

Efficient Task Offloading and Resource Allocation in an Intelligent UAV-MEC System

Benedetta Picano and Romano Fantacci

Abstract—Nowadays, the functional integration of digital twin (DT) technology and artificial intelligence (AI) methodologies has enabled reliable predictions of many random processes, supporting efficient control and optimization procedures. In line with this trend, this paper explores the joint use of these technologies in an AI-empowered DT framework for an unmanned aerial vehicle-aided multi-access edge computing (UAV-MEC) system. Specifically, this approach defines an intelligent UAV-MEC system capable of significantly improving service quality and deployment flexibility. The focus is on a UAV-MEC network consisting of multiple elementary service areas, where DTs efficiently orchestrate and reduce congestion levels by utilizing UAVs with onboard processing capabilities. A potential architecture for the DTs is outlined, conceptualizing each DT as a collection of basic cyber entities. Additionally, a suitable framework utilizing a matching game approach is proposed to effectively manage task offloading, channel allocation, and the dynamic assignment of UAV support to congested service zones within the same area. Finally, comprehensive simulation results validate the efficacy of the proposed intelligent UAV-MEC system, as indicated by metrics such as task completion delay and accuracy in congestion prediction.

Index Terms—Digital twin, intelligence system, machine learning, mobile edge computing, unmanned aerial vehicle.

I. INTRODUCTION

UPCOMING sixth-generation (6G) networks are expected to trigger an ever-increasing demand for computation-intensive and latency-critical applications, i.e., virtual and augmented reality, online gaming, etc., posing new challenges in terms of both runtime monitoring and execution [1]. Within this context, integrated ground-air networks (IGANs) have emerged as a promising networking solution, capable of satisfying different service requests, anywhere and anytime, allowing wide-area reliable coverage. IGANs are usually characterized by suitable cooperative schemes between unmanned aerial vehicles (UAVs) and other ground mobile devices or network infrastructure nodes, i.e., edge computing nodes (ECs) or cloud. This allows to boost network performance, leading improvements on both network behavior and users' quality of experience [2], [3]. Integrating UAVs, equipped with computing capabilities, into a ground multiaccess edge computing (MEC) network offers numerous benefits. These include

Manuscript received April 6, 2024; revised September 2, 2024; approved for publication by Qiao, Daji Division 2 Editor, September 10, 2024.

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGeneration EU, partnership on "Telecommunications of the Future" (PE0000001 - program "RESTART").

B. Picano and R. Fantacci are with University of Florence, 50139 Firenze, Italy, email: {benedetta.picano, romano.fantacci}@unifi.it.

R. Fantacci is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2024.000050

enhanced deployment flexibility across various scenarios to meet real-time demands, as well as offering efficient computation offloading solutions in areas where network infrastructure is either overloaded or experiencing failures.

Typical UAV-MEC environments comprise a collection of ground EC nodes and a set of UAVs, equipped with computational and storage capabilities, strategically positioned to facilitate computations. Following the MEC paradigm, computational capabilities are decentralized to the network edges, such as wireless network small base stations (SBSs), thereby reducing latency and network congestion commonly associated with previous cloud-based solutions. Based on quality of service (QoS) constraints set by applications, tasks initiated by end devices (EDs) can be offloaded onto the UAV-MEC system according to specific rules. In such a context, an effective radio resource management is then essential to ensure efficient service provision and minimize channel interference, fostering beneficial collaboration between UAVs and ground MEC infrastructure.

This paper deals with an intelligent UAV-MEC system that pursues the functional integration of the digital twin (DT) technology with artificial intelligence (AI) capabilities in order to allow for significantly improving both the flexibility of its deployment and the performance achieved. In this context, the DT technology represents the next frontier in simulation, seamlessly integrating cyber and physical realities to create virtual models of physical objects in the digital environment [4]. This approach significantly enhances data-driven optimization of system and service planning, enabling unprecedented levels of performance and reliability through real-time monitoring, prediction, estimation, analysis of dynamic changes, and interaction between physical objects and their digital counterparts.

Furthermore, AI serves as a potent tool for facilitating data-driven decision-making, leveraging its capacity for self-learning to continuously enhance its intelligence through data analytics. This capability enables the analysis of information gathered from real systems, uncovering, interpreting, and extracting valuable insights from data. The aim of this paper is to highlight how the proposed UAV-MEC intelligent system enhances the effectiveness of task offloading, thereby reducing network congestion and ultimately improving performance¹.

The focus is on a 6G based networks partitioned into elementary UAV-MEC service areas consisting of a number

¹DTs may be also utilized to guarantee privacy and security issues as described in [5], by detecting intrusions and anomalies in UAV-MEC systems. However, an in-depth discussion of this topic is beyond the scope of this paper and is therefore left for further study.

of SBSs, each of which associated² with an EC according to the edge computing paradigm. With the goal of lowering the occasional congestion of a given SBS service area, an UAV, equipped with computational capabilities, can be allocated according to a suitable procedure. In particular, in our case, we have that the involvement of the DT directly drives suitable decision-making procedures based on the matching theory (MT) [6]–[8] that, in addition to a proper access channels assignment to lower users mutual interference, enables a suitable UAV assignment procedure to congested ECs .

The salient contributions of the paper, which has the ambition to be a computer simulation study in accordance with the state-of-the-art classification presented in [9], can be summarized as follows

- Proposal and analysis of a novel intelligent UAV-MEC system. The proposed framework aims at improving the congestion control and the offloading decision making policy over time, exploiting the functional integration between DT and a data-driven AI approach. In particular, a DT architectural scheme based on suitable functional entities is discussed in compliance with emerging standard ISO 23247 [10];
- Proposal of a sub-optimal optimization strategy, based on a suitable matching game, to perform planning about EDs task offloading and UAV allocation to the SBSs within a same elementary service area. The objective is to minimize the task outage probability, i.e., the probability that an ED experiences a task completion time greater than the deadline associated therewith;
- Integration of a machine learning based module to catch the traffic behavior within each SBS service areas, learning how service requests vary over time. In order to do that, we exploited an echo state network (ESN) [7] running on each DT;
- Performance analysis of the proposed task offloading procedure, in comparison to the alternative Kolkata Paise Restaurant game [11], and the random selection approaches. In addition to this, performance results are provided to highlight the effectiveness of the proposed intelligent UAV-MEC system in terms of congestion prediction accuracy and task outage probability.

The rest of the paper is organized as follows. In Section II an in-depth review of the related literature is provided. Section III presents the system model description, and Section IV presents the proposed intelligent UAV-MEC system offloading framework. Finally, Section V deals with the performance analysis, whereas conclusions are drawn in Section VI.

II. RELATED WORKS

In the past decade, many existing works have been proposed to solve various research problems of the UAV systems [12]–[14]. Authors in [15] proposed a heuristic method to maximize the cellular coverage by optimizing drones deployment and minimizing the communication cost among UAVs. A drone-as-a-service market model has been developed

in [16], in which a service algorithm has been designed, in order to properly meet the quality requirements in terms of cost and delay, expressed by users. In [17], the main focus is the limited UAV resources. In fact, authors analyze the feasibility of overcoming these constraints by combining and controlling multiple UAVs. In this reference, the paper explores programmable crowd-powered drones to create a federated cloud. Moreover, a scripting language is applied to coordinate flight trajectories of multiple drones, as well as multi-drone service management. Differently, in [18], a mixed integer programming problem has been formulated as a traveling salesman problem. The route distortion problem has been defined, and a lower bound of the number of drones needed to solve it has been proposed. The paper [19] proposes an UAV-assisted MEC network with air-ground cooperation, in which both UAV and ground access points exhibit a direct link towards devices and cooperate to execute tasks computation, aiming at minimizing the worst delay and optimizing the resource allocation by jointly controlling UAV-device matching, UAV horizontal and vertical position, bandwidth selection, and task splitting. A two-layered decision-making framework for the cooperation between one or more stations and one or more drones is presented in [20], maximizing profit, and minimizing the travel distance. DT design and application in intelligent systems, as far as we know, still remains an emerging and unsolved research problem. Among the available literature on this subject, paper [21] discusses the main objectives and challenges presented by the DT technology, especially in the vehicular edge computing network, highlighting the key role of the DT in the vehicular mobility monitoring. Similarly, paper [22] investigates the potential, in terms of performance improvements, in applying the DT in industrial manufacturing processes, enabling rapid and agile system testing. Differently, paper [4] proposes the contextualization of the DT technology to a vehicular edge computing landscape, where offloading and scheduling decisions are performed throughout a deep reinforcement learning strategy. Specifically, a multiagent deep reinforcement learning approach has been developed to reduce learning complexity and to provide resource utilization decisions minimizing the offloading cost, and taking into account the strict delay constraints due to the nature of the vehicular scenario. The DT has been also exploited in combination with a blockchain approach in [23], where an edge node selection policy is provided involving a Markov decision process, on the basis of which a deep-Q-learning algorithm is developed. Then, authors in [24] present a framework to minimize latency providing devices transmit power optimization, user-edge node association, and task offloading decisions. Finally, paper [25] addresses the problem of the clustering to reduce energy consumption into an unmanned aircraft system. A DT-based offloading scheme is also presented in [26], where a hybrid UAV-MEC system is considered to optimize user association, UAV trajectory, transmission power distribution and computation capacity allocation. A double deep Q-network is applied to solve problems of the UAV trajectory selection and the user association. Then, an iterative algorithm is designed to solve both the transmission power distribution and the computation capacity allocation problem. The applicability

²The terms SBS, EC will be used interchangeably, hereafter.

of the DT technology within 6G communication systems, is investigated in paper [27], presenting a wide discussion about DT architectures, use cases, its ability in improving system performance and in providing parameter optimization. Furthermore, an in-depth critical analysis of the state-of-the-art literature is included, as well as a discussion about existing open issues needing solution to make DT a concrete reality. Similarly, authors in [28] detailed main aspects concerning the DT software architecture principles, where an accurate analysis of works focusing on DT data extraction, synthesis process, and so on is presented. Authors identified 14 quality attributes that are relevant for DTs, as well as the most used object oriented 10 patterns applied stand-alone or in combination with other patterns. A 6G oriented resource management framework to support stateful applications, i.e., applications needing context data, is designed in [29]. In this picture, authors exploit the DT technology to collect context data required by stateful applications. The main objective of the paper is to jointly minimize the computation usage, the storage, the communication resources and the cost of re-configuration by assuming a multi-tier computing infrastructure as network architecture. Hybrid aerial-ground solutions are also considered in [30], where UAV visual target tracking is realized performing offloading of such a deep learning tasks to an EC node, to overcome the limited computational resource and energy capacity of the UAV. Multi-task learning is adopted in paper [31], where the problem of the unstable coverage and link performance of the aerial-terrestrial communication links is mitigated by proposing a reflecting intelligent surface (RIS)-assisted transmission policy. Authors design and develop an adaptive RIS-assisted transmission protocol, where the channel estimation, the transmission policy, and the data transmission are independently implemented in a frame. Furthermore, the multi-task learning is introduced to properly produce the optimal transmission strategy in near-real-time. Differently, in [32], a multi-task resource scheduling framework exploiting the deep reinforcement learning has been designed with the objective to minimize the energy consumption of all users and UAVs in the system. Authors in [33] integrate horizontal federated learning with double deep Q-network to solve the problem of the computation offloading and relay communication in air-ground integrated networks, considering emergency scenarios. In reference to the offloading problem, the objective is the minimization of the weighted sum of both delay and energy consumption. For the data transmission, the main goal is to maximize the minimum rate of relay links. In [34] artificial intelligence capabilities are used to manage computation offloading from a UAV to terrestrial MECs. This paper addresses various alternatives and provides useful methodologies to improve performance. However, the problem addressed is different from the one of interest here, where computation offloading is necessary from MEC nodes to a UAV to mitigate the effect of congestion in the service area. Towards this direction, this paper proposes the contextualization of DTs to an UAV-MEC 6G environment. Furthermore, [34] is not focused on forecasting but instead proposes an AI-based framework to obtain an optimal decision-making policy based on current

TABLE I
LIST OF MAIN NOTATIONS.

Notation	Description
S	Number of SBSs in a service area
\mathcal{U}	Set of Edge Devices (i.e., users) in a service area
\mathcal{W}	Channel bandwidth
$\mathcal{R}_{gh}(i)$	Instantaneous available rate for the communication channel h assigned to ED i
\mathcal{R}_d	Instantaneous available rate for the communication channel for the downlink channel used by the SBS
\mathcal{R}_{uav}	Instantaneous available rate for the communication channel connecting a SBS (i.e., EC node) to the UAV and vice-versa
P	Transmission power
g_B	Boltzmann constant
T_0	Temperature in Kelvin
$K(\nu_c)$	Global absorption coefficient of the medium
ν_c	Carrier frequency
d_i	Distance between the ED i and the linked EC
c	Light speed
T_h	Aggregate power of the interfering signals at the linked SBS on channel h
f_x	EC-CPU computational capacity
ϕ_u	UAV-CPU computational capacity
CA	Central aggregator of each elementary service area

network conditions that makes clear the difference with the proposed solution. Finally, we have to say that a network infrastructure similar to that of interest here was considered in [7], [8], [26]. However, this paper has important differences from the previous ones. In particular, in respect to [7], [8], the paper deals with a more current situation that considers the impact of mutual interference among users accessing the same channel in interfering SBS service areas. This leads to a different optimization problem to be solved for which the evaluation of implementation complexity has been also provided here. Furthermore, in relation to [8], this paper, although not aiming to provide a formal definition of a DT architecture, emphasizes the functional relationships of the proposed solution with the ISO 23247 standard.

III. PROBLEM STATEMENT

As stated before, the focus here is on a network partitioned into elementary service areas, each with an equal number of SBSs/ECs and with possibility of the potential use of the dedicated UAV with onboard processing capabilities to lower the congestion at the ground ECs. To this end, a DT approach is employed, and its possible hierarchical architecture is outlined according to the ISO 23247 recommendations. Specifically, we consider the integrated UAV-MEC intelligent system comprising four layers (or domains) as sketched in Fig. 1. Further refinement of these domains has led to the identification of several entities based on their functions, as outlined below.

A. Physical layer : Observable Manufacturing Elements

In ISO 23247, physical entities, such as the EDs, SBSs and the UAV related to the given service area in our case, are referred to as observable manufacturing elements (OMEs). The digital twin tracks its OMEs by gathering pertinent operational and environmental data, as detailed in the subsequent discussion. Moreover, the OME encompasses a set of elementary

service areas that align with 6G technology, hence, operating across THz channels. Each elementary service area includes of a group of SBSs, denoted as \mathcal{S} , which could potentially interfere with each other and SBSs of adjacent elementary service areas. This approach allows the service model to scale and adapt to different scenarios being able to define service areas based on specific needs. Moreover, it is also able to manage situations involving high user mobility thanks to the interaction with the core network, which allows users to remain connected with the SBS through which the offloading of their tasks to the appropriate UAV has been carried out. Each SBS provides a set of channels to grant EDs access to the associated ECs for task offloading purposes. The SBSs in a given area are deployed according to an isotropic homogeneous Matern hard core point process (MHCPP) with intensity ι and with the constraint of a minimum distance ρ , as the distance between SBSs cannot be arbitrarily small.

Furthermore, we have assumed that each EC belonging to the same elementary service area (i.e., ground ECs) is equipped with a central processing unit (CPU) with a specific speed, denoted as f_j with $j \in \mathcal{S}$. In addition, we have considered in each service area the presence of a set \mathcal{U} of EDs (i.e., users) needing to offload intensive task computations in relation to their local capabilities. With reference to this, we would like to note that our performance analysis is dependent solely on the computational load forwarded from EDs $\in \mathcal{U}$ and the workload this entails for the ECs. In some way, this makes our analysis independent of the number of EDs that can perform local computations. This eventually only impacts the number of active EDs in a cell. With reference to this, we would like to stress that providing results based on the variable number of EDs in \mathcal{U} , allows us to indirectly include in our analysis also the actual scenario of advanced EDs capable of performing computational intensive tasks, having this impact on the number of EDs that need offloading. Furthermore, with the aim at performing a worst case analysis we have not considered the possibility of a partial task offloading for EDs that have this capability. Finally, we assumed that each EDs interested in offloading their tasks are linked to the appropriate ECs through the associated SBSs by means of individual channels. To simplify our analysis, we have assumed that the number of channels available for accessing the SBS is not less than the number of EDs interested in offloading their tasks. This assumption is justified by the hypothesis of using a 6G technology network, which will have large access capacities, making the probability of blocking new incoming requests negligible³. However, the proposed methodology can be extended to the case where the number of access channels is fewer than the offloading requests. This can be done by generalizing the proposed matching strategy to share the available channels among the task offloading requests according

³The paper essentially refers to operational conditions that require completing a task's processing within the time a user stays in the respective service area of the connected SBS. However, in scenarios with high-mobility users who have offloaded tasks to a UAV and then handed off to other SBSs, the computation result is still sent to the original SBS and then, through the core network, to the SBS connected to the interested user. In this case, the additional communication delay for the end-to-end delay, considering that it typically involves fiber optic links, is negligible.

to a suitable policy, such as prioritizing tasks with tighter deadlines. Once the transfer of the first block of requests to the appropriate ECs is completed, the allocation of the remaining requests is performed using the same methodology by reusing the same block of channels. This process continues until all offloading is completed. The download of the computation results, which is executed according to the FIFO policy at each EC, can be managed without penalty by assigning the associated SBS the best available channel at that moment, as suggested by the associated DT. In this case, it is easy to foresee a degradation in performance with respect to the ideal case considered in our analysis in terms of outage probability. However, its impact can be limited or even made negligible, thanks to the reduced communication times allowed in 6G networks.

With the aim of simplifying the discussion, all the communication channels in the network are assumed having the same bandwidth \mathcal{W} . In accordance with [35]–[37], we assumed that the instantaneous available rate, $\mathcal{R}_{gh}(i)$ for the communication channel h assigned to the given ED i to communicate to the appropriate SBS (uplink) and vice-versa (downlink), is given by [7], [35], [36], [38]

$$\mathcal{R}_{gh}(i) = \mathcal{W} \log_2 \left(1 + \frac{PA_0 d_i^{-2} e^{-K(\nu_c) d_i}}{g_B T_0 + PA_0 d_i^{-2} (1 - e^{-K(\nu_c) d_i}) + \mathcal{I}_h} \right), \quad (1)$$

where P is the transmission power, assumed the same for all the situations considered here. Moreover, in (1) we have that g_B is the Boltzmann constant, T_0 the temperature in Kelvin, $K(\nu_c)$ the global absorption coefficient of the medium, $A_0 = (\frac{c}{4\pi\nu_c})^2$ [38], with ν_c the carrier frequency, d_i the distance between the ED i and the linked SBS/EC and, c is the speed of the light. Finally, the term \mathcal{I}_h in (1), representing the aggregate power of the interfering signals at the linked SBS due to the use of the same channel h by EDs linked to the other $S_{\mathcal{I}_h}$ interfering SBSs at a distance d_γ , irrespective of the elementary service area, is given by

$$\mathcal{I}_h = \sum_{\gamma=1, \gamma \neq i}^{S_{\mathcal{I}_h}} PA_0 d_\gamma^{-2} u_{\gamma,h}, \quad (2)$$

with the term $u_{\gamma,h}$ having value 1 if the channel h is allocated to an ED at the interfering SBS γ , and 0, otherwise. Note that values of terms $u_{\gamma,h}$ result from a certain rule, i. e., the channel allocation strategy proposed in Section IV.

In addition to this, we have to account for the task computation time at a ground EC. In particular, we have for a generic task i , the computation time on the (ground) EC j results in

$$t_{i,j} = \frac{s_i}{f_j} + \omega_j, \quad (3)$$

where s_i is the number of CPU cycle employed by task i to be computed, f_j is expressed as the number of CPU cycles per unit of time, and ω_j represents the queuing time on EC j , i.e., the time spent by task i waiting for receiving computation. Such a time is due to the presence of tasks previously offloaded on the same computation site by the other users.

Alternatively, when a task, i.e., task i , is offloaded to the UAV by the linked SBS, we have assumed that the instantaneous available rate, \mathcal{R}_{uav} for the communication channels

connecting a SBS (i.e., EC node) to the UAV and vice-versa are the same, and, specifically, given by :

$$\mathcal{R}_{uav} = \mathcal{W} \log_2 \left(1 + \frac{PA_0 d_{uav}^{-2} e^{-K(\nu_c) d_{uav}}}{g_B T_0 + PA_0 d_{uav}^{-2} (1 - e^{-K(\nu_c) d_{uav}})} \right), \quad (4)$$

where P , as in (1), is the power used in transmission, while d_{uav} is the distance, almost surely constant, between the UAV and any SBS. Please note that in (4), we have neglected the mutual interference effects. This assumption is based on the pursuit of an appropriate channel allocation policy (i.e., avoiding to allocate a same channel to link SBSs to the proper UAV in adjacent elementary service areas)⁴ and taking into account the attenuation effect due to the d_{uav} value. Furthermore, we have assumed the UAV equipped by an embedded CPU with computational capacity ϕ_u , less powerful than those residing on ground ECs [39] due to the weight and severe battery constraints. Hence, the corresponding computation time of a task i , offloaded on the UAV, i.e., $t_{i,u}$, is given by

$$t_{i,u} = \frac{s_i}{\phi_u} + \omega_{u,i}, \quad (5)$$

where $\omega_{u,i}$ represents the queuing time that task i has to wait on the UAV before achieving computation due to the service completion of the tasks previously offloaded to the UAV.

In particular, in the intelligent UAV-MEC system under consideration, an ED i needing task processing sends its task to the linked SBS. The task can receive service at the EC node, or if it is congested and the UAV has been allocated, onboard the UAV itself if this results in a more suitable choice according to the considered offloading procedure (see Section IV).

From above, it follows that the task completion time, i.e., the end-to-end delay defined as the total time spent by the user in submitting a task computation request and receiving back the corresponding outcome, under the assumption of the assignment of a given channel h to be linked to the local SBS, results as in (6), where $\beta_{i,j} = 1$ when task i is offloaded on EC j , and $\beta_{i,j} = 0$ when the computation is performed onboard the UAV. In particular, we assume that a task i is in outage if T_i exceeds its deadline δ_i , i.e., $T_i \geq \delta_i$.

B. Digital Twin Reference Architecture

According to the ISO 23247 standard the DT architecture is based on an entity reference model that mainly encompasses, in addition to the OME layer discussed before, three additional layers: i) Device communication entity (DCE); ii) Digital twin entity (DE), and iii) user entity (UE). The functional view of the DT architecture assumed for the intelligent UAV-MEC system under consideration is portrayed in Fig. 1. In our scenario, each SBS is linked to an individual DT, whose execution is managed by computation facilities within the corresponding EC, as described in [40]. The objective is to maintain its state consistent with the actual conditions of the service area of interest⁵. This includes factors such as the

⁴Note that two or more SBSs cannot be connected to the UAV within the same elementary service area simultaneously.

⁵In compliance with literature data are assumed collected real-time and automatically by the physical layer.

number of connected EDs, related number of computation requests, level of interference across all SBS channels, ECs workload, and other relevant parameters.

The DT, replicating and simulating the surrounding SBS service area based on periodically gathered data from the physical layer, is assumed to have a functional architecture compliant with the emerging ISO 23247 standard. This ensures that the proposed scheme maintains logical and functional compatibility with devices implementing ISO 23247, irrespective of third-party hardware manufacturers. The main entities, along with their related sub-entities and functionalities (Fig. 1), are as follows:

- **Device communication entity.** This entity comprises two sub-entities: the *Data Collection* sub-entity and the *Device Control* sub-entity. The former is responsible for collecting data from the OME concerning the availability of access channels and their interference level while the latter sends commands to implement an adequate access channels allocation and tasks to be computed based on the appropriate procedures described below.,
- **User entity.** makes use the service provided by the Digital Twin entity and host the channel and tasks allocation procedures outlined in Section IV. In addition to this, it has in charge to interact with the Control Aggregator (CA) of the elementary service area to support the selection of the SBS that, in relation to the predicted level of congestion, has the greatest need for UAV support.
- **Digital twin entity.** The DTE consists of six sub-entity blocks. The first is the *Operation* sub-entity, which is responsible for maintaining the DT operational and aligned with the current state of the OME. Additionally, it provides all the information to be passed to the UE in order to properly perform all the procedures in charge to this entity listed before. The *Application and Service* sub-entity refers to the functional entities performing prediction, simulation, and analysis of the associated SBS service area behavior. In particular, this sub-entity is empowered by a local eco state network (ESN), enabling it to monitor the congestion level of the physical service area corresponding to a given EC, i.e., EC j , in terms of the predicted number of computation requests, denoted as Ω_j . It's important to note that the DT has the capability to collect data over time through its data collection sub-entity. Once the data gathering process is complete, this sub-entity is also responsible for preprocessing and cleaning the data before passing it to the AI module via the device control sub-entity for training the ESN, as described in Section IV-B. Specifically, each DT possesses its own time-series dataset, denoted as \mathcal{D}_j , in the data collection entity, containing historical data about the individual EC congestion level Ω_j . *Resource access and interchange* enables the integration and the dynamical connection between DTs and the physical counterpart.

$$T_i = \left(\frac{2s_i}{\mathcal{R}_{gh}(i)} + t_{i,j} \right) \beta_{i,j} + \left(\frac{2s_i}{\mathcal{R}_{gh}(i)} + t_{i,u} + 2 \cdot \frac{s_i}{\mathcal{R}_{uav}} \right) (1 - \beta_{i,j}) \quad (6)$$

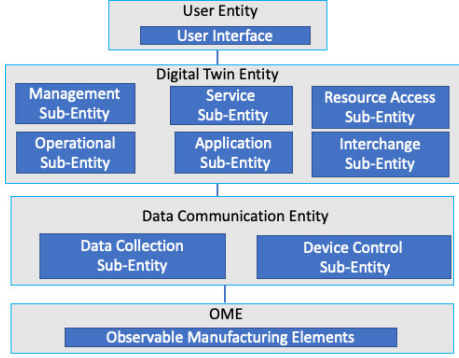


Fig. 1. Four layers of domains according to ISO 23247 standard.

IV. INTELLIGENCE UAV-MEC OFFLOADING FRAMEWORK

From a software standpoint, the DT can be considered as a collection of cyber-entities, each mirroring the digital representation of its real-world counterpart. Essentially, physical elements like EDs channels, processors, and so forth, form the complete set of features defining the operational domain linked with each EC j . By employing this method, the DT related to the EC j can gather data about the EC j service area to develop an approximate model as:

$$\Psi_j^t(f_j, \Omega_j, \mathcal{M}_j, \chi_S, \omega_j), \quad (7)$$

where f_j represents the computational capability of the EC j , expressed in terms of CPU frequency, and \mathcal{M}_j is a vector with a number of elements equal to the number of the available channels that provides a virtual model mirroring channel interference level in the EC (SBS) j service area, to trigger an efficient channel allocation procedure as detailed in Section IV. The χ_S term refers to a channel map that, for each SBS in \mathcal{S} , keeps updated the list of available channels and their interference level by gathering periodical measurements performed by the interfering SBSs themselves.

A. Problem Formulation

Considering an interval time Γ_t , the aim of this paper is to provide an optimized planning over Γ_t thorough proper tasks offloading strategy (selecting computation site between the ECs and the UAV in a specific elementary service area) and the channel allocation policy to link EDs to the SBSs in the same area. In particular, the main objective of the considered intelligent system is the minimization of the task outage probability, i.e., the number of EDs requests in outage in any elementary service area, defined as

$$\mathcal{G}(\Gamma_t) = \{i \in \mathcal{U} | T_i \geq \delta_i\}. \quad (8)$$

Hence, our problem can be formulated as

$$\min_{\mathbf{B}, \mathbf{X}, \mathbf{H}, \mathbf{r}} |\mathcal{G}^{\Gamma_t}|, \quad (9)$$

$$s.t. \quad \sum_{j=1}^{|\mathcal{S}|} \beta_{i,j} + x_{i,j} = 1, \forall i \in \mathcal{U} \quad (10)$$

$$\sum_{j=1}^{|\mathcal{C}|} u_{i,h} = 1, \forall i \in \mathcal{U} \quad (11)$$

$$\sum_{j=1}^{|\mathcal{S}|} r_j \leq 1, \quad (12)$$

denoting with $\mathbf{B} \in \{0, 1\}^{|\mathcal{U}| \times |\mathcal{S}|}$, where the operator $|\cdot|$ expresses the cardinality of the set, i.e., the number of elements belonging to the set within the operator. Λ is the edge offloading matrix, whose generic element is 1 if user i offloads task on EC j , zero, otherwise. Then, matrix $\mathbf{X} \in \{0, 1\}^{|\mathcal{U}| \times |\mathcal{S}|}$ represents the UAV selection matrix, where the element $x_{i,j} = 1$ if task i is offloaded on the UAV through EC j , zero otherwise with $|\mathcal{C}|$ be the number of available channels. The matrices $\mathbf{H} \in \{0, 1\}^{|\mathcal{U}| \times |\mathcal{C}|}$ is the channel allocation matrix, respectively, and, as before, $u_{i,h} = 1$ if user i is transmitting over channel h , zero otherwise. The vector of the UAV position is given by $\mathbf{r} \in \{0, 1\}^{|\mathcal{S}|}$, and its generic element r_j is 1 if the UAV is assigned to the j th SBS in the proper elementary service area⁶. Constraint (10) expresses that each task can be offloaded on only the tagged EC or the UAV. Similarly, constraint (11) points out that each user can be allocated on only one channel. Constraint (12) claims that the UAV can support in computation only one EC j within its operational area, during each Γ_t time period. It is important to highlight here that the proposed solution does not aim to achieve a global optimum, which, even if achievable, would be challenging to define. Specifically, the proposed algorithm embodies a distributed and sub-optimal strategy due to the established matching game involving externalities, as better detailed in what follows.

To make the proposed decision-making strategy truly effective we have to identify, thanks to the local DT, the most congested SBS within a given elementary service area, and, to perform an efficient offloading strategy from the associated EC and the temporary allocated UAV. The goodness of the approach outlined here will be validated later by providing performance comparisons with some possible alternatives.

In particular, (9) entails defining the UAV position vector \mathbf{r} , the offloading and the selection matrices \mathbf{B} , \mathbf{X} , and, finally, the channel allocation matrix \mathbf{H} .

With the aim at providing an effective decision-making strategy, as detailed in Section IV-B, the traffic congestion in a given elementary service area is monitored by the local control aggregator (CA) on the basis of interactions with the

⁶Optimal UAV flight paths to switch to from a given service area to a new one are assumed to be preloaded in order to lower the energy consumption (e.g., according to the procedure outlined in [3]).

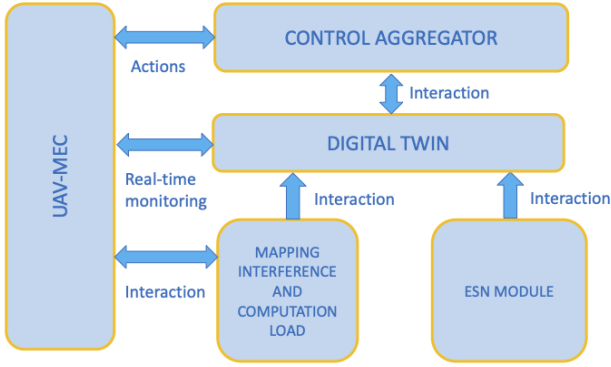


Fig. 2. Functional interaction among entities.

DTs related to the SBSs/ECs service area. In particular, this encompass to make use of the prediction provided to the DTs by the local ESNs in order to provide to the CA reliable predictions about congestion level of each EC over time. As a consequence, the CA can take the most proper decisions about the EC to be supported by the UAV to reduce its congestion. Furthermore, the local DT enables the definition and updating of the matrices \mathbf{B} , \mathbf{X} , and, \mathbf{H} according to (9)–(12) as outlined in Section IV-C.

B. AI Empowered DTs Congestion Monitoring

In the intelligent UAV-MEC system under consideration, each SBS is augmented with AI capabilities through a local ESN. As previously mentioned, we assume that the training phase of ESNs relies on available data collected by monitoring congestion behavior over a suitable observation interval where UAV support is absent. The functional interplay among the relevant entities is depicted in Fig. 2.

As for the ESN, we can say that it embodies a specific implementation of the broader Reservoir Computing paradigm. It inherits advantages from recurrent neural networks (RNNs), particularly their proficiency in handling inputs with temporal dependencies [41]. The key components of the ESN, as outlined in [41], are as follows:

- Neurons randomly connected;
- Sparse connection links;
- Large number of neurons;
- Low in energy and time demand.

The updating rule for the reservoir weight matrix follows the equation outlined in [41]. Due to its utilization of reservoir computing, a learning paradigm characterized by sparsely connected random links, ESNs are well-suited to circumvent the issue of vanishing gradients commonly encountered in training recurrent neural networks (RNNs) like the long short-term memory (LSTM) architecture. Moreover, the stochastic nature of ESNs contributes to efficient training, effectively managing the computational overhead associated with the learning phase, which often poses a bottleneck for neural network approaches.

In accordance with [41], the ESN consists of three components: the input weight matrix I , the reservoir weight matrix R , and the output weight matrix W . We denote with $\mathbf{x}^{q \times 1}$ the input vector, assuming as reservoir weight matrix updating rule the following equation [41]

$$u^{s \times 1}(q) = \tanh(\mathbf{W}_{in}^{s \times q} \mathbf{x}^{q \times 1}(q) + \mathbf{W}_r^{s \times 1}(q-1)), \quad (13)$$

where $u^{s \times 1}$ represents a vector of internal units in the reservoir part, and $\mathbf{W}_{in}^{s \times q}$ is the weights matrix associated to the connections existing between the input layer and the reservoir level. Then, the $\mathbf{W}_r^{s \times 1}(q-1)$ is the recurrent weights matrix. Denoting with $v(q)$ the output vector and $\mathbf{W}_{out}^{q \times s}$ the weight matrix associated to the connection between the reservoir and the output layer, the relationship between the reservoir and the output level can be described as

$$v(q) = \mathbf{W}_{out}^{q \times s} u^{s \times 1}(q). \quad (14)$$

Given the ESN capability to capture temporal relationships among successive samples, it leverages historical data to predict forthcoming traffic conditions for each elementary service area, quantified by the number of task computation requests originating in the respective area. Let C_n represent the critical threshold for such a number below which an area is deemed uncongested. The local CA by interactions with the DTs of its elementary service area through the appropriate UEs acquires the data provided by the local ESNs in order to identify the SBS with the highest congestion level surpassing C_n among the S in the same elementary service area. Note that future network conditions are assumed coherent with the historical behavior of the network. However, more dynamic network conditions can be predicted with the support of a foundation model such as Timesfn [42].

C. Offloading Procedure

The tasks offloading procedure outlined here is derived by resorting to the MT that is a suitable mathematical framework able to match together elements belonging to two opposite sets, by resorting to the definition of preferences lists built by players, expressing the level of achievement of each participant in being matched to each element of the opposite set and vice-versa. Consequently, the matching procedure gives rise to an effective trade-off between the preferences exhibited by players.

In this paper, a proper matching game is proposed to take the decision on the access channel assignment and computation site, i.e., EC or UAV, selection, whenever this alternative arises. In particular, when the UAV is assigned to a given SBS, we have to take decision about the most convenient task offloading alternative between the local EC or the allocated UAV. More in depth, a matching game is developed for each decision to be taken. Note that the assignment is provided for the time interval Δ_t , and it is enabled by the interaction with the local DT.

As for the assignment of EDs to access channels, the following preference lists are built.

- *Users preferences over channels:* For each user i , in a giving cell j , the channel is selected by accessing the

Algorithm 1: Channel allocation

input : set of users \mathcal{U} , set of channels \mathcal{C}
output: assignment matrix \mathbf{H}

```

1  while  $\exists i \in \mathcal{U}$  unmatched do
2      compute users preference list;
3      compute channels preference lists;
4      foreach unmatched customer  $i$  do
5          propose assignment to the most favorite
           channel;
6          foreach Channel  $h$  receiving proposals do
7              select the most favorite user  $i^*$ ;
8               $\mathbf{H}(i^*, h^*) = 1$ 
9  return  $\mathbf{H}$ 
    
```

information from the local DT regarding the available channels and their level of interference due to the simultaneous use in adjacent (i.e., interfering) cells.

- *Channel preferences over users*: For each available channel, the preferences over users are defined sorting users in ascending order, considering that for the given channel h the most preferred user i^* is that for which we have

$$i^* = \min_i \left(\sum_i \frac{s_i}{\mathcal{R}_{gh}(i)} \right). \quad (15)$$

The channel allocation algorithm, whose pseudocode is reported in Algorithm 1, is given by

- 1) Both users and channels build their own preferences list on the opposite set;
- 2) Each user proposes allocation to the most preferred channel h^* ;
- 3) Among the received users proposals, h^* accepts the most favorite i^* ;
- 4) Each channel h receiving at least one proposal, accepts its most favorite user i^* for allocation;
- 5) Repeat steps 1)-5) until all users are allocated.

Regarding the computational complexity of the proposed approach, we can focus on a worst-case scenario, wherein each round sees all users proposing association to the same channel. In this case, giving the computational complexity of the algorithm in terms of number of steps to be performed, we have that the algorithm concludes within a number of steps equivalent to the total number of users $|\mathcal{U}|$. Once the UAV is assigned to a given SBS j , and the EDs in \mathcal{U} have assigned a channel to be linked to the SBS j , a new matching game starts to take decision about the splitting requests between the local EC and the UAV. Such a matching game acts as follows

- *Users preferences over computation sites*: For each user $i \in \mathcal{U}$, the preference list on the set of computation sites is built considering the time spent to receive service exploitation from each computation alternative. Therefore, the most preferred computation site is j^* , under the assumption that channel h has been assigned to ED i , such that

$$j^* = \arg \min_{j,u} \left\{ \tau_j, \tau_{uav} \right\}, \quad (16)$$

Algorithm 2: Computation nodes selection

input : set of users \mathcal{U} , set of computation nodes \mathcal{S}
output: assignment matrix \mathbf{B}

```

1  while  $\exists i \in \mathcal{U}$  unmatched do
2      compute users preference list;
3      compute computation nodes preference lists;
4      foreach unmatched customer  $i$  do
5          propose assignment to the most favorite
           node;
6          foreach Computation node  $j$  receiving
           proposals do
7              select the most favorite user  $i^*$ ;
8               $\mathbf{B}(i^*, j^*) = 1$ 
9          ;
10 return  $\mathbf{B}$ 
    
```

where

$$\tau_j = 2 \frac{s_i}{\mathcal{R}_{gh}(i)} + t_{i,j}, \quad (17)$$

and

$$\tau_{uav} = 2 \frac{s_i}{\mathcal{R}_{gh}(i)} + t_{i,uav} + 2 \frac{s_i}{\mathcal{R}_{uav}} + \tilde{t}_{uav,i}, \quad (18)$$

where $\tilde{t}_{uav,i}$ is an additional delay contribution due to the overall transmission time of the tasks previously offloaded to the UAV.

- *Computation sites preferences over users*: In order to construct the preference list of computation sites on users, each computation evaluates the deadline associated to each user. In so doing, each computation site sorts the EDs requests in ascending order on the basis of the associated deadline δ_i . As a consequence, the most favorite user i^* results to be given by

$$i^* = \arg \min_i \delta_i. \quad (19)$$

The users-computation node association algorithm, whose pseudocode is detailed in Algorithm 2, consists of the following steps.

- 1) Both users in \mathcal{U} and computation nodes in \mathcal{S} build their own preferences list on the opposite set;
- 2) Each user proposes allocation to the most preferred computation node j^* ;
- 3) Among the received users proposals, each computation node accepts the most favorite i^* ;
- 4) Each computation node receiving at least one proposal, accepts its most favorite user i^* for allocation;
- 5) Repeat steps 1)–4) until all the users are allocated.

Also in this case, in the worst case scenario, the algorithm terminates in a number of steps equal to $|\mathcal{U}|$.

D. Practical Consideration

The whole implementation cost of the intelligent UAV-MAC system under consideration is a very challenging point, since it is the outcome of a rich combination of different factors. First of all, the complexity of the DT implementation intimately depends on the software design choices, e.g. monolithic software

architectures versus micro-service architecture. These choices not only impact on the cost, but also impose architectural constraints over the deployment of the DT software components (for example due to functional dependencies involved in the business logic of the software architecture implementing the DT). In this reference, it is important to clarify that the deep analysis of the architectural cost in realizing and deploying a DT software architecture is out of the scope of this paper in its current form. In addition, today's technological constraints poses scalability challenges to the proposed approach. In fact, due to the storage cost, transmission and computation time, a large-scale actual implementation of the framework designed is not feasible at the moment. However, it results to be affordable for small cells deployment, typical of upcoming 6G networks. More in details, the primary inputs for the DT consist of the interference power level characterizing each available channel, in order to evaluate the impact of new allocations. This means that a continuous information-exchange process is required to keep the DT network map updated with the real-world interference conditions. In addition, it subtends data exchange between This actually increases the signaling between the network layers. However, as commonly recognized in literature, it is reasonable to assume the presence of a dedicated channel for DT signaling. Furthermore, in our paper, the local DT also has to collect data concerning the congestion predictions of each service area, in order to build the dataset exploited by the local ESN to perform predictions. Since both the local DT and the ESN reside on the same EC node, we can assume that they are connected to each other via a bus. Although it is not easy to quantify precisely, it can be assumed that for small-medium scale scenario the signaling overhead is affordable. In terms of computational complexity, the main contribution is due to the presence of the ESN. However, the ESN is a lightweight RNN, efficient in terms of training time and energy consumption. In particular, as credited by literature [43], the ESN represents a recurrent neural network whose architecture exhibits several benefits as the low computation time, energy cost, and quick training. This is mainly due to the fact that connections are randomly set, and only the output layer undergoes training using regularized least squares, allowing for the exploration of numerous hyperparameter and reservoir matrix size combinations [43]. Also in this case, in the authors' opinion, the proposed solution can still be applicable in several controlled practical settings. In order to discuss the computational complexity of the proposed approach, we have to consider that the most expensive operation in both Algorithm 1 and Algorithm 2 is the preference list construction process. In Algorithm 1, the cost spent by each user for sorting the the set of channels is

$$\mathcal{O}(|\mathcal{C}| \log |\mathcal{C}|), \quad (20)$$

which, iterated for all users in \mathcal{U} becomes

$$\mathcal{O}(|\mathcal{U}| |\mathcal{C}| \log |\mathcal{C}|). \quad (21)$$

Similarly, the computational cost incurred by channels in building the preference lists is

$$\mathcal{O}(|\mathcal{C}| |\mathcal{U}| \log |\mathcal{U}|). \quad (22)$$

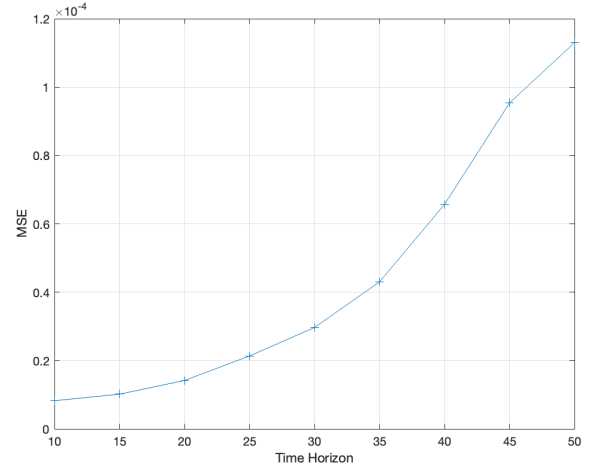


Fig. 3. ESN forecasting error as a function of the time horizon expressed in ms.

Therefore, the overall computational cost of Algorithm 1 is

$$\mathcal{O}(|\mathcal{U}| |\mathcal{C}| \log |\mathcal{C}|) + \mathcal{O}(|\mathcal{C}| |\mathcal{U}| \log |\mathcal{U}|). \quad (23)$$

Since, typically, $|\mathcal{C}| \ll |\mathcal{U}|$, the computational cost of Algorithm 1 results to be

$$\mathcal{O}(|\mathcal{C}| |\mathcal{U}| \log |\mathcal{U}|). \quad (24)$$

The same derivation can be applied to Algorithm 2, whose complexity is given by

$$\mathcal{O}(|\mathcal{S}| |\mathcal{U}| \log |\mathcal{U}|). \quad (25)$$

Consequently, the computational complexity of the proposed framework grows as

$$\mathcal{O}(|\mathcal{C}| |\mathcal{U}| \log |\mathcal{U}|) + \mathcal{O}(|\mathcal{S}| |\mathcal{U}| \log |\mathcal{U}|). \quad (26)$$

V. PERFORMANCE ANALYSIS

This section focuses on conducting a thorough performance analysis to accurately examine the behavior of the proposed intelligence UAV-MEC system. All results presented below have been derived from averaging 10^3 independent simulation runs, with system parameters modified as outlined later on. Note that the dataset used to perform the ESN training was obtained empirically by running the simulator developed, where system parameters were sets in [3], [7], and [35] and reported in Table II.

Moreover, we have considered $|\mathcal{S}| = 5$ and $|\mathcal{U}| = 150$, unless otherwise specified in the text. ECs CPUs configurations have been selected with an equal probability among five possible Intel processor cores alternatives: the Core i7, Core i5, Core i3, Pentium and Celeron, with a CPU clock rate of 3.6 GHz, 2.7 GHz, 2.4 GHz, 1.9 GHz, 2.8 GHz. For the ESNs, the hyperbolic tangent has been assumed, having as loss function the mean squared error measure. To optimize hyperparameters, we have resorted to the procedure presented

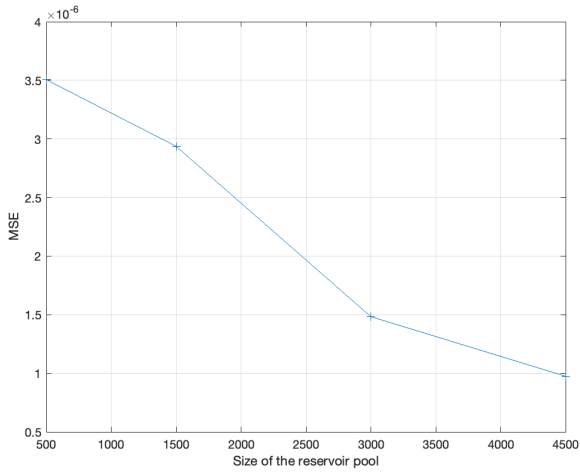


Fig. 4. MSE behavior as a function of size of the reservoir pool for a time horizon of 1 ms.

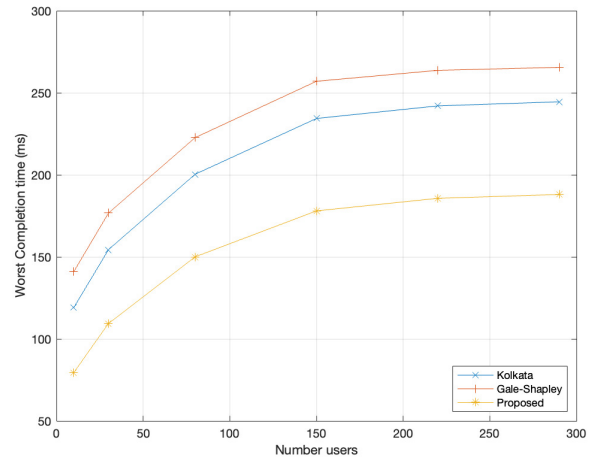


Fig. 6. Worst completion time as a function of the number of users for the system parameters values given in Table II and number of SBSs/ECs in each elementary service area equal to 5.

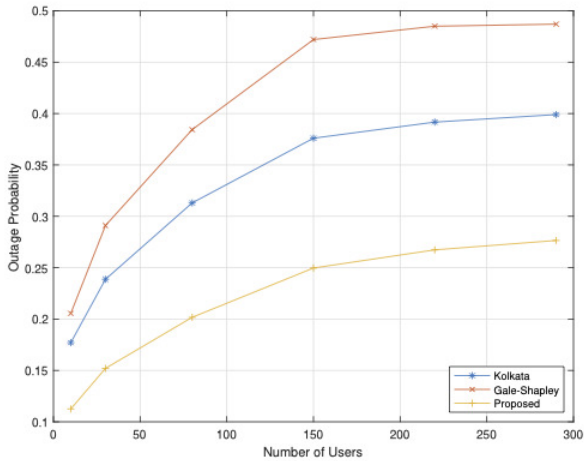


Fig. 5. Outage probability as a function of the number of users for the system parameters values given in Table II and number of SBSs/ECs in each elementary service area equal to 5.

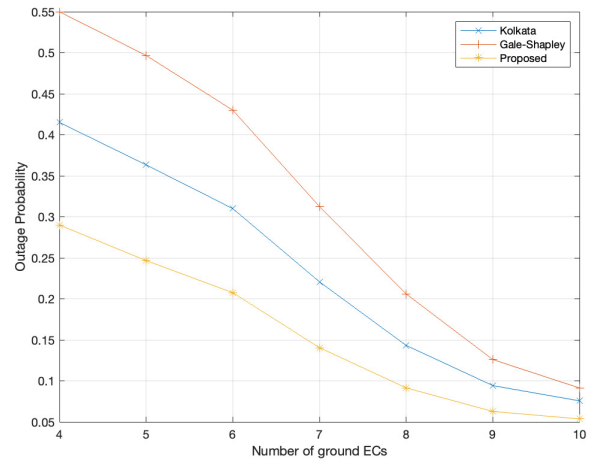


Fig. 7. Outage probability as a function of the number of ECs for the system parameters values given in Table II and number of EDs in each elementary service area equal to 150.

in [44], assuming a population of 100 components, the elite count and crossover function equal to 5 and 0.78, respectively, $q = 15$, $R \in [40, 300]$, and with a learning rate α within the interval $[0.75, 1.4]$. This permits to optimize the choice of the learning rate exploiting the genetic algorithm as in [44]. EDs are spatially distributed in accordance with a homogeneous Poisson point process distribution between 2 m and 100 m in the SBS service area of radius⁷ of 100 m. In conducting our analysis, we have made the abstraction of assuming that the UAV always has sufficient energy to complete the computation of the offloaded tasks. We can justify this assumption by considering the following methodologies [45], [46]:

⁷This is in accordance with the recent developments by Japan’s telecommunications company NTT, the Japanese mobile phone operator DOCOMO, and electronics corporations NEC and Fujitsu. Even if, the work in this filed is in progress with a current trade-off in terms of range, the recent experiment results have shown a stable connection over 100 meters for sub-TeraHertz bands .

- An appropriate management policy that includes the possibility of a second UAV. In this regard, a useful AI based methodology that allows for anticipating the UAV computation system going out of service, thereby enabling a proactive offloading of ongoing applications and ensuring continuity of service, has been proposed in [47]. This can be considered as an additional module for the *DTE - Application and Service sub-entity*;
- Appropriate supply power solutions, i.e., by resorting to the emerging technologies of lightweight solar panels or tether cable UAV that guarantee a very long energy autonomy, thus ensuring complete continuity of service.

Alternatively, if the assumed hypothesis is not applicable, one could consider modifying the offloading procedure by taking into account the residual energy capacity of the UAV when accepting tasks or by introducing an additional parameter in the definition of the matching procedure that considers the

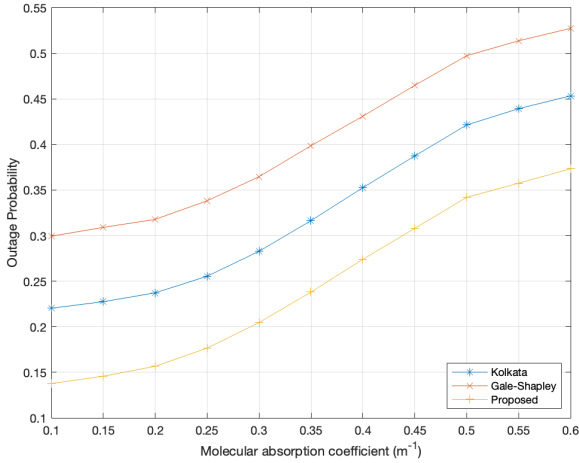


Fig. 8. Outage probability as a function of the molecular absorption coefficient for the system parameters values given in Table II, number of SBSs/ECs and EDs in each elementary service area equal to 5 and 150, respectively.

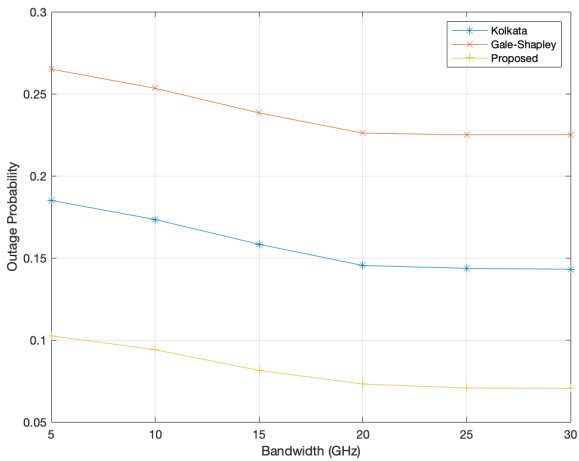


Fig. 9. Outage probability as a function of the channel bandwidth for the system parameters values given in Table II, number of SBSs/ECs and EDs in each elementary service area equal to 5 and 150, respectively.

energy consumption that the computation of a task entails in relation to the residual availability. This topic is undoubtedly interesting and, deserving an in-depth analysis, which, due to time and space constraints, it has been left as a future extension of the present work.

The efficacy of the proposed approach is assessed here by employing both the Kolkata Paise Restaurant Game algorithm (referred to as Kolkata) [11] and the random selection procedure where the SBSs within a same elementary service area are randomly selected to be supported by the assigned UAV. Furthermore, the Kolkata algorithm is regarded to function as a repeated game, where the proposing set (EDs in this scenario) generates preference lists concerning channels and computation sites using predetermined metrics for the matching process. Conversely, in the Kolkata game, the non-proposing set randomly chooses tasks offered by the proposers.

To begin, the accuracy of the ESN is depicted in Fig. 3,

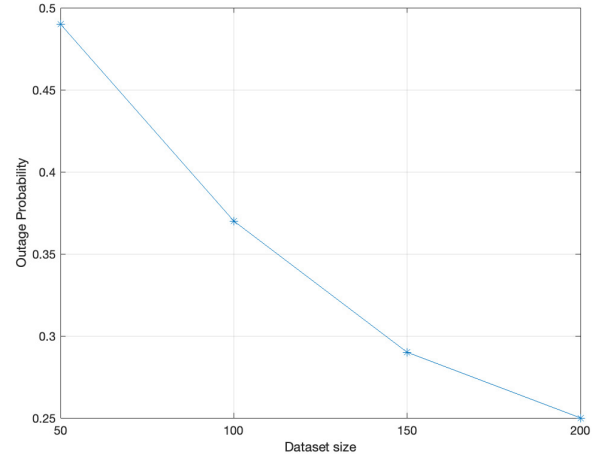


Fig. 10. Outage probability as a function of dataset size for the system parameters values given in Table II, number of SBSs/ECs and EDs in each elementary service area equal to 5 and 150, respectively.

TABLE II
SIMULATION PARAMETERS VALUES.

Parameter	Value
Carrier frequency	0.2 THz
Channel bandwidth	13 GHz
Transmission power	1 W
Service area radius	100 m
Global absorption coefficient of the medium	$2 \cdot 10^{-4} m^{-1}$
Content packet size	100 Mbits
Water vapor percentage	1 %

showcasing various prediction horizons. Fig. 3 presents the mean squared error (MSE), which is defined as

$$MSE = \frac{1}{|\mathcal{S}|} \sum_{l=1}^{|\mathcal{S}|} (\hat{\epsilon}_l - \epsilon_l)^2, \quad (27)$$

and $\hat{\epsilon}_l$ and ϵ_l are the predicted and the actual value at step l , respectively. As is often expected, increasing the prediction horizon results in higher error, mainly due to the inherent challenge of forecasting long-term behavior. This drawback arises because short-term forecasting involves fewer variables to predict. Furthermore, Fig. 4 shows the MSE behavior as a function of the size of the reservoir pool and considering as time horizon 1 ms. Intuitively, the greater the size of the reservoir containing learning units, the better the accuracy reached.

The curves depicted in Fig. 5 emphasize the notable performance of the proposed framework compared to the task outage probability values obtained by employing both the Kolkata algorithm and the Gale-Shapley alternative. The rationale behind this observation lies in the fact that the Kolkata algorithm aligns with the matching game proposed for constructing user preference lists. In contrast, unlike the proposed scheme, in the Kolkata strategy, computation sites randomly select tasks to accept from those received. This deviation from the developed matching algorithm significantly influences the performance trend, underscoring the importance of prudent task selection. Similarly, the Gale-Shapley algorithm faces a disadvantage as

it does not update preferences. This behavior is confirmed by the results provided in Fig. 6 where it appears evident that a higher mean completion time implies a greater probability of experiencing a completion time exceeding the corresponding deadline.

Fig. 7 illustrates the trend of the outage probability in the considered scenario as the number of ECs in a same elementary service area increases. As it is easy to foresee, the task outage probability decreases as the number of ECs grows. This trend arises because, with the number of task computation requests held constant at 150, increasing the number of ECs boosts computational resources, providing more support to tasks and consequently reducing outage probability. Once again, the advantageous performance of the proposed intelligent UAV- MEC system is confirmed compared to the alternative Kolkata algorithm and the Gale-Shapley scheme.

Additionally, with the aim at investigating the influence of the 6G communication channel on the performance of the intelligent system under consideration, Fig. 8 depicts the behavior of outage probability as a function of the molecular absorption coefficient. It is evident that the curve worsens as the molecular absorption coefficient increases. This deterioration occurs because channels experience greater disturbances with higher values of the molecular absorption coefficient so decreasing the possible $R_{gh}(i)$ and R_{uav} values given by (1) and (4), respectively. Furthermore, Fig. 9 illustrates the outage probability as a function of the bandwidth. As expected, the outage probability improves as the channel bandwidth increases. This Figure also demonstrates how our approach allows for spectrum saving by not showing significant improvements increasing the channel bandwidth beyond 20 GHz. Finally, Fig. 10 illustrates the behavior of the UAV-MEC system performance as a function of the computation cost that in our case as been assumed given by the dataset size using for the training of the considered ESNs. This figure shows that as the dataset size (i.e., cost in our case) increases, the performance of the proposed system improves. A dataset size of 200 is considered optimal because increasing it further does not lead to additional performance gains.

VI. CONCLUSION

The paper has dealt with the functional integration of the DT technology and AI capabilities within an intelligent UAV-MEC system, aiming to efficiently manage task offloading and lower congestion at the EC nodes. A suitable matching game is formulated to efficiently assign channels, allocate UAVs to congested SBSs, and select computation sites within the same elementary service area. The aim is to minimize the number of task computation requests experiencing outage with respect to a given deadline. Deep integration and cooperation between the physical and digital network layers are designed, outlining the DT architecture and responsibilities among involved entities, following the emerging standard ISO 23247. The provided performance analysis confirms the notable advantages arising by the functional integration of the DT technology with AI capabilities into the considered intelligent UAV-MEC system.

REFERENCES

- [1] G. Mitsis, E. E. Tsiropoulou, and S. Papavassiliou, "Price and risk awareness for data offloading decision-making in edge computing systems," *IEEE Syst. J.*, vol. 16, no. 4, pp. 6546–6557, 2022.
- [2] M. Alwateer, S. W. Loke, and N. Fernando, "Enabling drone services: Drone crowdsourcing and drone scripting," *IEEE Access*, vol. 7, pp. 110035–110049, 2019.
- [3] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 2019.
- [4] K. Zhang, J. Cao, and Y. Zhang, "Adaptive digital twin and multi-agent deep reinforcement learning for vehicular edge computing and networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1405–1413, 2022.
- [5] B. Fraser, S. Al-Rubaye, S. Aslam, and A. Tsourdos, "Enhancing the security of unmanned aerial systems using digital-twin technology and intrusion detection," in *Proc. IEEE/AIAA DASC*, 2021.
- [6] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12240–12251, 2020.
- [7] B. Picano and R. Fantacci, *Edge Intelligent Computing Systems in Different Domains*, 1st ed., ser. SpringerBriefs in Computer Science. Cham :: Springer Nature Switzerland, 2024. [Online]. Available: <https://doi.org/10.1007/978-3-031-49472-7>
- [8] B. Picano and R. Fantacci, "AI-driven digital twins for tasks offloading in 6G UAV-aided MEC networks," in *Proc. IEEE Globecom Wkshps*, 2023.
- [9] K.-J. Stol and B. Fitzgerald, "The abc of software engineering research," *ACM Trans. Softw. Eng. Methodol.*, vol. 27, no. 3, 2018.
- [10] G. Shao, (2021). *Use case scenarios for digital twin implementation based on iso 23247* [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=932269
- [11] A. S. Chakrabarti, B. K. Chakrabarti, A. Chatterjee, and M. Mitra, "The kolkata pause restaurant problem and resource utilization," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 12, pp. 2420–2426, 2009.
- [12] S. Shen, Y. Fu, and K. Yang, "Joint resource allocation in digital-twin-assisted and UAV-enabled mobile edge computing system," in *Proc. IEEE ICETCI*, 2023.
- [13] Q. Guo, F. Tang, and N. Kato, "Federated reinforcement learning-based resource allocation for D2D-aided digital twin edge networks in 6G industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 5, pp. 7228–7236, 2023.
- [14] P. Wang, N. Xu, W. Sun, G. Wang, and Y. Zhang, "Distributed incentives and digital twin for resource allocation in air-assisted Internet of vehicles," in *Proc. IEEE WCNC*, 2021.
- [15] H. Huang and A. V. Savkin, "An algorithm of efficient proactive placement of autonomous drones for maximum coverage in cellular networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 994–997, 2018.
- [16] B. Shahzaad, A. Bouguettaya, S. Mistry, and A. G. Neiat, "Composing drone-as-a-service (DaaS) for delivery," in *Proc. IEEE ICWS*, 2019.
- [17] M. Alwateer, S. W. Loke, and N. Fernando, "Enabling drone services: Drone crowdsourcing and drone scripting," *IEEE Access*, vol. 7, pp. 110035–110049, 2019.
- [18] S. Kim and I. Moon, "Traveling salesman problem with a drone station," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 1, pp. 42–52, 2019.
- [19] J. Huang, S. Xu, J. Zhang, and Y. Wu, "Resource allocation and 3D deployment of UAVs-assisted MEC network with air-ground cooperation," *Sensors*, vol. 22, no. 7, 2022.
- [20] M. Alwateer and S. W. Loke, "A two-layered task servicing model for drone services: Overview and preliminary results," in *Proc. IEEE PerCom Workshops*, 2019.
- [21] Y. Dai and Y. Zhang, "Adaptive digital twin for vehicular edge computing and networks," *J. Commun. Informat. Netw.*, vol. 7, no. 1, pp. 48–59, 2022.
- [22] Z. Gao, A. Paul, and X. Wang, "Guest editorial: Digital twinning: Integrating AI-ML and big data analytics for virtual representation," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1355–1358, 2022.
- [23] T. Liu, L. Tang, W. Wang, Q. Chen, and X. Zeng, "Digital-twin-assisted task offloading based on edge collaboration in the digital twin edge network," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1427–1444, 2022.
- [24] T. Do-Duy, D. Van Huynh, O. A. Dobre, B. Canberk, and T. Q. Duong, "Digital twin-aided intelligent offloading with edge selection in

- mobile edge computing,” *IEEE Wireless Commun. Lett.*, vol. 11, no. 4, pp. 806–810, 2022.
- [25] W. Wang, X. Li, L. Xie, H. Lv, and Z. Lv, “Unmanned aircraft system airspace structure and safety measures based on spatial digital twins,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2809–2818, 2022.
- [26] B. Li, Y. Liu, L. Tan, H. Pan, and Y. Zhang, “Digital twin assisted task offloading for aerial edge computing and networks,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10863–10877, 2022.
- [27] N. P. Kuruvatti *et al.*, “Empowering 6G communication systems with digital twin technology: A comprehensive survey,” *IEEE Access*, vol. 10, pp. 112158–112186, 2022.
- [28] E. Ferko, A. Bucaioni, and M. Behnam, “Architecting digital twins,” *IEEE Access*, vol. 10, pp. 50335–50350, 2022.
- [29] C. Zhou, J. Gao, M. Li, X. Sherman Shen, and W. Zhuang, “Digital twin-empowered network planning for multi-tier computing,” *J. Commun. Informat. Netw.*, vol. 7, no. 3, pp. 221–238, 2022.
- [30] B. Yang, X. Cao, C. Yuen, and L. Qian, “Offloading optimization in edge computing for deep-learning-enabled target tracking by Internet of UAVs,” *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9878–9893, 2021.
- [31] X. Cao *et al.*, “Reconfigurable intelligent surface-assisted aerial-terrestrial communications via multi-task learning,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3035–3050, 2021.
- [32] F. Jiang, Y. Peng, K. Wang, L. Dong, and K. Yang, “Mars: A DRL-based multi-task resource scheduling framework for UAV with IRS-assisted mobile edge computing system,” *IEEE Trans. Cloud Comput.*, vol. 11, no. 4, pp. 3700–3712, 2023.
- [33] X. Song, M. Cheng, L. Lei, and Y. Yang, “Multitask and multiobjective joint resource optimization for UAV-assisted air-ground integrated networks under emergency scenarios,” *IEEE Internet Things J.*, vol. 10, no. 23, pp. 20342–20357, 2023.
- [34] G. Fragkos, N. Kemp, E. E. Tsiropoulou, and S. Papavassiliou, “Artificial intelligence empowered UAVs data offloading in mobile edge computing,” in *Proc. IEEE ICC*, 2020.
- [35] C. Chaccour, M. N. Soorki, W. Saad, M. Bennis, and P. Popovski, “Can terahertz provide high-rate reliable low-latency communications for wireless VR?” *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9712–9729, 2022.
- [36] J. Kokkonen, J. Lehtomäki, K. Umehayashi, and M. Juntti, “Frequency and time domain channel models for nanonetworks in terahertz band,” *IEEE Trans. Antennas Propag.*, vol. 63, no. 2, pp. 678–691, 2015.
- [37] C. Han, W. Tong, and X.-W. Yao, “MA-ADM: A memory-assisted angular-division-multiplexing MAC protocol in terahertz communication networks,” *Nano Commun. Netw.*, vol. 13, pp. 51–59, 2017.
- [38] V. Petrov, D. Moltchanov, and Y. Koucheryavy, “Interference and SINR in dense terahertz networks,” in *Proc. VTC2015-Fall*, 2015.
- [39] W. Saad, M. Bennis, M. Mozaffari, and X. Lin, *Wireless Communications and Networking for Unmanned Aerial Vehicles*. Cambridge University Press, 2020. [Online]. Available: <https://books.google.it/books?id=02-nygEACAAJ>
- [40] W. Sun, H. Zhang, R. Wang, and Y. Zhang, “Reducing offloading latency for digital twin edge networks in 6G,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12240–12251, 2020.
- [41] H. Peng, C. Chen, C.-C. Lai, L.-C. Wang, and Z. Han, “A predictive on-demand placement of UAV base stations using echo state network,” in *Proc. ICC*, 2019.
- [42] A. Das, W. Kong, R. Sen, and Y. Zhou, “A decoder-only foundation model for time-series forecasting,” in *Proc. ICML*, 2024.
- [43] B. Whiteaker and P. Gerstoft, “Reducing echo state network size with controllability matrices,” *Chaos: An Interdisciplinary J. Nonlinear Sci.*, vol. 32, no. 7, p. 073116, 2022.
- [44] A. A. Ferreira and T. B. Ludermit, “Genetic algorithm for reservoir computing optimization,” in *Proc. IEEE IJCNN*, 2009.
- [45] J. Zhang, J. F. Campbell, D. C. Sweeney II, and A. C. Hupman, “Energy consumption models for delivery drones: A comparison and assessment,” *Transp. Research Part D: Transp. Environment*, vol. 90, p. 102668, 2021.
- [46] A. Ali and M. O. Hasna, “Energy harvesting schemes for UAV based communications,” in *Proc. IEEE CCNC*, 2019.
- [47] M. Catelani, L. Ciani, R. Fantacci, G. Patrizi, and B. Picano, “Remaining useful life estimation for prognostics of lithium-ion batteries based on recurrent neural network,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.



Benedetta Picano (Member, IEEE) received the M.S. degree in Computer Engineering at the University of Florence, Italy, in 2016, and the Ph.D. degree in Information Engineering at the University of Florence, Italy, in 2020. She was a Visiting Scholar at the Houston University, Houston, USA. Presently, she is a Visiting Researcher at the University of Pisa, Italy. Her research interests include computer networking, wireless communications, and machine learning application, with emphasis on system performance optimization and generation of multimodal

trajectories with generative flow networks. Dr. Picano served on the Editorial Boards of the IEEE Transactions on Vehicular Technology, Peer-to-peer Networking and Applications journal, and she was Guest Editor for the Special Issue: Applications of Internet of Things Networks in 5G and Beyond, in Sensors. Finally, she was Workshop Program Chair for ISSRE 2023, and Co-chair of Track 11: Wireless Networks: Protocols, Security and Services, for IEEE Vehicular Technology Conference 2024.



Romano Fantacci (LF23) is a Full Professor of Computer Networks at the University of Florence, Florence, Italy. He was elected Fellow of the IEEE in 2005 for contributions to wireless communication networks. His research focuses on wireless communication networks, Edge intelligent networks, networks modeling and analysis. He has received several awards in recognition of his research contributions. These awards include 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 Best Paper awards at IEEE international conferences. He has actively participated in the Organization and Technical Program Committees of numerous IEEE international conferences. Additionally, he is a Member of the Editorial Board for IEEE COMSOC Technical Journals. Currently, he is an IEEE Life Fellow and holds positions on the Steering Committee of IEEE Wireless Letters and the IEEE Fellows Committee.