

FEDERATED LEARNING FOR IOE ENVIRONMENTS: A SERVICE PROVIDER REVENUE MAXIMIZATION FRAMEWORK

Benedetta Picano¹, Romano Fantacci¹, Tommaso Pecorella¹, Adnan Rashid¹

¹Dpt. Information Engineering, Università di Firenze, Italy

NOTE: Corresponding author: Benedetta Picano, benedetta.picano@unifi.it

Abstract – In accordance with the Internet of Everything (IoE) paradigm, millions of people and billions of devices are expected to be connected to each other, giving rise to an ever increasing demand for application services with a strict quality of service requirements. Therefore, service providers are dealing with the functional integration of the classical cloud computing architecture with edge computing networks. However, the intrinsic limited capacity of the edge computing nodes implies the need for proper virtual functions' allocations to improve user satisfaction and service fulfillment. In this sense, demand prediction is crucial in services management and exploitation. The main challenge here consists of the high variability of application requests that result in inaccurate forecasts. Federated learning has recently emerged as a solution to train mathematical learning models on the users' site. This paper investigates the application of federated learning to virtual functions demand prediction in IoE based edge-cloud computing systems, to preserve the data security and maximise service provider revenue. Additionally, the paper proposes a virtual function placement based on the services demand prediction provided by the federated learning module. A matching-based tasks allocation is proposed. Finally, numerical results validate the proposed approach, compared with a chaos theory prediction scheme.

Keywords – Edge computing, federated learning, Internet of Everything, matching theory, revenue maximization, virtual function placement

1. INTRODUCTION

The emergence of new network paradigms such as Edge Computing (EC) [1, 2, 3, 4], for which the limitations typical of the cloud architecture have been bypassed moving computation nodes to the network edges close to the end users, has given rise to a wide range of challenges in many research areas [5, 6]. Consequently, several new issues, such as user mobility, heterogeneity in Quality of Service (QoS) or service requirements, massive volume of data, user privacy, diversity on data types and so on, have led to numerous efforts from both academia and industry in providing highly effective and efficient solutions [7, 8, 9, 10, 11]. In particular, there exists a significant branch of literature regarding possible solutions to improve EC Network (ECN) performance in order to guarantee a high level of user satisfaction and to provide dynamic and flexible network resource allocation and decision-making strategies. Within this context, the Internet of Everything (IoE) paradigm, in which people, process, data, and things are connected and exchange data, has given rise to systems with increasing complexity and applications involving strict real-time requirements and sensitive data [12], heterogeneous traffic. Generally speaking, heterogeneity in data flow types implies different QoS or service requirements. Furthermore, from a Service Provider (SP) perspective, such diversity triggers new data flow management policies, service provision costs and selling prices. In this respect, the SP revenue maximization is strictly related to the adopted management and administration policy.

Indeed, a proper resource exploitation planning is essential to guarantee elevated levels of network efficiency, user satisfaction and consequent high SP revenues, as highlighted by literature such as [13], [14]. In particular, having an a priori knowledge about the data flow service demand can be properly exploited to perform suitable resource infrastructure planning with maximum income. In order to pursue this objective, Machine Learning (ML) [15, 16, 17, 18, 19] has emerged by providing many techniques to perform data behavior interpretation and analysis. The ability of ML techniques in catching data trends, patterns and hidden features, has ensured its applicability to many problems. However, although the knowledge and extrapolation of user data characteristics positively impacts many application areas, it may result in being non-compliant with some specific user privacy constraints [20]. In this respect, if on the one hand the users' data analysis may lead to remarkable advantages in reference to the network resources planning and exploitation, on the other the user data gathering may trigger user dissention, due to privacy concerns and violation. Within this context, a data-manipulation framework able to collect users' data without contravening users' privacy is a priority. In this respect, Federated Learning (FL) [21, 20, 8, 22, 23, 24] has recently emerged as a promising tool to perform, locally on the users' devices, statistical and mathematical training models based on ML methodologies without losing users privacy constraints. The FL framework consists of the devices level, generally indicated in literature as clients, and a central server unit

which aggregates and merges the data preliminary processed by the clients. Typically, FL has the following matters to face with [25]

- **Non-Independent Identically Distributed Data** The clients have different training datasets, therefore a single dataset cannot be considered representative of the other clients datasets;
- **Unbalanced Datasets** Different clients have different datasets, and each dataset may have a diverse number of elements in comparison to other clients datasets;
- **Large-Scale Distribution** The number of clients involved in the FL training procedure is generally higher than the amount of data processed at the client level;
- **Limited Communication** Mobile devices may or may not be available for data training and the computational capability or communication conditions could be poor.

In reference to the proposed contextualization, we have assumed here that sensitive user data may be derived from historical users functions utilization. In this perspective, sharing data about daily users habits may expose the users to undue risks. For this reason, the FL framework may represent a useful tool to counteract such a problem. However, a deep investigation of the privacy issues are out of the scope of this paper. The paper proposes the application of the FL framework, in order to forecast the service demands, without losing the user privacy constraints, in an IoE scenario. Moreover, on the basis of service demand forecasting, this paper proposes a suitable Virtual Functions (VFs) placement both on the ECN and cloud. Summarizing, the contributions of this paper are

- Application of the FL strategy to forecast the network VFs demand, in order to take into account the users privacy;
- Formulation of the SP maximum revenue problem, by considering Service Requests (SRs) with a different priority and hence, different cost and price. In particular, the SP can accept the data SRs with low priority if all the high priority flows have been satisfied;
- Proposal of a VFs placement strategy and a suitable matching-based SRs allocation algorithm based on the considered FL and the previously provided VFs forecasting scheme;
- Performance evaluation of the proposed approach and the comparisons with a centralized Chaos Theory (CT)-based prediction scheme, by resorting to extensive computer simulation runs.

The rest of paper is organized as follows. In Section 2 an in-depth review of the related literature is presented. Section 3, discusses the problem statement, while in Section 4 the FL framework and the placement strategy are presented. Then, in Section 5 the experimental results are analyzed and the alternative CT predictive approach explained. Finally, the conclusions are presented in Section 6.

2. RELATED WORKS

Recently, ML techniques have found extensive applications in big data analysis in fog/edge networks research area.

An overview of the ML techniques applied to fog is presented in paper [26]. Then, paper [26] investigates the ability of the ML strategies in detecting malicious attackers in fog networks, while paper [27] focuses on the ML solutions to evaluate the advantages deriving from an edge caching solution, taking into account user satisfaction perspective and energy efficiency. The improvement in sensing reliability and network latency is the aim of paper [28], in which the authors implement a multi-hidden multi-layer convolutional neural network solution to provide data authentication in a mobile crowd-sensing environment. The tree decisions strategy combined with the k-nearest neighbors method is applied in [29], in which authors deal with the position-based confidentiality problem in high real-time industrial application scenarios.

In a different way, SP maximization is the objective of paper [30], in which a deep supervised learning approach is applied to perform the minimization of the total network cost. A fog blockchain network is analyzed in paper [31], which formulates a solution based on the auction theory, where deep learning is applied to the maximization of the edge computing SP revenue.

Additionally, distributed ML is adopted in papers [32, 33, 34, 35]. In paper [32], a distributed version of the well-known support vector machine method is implemented to investigate its applicability. The reinforcement learning, and more in depth the Q-learning algorithm, is applied in paper [33], in order to minimize the users' outage in heterogeneous cellular networks scenarios. The control in crowd-sensing problem is the main objective of paper [34], exploiting the human in the loop methodology to propose a hierarchical crowd sensing framework with the aim of reducing cloud congestion and promoting the balancing of the data traffic. Then, the distributed stochastic variance reduced gradient is applied in paper [35], in which a target accuracy is fixed, and the optimization of the number of collection points to make data analysis provided. Furthermore, paper [35] proposes the minimization of the amount of network traffic sent towards the collection points. In a different way, the maximization of SP profit in a Mobile Edge Computing (MEC) blockchain network has been studied in paper [31], in which an auction strategy combined with deep learning is formulated to

perform edge resource allocation. Similarly, the auction theory is also applied to the profit maximization problem in [36], in which a novel combined optimal pricing and data allocation problem is solved with the Bayesian auction approach. The profit maximization in the cognitive virtual operator is addressed in paper [37], in which a dynamic network scenario is considered. Paper [37] develops a low complexity online control scheme to perform decisions about price and resource planning. A cloud allocation scheme for three classes of virtual machines is presented in [38], with the aim of maximising cloud provider profit.

Recently, FL has gained attention and papers [8, 39, 40, 41, 42, 20] provide its application to different contexts and situations. Paper [8] and paper [39] contextualize the FL in MEC networks, optimizing with the distributed gradient descent method the trade-off between local updates and global aggregations, formulating a loss function minimization problem, and introducing some resource constraints. Papers [8] maximize the number of clients involved in the aggregation process, aiming at minimizing the aggregation error. The MEC scenario is taken into account also in paper [39] which addresses the popularity content caching problem throughout the adoption of the hybrid filtering on stacked encoders to forecast content requests trend. Authors in [40] exploit the signal superposition property of wireless channels on the basis of which a novel aggregation data strategy for the over-the-air computation is presented. Furthermore, the model proposed in [20] is applied in [20] with the stochastic gradient descent algorithm as optimizer, aiming at training data in a distributed fashion by limiting the communication costs. The multi-task learning problem is solved with the FL and the novel Mocha context-aware optimization algorithm is presented in paper [22], while a blockchained FL architecture is proposed in [41]. Then, this architecture is designed to implement a distributed consensus strategy, by taking into account the blockchain end-to-end delay. Finally, a hybrid IoT-MEC network is considered for the application of FL in [42]. Paper [42] provides transmission and computational costs optimization, applying multiple deep reinforcement learning agents. Authors in [43] propose a QoE-driven delivery approach, in which there is cooperation between the Over-The-Top and Internet service providers, aiming at maximizing the revenue. Similarly, paper [44] addresses the economic aspects of a collaborative services management between Over-The-Top and Internet service providers. Consequently, authors propose an architecture to realize their collaboration, defining three different approaches on the basis of which the profit maximization of different customers is pursued. Then, the main objective of paper [45], is the investigation of the management procedures for multimedia services, proposing a collaborative zero-rated QoE approach to model the close cooperation between mobile network operators and the Over-The-Top service providers.

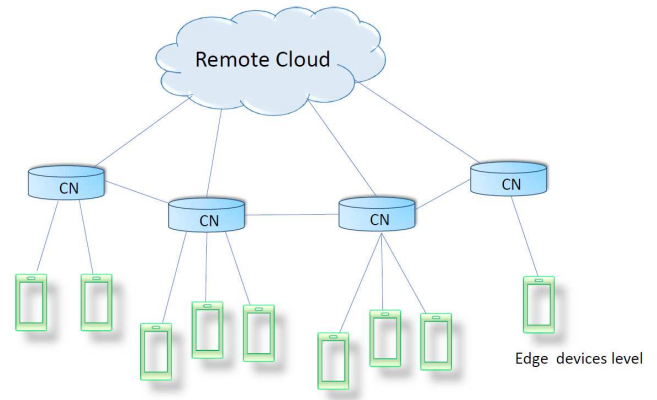


Fig. 1 – Hybrid cloud-fog network architecture

As summarized in Table 1, in contrast to papers [36, 37, 38], which provide profit maximization solutions without taking into account user privacy issues, we propose a revenue maximization framework based on data information elaborated locally on the users' devices, avoiding the typical privacy concerns of the other approaches. Hence, as in papers [40, 8, 39, 20, 41, 42], we propose an FL-based framework by using the gradient descent algorithm as optimizer. The motivation for this conservative choice resides in the fact that more complex methods may result in prohibitive consumption of the End Users' (EU) hardware resources, which is a crucial point in the distributed data training problems. Furthermore, in contrast to the previous up-to-date works, this paper contextualizes the application of the FL to the VFs deployment problem, by exploiting the FL framework to properly predict the application network demand, in order to maximize the SP revenue. Furthermore, a VFs placement and an SRs service allocation is provided to evaluate the actual validity of the proposed solution. In fact, the SRs service allocation algorithm, based on the matching theory, does not take into account the SP perspective, but only the users, i.e., the SRs, interests. Finally, to the best of our knowledge, this is the first paper which applies the FL to the SP revenue maximization problem, by considering even the users' perspective. The proposed approach performance has been evaluated by resorting to extensive numerical simulation and by providing comparison with the centralized CT-based predictive method.

3. PROBLEM STATEMENT

As an IoE reference scenario, we consider a single SP featuring an ECN constituted by \mathcal{N} Computation Nodes (CNs) located at the network edges, and a more powerful cloud located far from the ECN. We suppose that all the CNs are equipped with a Central Processing Unit (CPU) with the same computational capability and number of available Storage Resource Blocks (SRBs) S . In a different way, the cloud is assumed to have a storage capacity of U SRBs, with $S < U$. In addition, we assume the availability of high speed wired links between CNs and from

Table 1 – Literature contributions

Standard Literature	Paper contribution
[36, 37, 38]	Proposal of a revenue maximization framework based on data information elaborated locally on the users' devices, avoiding the typical privacy concerns of the other approaches.
[40, 8, 39, 20, 41, 42]	Contextualization of the application of the FL to the VFs deployment problem, by exploiting the FL framework to properly predict the application network demand, in order to maximize the SP revenue.

Table 2 – Main symbols

Notation	Description
CN	Computation node
VF	Virtual function
FL	Federated learning
SRB	Storage resource block
SR	Service request
S	Number of SRBs per CN
U	Cloud SRBs
ECN	Edge computing network
\mathcal{T}	High priority requests
\mathcal{M}	Low priority requests
τ_i	Time deadline
x_i	Number of req. demanding for service i
y_j	Number of req. demanding for service j
$\mathcal{X}(x_i, q_i)$	SP revenue for the high priority req.
$\mathcal{Y}(y_j, z_j)$	SP revenue for the low priority req.
T_r	Service accomplishment time
$\omega_{z,h}$	Waiting time on the CN
$\omega_{z,C}$	Waiting time on the cloud

any CN to the cloud¹. Furthermore, we guess that the ECN is able to support \mathcal{T} different high priority service types, which are characterized by different provision costs and selling prices. Each service type $i \in \mathcal{T}$ has associated a QoS level expressed as a time deadline τ_i before which the type i service accomplishment has to be completed. In addition, we consider the presence of \mathcal{M} service type requests with lower priority and without any time deadline constraint. The number of requests belonging to this class is indicated hereafter with y_j , with $j \in \mathcal{M}$.

Periodically, the SP updates the service demand and we assume that any new request does not arrive between two SP updates.

Let x_i be the number of SRs demanding for service i . We suppose that each SR is originated by an EU, and that an EU requires only one SR. Therefore, as a direct consequence, hereafter we assume interchangeable the SR and EU terms. Then, as regards the SP, the provision of a service has a cost mainly depending on x_i and following the model given by [46]

¹We have assumed that the connection towards the cloud is performed throughout the CN nearest to the SRs needing computation. Consequently, the communication latency cost between SRs and their nearest CN has no impact on the overall SR completion time and hence it has been neglected in defining (7).

$$c(x_i) = \begin{cases} 0, & x_i = 0, \\ \beta_{c,i} + \beta_{l,i}\mu_i^{x_i}, & x_i > 0, \end{cases} \quad (1)$$

in which $\beta_{c,i}, \beta_{l,i}, \mu_i$ are real valued parameters whose value changes on the basis of the request type. Similarly, the provision cost for providing y_j SRs of type j follows the rule [46]

$$b(y_j) = \begin{cases} 0, & y_j = 0, \\ \alpha_{c,j} + \alpha_{l,j}\nu_j^{y_j}, & y_j > 0, \end{cases} \quad (2)$$

where $\alpha_{c,j}, \alpha_{l,j}, \nu_j$ are, also in this case, real valued parameters.

Moreover, for each service type with high priority, the SP revenue results ruled by the following relation

$$U(x_i, q_i) = \frac{\log(1 + x_i)}{q_i}, \quad (3)$$

with $q_i = |x_i - k_i|$, where k_i is the number of SRs for which τ_i has been respected. Then, the SP revenue for the low priority SRs is given by

$$U(y_j, z_j) = \frac{\log(1 + y_j)}{z_j}, \quad (4)$$

where z_j is the number of SRs among y_j accepted by the network for their service. Hence, the SP revenue, corresponding to the provision of the i -th and the j -th service type, can be expressed as

$$\mathcal{X}(x_i, q_i) = U(x_i, q_i) - c(x_i), \quad (5)$$

and

$$\mathcal{Y}(y_j, z_j) = U(y_j, z_j) - b(y_j), \quad (6)$$

respectively.

Both the SRs with high and low priority, in order to be accomplished, require the presence of a VF in set \mathcal{V} which has to be preliminary loaded on at least one CN of the network or on the far cloud. The loading process requires the CN or cloud availability in terms of SRBs, since each VF $v \in \mathcal{V}$ requires a number a_v of SRBs, different for each VF. Consequently, the time required for the service accomplishment (TSA) of a generic SR r , independently by its priority, is given by

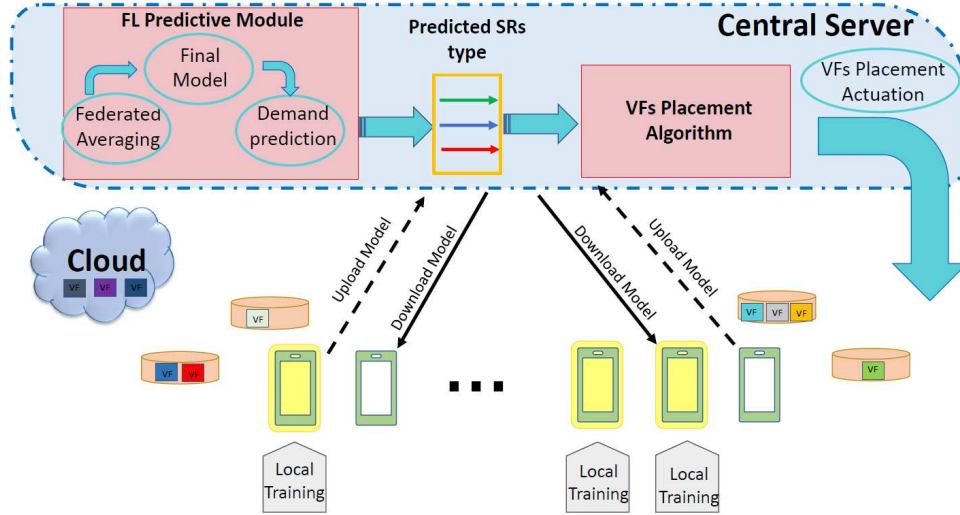


Fig. 2 – FL framework for the VFs placement

$$T_r = \sum_{v \in \mathcal{V}} \sum_{h \in \mathcal{N}} (\gamma_z + \omega_{z,h}) \rho_{r,h} \theta_{v,h} + (1 - \rho_{r,h}) \zeta_{v,C} (\gamma_C + \omega_{z,C}), \quad (7)$$

where γ_z and γ_C are the execution time spent by the SR z on the CPU of a CN and of the cloud, respectively. It is important to note that both the execution times γ_z and γ_C mainly depend on the size of the SR z , the CPU frequency of the node hosting its elaboration, and the time spent by the SR on that node waiting for the actual computation. Therefore, $\omega_{z,h}$ and $\omega_{z,C}$ represent the queuing time experienced by the SR z waiting for its execution on the CN h and cloud, respectively². Furthermore, $\rho_{r,h}$ is a binary value equal to 1 if the SR j is executed on the CN h , 0 otherwise. Similarly, $\theta_{v,h}$ is equal to 1 when the VF v is present on CN h , 0 otherwise. Finally, $\zeta_{v,C}$ is equal to 1 if the VF v is loaded on cloud, 0 otherwise. It is important to make evident that the TSA in (7) strongly depends on the queuing time experienced by the SR on the service provision site. In fact, a proper deployment of VFs on the ECN may drastically reduce the TSA time.

In formal terms, the aim of this paper is the maximization of the SP revenue by providing decision making on the VFs placement, in order to satisfy the SRs. Therefore, the main goal of the paper is given by

$$\min_{\mathbf{q}, \mathbf{z}} \sum_{i=1, \dots, \mathcal{J}} \mathcal{X}(x_i, q_i) + \sum_{j=1, \dots, \mathcal{M}} \mathcal{Y}(y_j, z_j), \quad (8)$$

s.t.

$$T_i \leq \tau_i, \forall i = 1, \dots, \mathcal{J}, \quad (9)$$

$$\sum_{v \in \mathcal{V}} \theta_{v,h} a_v \leq S, \forall h \in \mathcal{N}, \quad (10)$$

$$\sum_{v \in \mathcal{V}} \zeta_{v,C} a_v \leq U. \quad (11)$$

²The CPU queue has been modeled with the first-in-first-out service policy.

In problems (4)-(10), constraint (9) expresses the fact that each SR with a high priority has to be served, while constraints (10) and (11) represent that the VFs allocation has to respect the storage limit of CNs and cloud, respectively.

4. FEDERATED LEARNING FRAMEWORK

4.1 The learning problem

The aim of ML is the exploitation of some data used for training, to learn models. In order to do that, typically, ML involves the definition of a loss function representing the error implicitly resulting from the model training [8]. The loss function depends on the data sample z and a parameter vector \mathbf{w} , and it is named hereafter as $f_z(\mathbf{w})$. As previously introduced, this paper supposes the presence of L SRs, with $L = \mathcal{J} + \mathcal{M}$, deriving from an underlying level of EUs, each of which disposes of a local dataset Θ_l , $l = 1, \dots, L$. Therefore, as assumed in [8, 20], we suppose the collective loss function equals to

$$F_l(\mathbf{w}) = \frac{1}{|\Theta_l|} \sum_{z \in \Gamma_l} f_z(\mathbf{w}), \quad (12)$$

where $|\Gamma_l|$ is the number of elements belonging to Γ_l , referred as the cardinality of the Γ_l set. Respectively, the global function evaluated at the central server site, the global loss function, based on the distributed local dataset Θ_l and defined as [8, 20], is expressed by the following relation

$$F(\mathbf{w}) = \frac{\sum_{l=1, \dots, L} |\Theta_l| F_l(\mathbf{w})}{\sum_{l=1, \dots, L} |\Theta_l|}. \quad (13)$$

Therefore, the objective here is to find \mathbf{w}^* such that [8]

$$\mathbf{w}^* = \operatorname{argmin} F(\mathbf{w}). \quad (14)$$

Accordingly, with numerous contemporary papers [20, 8] recently proposed in literature, the optimization of (14) limiting the computational complexity, is pursued by applying the gradient descent method.

4.2 Federated learning framework

Algorithm 1 Client Side

```

1: for each NE involved in learning process do
2:   update  $\mathbf{w}_\chi(u) = \hat{\mathbf{w}}_\chi(u-1) - \xi \nabla F_\chi(\hat{\mathbf{w}}_\chi(u-1))$ ;
3:   return  $\mathbf{w}_\chi(u)$  to the central server;
4: end for
    
```

Algorithm 2 Server Side

```

1: initialize  $\mathbf{w}_0$ ;
2: for each NE involved in learning process in parallel do
3:   Receive and update  $\mathbf{w}_\chi(u)$ 
4: end for
5: update global model  $\mathbf{w}(u) = \frac{\sum_{\chi \in \mathcal{X}} |\Theta_\chi| \mathbf{w}_\chi}{\sum_{\chi \in \mathcal{X}} |\Theta_\chi|}$ .
    
```

Algorithm 3 VFs Placement Planning

```

1: Input: predicted application popularity vector  $\mathbf{p}$ ;
2: for each VF  $v \in \mathbf{p}$  do
3:   for each CN  $h \in \mathcal{N}$  do
4:     if  $h$  has enough SRBs then load  $v$  on  $h$ ;
5:     else
6:       if cloud has enough SRBs then
7:         load  $v$  on cloud;
8:       end if
9:     end if
10:  end for
11: end for
    
```

As represented in Fig. 2, the proposed FL framework consists of the client level, responsible for the distributed local data training, and of a server side. The server side is typically represented by a base station or a more general central unit, set up for improving the global learning model, and to merge the locally trained EU models. The client and server sides interact with each other, throughout a series of iteration rounds u . It is important to highlight that the number of EUs involved in the training procedure are a subset of the totality of the EUs.

The FL procedure consists of the following steps

- Let \mathcal{K} be the set of the EUs involved in the training process. In parallel, each EU belonging to \mathcal{K} , i.e. EU χ , updates its local parameter vector $\mathbf{w}_\chi(u)$, which depends on its local dataset Θ_χ , accordingly with the following rule [8]

$$\mathbf{w}_\chi(u) = \hat{\mathbf{w}}_\chi(u-1) - \xi \nabla F_\chi(\hat{\mathbf{w}}_\chi(u-1)), \quad (15)$$

where ξ is the learning rate and $\hat{\mathbf{w}}_\chi(u-1)$ represents the term $\mathbf{w}_\chi(u-1)$ after global aggregation.

- As detailed in [20], the server side computes the weighted average expressed by

$$\mathbf{w}(u) = \frac{\sum_{\chi \in \mathcal{X}} |\Theta_\chi| \mathbf{w}_\chi}{\sum_{\chi \in \mathcal{X}} |\Theta_\chi|}. \quad (16)$$

It is important to make evident that EUs, in performing distributed data training accordingly with the FL framework, achieve numerous advantages in terms of client privacy, and limited exploitation of their computational resources. This is directly connected to the fact that training data locally on the client's site, helps users to keep their sensitive and personal information reserved, since the uploading of the EU χ parameter vector \mathbf{w}_χ does not expose the client to any sort of privacy matter. More specifically, from \mathbf{w}_χ , it is not elementary to retrieve Θ_χ . Finally, each algorithm iteration round involves just a part of the whole EUs' set, reducing the message passing between client and central server entities. Strongly connected with this aspect, the usage of the gradient descent algorithm is able to afford the learning problem without implying an excessive resource consumption, meeting the limited computational capabilities intrinsic of each mobile device.

Algorithms 1 and 2 exhibit the pseudocode corresponding to the client and server sides, respectively.

4.3 VFs placement planning

Once the FL framework is applied to obtain SRs prediction on the basis of the historical EUs' information, properly aggregated by the central server, the VFs' placement planning strategy starts. The placement acts on the basis of the VFs popularity, expressed with the popularity vector \mathbf{p} . The popularity vector \mathbf{p} has length equal to \mathcal{V} and contains the type of the VFs sorted by descending order on the basis of the occurrence frequency of each VF type in the pool of the whole network requests.

In order to validate the benefits of the proposed framework to the VFs placement problem, we propose a straightforward placement strategy strictly dependent on \mathbf{p} . Supposing that the predicted network SRs are given in terms of the VFs' popularity and expressed with the popularity vector \mathbf{p} , the VFs' placement is realized through the following steps

1. Process the popularity vector \mathbf{p} starting from the most popular VF in \mathbf{p} , i.e., r^* , hence from the most requested VF;
2. Deploy r^* on the first CN with enough available SRBs to host r^* ;
3. Deploy r^* on the cloud if it has enough available SRBs to host r^* ;
4. If r^* cannot be loaded neither on the CNs nor on the cloud

- (a) if the VF \hat{r} which can be hosted by a CN or cloud does not exist in \mathbf{p} , then terminate placement;
- (b) Otherwise repeat steps 1) – 4).

The pseudocode of the VFs planning strategy is detailed in Algorithm 3.

Algorithm 4 SRs Allocation Planning

- 1: **until** all the SRs are not allocated **repeat**
 - 2: **for each** unallocated SR r **do**
 - 3: builds its preferences on \mathcal{C}_r and proposes to its favorite element in \mathcal{C}_r ;
 - 4: **end for**
 - 5: **for each** computation site **do**
 - 6: accepts the SR requiring the VF type with the more stringent deadline;
 - 7: updates the corresponding queuing time;
 - 8: **end for**
 - 9: **end repeat**
-

4.4 SRs allocation planning

The designed SRs allocation policy is based on the matching theory principles [47, 48], and consider the EUs' perspective. In order to better explain this point, it is important to highlight that the SRs allocation strategy is based on metrics which do not consider the SP revenue, but only the EUs' interests. In this regard, the two parts involved in the matching are the SRs and the computational sites, referred hereafter, for each SR r , as \mathcal{C}_r . The set of the computational sites may be different for diverse SRs since, given the SR r , \mathcal{C}_r consists of the CNs which contain the VF requested by r and of the cloud, if this contains the desired VF. Each SR r expresses the preference in being matched, i.e., in being computed, with each element of \mathcal{C}_r and vice versa. The SRs aim at minimizing their own TSA defined as in (7), hence they prefer to be executed on computational sites which lower (7). By contrast, the computational sites prefer SRs requiring VFs with stringent deadline requirements.

Therefore, the matching algorithm consisting of a modified version of the Gale-Shapley [47] algorithm can be summarized through the following steps

1. Each SR builds its preference on the elements belonging to \mathcal{C}_r ;
2. Each SR r , proposes to be computed on its most preferred computational site;
3. Each computational site, among the received computational proposals, accepts the SR requiring the VF type with the closest deadline, and discards the other proposals;
4. Update queuing time on each CN;
5. Update preferences of the unallocated SRs;

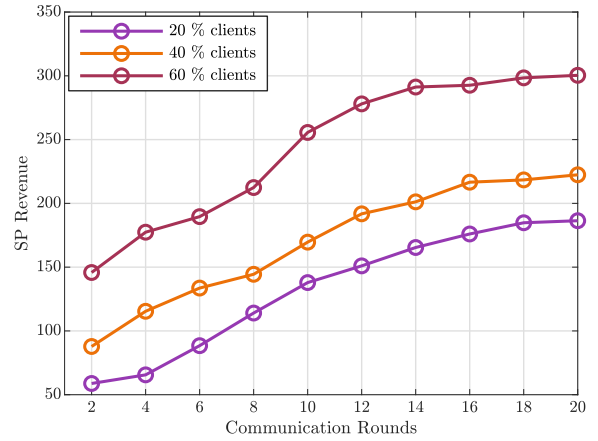


Fig. 3 – SP revenue by varying communication rounds, considering 100 SRs and 20 VFs

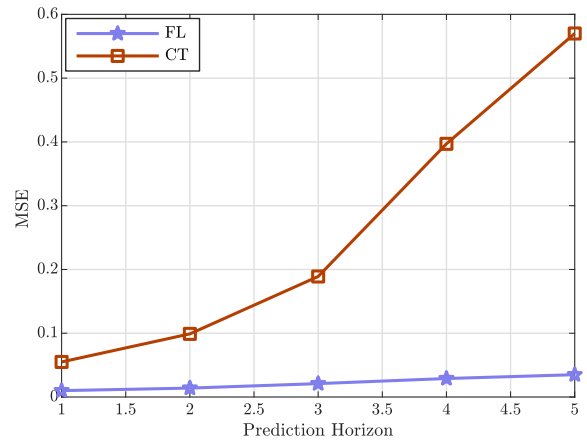


Fig. 4 – MSE by varying the time prediction horizon for type 1 SRs

6. repeat steps 2) – 6) until all the SRs are allocated.

Algorithm 4 explains in more detail the SRs allocation planning procedure.

5. NUMERICAL RESULTS

The proposed FL-based framework has been tested by resorting to numerical simulations in the Tensorflow environment. We supposed an IoE scenario consisting of $\mathcal{N} = 3$ CNs, equipped with a CPU frequency equals to 2.4 GHz, while the cloud has been equipped with a CPU frequency equals to 4.6 GHz. Furthermore, we set $S = 70$ and $U = 120$.

The VFs required by SRs have been modeled in a similar way as in [39, 49, 50], and we considered the presence of two priorities, corresponding to the set MovieLens 1M dataset [51] and MovieLens 100K dataset [51], respectively. We modeled 10 VFs, each of which needs a number of SRBs uniformly distributed in [50, 80]. All the FL network hyperparameters and the neural architecture have been assumed to be the same as those in [39]. Each SR has been modeled as a number of 64 bits format instructions uniformly distributed in [250, 800], needing 8 CPU

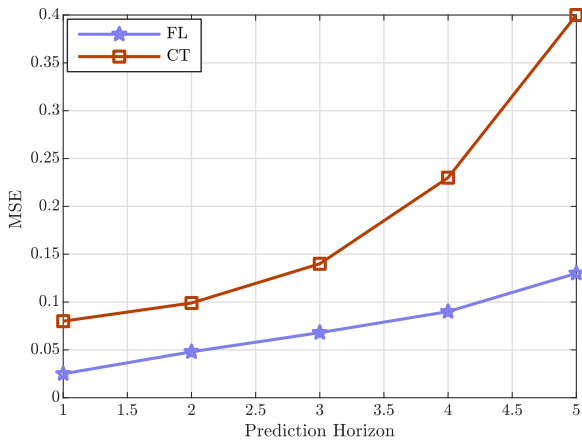


Fig. 5 – MSE by varying the time prediction horizon for type 2 SRs

cycles per instruction. Furthermore, as loss function, we adopted the Mean Squared Error (MSE) which, for each data ι_ϕ in Θ_χ is defined as

$$MSE = \frac{1}{\Phi} \sum_{\phi=1}^{\Phi} (\hat{\iota}_\phi - \iota_\phi)^2, \quad (17)$$

where Φ is the number of samples in the test data, and $\hat{\iota}_\phi$ is the predicted value. Then, to test the effectiveness of the proposed approach, we made comparison in terms of accuracy of our strategy, with the prediction scheme based on the application of the CT principles by performing the phase space reconstruction method as explained in [52, 53], and by using the predictive model of the k-neighbors discussed in [54]. It is important to note that the CT approach is performed on the central server site, on which all the user data is gathered without considering the preservation of their privacy.

Fig. 4 and Fig. 5, which exhibit the MSE behavior by varying the prediction horizon, confirm the greater accuracy of the proposed model in comparison to CT. As it is evident in Fig. 4 and Fig. 5, the MSE grows as the prediction horizon increases. This is a direct consequence of the natural difficulty in predicting the long-term behavior of the series. Nevertheless, both the figures show the superiority of the proposed approach in comparison with the alternative here considered.

Then, Fig. 3 makes clear the significant improvement obtained by increasing the number of communication rounds, i.e., information updates, between the server and the clients, for different numbers of EUs involved in the FL process. The direct implication is that higher is the number of the EUs taking part in the learning process, the greater the levels of accuracy on the acquired information on which the VFs placement strategy is based. Moreover, the SP revenue improves its trend. It is important to highlight here that the FL requires a converge time of 12.42 seconds to converge, against the 6.17 seconds required by the CT approach. Fig. 6 shows the SP revenue behavior by increasing the number of SRs. As it is straightforward to note, the SP revenue tends to grow by increas-

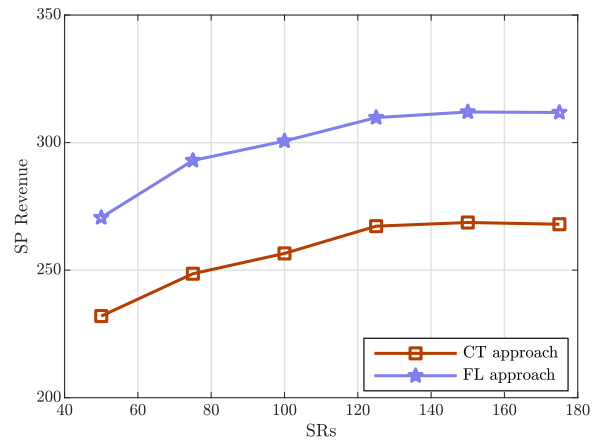


Fig. 6 – SP revenue by varying the number of SRs, considering 10 VFs

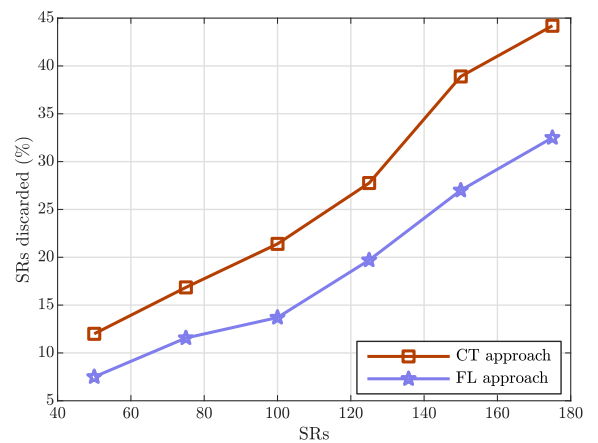


Fig. 7 – Percentage of SRs discarded, by increasing the SR number

ing the number of SRs, until the network infrastructure is not saturated and consequently it cannot accept new SRs. Such a situation is clearly a consequence of the physical resources limitation of the network. Finally, Fig. 7 depicts the behavior of the percentage of the SRs discarded, i.e., the percentage of the SRs which have not been served by the network infrastructure since their computation is not finished before the expiration of their deadline. In conclusion, the resulting system performance makes clear the validity of the FL application for our problem, highlighting the importance of considering the data expressing the users' preferences and daily habits.

6. CONCLUSION

This paper has dealt with a framework based on the federated learning paradigm to maximise SP revenue, in a hybrid cloud-edge system, arranged to support IoE applications. The proposed framework resorts to the use of the FL approach to predict the SRs demand, in compliance with the users' privacy. Furthermore, a VFs placement on the basis of the obtained SRs demand prediction has been performed and, the related SRs allocation, modeled as a matching game problem, has been hence

accomplished. The effectiveness of the proposed framework has been finally validated by providing performance comparisons with an alternative predictive approach based on the chaos theory. In reference to the future research directions, a very interesting topic needing further exploration may be represented by the definition of novel solutions and methodologies to allow the design of privacy-based learning and inference of deep learning and advanced signal processing in heterogeneous hardware architectures. Such a privacy-preserving approach will rely on Homomorphic Encryption that enables processing directly on encrypted data.

REFERENCES

- [1] H. Tianfield. "Towards Edge-Cloud Computing". In: *2018 IEEE International Conference on Big Data (Big Data)*. Dec. 2018, pp. 4883–4885. DOI: 10 . 1109/BigData.2018.8622052.
- [2] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. de Foy, and Y. Zhang. "Mobile Edge Cloud System: Architectures, Challenges, and Approaches". In: vol. 12. 3. Sept. 2018, pp. 2495–2508. DOI:10 . 1109/JSYST.2017.2654119.
- [3] X. Shan, H. Zhi, P. Li, and Z. Han. "A Survey on Computation Offloading for Mobile Edge Computing Information". In: *2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*. May 2018, pp. 248–251. DOI:10 . 1109/BDS/HPSC/IDS18.2018.00060.
- [4] P. Mach and Z. Becvar. "Mobile Edge Computing: A Survey on Architecture and Computation Offloading". In: *IEEE Communications Surveys Tutorials* 19.3 (2017), pp. 1628–1656. ISSN:1553-877X. DOI:10 . 1109/COMST.2017.2682318.
- [5] S. Singh. "Optimize cloud computations using edge computing". In: *2017 International Conference on Big Data, IoT and Data Science (BIG)*. Dec. 2017, pp. 49–53. DOI:10 . 1109/BID.2017.8336572.
- [6] P. Corcoran and S. K. Datta. "Mobile-Edge Computing and the Internet of Things for Consumers: Extending cloud computing and services to the edge of the network". In: *IEEE Consumer Electronics Magazine* 5.4 (Oct. 2016), pp. 73–74. ISSN:2162-2248. DOI:10 . 1109/MCE.2016.2590099.
- [7] M. Chiang and T. Zhang. "Fog and IoT: An Overview of Research Opportunities". In: vol. 3. 6. Dec. 2016, pp. 854–864. DOI:10 . 1109/JIOT.2016.2584538.
- [8] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. S. Chan. "When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning". In: 2018, pp. 63–71.
- [9] R. Fantacci and B. Picano. "A Matching Game With Discard Policy for Virtual Machines Placement in Hybrid Cloud-Edge Architecture for Industrial IoT Systems". In: *IEEE Transactions on Industrial Informatics* 16.11 (2020), pp. 7046–7055. DOI:10 . 1109/TII.2020.2999880.
- [10] F. Chiti, R. Fantacci, and B. Picano. "A matching game for tasks offloading in integrated edge-fog computing systems". In: *Transactions on Emerging Telecommunications Technologies* 31.2 (2020). e3718 ett.3718, e3718. DOI: [https : / / doi . org / 10 . 1002 / ett . 3718](https://doi.org/10.1002/ett.3718). eprint: [https : / / onlinelibrary . wiley . com / doi / pdf / 10 . 1002 / ett . 3718](https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.3718). URL:<https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3718>.
- [11] R. Fantacci and B. Picano. "When Network Slicing Meets Prospect Theory: A Service Provider Revenue Maximization Framework". In: *IEEE Transactions on Vehicular Technology* 69.3 (2020), pp. 3179–3189. DOI: 10 . 1109 / TVT . 2019 . 2963462.
- [12] W. Saad, M. Bennis, and M. Chen. "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems". In: *IEEE Network* 34.3 (2020), pp. 134–142. DOI: 10 . 1109 / MNET . 001 . 1900287.
- [13] C. Tselios, I. Politis, M. Tsagkaropoulos, and T. Dag-iuklas. "Valuing Quality of Experience: A Brave New Era of User Satisfaction and Revenue Possibilities". In: *2011 50th FITCE Congress - "ICT: Bridging an Ever Shifting Digital Divide"*. 2011. DOI:10 . 1109 / FITCE . 2011 . 6133422.
- [14] F. Conti, S. Colonnese, F. Cuomo, L. Chiaraviglio, and I. Rubin. "Quality Of Experience Meets Operators Revenue: Dash Aware Management for Mobile Streaming". In: *2019 8th European Workshop on Visual Information Processing (EUVIP)*. 2019, pp. 64–69. DOI:10 . 1109/EUVIP47703.2019.8946152.
- [15] C. Zhang, P. Patras, and H. Haddadi. "Deep Learning in Mobile and Wireless Networking: A Survey". In: vol. abs/1803.04311. 2018. arXiv: 1803 . 04311. URL:<http://arxiv.org/abs/1803.04311>.
- [16] S. Athmaja, M. Hanumanthappa, and V. Kavitha. "A survey of machine learning algorithms for big data analytics". In: *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. Mar. 2017, pp. 1–4. DOI: 10 . 1109/ICIIECS.2017.8276028.
- [17] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani. "Deep Learning for IoT Big Data and Streaming Analytics: A Survey". In: *IEEE Communications Surveys Tutorials* 20.4 (2018), pp. 2923–2960. ISSN: 1553-877X. DOI: 10 . 1109 / COMST . 2018 . 2844341.

- [18] M. A. Alsheikh, S. Lin, D. Niyato, and H. Tan. "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications". In: *IEEE Communications Surveys Tutorials* 16.4 (2014), pp. 1996–2018. ISSN: 1553-877X. DOI:10.1109/COMST.2014.2320099.
- [19] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza. "A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks". In: *IEEE Communications Surveys Tutorials* 19.4 (2017), pp. 2392–2431. ISSN: 1553-877X. DOI:10.1109/COMST.2017.2727878.
- [20] H. B. McMahan, E. Moore, D. Ramage, and B. Agüera y Arcas. "Federated Learning of Deep Networks using Model Averaging". In: vol. abs/1602.05629. 2016. arXiv: 1602.05629. URL: <http://arxiv.org/abs/1602.05629>.
- [21] Q. Yang, Y. Liu, T. Chen, and Y. Tong. "Federated Machine Learning: Concept and Applications". In: vol. abs/1902.04885. 2019. arXiv: 1902.04885. URL: <http://arxiv.org/abs/1902.04885>.
- [22] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar. "Federated Multi-Task Learning". In: vol. abs/1705.10467. 2017. arXiv: 1705.10467. URL: <http://arxiv.org/abs/1705.10467>.
- [23] N. H. Tran, W. Bao, A. Zomaya, N. Minh N.H., and C. S. Hong. "Federated Learning over Wireless Networks: Optimization Model Design and Analysis". In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. Apr. 2019, pp. 1387–1395. DOI:10.1109/INFOCOM.2019.8737464.
- [24] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi. "Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning". In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*. Apr. 2019, pp. 2512–2520. DOI:10.1109/INFOCOM.2019.8737416.
- [25] H. B. McMahan, E. Moore, D. Ramage, and B. Agüera y Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Vol. 54. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Apr. 2017, pp. 1273–1282. URL: <http://proceedings.mlr.press/v54/mcmahan17a.html>.
- [26] P. Subramaniam and M. J. Kaur. "Review of Security in Mobile Edge Computing with Deep Learning". In: (Mar. 2019), pp. 1–5. DOI:10.1109/ICASET.2019.8714349.
- [27] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi. *Learn to Cache: Machine Learning for Network Edge Caching in the Big Data Era*. Vol. 25. 3. June 2018, pp. 28–35. DOI:10.1109/MWC.2018.1700317.
- [28] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez. "Robust Mobile Crowd Sensing: When Deep Learning Meets Edge Computing". In: vol. 32. 4. July 2018, pp. 54–60. DOI:10.1109/MNET.2018.1700442.
- [29] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hosain, and G. Muhammad. "Enforcing Position-Based Confidentiality with Machine Learning Paradigm through Mobile Edge Computing in Real-Time Industrial Informatics". In: *IEEE Transactions on Industrial Informatics* (2019), pp. 1–1. ISSN: 1551-3203. DOI:10.1109/TII.2019.2898174.
- [30] S. Yu, X. Wang, and R. Langar. "Computation offloading for mobile edge computing: A deep learning approach". In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Oct. 2017, pp. 1–6. DOI:10.1109/PIMRC.2017.8292514.
- [31] N. C. Luong, Z. Xiong, P. Wang, and D. Niyato. "Optimal Auction for Edge Computing Resource Management in Mobile Blockchain Networks: A Deep Learning Approach". In: (May 2018), pp. 1–6. ISSN: 1938-1883. DOI:10.1109/ICC.2018.8422743.
- [32] T. Tuor, S. Wang, T. Salonidis, B. J. Ko, and K. K. Leung. "Demo abstract: Distributed machine learning at resource-limited edge nodes". In: *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Apr. 2018, pp. 1–2. DOI:10.1109/INFOCOMW.2018.8406837.
- [33] T. Kudo and T. Ohtsuki. "Cell selection using distributed Q-learning in heterogeneous networks". In: *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. Oct. 2013, pp. 1–6. DOI:10.1109/APSIPA.2013.6694368.
- [34] L. Li, K. Ota, and M. Dong. "Human in the Loop: Distributed Deep Model for Mobile Crowdsensing". In: *IEEE Internet of Things Journal* 5.6 (Dec. 2018), pp. 4957–4964. ISSN: 2327-4662. DOI:10.1109/JIOT.2018.2883318.
- [35] L. Valerio, A. Passarella, and M. Conti. "Optimal trade-off between accuracy and network cost of distributed learning in Mobile Edge Computing: An analytical approach". In: (June 2017), pp. 1–9. DOI:10.1109/WoWMoM.2017.7974310.
- [36] Y. Jiao, P. Wang, D. Niyato, M. Abu Alsheikh, and S. Feng. "Profit Maximization Auction and Data Management in Big Data Markets". In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. Mar. 2017, pp. 1–6. DOI:10.1109/WCNC.2017.7925760.

- [37] S. Li, J. Huang, and S. R. Li. "Dynamic Profit Maximization of Cognitive Mobile Virtual Network Operator". In: *IEEE Transactions on Mobile Computing* 13.3 (Mar. 2014), pp. 526–540. ISSN: 1536-1233. DOI:10.1109/TMC.2013.10.
- [38] M. Li, Y. Sun, H. Huang, J. Yuan, Y. Du, Y. Bao, and Y. Luo. "Profit maximization resource allocation in cloud computing with performance guarantee". In: *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*. Dec. 2017, pp. 1–2. DOI:10.1109/PCCC.2017.8280482.
- [39] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas. "Federated Learning Based Proactive Content Caching in Edge Computing". In: *2018 IEEE Global Communications Conference (GLOBECOM)*. 2018, pp. 1–6. DOI:10.1109/GLOCOM.2018.8647616.
- [40] K. Yang, T. Jiang, Y. Shi, and Z. Ding. "Federated Learning via Over-the-Air Computation". In: Dec. 2018.
- [41] H. Kim, J. Park, M. Bennis, and S. Kim. "Blockchained On-Device Federated Learning". In: 2019, pp. 1–1. DOI:10.1109/LCOMM.2019.2921755.
- [42] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang. "Federated Learning-Based Computation Offloading Optimization in Edge Computing-Supported Internet of Things". In: vol. 7. 2019, pp. 69194–69201. DOI:10.1109/ACCESS.2019.2919736.
- [43] A. Ahmad, A. Floris, and L. Atzori. "QoE-centric service delivery: A collaborative approach among OTTs and ISPs". In: *Computer Networks* 110 (2016), pp. 168–179. ISSN: 1389-1286. DOI:10.1016/j.comnet.2016.09.022.
- [44] A. Floris, A. Ahmad, and L. Atzori. "QoE-Aware OTT-ISP Collaboration in Service Management: Architecture and Approaches". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 14.2s (Apr. 2018). ISSN: 1551-6857. DOI:10.1145/3183517.
- [45] A. Ahmad and L. Atzori. "MNO-OTT Collaborative Video Streaming in 5G: The Zero-Rated QoE Approach for Quality and Resource Management". In: *IEEE Transactions on Network and Service Management* 17.1 (2020), pp. 361–374. DOI:10.1109/TNSM.2019.2942716.
- [46] S. Wang, R. Uргаonkar, M. Zafer, T. He, K. S. Chan, and K. K. Leung. "Dynamic Service Migration in Mobile Edge-Clouds". In: CoRR abs/1506.05261 (2015). arXiv: 1506.05261. URL: <http://arxiv.org/abs/1506.05261>.
- [47] S. Bayat, Y. Li, L. Song, and Z. Han. "Matching Theory: Applications in wireless communications". In: *IEEE Signal Processing Magazine* 33.6 (Nov. 2016), pp. 103–122. ISSN:1053-5888. DOI:10.1109/MSP.2016.2598848.
- [48] A. E. Roth and M. Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Cambridge University Press, UK, 1990.
- [49] S. Müller, O. Atan, M. van der Schaar, and A. Klein. "Context-Aware Proactive Content Caching with Service Differentiation in Wireless Networks". In: *IEEE Transactions on Wireless Communications* PP (June 2016). DOI:10.1109/TWC.2016.2636139.
- [50] S. Li, J. Xu, M. van der Schaar, and W. Li. "Popularity-driven content caching". In: *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. Apr. 2016, pp. 1–9. DOI:10.1109/INFOCOM.2016.7524381.
- [51] F. M. Harper and J. A. Konstan. "The MovieLens Datasets: History and Context". In: *ACM Trans. Interact. Intell. Syst.* 5.4 (Dec. 2015), 19:1–19:19. ISSN: 2160-6455. DOI:10.1145/2827872. URL: <http://doi.acm.org/10.1145/2827872>.
- [52] Z. Liu. "Chaotic Time Series Analysis". In: *Mathematical Problems in Engineering* (Feb. 2010).
- [53] F. Takens. "Detecting strange attractors in turbulence". In: *Dynamical Systems and Turbulence, Warwick 1980*. Ed. by David Rand and Lai-Sang Young. Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 366–381. ISBN:978-3-540-38945-3.
- [54] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis*. 2nd. Cambridge: Cambridge University Press, 2003. DOI:10.1017/CB09780511755798.

AUTHORS



Benedetta Picano (Member, IEEE) has been a postdoctoral research fellow in the Department of Information Engineering at the University of Florence (Italy), since 2019. She received a Ph.D. degree in Information Engineering from the Department of Information Engineering at the University of Florence, in February 2020. She

received the 1st and 2nd level Laurea Degree in Computer Networks and Computer Engineering at the University of Florence, in July 2013, and October 2016, respectively. She was a visiting researcher at the University of Houston. Her research fields include matching theory, nonlinear time series analysis, resource allocation in edge and fog computing infrastructures, and machine learning.



Romano Fantacci is a Full Professor of Computer Networks at the University of Florence, Florence, Italy, where he heads the Wireless Networks Research Lab. He is the founding director of the Information Communication Technology Inc. (TiCOM), with Leonardo spa and of the Wireless Communications Research Centre (LiRS) with TIM -Telecom Italia. He received an M.S. degree in Electrical Engineering from the University of Florence, Italy and a Ph.D. degree in Computer Networks from the University of Florence, Italy. Dr. Fantacci was elected Fellow of the IEEE in 2005 for contributions to wireless communication networks. He received several awards for his research, including the IEE Benefactor Premium, the 2002 IEEE Distinguished Contributions to Satellite Communications Award, the 2015 IEEE WTC Recognition Award, the IEEE sister society AEIT Young Research Award and the IARIA Best Paper Award, the IEEE IWCMC'16 Best Paper Award and the IEEE Globecom'16 Best Paper Award. He served as Area Editor for IEEE Trans. Wireless Commun., Associate Editor for IEEE Trans. on Comm., IEEE Trans. Wireless Comm., Area Editor for IEEE IoT Journal, Regional Editor for IET Communications and Associate Editor for several non-IEEE Technical Journals. He guest edited special issues for IEEE Journals and Magazines and served as symposium chair of several IEEE conferences, including VTC, WCNC, PIRMC, ICC and Globecom. Dr. Fantacci currently serves as a member of the Board of Governors of the IEEE sister society AEIT, and of the Steering Committee of IEEE Wireless Comm. Letters.



Adnan Rashid (IEEE and Internet Society member) is currently working as a Ph.D. Research Scholar in the Department of Information Engineering (DINFO), University of Florence, Italy, since November 2018. He received a Master's degree in Computer Engineering from the Department of Electrical Engineering, Capital University of Science & Technology (CUST), Islamabad, Pakistan, in December 2015. He received a Bachelor of Science in Computer Science (BSCS) from the Federal Urdu University of Arts, Science, and Technology (FU-FAST), Islamabad, Pakistan, 2009. He was a visiting faculty member from 2017-to-2018 in the Department of Computer Science at the PMAS-Arid Agriculture University, Rawalpindi, Pakistan. His research activity focused on the security and resilience of IoT systems, Software Defined Networking (SDN), and Fog Networks. He is involved in multiple IETF working groups and doing research to standardize the IPv6 and 6LoWPAN-ND features. He is collaborating with the open-source ns-3 simulator maintainers for the development of the 6LoWPAN-ND protocol.



Tommaso Pecorella (Senior member, IEEE) received Ph.D. and M.Sc. degrees in Electronic Engineering (Telecommunications track) from the Department of Information Engineering at University of Florence (Italy) in 2000 and 1996 respectively. From 2001 to 2007 he was a researcher at Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT). Since November 2007 he is a tenure-track Assistant Professor in the Department of Information Engineering at University of Florence (Italy). In 2018 and 2019 he was also visiting professor at University of Saint Louis, Missouri (USA). He received the Best paper award at the IEEE GLOBECOM 2016, and in 2021 got the Italian Habilitation (Abilitazione Scientifica Nazionale) for Associate Professorship in Telecommunication Engineering. He is the author of more than 90 publications between conference papers and journals. His research interests focus on IoT communication systems, network security, and application of machine learning to networking systems.