) in MAV and 4 out of 8 (50%) in WESAD, respectively, preserving the informative content of data.**

***Index Terms*—Artificial intelligence, autonomous system, feature selection, federated learning, human state monitoring, Internet of things, machine learning.**

I. Introduction

AUTOMATION enables a Cyber Physical System of Systems (CPSoS) to run with a minimum human assistance and evolves into autonomy when the human is taken out of the sensing, decision, and actuation loop. Automation can be used to operate a CPSoS comprising complex, dynamic, virtual and physical resources, such as telecommunication networks, computing units, software, sensors, and machines [1]. Humans can interact with an autonomous system either as passive end-users (such as passengers in autonomous transportation system) or rather as active co-operators in a mutual empowerment relationship towards a shared goal. Such cooperative, connected, and autonomous systems have the potential to be a game-changer in multiple domains if they will be capable of positively exploiting such an inescapable human factor. The increasing development of semi-Autonomous Driving Systems (ADSs) poses the challenge of taking the end-user, in the middle of the evolution process toward fully ADSs. Aside from vehicle control, a CPSoS needs to monitor the comfort/discomfort of the passenger, as well, to improve its well-being and to acknowledges the degree of safety and satisfaction perceived about the ADS. Artificial Intelligence (AI) is a fundamental technology for deploying the future CPSoS for ADSs [2]. The stringent computational and memory requirements for Machine Learning (ML) algorithms will impose a significant rethinking of the underlying computing and communication system and will have to fit the constraints of the onboard units. Information extraction should follow as much as possible optimal criteria, cooperating with the inherently distributed nature of the automotive scenario.

Moreover, local processing of information can also be an advantage in specific scenarios with intermittent connectivity or when data privacy is a key issue [3]. Hence, reducing the transfer time needed of either raw data or the relative features is of the utmost importance in determining the performance of computation offloading. Intuitively, traditional data compression techniques [4] could reduce such a delay component, but will also degrade the relative classification performance [5], prolonging the training phases as well as degrading the inference performance.

Conversely, when information extraction algorithms produce massive streams of features, selecting the most relevant ones to feed a ML model becomes very convenient, both in terms of compression and accuracy preservation. Such an operation is known as Feature Selection (FS) [6] and allows for achieving simpler and, therefore, more efficient ML-based models [7].

This work focuses on feature selection efficiency within a fleet of Autonomous Vehicles (AVs), which collect, through their sensors, multi-modal raw measurements. Collected data need to be pre-processed and delivered to feed a remote edge server for inference tasks. Such a procedure can introduce information redundancy, which leads to a waste of computing and communication resources. The AV ensemble aims at limiting the transmission to the top relevant features only. However, just a subset of the top-features can be extracted from each local data collection w.r.t. the whole top-set extracted from the union of all the local datasets but in a centralized manner. In fact, the former case may lead to an inconsistent model *w.r.t.* to the

Manuscript received 8 September 2021; revised 15 February 2022; accepted 2 May 2022. Date of publication 27 May 2022; date of current version 19 September 2022. This work was supported in part by MIUR PON Project OK-INSAID under Grant GA #ARS01_00917, and in part by H2020 Projects TEACHING under Grant GA #871385, in part by HumanAI-Net under Grant GA #952026, in part by MARVEL under Grant GA #957337, and in part by SoBigData++ under Grant GA #871042. The review of this article was coordinated by Dr. Tomaso De Cola. *(Corresponding author: Pietro Cassarà.)*

Pietro Cassará and Alberto Gotta are with the National Research Council, Institute of Information Science and Technologies, 56124 Pisa, Italy (e-mail: pietro.cassara@isti.cnr.it; alberto.gotta@isti.cnr.it).

Lorenzo Valerio is with the National Research Council, Institute of Informatics and Telematics, 56124 Pisa, Italy (e-mail: lorenzo.valerio@iit.cnr.it).

Digital Object Identifier 10.1109/TVT.2022.3178612

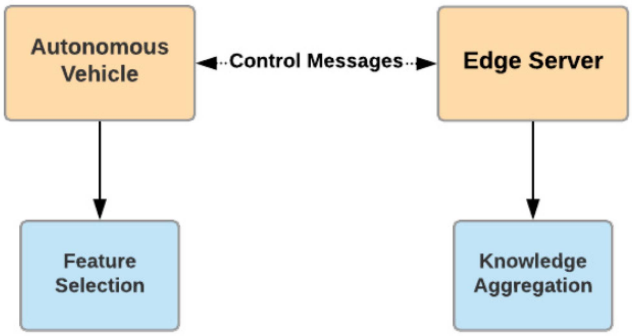

Fig. 1. Feature selection and aggregation components of the proposed FFS system.

latter. Therefore, the AVs, shall participate to a collaborative FS process, in order to exploit the whole information in a federated manner. We tackle this problem, by proposing, for the first time, a Federated-Feature Selection (FFS) algorithm, exploiting a distributed computing paradigm applied to AVs. In FFS all AVs collaborate to come up with the minimal set of features selected from their local datasets.

The proposed FFS system is made up of two components provided in Fig. 1:

- a local FS process runs on each AV and aims at generating a local distribution probability that ranks the information associated to a given feature, according to the Mutual Information (MI) metric [8], [9], which is solved by using the Cross-Entropy (CE) [10].
- An aggregation algorithm executed on the Edge Server (ES) that combines the local estimates received from the AVs. The aggregation algorithm is based on a Bayesian approach to merge the local information into a global one.

The messages delivered by AVs contain probability vectors where each element is the probability to select that feature. The ES returns the "federated" probability vector which is derived by the aggregation of the vectors received by the AVs, as detailed in the following, to replace each of the local ones. Note that, the proposed approach does not need to share any local raw data but only the estimates of the local most informative features. Moreover, it guarantees that all the AVs reach a consensus on the subset of the most informative features, after a finite number of communication rounds, i.e., the messages exchanged between the AVs and the ES.

As we show in the paper, the proposed algorithm *(i)* significantly limits the control messages exchanged during the FFS process and *(ii)* provably let the AVs converge to a subset of top features, which effectively reduce the information stored and transmitted by the AVs. Specifically, numerical results show that, on reference benchmarks, our solution limits data processing and transmission, by removing up the to 99% of redundant features from the selected datasets, without loss of accuracy on the learning model.

Summarising, the novel contributions of this paper are:

- A novel FFS algorithm based on the MI CE (client-side) on the AV and a Bayesian aggregation approach on the ES.
- The theoretical proof that such an algorithm converges to a stable solution in a fixed number of iterations.
- An extensive numerical evaluation tested on two real-world datasets that shows the efficiency of our solution.

The paper is organized as follows: related works are presented in Section II; the reference scenario and the system assumptions are presented in Section III; the theoretical background underlying the proposed feature selection approach is presented in Section IV; the federated version of the feature selection algorithm in presented in Section V; Section VI presents the experimental results of a study case with two real world datasets, belonging to different application domains; conclusions in Section VII.

II. Related Works

A. Feature Selection

Many FS procedures have been proposed in the literature. In [6], [11], [12] authors provide a comprehensive overview of the existing methods. Additionally, they consider the most important application domains and review comparative studies on feature selection therein, in order to investigate, which methods outperform for specific tasks. Authors highlight that FS is based on the identification of the relevance and redundancy provided by the features with respect to a class attribute function. The main approaches of FS fall into three categories: filtering, wrapping, and embedded methods. This categorisation is based on the interaction between the selected features and the learning model adopted to take a decision. The output of the wrapping and embedded methods is tightly connected to the learning model that uses the selection. Therefore, with these methods FS and model training cannot be uncoupled. Conversely, filtering methods are suitable for being used regardless the presence of a learning model to train.

As shown in [6], [11], [12], most of the well-known filtering algorithms use information-based metrics for FS, and can deal with samples of variable lengths, as presented in [13], [14]. A suitable information-based metric for the FS is the MI. MI has gained increasing popularity in data mining, for its ease to use, effectiveness, and strong theoretical foundation. mRMR [15] and HJMI [16] are some of the most used methods that exploit MI. These approaches rank the features according to the maximization of the MI and let the user to select a desired subset k. Differently, the proposed algorithm automatically select a minimal subset of relevant features, also capturing the mutual dependencies. Note that the formulation of the underlying optimization problem is NP-Hard [8], [9], i.e., MI-based feature selection problem involves the integer programming or, in some cases, the quadratic integer programming. In [17]–[19] authors show how to adopt the CE approach to address such native computational complex problems, for different application scenarios. Beyond MI, other filtering methods can use different metrics, such as in [20] where the authors evaluate the variance of all the features to measure the impact that each of them has on the learning process. This method relies on the concept that the features with zero variance add no information, by considering the relation between the target variable and feature vectors.

To the best of our knowledge, all these algorithms are designed for being executed in a centralised setting, i.e., under the assumption that the whole dataset is available to the learning agent.

B. Distributed Learning

Distributed learning is considered from several perspectives in the literature. A very consistent body of work deals with distributed learning based on the Federated Learning (FL) framework. FL is a distributed learning framework initially proposed by Google, where a large number of mobile or edge devices participate in a collective and distributed training of a shared model. [21], [22]. FL is an iterative procedure spanning over several communication rounds until the convergence is reached. Based on this paradigm, several modifications have been proposed concerning (i) new distributed optimisation algorithms [23]–[26], and (ii) privacy-preserving methods for FL [27], [28]. Alternatively, other approaches do not rely on a centralised coordinating server. In [29], [30], authors propose a distributed and decentralised learning approach based on Hypothesis Transfer Learning. Similarly to the FL framework, authors assume that several devices hold a portion of a dataset to be analysed by some distributed machine learning algorithms. The aim of [29], [30] is to provide a learning procedure able to train, in a decentralised way, an accurate model while limiting the network traffic generated by the learning process. The vast majority of the distributed learning solutions, presented in the literature, focus on the model's training, giving the feature engineering phase for granted. Until now, the idea of performing FS, directly, on edge devices remains unexplored.

In the literature only few approaches cope with FS in distributed settings. In [31], authors present a distributed algorithm for FS based on the Intermediate Representation, which aims at preserving the privacy of data, allowing the node to exchange each other the data they hold. Therefore, in this method FS is performed under the assumption that all data are available to the FS algorithm. Moreover, the method presented by the author depends from the specific learning model that uses the selected features.

In [32], the authors propose an information-theoretic FFS approach called Fed-FiS. Fed-FiS estimates feature-feature mutual information and feature-class mutual information to generate a local feature subset in each user device. Then a central server ranks each feature and generates a global dominant feature subset using a classification approach. This approach has some commonalities with ours, such as the adopted metric (MI) and the federated settings. However, differently from [32] *(i)* we provide directly the minimum set of relevant features instead of a ranking, *(ii)* we propose an aggregation based on Bayes' theorem that does not rely on any Machine Learning scheme to finalise the selection (i.e., no regression or classification methods are adopted in our solution), resulting in a computationally more suitable approach for vehicular scenarios.

In light of this and to the best of our knowledge, this is the first paper that proposes a federated mechanism of feature selection explicitly designed to meet the requirements of the CPSoS context.

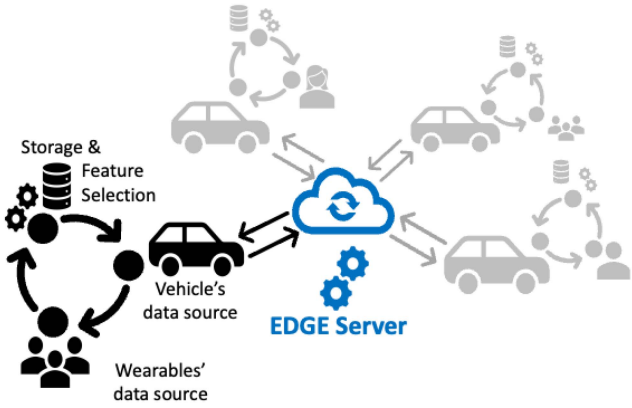

Fig. 2. System architecture. Data sources characterize two different Cyber Physical Systems (CPSs): the former that monitors the user through wearable sensors, the latter relative to the ADS.

III. System Assumptions

In this section, we describe the reference scenario and the system assumptions considered in this paper. As shown in Fig. 2, we consider a set of AVs, implementing an ADS each, collecting data generated by the sensors integrated in a CPSoS and that collaborates with the others ADSs to learn a minimal, and most informative set of features from their local datasets. To this end, the AVs execute an in-network data filtering process through our FFS approach to reach a consensus in identifying the most informative feature subset. Finally, the globally shared feature set is used like a compression scheme before transmitting it to an ES. Note that, in this system the AVs are only responsible for finding the best compression scheme applicable to the their local data in a collaborative way, based only on the control information they exchange with the ES. Moreover, the ES has a three-fold role: i) it acts as central coordinating entity in the FFS process whose purpose is to aggregate the partial control information sent by the AVs; ii) it acts as final collector for the compressed data, once the FFS is completed and, iii) runs the AI services to extract knowledge from data but that is used only for performance evaluation in this paper. We target two different user cases to validate the performance of the proposed FFS method. The former refers to the localization of an AV in the environment based on images and inertial measurements, and the latter regards the physiological-state monitoring of a passenger in the automotive domain. We define two different sub-systems part of the same CPSoS: the ADS of above, and an Human State Monitoring System (HSMS) to learn the feeling perceived from a passenger relatively to the ADS driving style. Therefore, we assume each AV to be equipped with a camera to capture images from the surrounding environment aside some inertial sensors for the former learning task, and a set of body sensors, such as, Electrocardiography (ECG), Electrodermal Activity (EDA), Electromyography (EMG), and Respiration (RSP) for the latter.

Each AV is able to locally synchronize the multi-sensory data such that, for each image, it is possible to associate the corresponding inertial measurements leading to an *enhanced Raw Input Datum* (eRID). Note that for the scope of this paper it is not important the specific semantic of the labelling, but it is enough

to assume a labelling process on the collected data. The AVs are also equipped with a relatively small edge computing unit (e.g., a RaspeberryPi or, at most, an Nvidia Jetson Nano) able to cache data and execute the FS task, before transmitting the features. Additionally, the AVs are endowed with a radio communication interface to communicate toward the ES. It must be noted that the task is not collecting images of the environment, or physiological parameters of the user but, conversely, retrieving the information associated to those images or to those physiological sensors, e,g., the position of the AV with respect to the surrounding or the user mood. In particular, the latter is labelled according to the classification scale provided by questionnaires like PANAS, SSSQ or SAM [33], which associates numerical labels to the physiological states.

IV. Feature Selection

In this section, we provide the theoretical background of the MI-based FS algorithm and the relative implementation based on the CE method.

A. Background Feature Selection Based on Mutual Information

To make the paper self-contained, we report in this Section the necessary theoretical background needed to get an intuition about the internal details of the CE-based FS method presented in Section IV-B.

First, let us define the FS problem as follows:

Definition (FS Problem): Given the input data matrix $\mathbf{X}$ composed by n samples of m features ($\mathbf{X} \in \mathbb{R}^{n \times m}$), and the target attributes' (or labels) vector $\mathbf{y} \in \mathbb{R}^n$, the FS problem is to find a k-dimensional subset $\mathbf{U} \subseteq \mathbf{X}$ with $k \leq m$, by which we can characterize $\mathbf{y}$.

The method we adopt in the paper performs the FS measuring, through the Mutual Information metric, the amount of information that a subset of features (or attributes) $\mathbf{U}$ expresses with respect to a specific target label $\mathbf{y}$.

Formally, the MI between random variables can be defined as [34], [35]:

$$\mathbf{I}(\mathbf{U};\mathbf{y}) = \mathbf{H}(\mathbf{y}) - \mathbf{H}(\mathbf{y}|\mathbf{U}), \tag{1}$$

where $\mathbf{U} = \{\mathbf{x}_1 \cdots \mathbf{x}_k \mid k \leq m\} \subseteq \mathbf{X}$, and $\mathbf{H}(\mathbf{y}|\mathbf{U})$ is the conditional entropy which measures the amount of information needed to describe $\mathbf{y}$, conditioned by the information carried by $\mathbf{U}$. Hence, $\mathbf{I}(\mathbf{U};\mathbf{y})$ represents the dependence between $\mathbf{U}$ and $\mathbf{y}$, i.e., the greater the value of $\mathbf{I}$, the greater the information carried by $\mathbf{U}$ on $\mathbf{y}$. We recall that the MI between two random variables $\mathbf{A}$ and $\mathbf{B}$ is strictly related to the entropy $\mathbf{H}(\cdot)$, which defines the amount of information held by the variables, i.e., the entropy of a random variable $\mathbf{A}$ (i.e., $\mathbf{H}(\mathbf{A})$) and its probability are inversely proportional: the greater the entropy of a random variable $\mathbf{A}$, the greater its unpredictability and vice-versa. Hence, we can assert that the entropy measures the diversity of $\mathbf{A}$ in terms of the uncertainty of its outcomes.

In MI-based FS the features to be selected are those that maximise (1). These features are typically referred as Essential Attributes (EAs). By solving the following optimization problem

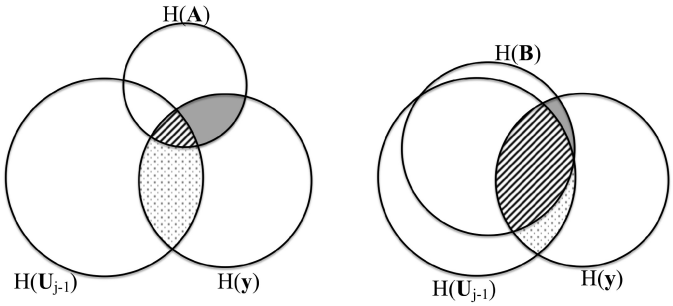

Fig. 3. Example of the relationship between Mutual Information and Entropy.

we would obtain the optimal global solution to the FS problem defined in IV-A:

$$\arg\max_{\mathbf{U}} \mathbf{I}(\mathbf{U};\mathbf{y})$$
$$\mathbf{U} = \{\mathbf{x}_1 \cdots \mathbf{x}_k \mid k \leq m\} \subseteq \mathbf{X} \tag{2}$$

Note that the problem (2) belongs to the class of Integer Programming (IP) optimization problems and finding its optimal solution is NP-hard [36], i.e., the optimal solution $\mathbf{U}$ would be found among all combinations of feature indices of the native set $\mathbf{X}$.

The problem (2) becomes computationally tractable if approached through an iterative algorithm which selects and adds to the subset $\mathbf{U}$ one feature at a time. Therefore, instead of solving 2, we address the problem defined in (3):

$$\arg\max_{\mathbf{x}_j \in \mathbf{X} \setminus \mathbf{U}} \mathbf{I}(\mathbf{x}_j;\mathbf{y}|\mathbf{U}),$$
$$\mathbf{U} = \{\mathbf{x}_1 \cdots \mathbf{x}_{k-1} \mid k \leq m\} \subseteq \mathbf{X}. \tag{3}$$

For the sake of clarity, we provide an intuitive example based on the relation between MI and the entropy. Considering Fig. 3, the circles are the entropy of the random variables $\mathbf{A}, \mathbf{B}, \mathbf{U}, \mathbf{y}$, and the grey regions are the information carried by the variable $\mathbf{A}$ (or $\mathbf{B}$) on $\mathbf{y}$. The dashed area shows the information redundancy of the variable $\mathbf{A}$ (or $\mathbf{B}$) given the already selected variables in $\mathbf{U}_{j-1}$. In this example, the variable $\mathbf{A}$ should be added to the set $\mathbf{U}$ since it is more informative than $\mathbf{B}$ on $\mathbf{y}$, i.e., its grey area is larger than $\mathbf{B}$'s, and it is less redundant than $\mathbf{B}$ w.r.t. to $\mathbf{U}_{j-1}$.

The main drawback of this approach is that it might end up with a sub-optimal solution because, by selecting the features one by one, the algorithm makes the implicit assumption that they are independent, which might not hold true. Theoretical foundations for the incremental version of the FS algorithms has been proven by the authors in [34], [35]. It is worth mentioning that a connected issue with problem (3) regards the efficient evaluation of the MI, which might become prohibitive even for datasets with a small number of samples. We overcome this problem by adopting the MIToolbox [37], a state-of-the-art tool for numerical optimization.

B. CE-Based Feature Selection Algorithm

In this section, we describe the CE-based algorithm that finds, in a finite number of steps, a solution that well approximates the one found by solving problem (2), while making negligible

the assumption of independence among features introduced in problem (3). In other words, with CE-based FS, instead of selecting one EAs at a time, we select a set of EAss *jointly*.

The CE-based algorithm is based on the following intuition: if the set $\mathbf{U}$ contains only EAss, then $\mathbf{I}(\mathbf{U};\mathbf{y}) \rightarrow \mathbf{H}(\mathbf{y})$, which implies that $\mathbf{H}(\mathbf{y}|\mathbf{U}) \rightarrow 0$ [34], [35]. Note that with our approach, we avoid the greedy research of the set $\mathbf{U}$ among all the possible $\binom{m}{k}$ solutions which realizes $\mathbf{H}(\mathbf{y}|\mathbf{U}) \rightarrow 0$. Instead, we adopt the stochastic approach. Precisely, we associate each i-th feature with a random variable $z_i \sim \text{Bernoulli}(p_i)$. The CE-based algorithm identifies which variables $z_i,\ i = 1, \ldots, m$ must have $p_i \rightarrow 1$, so that the objective function $\mathcal{O}(\mathbf{U}(\mathbf{z})) = \mathbf{H}(\mathbf{y}|\mathbf{U})$ gets close to 0. This is called Associated Stochastic Problem (ASP) [10]. In this way, we get the optimal distribution of the binary vector $\mathbf{z}$ through which we identify the features to be selected, i.e. the i-th feature is selected if $p_i \rightarrow 1$. It is worth noting that searching for the solution of the optimization problem through the definition of the ASP has the advantage of addressing the native problem in (2) as a convex problem.[1]

We formulate the ASP as a minimization problem, as shown in (5). In the following we present the essential steps that brings to its formulation. Briefly, we need to find the probability distribution $g(\mathbf{z}, \mathbf{p})$ of the values in $\mathbf{z}$ equal to 1 that solves the equation:

$$\Pr(\mathcal{O}(\mathbf{U}(\mathbf{z})) \leq \gamma) = \sum_{\{\mathbf{z}\}} \mathcal{I}(\mathcal{O}(\mathbf{U}(\mathbf{z})) \leq \gamma)\, g(\mathbf{z}, \mathbf{p}) \quad (4)$$

where $\mathcal{I}(\cdot)$ is the indicator function of the event $\mathcal{O}(\mathbf{U}(\mathbf{z})) \leq \gamma$, and γ is the minimum value for our objective function. Precisely, γ at step t is calculated as the percentile $1-\beta$ of the objective function calculated by using the samples drawn from the distribution $g(\mathbf{z}, \mathbf{p})$ at step t. Note that, the authors in [10] recommend to set β in the range $0.9 - 0.95$. The indicator function is equal to 1 for all the possible configurations in $\mathbf{z}$ that verify the event $\mathcal{O}(\mathbf{U}(\mathbf{z})) \leq \gamma$, and 0 otherwise.

We estimate $g(\mathbf{z}, \mathbf{p})$ through the Likelihood Ratio (LR) estimator with reference parameter $\mathbf{p}$. Precisely, we apply the LR theory of estimation [10] to define the following optimization problem and to obtain the optimal value $\mathbf{p}^*$ for the distribution.

$$\mathbf{p}^* = arg \min_{\mathbf{p}} \frac{1}{S} \sum_{j=1}^{S} \mathcal{I}(\mathcal{O}(\mathbf{U}(\mathbf{z}_j)) \leq \gamma) \ln(g(\mathbf{z}_j, \mathbf{p})) \quad (5)$$

where $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_S\}$ is a set of possible samples drawn from the distribution $g(\mathbf{z}, \mathbf{p})$.

As stated above $\mathbf{z}_j = [z_{1j} \cdots z_{mj}]$ is a vector of independent Bernoulli random variables where z_{ij} takes value equal to 1 with probability p_i and 0 with probability $1 - p_i$. Hence, $g(\mathbf{z}_j, \mathbf{p})$ can be written as:

$$g(\mathbf{z}_j, \mathbf{p}) = \prod_{i=1}^{m} p_i^{z_{ij}} (1-p_i)^{(1-z_{ij})}\ ;\ z_{ij} \in \{0, 1\} \quad (6)$$

[1]More details are in Section 4 of [10].

Given that the objective function of problem (5) is concave,[2] we can solve it in closed form by imposing:

$$\frac{\partial}{\partial p_i} \frac{1}{S} \sum_{j=1}^{S} \mathcal{I}(\mathcal{O}(\mathbf{U}(\mathbf{z}_j)) \leq \gamma) \ln(g(\mathbf{z}_j, \mathbf{p})) = 0,$$

leading to:

$$p_i = \frac{\sum_{j=1}^{S} \mathcal{I}(\mathcal{O}(\mathbf{U}(\mathbf{z}_j)) \leq \gamma) z_{ij}}{\sum_{j=1}^{S} \mathcal{I}(\mathcal{O}(\mathbf{U}(\mathbf{z}_j)) \leq \gamma)} \quad i = 1 \cdots m; \quad (7)$$

In the CE-base algorithm the result in the (7) is used for updating the distribution $\mathbf{p}$ as follows:

$$p_i = (1-\alpha) p_i + \alpha \frac{\sum_{j=1}^{S} \mathcal{I}(\mathcal{O}(\mathbf{U}(\mathbf{z}_j)) \leq \gamma) z_{ij}}{\sum_{j=1}^{S} \mathcal{I}(\mathcal{O}(\mathbf{U}(\mathbf{z}_j)) \leq \gamma)}. \quad (8)$$

The mathematical analysis about the choice of the parameter α is provided in the Appendix VII-A of this work. Further indications on the choice of α can be found in [10], [38], [39]. The derivation of equations (5-7), as well as, the optimality of $g(\mathbf{z}_j, \mathbf{p})$ are proven in [10].

The solution of the problem defined in (5) is achieved through Algorithm 1: it starts with an initial guess of $\mathbf{p}_G$; S Bernoulli random samples of size m each (line 4) are drawn at each step t. For each sample $\mathbf{z}_s$, the values of the conditional entropy (line 7) are computed on the dataset where the only active features are those corresponding to the elements equal to one (line 6) in $\mathbf{z}_s$. The subset selection is shown in the procedure GETSUBSET$(\mathbf{X}, \mathbf{z})$ (lines 15-26). Then we compute $\mathbf{p}(\mathbf{Z}_t)$ (lines 9-10) as in (7) and finally we update the current estimate of the probability vector $\mathbf{p}$ (line 11) as in (8).

V. Federated Feature Selection

In this section we present how we exploit the CE-based FS algorithm presented in Section IV and summarised in Algorithm 1 to design our FFS algorithm FFS, described in Algorithms 2 and 3. They cover, respectively, the two functional blocks of FFS, i.e., Algorithm 2 is executed by the ES to coordinate the distributed FS and Algorithm 3 runs on the clients. The FFS is an iterative procedure. At the beginning, the ES sends to the clients involved in the process a vector $\mathbf{p}_G \in \mathbb{R}^m$ where each element represents the probability that each feature has to be selected according to its importance (lines 8-10 of Alg.2). Each element of $\mathbf{p}_G$ is initialized to 0.5, i.e., this is a common choice when using the CE algorithm. The vector $\mathbf{p}_G$ represents a piece of global information that the ES shares with the client nodes. Each client l uses $\mathbf{p}_G$ to initialize its local copy of the probability vector, i.e., $\mathbf{p}_l \leftarrow \mathbf{p}_G$ and runs the local FS procedure based on its local data (lines 2-3 of Algorithm 3). At the end of the local FS, the l-th client sends to the ES the locally updated probability vector $\mathbf{p}_{l_{new}}$ and a control information regarding the cardinality of its local data n_l whose purpose will become clear in the following. The ES computes the new global probability vector (line 13 of Algorithm 2) by aggregating the ones received

[2]The logarithm is a concave function, the indicator function is 0 or 1 so the weighted sum of concave functions gives still a concave function.

Algorithm 1: CE-based Algorithm for FS.

procedure CE$\mathbf{X}, \mathbf{y}, \mathbf{p}$, T,S
  for all $t = 1, \ldots, T$ **do**
    $\mathbf{Z}_t \leftarrow$ GENRNDSAMPLE $(S, \mathbf{p})$ $\triangleright\ \mathbf{Z} \in \{0,1\}^{S \times m}$
    $\mathbf{u} \leftarrow \{\}$
    for all $\mathbf{z}_s \in \mathbf{Z}_t$ **do** $\triangleright \mathbf{z}_s \in \{0,1\}^{1 \times m}$
      $\mathbf{U} \leftarrow$ GETSUBSET$(\mathbf{X}, \mathbf{z}_s)$
      $\mathbf{u} \leftarrow \mathbf{u} \cup \mathbf{H}(\mathbf{y}|\mathbf{U})$
    end for
    $\gamma \leftarrow$ COMPUTEPERCENTILE$(\mathbf{u}, 1-\beta)$
    $\mathbf{p}(\mathbf{Z}_t) \leftarrow$ COMPUTENEWPROB$(\mathbf{u}, \gamma, \alpha)$ $\triangleright(7)$
    $\mathbf{p} \leftarrow (1-\alpha)\mathbf{p} + \alpha\mathbf{p}(\mathbf{Z}_t)$ $\triangleright(8)$
  end for
  return $\mathbf{p}$
end procedure
procedure getSubset$\mathbf{X}, \mathbf{z}$
  $\mathbf{U} \leftarrow \{\}$
  for all $\mathbf{x} \in \mathbf{X}$ **do**
    $\mathbf{u} \leftarrow \{\}$
    for all $j = 1, \ldots, m$ **do**
      if $z_j == 1$ **then**
        $\mathbf{u} \leftarrow \mathbf{u} \cup x_j$
      end if
    end for
    $\mathbf{U} \leftarrow \mathbf{U} \cup \mathbf{u}$
  end for
end procedure

from the clients as expressed in (9) and discussed later on. The updated vector $\mathbf{p}_G$ is transmitted to the nodes that run Algorithm 3 by updating the local probability vector with the new global one. This procedure iterates until the distribution global probability vector converges to a stable one. In FFS we check convergence by comparing the distribution of the current global probability vector $\mathbf{p}_G$ to the previous one $\mathbf{p}_{G_{old}}$ using the Kolmogov-Smirnov statistical test for two one-dimensional samples (KS-test). The procedure stops when *(i)* the p-value of the KS-test is greater than a fixed threshold[3] $\tau_1 = 0.995$ and, *(ii)* its variation from the previous one is less than $\tau_2 = 10^{-6}$ (line 7 of Algorithm 2).

The core point of Algorithm 2 regards the aggregation step (line 13 of Algorithm 2) where the ES merges the local probability vectors into the global one which, in our solution, is defined as a weighted average. The main idea is to merge the local probability vectors by a weighted average where the weights (computed as in (10)) serve the twofold purpose of *(i)* considering more (or less) those vectors that are computed from larger local datasets and *(ii)* defining a common support among all the probability vectors. This second aspect is quite crucial for the consistency of the computation in (9).

Formally, we assume that each node acquires a number of *i.i.d.* records n_l to perform the FS, and that the nodes share the same set of features $\mathbf{X}$. The global probability $\mathbf{p}_G$ used for the FS can be written as follows:

$$\mathbf{p}_G = \sum_l \mathbf{p}_l \omega_l, \tag{9}$$

where $\mathbf{p}_l$ is the solution of problem (5) at node l obtained by using Algorithm 1, and ω_l weights $\mathbf{p}_l$ *w.r.t.* the other nodes, whose formal definition is:

$$\omega_l = \frac{n_l}{\sum_l n_l}. \tag{10}$$

As anticipated, according to (10), we weight the probability vector $\mathbf{p}_l$ of node l proportionally to the size of its local dataset compared to the whole amount of data present in the system. In this way, we can contrast situations where local datasets are heterogeneous w.r.t. the size.

In FFS, the updating scheme can be, at least in principle, both synchronous and asynchronous, provided that the set of nodes involved in the process does not change over time.[4] Precisely, we assume a system where the ES after having sent the updated global probability vector, expects the nodes to receive their local updates within a fixed time slot, after which, it begins the aggregation step using only the information received. Therefore, the number of updates used to compute the new global probability vector might change because a subset of nodes could not communicate their updates within the deadline set by the ES. Regardless of the number of nodes that contributed to the aggregation step during one round of communication, the ES broadcasts the new global probability vector $\mathbf{p}_G$ to *all* nodes in the system. In this way, all nodes start the new round of local computation from the same starting point, and, consequently, we dramatically limit the potentially detrimental effects deriving from the aggregation of outdated local probability vectors. Moreover, as proved by the convergence analysis provided in AppendixVII-A and AppendixVII-B, independently from the updating scheme, FFS converges in a finite number of steps to the very same solution as running the CE in centralised settings i.e., with complete access to the entire dataset.

It's worth noting that our solution is able to cope with feature redundancy in federated settings. Precisely, this represents an issue that might prevent the possibility of performing the FS in federated settings. In fact, running a standalone FS algorithm on different local datasets where there is redundancy between features, different FSs might occur but with an equivalent information content across all the AVs. This aspect makes all the local selections completely useless regarding the communication efficiency, due to the consequent lack of agreement on the FS between the AVs. Conversely, since in FFS the AVs share at each communication round their local information, they may come up with a final agreement on the FS. Summarising, even if there is redundancy between features, the final selection is consistent among all the AVs and, according to results presented in Section VI, it is also accurate if compared to the centralized FS (i.e., when all the local raw data are transferred onto the ES).

[3]We empirically observed that the closer τ_1 to one, the more accurate the solution.

[4]Note that this condition does not imply that all nodes must be active during the entire process. In fact, as we will show in Section VI our system is robust to the presence of churning nodes.

Algorithm 2: Server Side FFS Algorithm.

procedure Server-Node
  $v \leftarrow 0$ ▷p-value of Kolmogorov-Smirnov test
  $\tau_1 \leftarrow .995$
  $\tau_2 \leftarrow 10^{-6}$ ▷Thresholds to check convergence
  $\mathbf{p}_G \leftarrow \{1/2 \mid \forall\, p_i\ i = 1, \ldots, m\}$
  do
    for all $l \in L$ **do**
      SENDTOCLIENT$(l, \mathbf{p}_G)$
    end for
    RECEIVEFROMCLIENTS$(\mathbf{p}_{l_{new}}, n_l)$
    $\mathbf{p}_{G_{old}} \leftarrow \mathbf{p}_G$
    $\mathbf{p}_G \leftarrow$ UPDATEGLOBALPROBABILITY() ▷(9)
    $v_{old} \leftarrow v$
    $v \leftarrow$ KOLMOGOROVSMIRNOVTEST $(\mathbf{p}_G, \mathbf{p}_{G_{old}})$
  while $v \geq \tau_1 \wedge |v - v_{old}| \leq \tau_2$ ▷repeat until convergence is met
end procedure

Algorithm 3: Client side Federated Feature Selection algorithm.

procedure Client-Node
  $\mathbf{p}_l \leftarrow$ RECEIVEFROMSERVER$(\mathbf{p}_G)$
  $\mathbf{p}_{l_{new}} \leftarrow$ CE$(\mathbf{X}_l, \mathbf{y}_l, \mathbf{p}_l$,T,S) ▷Algorithm 1
  SENDTOSERVER$(\mathbf{p}_{l_{new}}, n_l)$
end procedure

VI. NUMERICAL EVALUATION

In this section, we present the numerical results of our compression method based on the FFS algorithm presented in Section V. Before going through the results, we introduce the datasets, the simulation settings, the methodology, and the metrics used to evaluate our solution's performance.

A. Dataset Description and Simulation Settings

We based the performance evaluation of FFS on two datasets, each one mapping one of the two use cases described in Section III. The first one called MAV[5] is a publicly available dataset containing both 64×64 images and 6 Inertial Measurement Units (IMUs) collected by a AV during a mission in a controlled environment. The second dataset called WEarable Stress and Affect Detection (WESAD) is a collection of data sampled from heterogeneous biophysical sensors: ECG, EDA, EMG, Temperature, Respiration and Inertial Measurements on the three axes.

a) MAV dataset: both images and inertial measurements are synchronised to obtain a set of eRIDs. We pre-process the raw images to extract more informative features as it is customary in the computer vision domain. Feature extraction eases the training of a machine learning model and, performs a preliminary step of data compression. In fact, a raw image is made of 4102 floats (64×64 pixels + 6 IMU readings) while, after the feature extraction, it shrinks down to a vector of size 2166 floats. In our settings, we extract the Histogram of Oriented Gradient (HOG) features,[6] and we assume that the feature extraction is accomplished directly on the AV, which might be possible if equipped with a board of the kind discussed in [40]. Note that the original dataset is unlabeled. Therefore we labelled it in a way compatible with the original context of positioning. To this end, we associated with each eRID a label corresponding to the corresponding voxel.[7] Table I shows the structure of an eRID for the MAV; the first 2160 feature are HOG while the last 6 are IMUs, i.e., acceleration (ACC) and angular velocity (AV). The whole dataset contains 2911 labelled records. To simulate the federated data collection, we split it into 10 disjoint partitions of size 291 records such that each partition is i.i.d. w.r.t. the entire dataset. Each subset represents a AV. The data collection is slotted; hence, the AVs draw with replacement a random sample from their local dataset for each time slot. This sample is used to perform the local computation of the distributed algorithm followed by a communication round for synchronising the AVs on the local FS. Each random draw's size is accumulated to trace the cache necessary for storing data until the completion of the distributed FS.

TABLE I
STRUCTURE OF A MAV ERID

0	1	2	...	2158	...
HOG #					
2160	2161	2162	2163	2164	2165
ACC_x	ACC_y	ACC_z	AV_x	AV_y	AV_z

TABLE II
STRUCTURE OF A WESAD ERID

0	1	2	3	4	5	6	7
ACC_x	ACC_y	ACC_z	ECG	EMG	EDA	TEMP.	RSP

b) WESAD dataset it provides data in terms of features and labels already useful to perform the detection of stress and affection state of human subjects. The dataset contains readings from two devices, i.e., Respiban and Empatica E4, positioned i) on the chest and ii) on the wrist of human subjects. Each device is equipped with multiple sensors monitoring several physiological parameters. Since the two devices have different operating settings, we focused on the Respiban, whose collection rate is homogeneous for all its sensors. The dataset contains readings collected from 17 human subjects, which perform a predetermined protocol to induce the body in one of the following states: 0-baseline, 1-amusement, 2-stress, 3-meditation, 4-recovery. The data collected for each subject amounts to ∼3.6 M records, equivalent to ∼220 MB. A complete description of the dataset is provided in [33]. Table II shows the structure of an eRID for the WESAD.Due to the huge size of the dataset we used the data from 5 out of 17 subjects, corresponding to ∼1.1 GB.

[5]dataset. [Online]. Available: https://tinyurl.com/mavmr01

[6]HOG is a standard feature extraction methodology used in computer vision and image processing to create an image descriptor that captures the spatial relations between different portions of it [40].

[7]A voxel represents a value on a regular grid in three-dimensional space.

The data is already partitioned according the subject ID, thus we keep the original partitions. In our simulated scenario, each partition corresponds to an edge device holding the data of only one subject, i.e., no artificial data re-distribution is performed. As for the previous scenario, each device executes FFS using only its own data.

We evaluate the performance of our methodology according to two metrics:

- *accuracy:* to assess the quality of the distributed FS
- *network overhead (N_{OH})*: to evaluate the impact in terms of network traffic generated by our methodology

Our target is to compress the data to be transmitted, without significantly degrading its informative content.

Accuracy metric: The quality assessment is a two-stage procedure. First, we set the baseline validating the quality of the features selected by CE executed in a centralised setting, i.e., we train a classifier using the set of selected features (CE-CFS) on the entire dataset, and we compare its prediction performance with that of a second classifier trained on the whole set of features (NO-FS). If the CE-CFS performance on a smaller group of features is comparable or equivalent with the one identified by NO-FS, we consider the FS valid. To strengthen this initial evaluation, we compare the centralised results of CE-CFS with other three reference FS algorithms: mRMR [15], HJMI [16] and ANOVA [20]. As we will show in the following, for all these benchmarks we have to specify the size k of the features selection. Since we are interested in assessing the quality of the FS and for the sake of fairness, we set k equal to the size of the FS obtained by CE-CFS (which finds such a number in a completely autonomous way).

Then, we repeat the same procedure training another classifier on the subset of features obtained from our FFS and we compare its performance with all the centralised methods. We split the dataset in train (80%) and test set (20%). The train set is used for both FS and model training, while the test is used for performance evaluation only. The accuracy is defined as the average of correctly classified records

$$A = \frac{1}{N}\sum_{i=1}^{N} I(\hat{y}_i = y_i), \tag{11}$$

where N is the size of the test set, I is the indicator function, $\hat{y}_i$ and y_i are the i-th predicted and true label, respectively. For the sake of statistical significance, the training is repeated ten times, changing the initialisation of the classifier and the composition of training and test set. The reported results are average values accompanied by confidence intervals at 95%.

Network Overhead: we measure the network traffic generated by our solution as follows. On the one hand, we compute the network overhead generated by the FFS network defined as:

$$N_{OH} = R * L * 2 * (z + 1 + b) \tag{12}$$

where R is the number of communication rounds before all the L AVs involved in the distributed FS converge to a solution, $z + 1$ is the number of nonzero floating point numbers belonging to the probability vector $\mathbf{p}_l$ in (9) exchanged between the AVs during each round plus the weight ω_l in (10). The symbol b is the size of the bit map used to reconstruct the position of the non-zero elements exchanged between the AVs and the edge server. On the other hand, we compute the compression obtained through the FS as:

$$C = |F|/|D| \tag{13}$$

where $F \subseteq D$ is the selected set, and D is the entire set of features.

TABLE III
COMPARISON BETWEEN NO-FS AND CE-CFS ON MAV AND WESAD DATASET

Dataset	Method	Size (# record)	FS (#)	C (%)	Accuracy (%)
MAV	NO-FS	2911	2166 (All)	-	97.5±0.4
	CE	2911	18	99	96.7±0.5
	MRMR	2911	k=18	99	95.0±0.5
	ANOVA	2911	k=18	99	95.0±0.4
	HJMI	2911	k=18	99	96.3±0.7
WESAD	NO-FS	$15{*}10^6$	8 (All)	-	94.3±0.7
	CE	$15{*}10^6$	4	50	94.6±0.8
	MRMR	$15{*}10^6$	k=4	50	94.3±0.8
	ANOVA	$15{*}10^6$	k=4	50	94.5±0.5
	HJMI	$15{*}10^6$	k=4	50	90.2±1.6

B. Settings the Baseline: FS in Centralised Settings

The following results regard the first stage of the validation, i.e., the accuracy of a classier trained using only the subset of features identified by the CE algorithm w.r.t the performance obtained by a classifier trained on the entire dataset. For this stage of validation, we train a Neural Network (NN). For MAV the NN is a multi-layer perceptron with two hidden layers of 300 and 100 neurons each. For WESAD, we used a deep NN with four hidden layers of 300,100,64,32 neurons each. The input layer's size depends on the number of features selected, while the size output layer is 37 and 5 for MAV and WESAD, respectively. The activation function is "ReLU"[8] and the optimizer is "Adam"[9] for both the models. These are very common settings which typically provides good performance [41].

Results in Table III show that CE algorithm executed on both datasets in centralised settings can autonomously identify a minimal set of features (i.e., 18 for MAV and 4 for WESAD) with the very same informative content of the whole feature set. The accuracy obtained by both the NN models trained on the CE's FS is statistically equivalent to the one obtained on the whole set of features, inducing a quite impressive compression rate (C): up to 99% and 50% of network traffic for MAV and WESAD, respectively. As a further confirmation of the CE results, we perform the FS using other three reference benchmarks, i.e., MRMR, ANOVA, HJMI. Note that all these approaches select a subset of features with the very same informative content of

[8]REctified Linear Unit
[9]Stochastic Gradient Descent with ADAptive Momentum

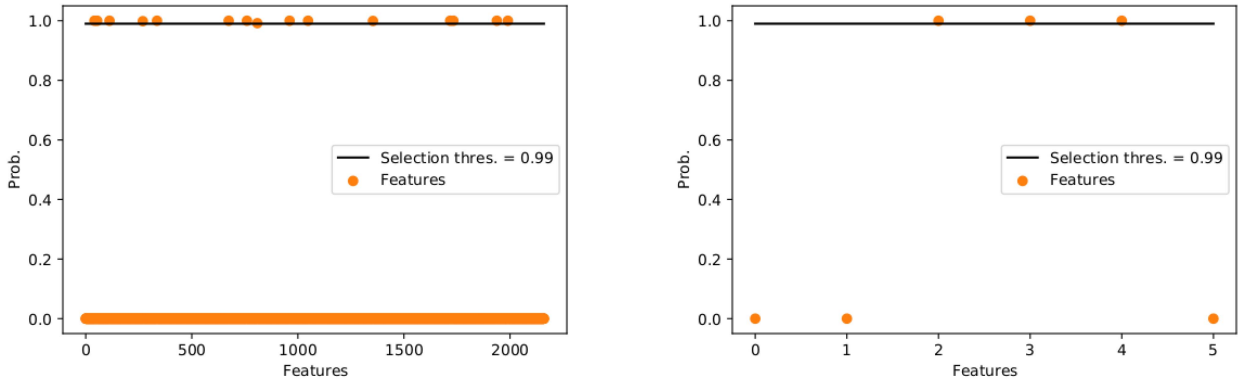

Fig. 4. Centralised FS probability for HOG and IMU. The selected features are those with probability greater than 0.99 (above threshold). (a) C-HOG. (b) C-IMU.

TABLE IV
COMPARISON BETWEEN CE-CFS AND FFS ON MAV AND WESAD

Dataset	Method	Size (# obs.)	FS (#)	C (%)	Accuracy (%±C.I.)
MAV	CE-CFS	2911	18	99	96.7±0.5
	FFS	291	24	99	96.7±0.4
WESAD	CE-CFS	15M	4	50	94.6±0.8
	FFS	3M	4	50	94.6±0.8

CE. However, we point out that for all of them we have to decide beforehand the number of features to be selected. This represent a major shortcoming that, instead, CE-based methods overcome by design, since the number of features to be selected is a byproduct of the CE algorithm. Finally, these results assess the suitability of the CE algorithm on both datasets, thus we can use them as a benchmark for the evaluation of our distributed FFS method.

C. Evaluation of Federated Feature Selection

We focus now on the analysis of our FFS method. We compare its performance to those obtained by CE executed in centralised settings (CE-CFS). We recall that, in federated (distributed) settings, each AV can process only the data it locally collects.

First we assess the performance of FFS in a static distributed scenario where the AVs have collected all the data and, before sending them to the ES, they perform the distributed FS in order to transmit only the very necessary information.

Table IV reveals that for MAV dataset, FFS finds a set of features that, although slightly larger than that found by CE-CFS (24 instead of 18), it has the very same informative content, i.e., the accuracy of the NN model trained on both subsets of features are statistically equivalent. As we can see, the results also hold for the WESAD dataset. Precisely, FFS selects the same number of features identified by CE-CFS. Specifically, FFS and CE-CFS select the same set, i.e., the features with indexes [1,2,5,6], explaining why the NN achieves the same prediction accuracy. We motivate such an exact correspondence between FFS and CE-CFS selection considering that the small size of the complete feature set of WESAD might prevent a high number of feature subset with equivalent informative content. Such an assumption also holds for the MAV dataset. In fact, as we can see in Figs. 4(b), Fig. 5(a) FFS and CE-CFS select the same subset of IMU features. Conversely, when the set of features is more redundant, as it is for the HOGs, there might exist several subsets holding the same informative content. The comparison in Figs. 5(a) and 4(a) confirms such a claim because the two feature sets are overlapping but not equal, yet the overall accuracy is comparable.

This result provides a preliminary insight regarding the effectiveness of the Bayesian aggregation used to merge the information extracted by the AVs from their local datasets. Precisely, Fig. 6 shows the number of selected features at each communication round for the MAV case. As we can see, in the beginning, the cardinality of FS remains almost constant. In this phase, due to the partitioning of data in separated datasets, the CE algorithm has not yet enough knowledge to identify the most informative features. However, the number of features added to the selection starts increasing following an almost-linear trend in a few communication rounds (16). The process ends after 44 rounds, i.e. when the distribution of probabilities indicating the most informative features becomes stable.

Our method's capability to converge quickly to the final and most informative set of features directly affects the amount of network traffic generated upon the completion of the FFS. To confirm such a claim, we performed a set of simulation in which we run FFS varying the size of the local dataset available at the edge device. In this way, we want to analyse our method's robustness when each edge device can access only a limited amount of data. In Table V we report the size of data used for each update (Size), the number of selected features (FS), the accuracy, the number of communication rounds upon convergence (R_c), the compression obtainable with FFS (C), the network overhead generated by FFS (N_{OH}), and the size of the cache needed to collect the data before starting the data transmission. Overall, we observe that, for both datasets, decreasing the size of data processed at each round does not affect significantly the number of communication rounds needed by FFS to converge to a solution, which results in limiting the network overhead generated during the process. Specifically, considering a

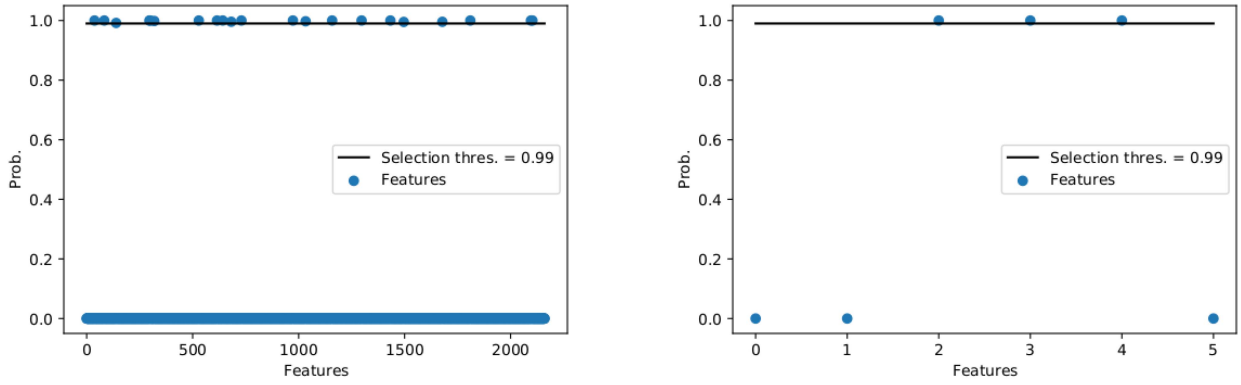

Fig. 5. FFS probability for HOG and IMU. The selected features are those with probability greater than 0.99 (above threshold). (a) F-HOG. (b) F-IMU.

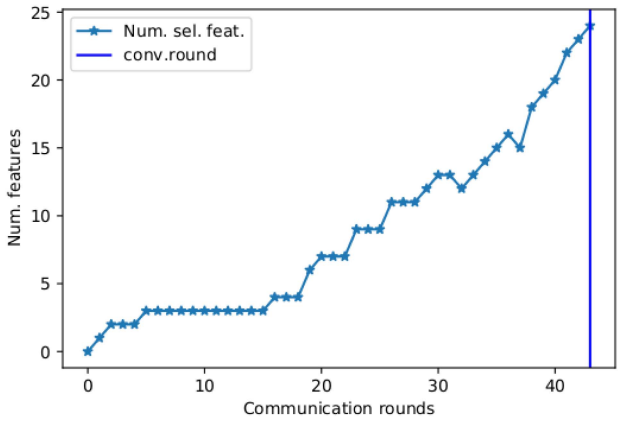

Fig. 6. FFS process on MAV dataset.

TABLE V
PERFORMANCE OF FFS VARYING THE DATA PROCESSED DURING A COMMUNICATION ROUND

Dataset	Size (# obs)	FS (#)	Accuracy (%±C.I.)	R_c (#)	C (%)	N_{OH} (MB)	Cache (MB)
MAV	291	24	96.7±0.4	44	99	16	217
	203	26	96.5±0.3	37	99	13	128
	145	34	97.0±0.3	55	98	20	134
	87	59	97.2±0.3	43	97	15	63
	29	41	97.2±0.4	53	98	19	26
WESAD	$3{\cdot}10^6$	4	93.6±0.8	11	50	0.009	$2{\cdot}10^3$
	$1{\cdot}10^6$	5	93.8±0.5	10	38	0.008	$1{\cdot}10^3$

dynamic data collection process as in the MAV-related use case, we see that the network overhead is always i) less than the storage needed to cache the data before starting the transmission and ii) negligible considering the compression achieved (i.e., up to 99%). Interestingly, the same holds also for the WESAD scenario. In this case, the network overhead can be considered negligible w.r.t. the size of the data processed ($<$ 1MB) if compared with the compression rate achieved by FFS (up to 50%).

In Table VII we show how the benchmark methods MRMR, ANOVA and HJMI behave when run in isolation on local dataset.

TABLE VI
PERFORMANCE OF FFS VARYING THE PERCENTAGE NON-FAULTY AVS PER COMMUNICATION ROUND

Dataset	ρ	FS (#)	Accuracy (%±C.I.)	R_c (#)	C (%)
MAV	0.2	48	97.0±0.4	37	98
	0.3	80	97.2±0.3	35	96
WESAD	0.2	4	94.5±0.4	9	50
	0.3	3	67.4±0.4	7	63

TABLE VII
LOCAL FS FROM COMPETITORS APPROACHES

Dataset	Method	Local FS intersection (%)
MAV (k=18)	MRMR	0
	ANOVA	0
	HJMI	0
WESAD (k=4)	MRMR	100
	ANOVA	100
	HJMI	100

Each method has been configured to select the optimal number of features found in a centralised setting. This is clearly an unrealistic situation that we use to demonstrate the limitations coming from running a non-FFS algorithm in federated settings (i.e., on partial datasets). Precisely, taking into account the MAV dataset, all the algorithms run in isolation on each AV, find a different subset of features (i.e., null pairwise intersection). No agreement between AVs on the subset of features means that all the local data must be transmitted to the ES, causing a non negligible waste of network resources. We motivate this behaviour with the fact that the original subset of features is redundant, as in MAV, running the FS in isolation on portions of data is not a winning strategy. Conversely, when the original subset of features is less noisy, as in WESAD, it is more likely that all the AVs find, completely by chance, the same subset of features, i.e., without a way to coordinate the features selection in a consistent and provable way, there are no guarantees for the AVs to identify a consistent and shared subset of features.

Finally we analyse the FFS performance in presence of *faulty nodes*, i.e., a node experiencing issues in transmitting successfully its updates to the ES. Note that, the causes preventing the updates' transmission might relate to either communication-related (i.e, a noisy channel) or the presence of a power-saving policy regulating the duty cycle of AVs switching off the network interface for a time corresponding to a communication round. The aim is assessing the robustness of FFS when few nodes cannot contribute to the distributed learning at each communication round. To this end, we simulate a scenario where, at each communication round, a random number of AVs fail to communicate their updates to the ES. We model the fault of a AV performing a random draw from a Bernoulli distributed random variable, with parameter ρ. At the beginning of the simulation we set ρ and, for each communication round and for each node, we perform a random draw, where 0 means faulty and 1 means non-faulty. This means that the updates of a faulty AV are not considered for the execution of Algorithm 2. We consider a fault rate ρ equal to 0.2 and 0.3, meaning that at each round there are, on average, 2 and 3 faulty AVs out of 10, respectively. Such values can be reasonably assumed as upper bounds to evaluate the performance of the system. Higher rates would reveal that the scenario is not reasonably set up to run with any sort of reliability.

In Table VI we report the performance of FFS, for both datasets. For the MAV dataset, although FFS selects $2\times$ and $3.3\times$ more features than the case when all the AVs contribute to the process (see Table IV), the compression rate deteriorates by 1% and 3%, respectively. The quality of the selection is confirmed by the accuracy that is statistically equivalent to the case without faulty AVs. Regarding WESAD, we notice that for $\rho = 0.2$ FFS performance is equivalent to the case with all non-faulty AVs. Conversely, for $\rho = 0.3$ FFS selects a smaller (i.e., 3 instead of 4) and less informative subset of features, as confirmed by the accuracy degradation. The reason is that the information collected by ES at each round is not enough to select, globally, the most informative features. A final comment is about the network overhead, which can be further reduced, limiting the number of *contributing* AVs during a communication round. In fact, Table VI suggests that there is a trade-off between accuracy, compression rate, and number of *contributing* AVs through which we might optimise both the compression and the resources spent to find it. Moreover, there is a limit below which saving resources becomes detrimental to the learning process. However, understanding the nature of such a trade-off is left to future works.

VII. Conclusion and Future Directions

The increasing development of ADSs can leverage AI to abstract both services and applications from the details of fast-flowing low-level data, such as sensor feeds. According to the Edge computing paradigm, a cyber physical system, namely AV, is deputed in collecting data from sensors and perform a lightweight round of computation, by extracting features from raw data and selecting those that maximise the knowledge on the learning task. Since the data gathering process is performed locally by each AV, the selected features might represent a partial subset of those that characterize the phenomenon and might be inconsistent to learn the model of the underlying process. We tackle this problem, by proposing a novel Federated Feature Selection (FFS) algorithm, exploiting a distributed computing paradigm applied to AVs. In FFS, AVs collaborate to iteratively come up with the minimal set of features selected from their local datasets, to be used as a compression schema for transmitting their data to the Edge Server. Feature selection is done by leveraging on the Mutual Information metric and the solution of the optimization problem is achieved through Cross-entropy method. The aggregation algorithm of the FFS solution is based on a Bayesian approach through which we merge the control information sent by the AVs to the ES. To test the proposed FFS algorithm we presented two different learning tasks, by using real-world datasets: MAV and WESAD. The former was suitable to test FFS with images and inertial measurements, which characterize the position of an AV in the environment. The latter was suitable to characterize time series produced by human state monitoring systems, like ECG, EDA, EMG, etc. The results show that our FFS algorithm identifies a minimal subset of informative features without sharing any raw data between AVs in the process. FFS is robust to feature redundancy, i.e., in presence of high rates of redundant features, all the AVs can reach a consensus on the FS achieving a compression rate up to 90x on the selected datasets. Finally, the quality of the feature selection is maintained, i.e., a learning model trained on the selected features is as accurate as a model trained on the whole feature set. Concluding, the proposed framework is general and modular, i.e., it can be applied to every incremental FS algorithm that associates a probability to each feature. We plan to investigate how to turn it into a framework to include more FS algorithms. Moreover, our solution is built on few simplifying assumptions: local datasets are *iid* and data are labelled. Therefore, for the future we plan to extend it to include non-*iid* data in possibly unsupervised or semi-supervised scenarios.

Appendix

A. Proof of Convergence of the Federated Method

In this section, we analyze the probability that the distribution $\mathbf{p}$ converges toward the optimal solution $\mathbf{p}^*$, when the Algorithm 1 is applied in a centralized way. Then, we extend this result for the proposed federated algorithm.

The convergence analysis is based on the results in [38], [39]: following that notation, we introduce some preliminary definitions. In the CE, the candidate solutions $\mathbf{Z}_t = \{\mathbf{z}_1 \cdots \mathbf{z}_S\}$ generated at iteration t are *iid* with distribution $g(\mathbf{z}, \mathbf{p}_{t-1})$.

We define $\mathcal{Z}_t := \{\mathbf{z}_{j,\tau} \neq \mathbf{z}^* \; j = 1 \cdots S, \; \tau = 1 \cdots t\} \subseteq \mathbf{Z}_t$ as the subset of $\mathbf{Z}_t$ of the samples generated up to t that do not provide the optimal solution $\mathbf{z}^*$. The probability $\Pr(\mathcal{Z}_t)$ that the optimal solution is not available until t can be found as in

the following:

$$\Pr(\mathcal{Z}_t) = \Pr(\mathcal{Z}_1)\prod_{\tau=2}^{t}\Pr(\mathcal{Z}_\tau|\mathcal{Z}_{\tau-1}) = \Pr(\mathcal{Z}_1)\prod_{\tau=2}^{t}\left(\Pr(\mathbf{z}_\tau \neq \mathbf{z}^*|\mathcal{Z}_{\tau-1})\right)^S \quad (14)$$

The (14) comes from the statistical independence of S identically distributed samples generated by the algorithm at iteration t. The upper bound for the probability $\Pr(\mathbf{z}_\tau \neq \mathbf{z}^*|\mathcal{Z}_{\tau-1})$ that the optimal solution was unavailable until τ is derived in [38], [39] as:

$$\Pr(\mathbf{z}_\tau \neq \mathbf{z}^*|\mathcal{Z}_{\tau-1}) \leq 1 - \Pr(\mathbf{z}_1 = \mathbf{z}^*)\prod_{i=1}^{\tau-1}(1-\alpha_i)^m \quad (15)$$

where

$$\Pr(\mathbf{z}_1 = \mathbf{z}^*) = \prod_{i=1}^{m}\left(p_i(z_i=0)\mathcal{I}(z_i^*=1) + (1-p_i(z_i=0))\,\mathcal{I}(z_i^*=0)\right) \quad (16)$$

Note that due to the definition of $\mathcal{Z}_1$ its probability is $\Pr(\mathcal{Z}_1) = 1 - \Pr(\mathbf{z}_1 = \mathbf{z}^*)$.

Combining equations (15) and (16), (14) becomes:

$$\Pr(\mathbf{z}_t \neq \mathbf{z}^*) \leq \left(1 - \prod_{i=1}^{m}\left(p_i(z_i=0)\mathcal{I}(z_i^*=1) + (1-p_i(z_i=0))\,\mathcal{I}(z_i^*=0)\right)\right)\cdot\prod_{\tau=2}^{t} \times \left(1 - \prod_{i=1}^{m}\left(p_i(z_i=0)\mathcal{I}(z_i^*=1)(1-p_i + (z_i=0))\mathcal{I}(z_i^*=0)\right)\prod_{j=1}^{\tau-1}(1-\alpha_j)^m\right)^S \quad (17)$$

The right side of the (17) is close to 0 for $t \to \infty$, if $\sum_{\tau=1}^{\infty}\prod_{j=1}^{\tau-1}(1-\alpha_j)^m \to \infty$, i.e., the sequence of the parameters α_j are generated by the function $\frac{1}{j\cdot m}$, as proven by authors in [42] (section 3.7). Note that, (17) can be used to determine numerically a combination of parameter values that yields a desired minimum probability of generating the optimal solution within a time t.

Therefore, Algorithm (1) definitely provides the optimal solution when applied in a centralized way. We extend this result for the federated approach as follows. The AVs draw distinct samples $\mathbf{z}_1 \cdots \mathbf{z}_S$ independently from an identical distribution, as stated in the section V. This means that the node l finds an optimal solution for its $\mathbf{z}_1 \cdots \mathbf{z}_S$ that differs for that obtained by the centralized algorithm. Hence, combining the local distributions into the global one as in (9), we need to prove that the local node can receive from the server a federated solution that is close to the solution provided by the centralized scenario, for $t \to \infty$.

Defining the Hamming's distance $\mathcal{L}(\mathbf{z}^*, \mathbf{z}_\tau)$ between the sample $\mathbf{z}_\tau$ at the time τ and the optimal solution $\mathbf{z}^*$, the set $\tilde{\mathcal{Z}}_t := \{\mathbf{z}_{i,\tau} \mid \mathcal{L}(\mathbf{z}^*, \mathbf{z}_{i,\tau}) = m_l\}$ contains the samples generated up to time t that differs for m_l entries from the optimal solution $\mathbf{z}^*$.

As in (14), we can calculate the probability $\Pr(\tilde{\mathcal{Z}}_t)$ as follows:

$$\Pr(\tilde{\mathcal{Z}}_t) = \Pr(\tilde{\mathcal{Z}}_1)\prod_{\tau=2}^{t}\Pr(\tilde{\mathcal{Z}}_\tau|\tilde{\mathcal{Z}}_{\tau-1}) \quad (18)$$

Exploiting again the results in [38], [39], and the statistical independence of the S identically distributed samples generated by the algorithm at a given iteration, the following equation holds for the conditional probability for the given node l:

$$\Pr(\tilde{\mathcal{Z}}_\tau|\tilde{\mathcal{Z}}_{\tau-1}) = \left[\binom{m}{m_l}\Pr(\mathbf{z}_1 = \mathbf{z}_1^*)\cdot\prod_{i=1}^{\tau-1}(1-\alpha_{i,l})^{m-m_l}\cdot \times \left(1 - \Pr(\mathbf{z}_1 = \mathbf{z}_1^*)\prod_{i=1}^{\tau-1}(1-\alpha_{i,l})^{m_l}\right)\right]^S \quad (19)$$

Note that, the result provided in (18) refers to the l-th node. Hence, the global solution is obtained as the weighted average over all the local probabilities $\Pr(\tilde{\mathcal{Z}}_{l,t})$ as:

$$\Pr_{\mathbf{G}}(\tilde{\mathcal{Z}}_t) = \sum_{l=1}^{L}\Pr(\tilde{\mathcal{Z}}_{l,t})\omega_l \quad (20)$$

where ω_l are computed as in (10).

The probability in (20) is close to 0, for $t \to \infty$, if $\sum_{\tau=1}^{\infty}\prod_{i=1}^{\tau-1}(1-\alpha_{i,l})^{m_l} \to \infty \;\forall\; l = 1, \ldots, L$. Note that, if the sum of products of $(1-\alpha_{i,l})^{m_l}$ is close to ∞ also the sum of products of $(1-\alpha_{i,l})^{m-m_l}$ is close to ∞. The sequences of the $\alpha_{i,l}$ parameters guarantee the convergence also in this case. Indeed, the parameters are generated locally by the node, using the function $\frac{1}{m\cdot t}$.

B. Analysis of the Global Probability Computational Effort

In this section, we analyze the probability distribution of the number of iterations t needed to evaluate the global probability in (9). We address this issue by exploiting the result in (20), which describes the probability that the global solution obtained at the iteration t differs by m_l entries from the optimal one. Hence, the probability that the global solution is reached within t can be written as follows:

$$\Pr_{\mathbf{G}}(\mathbf{z}_t = \mathbf{z}^*) = 1 - \sum_{m_l=1}^{m}\Pr_{\mathbf{G}}(\tilde{\mathcal{Z}}_t) \quad (21)$$

We can exploit the following inequality $(1-\alpha)^m \leq e^{-\alpha m} \mid 0 \leq \alpha \leq 1,\ m \geq 0$ to find the upper bound shown in

the following:

$$\Pr_{\mathbf{G}}(\mathbf{z}_t = \mathbf{z}^*) \leq 1 - \sum_{m_l=1}^{m} \sum_{l=1}^{L} \Pr(\tilde{\mathcal{Z}}_1) \prod_{\tau=2}^{t} \left[\binom{m}{m_l} \Pr(z_1 = z_1^*) \times \left(exp\left(-\sum_{i=1}^{\tau-1} \alpha_{i,l}(m - m_l) \right) - \Pr \times (z_1 = z_1^*)\, exp\left(-\sum_{i=1}^{\tau-1} \alpha_{i,l}\, m \right) \right) \right]^S \omega(l) \tag{22}$$

The difference between exponentials in (22) goes to zero faster than the binomial coefficient goes to infinity, as m increases, if the coefficient α satisfies the conditions verified in the previous appendix. Thus (22) can be used to evaluate the probability distribution of the number of iterations $t = 1, 2 \ldots, \infty$ required to converge to the optimal global solution. The numerical analysis shows an average value of 13 iterations to converge by using the parameters presented in the section VI, which is affordable for many edge devices like Nvidia Jetson Nano or RaspberryPi.

REFERENCES

[1] A. Bondavalli, S. Bouchenak, and H. Kopetz, *Cyber-Physical Systems of Systems: Foundations—A Conceptual Model and Some Derivations: The AMADEOS Legacy*, vol. 10099. Berlin, Germany: Springer, 2016.

[2] D. Bacciu *et al.*, "Teaching-trustworthy autonomous cyber-physical applications through human-centred intelligence," in *Proc. IEEE Int. Conf. Omni-Layer Intell. Syst.*, 2021, pp. 1–6.

[3] J. E. Siegel, D. C. Erb, and S. E. Sarma, "A survey of the connected vehicle landscape—architectures, enabling technologies, applications, and development areas," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2391–2406, Aug. 2018.

[4] L. Qingqing, J. P. Queralta, T. N. Gia, H. Tenhunen, Z. Zou, and T. Westerlund, "Visual odometry offloading in internet of vehicles with compression at the edge of the network," in *Proc. 12th Int. Conf. Mobile Comput. Ubiquitous Netw.*, 2019, pp. 1–2.

[5] X. Xie and K. H. Kim, "Source compression with bounded DNN perception loss for IoT edge computer vision," in *Proc. Int. Conf. ACM MobiCom*, 2019, pp. 1–16.

[6] Y. Saeys, I. Inza, and I. Larranaga, "A review of feature selection techniques in bioinformatics," *J. Bioinf. Rev.*, vol. 23, no. 19, pp. 2507–2517, Aug. 2007.

[7] S. Egea, A. R. Manez, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Intelligent IoT traffic classification using novel search strategy for fast-based-correlation feature selection in industrial environments," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1616–1624, Jun. 2018.

[8] G. Brown, "A new perspective for information theoretic feature selection," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2009, pp. 49–56.

[9] X. Nguyen, J. Chan, S. Romano, and J. Bailey, "Effective global approaches for mutual information based feature selection," in *Proc. Int. Conf. ACM Knowl. Discov. Data Mining*, 2014, pp. 1–10.

[10] D. Rubinstein and R. Y. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Berlin, Germany: Springer, 2004.

[11] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *Proc. Int. Conf. IEEE MIPRO*, 2015, pp. 1200–1205.

[12] B. Venkatesh and J. Anuradha, "A review of feature selection and its methods," *J. Cybern. Inf. Technol.*, vol. 19, no. 1, pp. 3–26, Feb. 2019.

[13] F. Chandrashekar and G. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.

[14] L. Miao and J. Niu, "A survey on feature selection," in *Proc. Int. Conf. Elsevier Inf. Technol., Quantitative Manage.*, 2016, pp. 919–926.

[15] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," in *Proc. IEEE Bioinf. Conf. Comput. Syst. Bioinf.*, 2003, pp. 523–528.

[16] A. Gocht, C. Lehmann, and R. Schöne, "A new approach for automated feature selection," in *Proc. IEEE Int. Conf. Big Data*, 2018, pp. 4915–4920.

[17] R. Petroccia, P. Cassará, and K. Pelekanakis, "Optimizing adaptive communications in underwater acoustic networks," in *Proc. Int. Conf. IEEE/MTS OCEANS*, 2019, pp. 1–7.

[18] G. F. Anastasi, P. Cassará, P. Dazzi, A. Gotta, M. Mordacchini, and A. Passarella, "A hybrid cross-entropy cognitive-based algorithm for resource allocation in cloud environments," in *Proc. Int. Conf. IEEE Self-Adaptive Self-Organizing Syst.*, 2014, pp. 11–20.

[19] F. Guarino, P. Cassará, S. Longo, M. Cellura, and E. Ferro, "Load match optimisation of a residential building case study: A cross-entropy based electricity storage sizing algorithm," *Elsevier J. Appl. Energy*, vol. 154, pp. 380–391, 2015.

[20] R. M. Heiberger and E. Neuwirth, "One-way anova," in *R Through Excel*. Berlin, Germany: Springer, 2009, pp. 165–191.

[21] J. Konečý, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," 2015, pp. 1–5, *arXiv1511.03575*.

[22] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. AISTATS*, 2017, pp. 248–257.

[23] S. Wang *et al.*, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. Int. Conf. IEEE INFOCOM*, 2018, pp. 63–71.

[24] M. M. Amiri and D. Gunduz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," in *Proc. Int. Conf. IEEE ISIT*, 2019, pp. 1–12.

[25] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for on-device federated learning," 2019, pp. 1–30, *arXiv:1910.06378*.

[26] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1–11.

[27] Y. Mao, "Learning from differentially private neural activations with edge computing," in *Proc. Int. Conf. IEEE/ACM SEC*, 2018, pp. 90–102.

[28] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G Srivastava, "A survey on security and privacy of federated learning," *Elsevier J. Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, 2021.

[29] L. Valerio, A. Passarella, and M. Conti, "Hypothesis transfer learning for efficient data computing in smart cities environments," in *Proc. Int. Conf. SMARTCOMP*, 2016, pp. 1–8.

[30] L. Valerio, A. Passarella, and M. Conti, "A communication efficient distributed learning framework for smart environments," *Elsevier J. Pervasive Mobile Comput.*, vol. 41, pp. 46–68, 2017.

[31] X. Ye, H. Li, A. Imakura, and T. Sakurai, "Distributed collaborative feature selection based on intermediate representation," in *Proc. Int. Conf. IJCAI*, 2019, pp. 4242–4149.

[32] S. Banerjee, E. Elmroth, and M. Bhuyan, "Fed-FiS: A novel information-theoretic federated feature selection for learning stability," in *Neural Information Processing*, T. Mantoro, M. Lee, M. A. Ayu, K. W. Wong, and A. N. Hidayanto, Eds., Berlin, Germany: Springer, 2021, pp. 480–487.

[33] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. Van Laerhoven, "Introducing WESAD, a multimodal dataset for wearable stress and affect detection," in *Proc. Int. Conf. ACM ICMI*, 2019, pp. 1–9.

[34] R. McEliece, *The Theory of Information and Coding: A. Mathematical Framework for Communication* (Encyclopedia of Mathematics and Its Applications 3). Reading, MA, USA: Addison-Wesley, 1977.

[35] J. Cover and T. M. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 1991.

[36] A. Chaovalitwongse, I. Androulakis, and P. Pardalos, "Quadratic integer programming: Complexity and equivalent form," in *Encyclopedia of Optimization*. Berlin, Germany: Springer, 2009, pp. 3153–3159.

[37] G. Brown, A. Pocock, M. Zhao, and M. Zheng, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *Springer J. Mach. Learn. Res.*, vol. 13, pp. 27–66, 2012.

[38] A. Costa, O. D. Jones, and D. Kroese, "Convergence properties of the cross-entropy method for discrete optimization," *Elsevier Oper. Res. Lett.*, vol. 35, no. 7, pp. 573–580, Sep. 2007.

[39] Z. Wu and M. Kolonko, "Asymptotic properties of a generalized cross-entropy optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 658–673, Oct. 2014.

[40] P. Chen, C. Huang, C. Lien, and Y. Tsai, "An efficient hardware implementation of HOG feature extraction for human detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 656–662, Apr. 2014.

[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[42] K. Knopp, *Infinite Sequences and Series*. New York, NY, USA: Dover, 1956.

Pietro Cassará received the M.Sc. degrees in telecommunication and electronic engineering from University of Palermo, in 2005, and the Ph.D. degree from the State University of New York, in 2010. He is currently a Staff Member with the Institute of Science and Information Technologies (ISTI), National Research Council (CNR), Pisa, Italy, and since 2017, he has been a Temporary Staff Member with the CMRE lab, NATO of La Spezia. He has been participating in European, ESA, and National funded projects. His research interests include wireless sensor network and machine learning for communication and networking.

He is currently a member of the IEEE ComSoc and VTS, vice-chair of the Italian Council of IPV6, and member of IETF for the ISG IPE. He serves in the TPCs of flagship ComSoc and VTS conferences, he also contribute as a Reviewer and into the editorial board of IEEE and MDPI journals.

Alberto Gotta (Member, IEEE) received the M.Sc. and Ph.D. degrees, in 2002 and 2007, respectively. He is currently a Researcher with the Institute of Information Science and Technologies (ISTI), National Research Council (CNR), Italy. He has been Principal investigator of several EU, national, regional, and ESA funded R&D projects. He coauthored more than 100 papers among which the most cited article in 2012 and 2013 on Elsevier Computer Communications and the second most cited article from 2019 on Elsevier Array. His research interests include traffic engineering, satellite and nonterrestrial networks, aerial networks, IoT, sensor networks, and machine learning for communications. He serves the TPCs of flagship ComSoc conferences and symposia and also the editorial boards of MDPI Sensors, MDPI Network, the *International Journal of Informatics and Communication Technology* (IJ-ICT), the *International Journal of Power Electronics and Drive Systems* (IJPEDS), and the *Journal of Computer Networks and Communications* (CNC).

Lorenzo Valerio is currently a Researcher with IIT-CNR. His research activity focuses on the design of communication efficient distributed learning solutions, machine learning solutions for resource-constrained devices and machine learning based cellular traffic offloading solutions for the Future Internet. He has authored or coauthored in journals and conference proceedings more than 40+ papers. He has served as Workshop co-chair for IEEE AOC'15 and IEEE PerConAI'22. He has also been Guest Editor for the *Elsevier Computer Communications* and *Elsevier Pervasive and Mobile Computing*. He has been recipient for one Best Paper Award at IEEE WoWMoM 2013 and one Best Paper Nomination at IEEE SMARTCOMP 2016. He is currently in the editorial board of *Elsevier Computer Communications*.

Open Access provided by 'Consiglio Nazionale delle Ricerche' within the CRUI CARE Agreement