Computer Communications 253 (2026) 108526

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

The human digital twin for service management: Architecture and user modeling☆

Matteo Fratta, Alessandro Floris, Simone Porcu *, Luigi Atzori

DIEE, University of Cagliari, 09123 Cagliari, Italy
CNIT, University of Cagliari, 09123 Cagliari, Italy

ARTICLE INFO

Keywords:
Human digital twin
HDT architecture
Quality of experience
Human emotions
Digital user representation

ABSTRACT

Human Digital Twins (HDTs) are increasingly adopted across various domains, yet their application to network and service management remains limited. Nevertheless, HDTs offer significant potential for optimizing service configurations based on human behavior, preferences, and profiles. In this paper, we analyze the role of HDTs in network and service management, identifying key functionalities such as collaborative learning for user modeling, Quality of Experience (QoE) and emotion prediction, application personalization, and behavioral forecasting for network "what-if" analysis. We propose an architectural framework designed to monitor user status and generate a corresponding digital replica that interacts with other network components to enhance service delivery. Our solution integrates collaborative learning for QoE modeling and applies it to service optimization. By aggregating user data from multiple HDTs, the approach improves prediction accuracy and resource optimization. Extensive performance evaluations demonstrate that the proposed collaborative HDT framework enhances the final utility function that considers perceived quality and resource usage by 27% compared to non-collaborative methods.

1. Introduction

The Digital twin (DT) is an accurate digital representation of a physical entity, which has been used in diverse research sectors not only to replicate system's detailed operations virtually but also to analyze the current condition, predict the future behavior based on past observation, and enhance control optimization. Adopting a DT in human-centered applications and systems creates a new concept known as the Human DT (HDT), which possesses unique characteristics compared to DT, such as dealing with human physiological states and behaviors [1]. This technology is revolutionizing various fields, particularly healthcare, workplace optimization, sports performance, and personalized digital experiences. By continuously monitoring an individual's health metrics, an HDT can help in early disease detection, offering proactive intervention before symptoms manifest. Doctors can use these digital twins to simulate surgeries and predict patient responses to treatments, making medicine more personalized and effective [2,3]. In workplace environments, HDTs contribute to increased well-being and productivity. They can analyze an employee's posture, fatigue levels, and stress to suggest ergonomic improvements or work schedule optimizations [4]. This helps prevent workplace injuries, improves job satisfaction, and enhances overall efficiency. In the realm of sports and human performance, HDTs are transforming athlete training and rehabilitation [5]. By replicating an athlete's biomechanics, they help optimize training routines, prevent injuries, and aid in post-injury recovery by tracking rehabilitation progress. Beyond physical applications, HDTs also enhance personalized digital experiences. They can adapt user interfaces based on an individual's emotional state, cognitive load, or fatigue, making digital interactions more intuitive. Despite some challenges related to privacy and ethics, Human Digital Twins represent a transformative leap toward a more personalized, data-driven future in multiple industries.

Whereas many are the domains where the HDTs are effectively adopted, its usage in the context of network and service management is very limited. Still, the offered advantages are many as by leveraging real-time data, artificial intelligence, and digital replicas of users, HDTs can significantly improve how networks adapt to human behaviors, preferences, and requirements. One of the primary benefits of HDT in network service optimization is its ability to provide personalized network experiences. By modeling an individual's usage patterns, mobility, and application preferences, a digital twin can help dynamically allocate bandwidth, prioritize network traffic, and adjust service quality based on real-time demand [6]. For example, a digital twin of a

☆ This article is part of a Special issue entitled: 'DT Orchestration' published in Computer Communications.

* Corresponding author.
E-mail address: simone.porcu@unica.it (S. Porcu).

https://doi.org/10.1016/j.comcom.2026.108526
Received 29 October 2025; Received in revised form 20 February 2026; Accepted 9 April 2026
Available online 14 April 2026

remote worker could predict peak usage times and optimize network resources to ensure seamless video conferencing without interruptions. Additionally, affective computing research studies have demonstrated that the human emotional state (which can be provided by a personalized HDT) can provide precious insights related to the user's satisfaction with the service and therefore with the QoE. In particular, the emotional and behavioral assessments provide deeper insight into how users perceive quality and react to digital services, especially in domains like multimedia consumption, virtual environments, and human–computer interactions [7–9]. HDTs also play a crucial role in Quality of Experience (QoE) prediction and enhancement. Since digital twins continuously monitor human interaction with the services and network, they can anticipate potential service degradations and proactively suggest or implement adjustments [10]. For instance, if an HDT detects that a user is about to start a high-bandwidth task, such as streaming or cloud gaming, it can preemptively optimize the network connection, reducing latency and buffering. From a network operations and maintenance perspective, HDTs can significantly improve efficiency [11]. Furthermore, HDTs are instrumental in 5G and beyond-5G networks, where Network Digital Twins (NDTs) are adopted and require highly adaptive and intelligent management. With digital twins of both users and network components, providers can dynamically reconfigure network slices to meet the needs of different applications, such as IoT, augmented reality (AR), or autonomous vehicles.

For these reasons, in this paper, we analyze the role of the Human Digital Twin in network and service management and we identify the major functionalities, i.e., collaborative learning for user modeling, QoE and emotion prediction, support in personalization of applications, prediction of user behavior for network what if analysis. We then propose an architectural solution which is intended to observe the user status to realize the relevant digital replica which collaborates with other network components for service enhancement. Specifically, the paper contributions are the following:

- We conducted a review of the literature studies proposing HDT architectures to identify and analyze the current key gaps, essential components and functionalities in the development of a fully functional HDT capable of taking a key role in the management of communication services.
- We proposed a fully functional HDT architecture capable of unobtrusively evaluating the user status in terms of emotional states, perceived QoE and activities, leveraging this data to make informed decisions that optimize service allocation by delivering tailored services.
- We provide a solution for collaborative learning in QoE modeling and its application to service optimization. This approach allows for improving the prediction accuracy putting together information on users obtained by different digital twins.
- We implemented a case study focusing on quality management for WebRTC-based audio-visual communications. The role of the HDT in service management has been highlighted and the performance of different approaches for QoE modeling have been analyzed. It results that the use of an HDT collaborative learning approach allows for improving the final utility function by 27% with respect to a non-collaborative approach.

The paper is structured as follows. Section 2 discusses the related work on HDT architecture and highlights the research challenges in this area. Section 3 discusses the different functionalities of the HDT in service management. Section 4 presents the proposed HDT architecture including the main components and functionalities. Section 5 presents a collaborative learning approach for QoE prediction. Section 6 extensively evaluates the performance of the proposed HDT for quality monitoring and quality control in WebRTC-based audio-video communications. Finally, Section 7 concludes the paper.

2. Related work

This section thoroughly examines the existing literature on HDT architecture by delving into key studies and findings that have shaped the understanding of this field. Following this review, we identify and discuss the research gaps from past related works, including HDT essentials, relevance and potential implications for future research.

2.1. Past works on human digital twin solutions

The concept of HDT has been frequently applied to various application domains, but only to a limited extent to the network and service management domain. In the following Section, we briefly review the major solutions that have been proposed so far to identify major features that could be also brought to our reference domain.

An early example of DT systems is presented in [12], where a virtual replica of an industrial collaborative workspace is continuously synchronized with its physical counterpart. The framework models both the robotic processes and the human operator, enabling safety monitoring and adaptive decision support through ML models, trained using either real observations and simulated ones. In this sense, the DT captures human behavior within the operational context, and supports predictive actions based on the state of the user. Building on this perspective, more domain-specific solutions have been proposed to exploit digital representations of users in service-oriented scenarios. Building on this perspective, more domain-specific solutions have been proposed to exploit digital representations of users in service-oriented scenarios.

Following this direction, [4] proposes an innovative HDT architecture tailored for operators in production and intralogistics environments. This architecture centralizes human-related data and models to enable real-time prediction of operator intentions and movements, enhancing human–machine collaboration. In the demonstrated case, the HDT of a maintenance technician uses position data to forecast motion trajectories and intended actions. These predictions provide insight into the operator's current and future state, allowing Automated Guided Vehicles (AGVs) to anticipate movements and avoid collisions, while workstations can preemptively initiate maintenance routines. The study highlights the need for standardized reference architectures to improve interoperability, while acknowledging challenges such as manual data entry and the protection of sensitive personal data.

Extending the application of HDTs beyond industrial environments, [13] explores their use in military training and mission readiness assessment. By integrating physiological measurements, behavioral observations, and cognitive indicators, the authors develop a digital representation of the personnel, which is then used to simulate responses to operational scenarios and environmental stress. Through this approach, the HDT enables predictive evaluation of fatigue, performance limitations, and coordination capability, supporting planning and real-time decision processes.

In a different application context, [14] presents a DT framework for adaptive learning environments, where the learner is represented through a digital profile that continuously updates. The system collects interaction data and behavioral patterns during educational sessions, and uses them to adjust the learning process in real time. By updating the virtual counterpart according to user's behavior, the platform is able to personalize feedback and improve engagement.

The study presented in [2] proposes a HDT framework for personalized healthcare, digitally replicating patients to enable predictive and adaptive medical services. The cloud–edge architecture integrates multi-source health data, including physiological, genetic, behavioral, and environmental information, collected from IoT sensors and medical records. The HDT supports real-time monitoring, disease prediction, and personalized treatment planning, thus forecasting the future health states of the patient.

In [15], the authors present a HDT framework for monitoring ergonomic performance in a manual assembly workstation, with the objective of mitigating Musculo-Skeletal Disorders (MSDs). Motion data was collected through a wearable inertial tracking system and integrated into a virtual 3D environment, where 60 consecutive work cycles were digitally replicated. The HDT enables automated ergonomic risk assessment, real-time visualization of worker movements, and simulation-based evaluation of potential workstation modifications. Predictive analysis targets four key indicators of the physical state of the worker: working postures, exerted forces, manual material handling, and repetitive upper-limb actions. Nevertheless, the study does not address data-security requirements and does not guarantee fully real-time performance of the system.

The study outlined in [16] introduces a HDT framework integrated into SmartFit software, for monitoring athletes in non-professional sports teams. The use case focuses on leveraging IoT-based wearable and app-logged data (including physical activity, sleep quality, food intake, and mood) to continuously represent the fitness status of the athletes. The HDT predicts training performance and automatically suggests behavioral adjustments to improve results, supporting coaches with monitoring, visualization, and rule-based notifications (i.e. automatic alerts triggered by certain conditions). Although effective in demonstrating personalized fitness management, the evaluation was limited to a small group of athletes and required manual data logging for some features, which may reduce data reliability and hinder generalization.

In [17], the authors introduce a DT-based collaborative learning architecture for mobile networks optimization, organized across client, edge, and cloud layers. Their approach, that leverages split federated learning together with reinforcement learning to coordinate distributed training, is specifically designed to preserve the privacy of local data. In this framework, the DT coordinates the learning process by adapting training strategies and aggregation schedules based on network conditions and resource constraints. While this solution optimizes how models are trained across the network infrastructure, it does not exploit human-related features such as behavioral or perceptual indicators, and thus does not directly target QoE prediction.

On the other hand, the study in [18] proposes a HDT-inspired Digital Agent framework, for 6G networks to support QoE-driven network management. The use case focuses on providing personalized and experience-centric service delivery by modeling each user's interaction with the network. The HDT leverages multi-modal data, including QoS metrics (delay, throughput, packet loss), behavioral dynamics, environmental complexity, and application context, to represent the current state of the user. It offers key functionalities such as real-time QoE prediction, tailored resource scheduling, and adaptive network slicing. The prediction relies on features describing the satisfaction of the user and context-dependent requirements. Lastly, the architecture enables collaboration across a hierarchy of user-specific and network-level HDTs, which interact to coordinate global resource allocation.

2.2. Research gap

The aforementioned studies demonstrate that there are different fields where the HDTs are adopted to improve the performances of the system. Most of them fall in the areas of human–system interaction, personalized healthcare, ergonomic system design, fitness and QoE-driven network management. The types of data sources exploited for building the HDT are variegated and include both static (personal characteristics) and more ephemeral data such as: body movement, physiological signals, posture, body temperature, or facial expression. Most of the times, the proposed solutions are tailored to the specific data sources that have been identified during the design phase, leaving limited space for extending the types of data sources that could be considered. Part of the proposed solutions also suffers from a limited inter-operability with external systems provided by other service providers or with other HDTs for an inter-HDT collaboration. Table 1 summarizes the main features of interest of the considered studies, in comparison to our case.

From this analysis it arises that, while recent DT collaborative learning solutions have been proposed for network optimization, most of them mainly focus on coordinating distributed training and resource efficiency, rather than modeling human perception and behavior. Only [18] monitors the status of the users and exploit the concept of HDT (here called digital agent) in network management. With respect to this work, we introduce the following key innovations: (i) we extend the user status monitoring by introducing the device layer where different types of devices can be exploited to send information about the user status (in the experiments we specifically demonstrate the exploitation of facial expression and speech data); (ii) we introduce the concept of collaboration among HDTs to enhance the quality prediction accuracy; (iii) we demonstrate the performance improvements introduced with the proposed HDT solution for quality monitoring and resource optimization in different relevant scenarios.

3. Functionalities of the HDT in service management

In this section, we describe the main functionalities that the HDT should provide in the context of service management, as also sketched on the left side of Fig. 1.

3.1. Collaborative learning in QoE and emotion prediction

The development of models for the prediction of the users' emotional status and perceived QoE requires the collection of a huge amount of data that can cover different scenarios, in terms of: application characteristics, user profiles, network configurations, and different content features, just to cite a few influencing factors. These models are needed for the development of effective network and service management strategies which are aware of the user's emotion and QoE. It is also important to continuously collect relevant data as the mentioned features change over time due to the appearance of novel applications and technologies. The HDT can help in this by supporting the collection of the data and the sharing of the relevant knowledge by feeding collaborative and distributed learning frameworks, which should, however, at the same time respect the user's privacy.

It is also important to highlight that the current models are very often fed with data generated with laboratory subjective tests, which present some limitations in terms of ecological validity. On the other hand, the collection of data in real environments requires to be able to collect key data on the different factors characterizing the observed scenarios (environment, user, system and application factors) [19].

3.2. QoE and emotion prediction

This HDT functionality is strictly linked with the previous one and consists of making available to the network and service management and orchestration modules the predicted QoE and emotion prediction models. These are essential for the optimization of the services. It may result that a user is often affected by cyber-sickness so that the immersive contents should be provided with slow motion features whereas another user may be more sensitive to video resolution issues; another one could be more interested in the audio quality. The prediction models built through HDT-collaborative learning are used as the basis which is then further personalized. Personalization is achieved in two ways: (i) by considering a large number of influence factors that characterize the user (the age, the gender, the content preferences and the social status), it is possible to make predictions that are accurate for the user profile under consideration; (ii) the shared data are used to create a generic model which is then personalized with the only user personal data. This allows for the provisioning of personalized services to the user which optimize his personal perception of the service quality. Not only are

Table 1
Summary of representative HDT studies across various domains, including the proposed QoE-driven HDT framework.

Study	Use case	Used data	Provided functionalities	Predicted features	Inter-operability level
Löcklin et al. [4]	Human–AGV (Automated Guided Vehicles) interaction	Real-time positions and interactions, vital signs and organizational data	AGV control, workstation control	Human movement trajectories, intentions, and behavioral patterns	Interaction with the DTs of the AGVs
Okegbile et al. [2]	Personalized healthcare	Physiological signals, genetic information, behavioral, environmental, and clinical records	Real-time health monitoring, personalized diagnosis/treatment, preventive care	Health status, disease progression, treatment outcomes, and physiological responses	Potential inter-HDT interaction
Greco et al. [15]	Ergonomic risk monitoring and improvement in manufacturing workstations	Motion data from wearable device, forces, weights, workstation layout, and task cycle timing	Near-real-time ergonomic assessment, risk index computation, and testing of alternative workstation designs	Worker posture, exerted force, material handling effort and repetitive motion fatigue	No interactions with other DTs or service providers
Barricelli et al. [16]	Fitness and wellness management for non-professional athletes	Wearable and app data: activity, nutrition, sleep, mood, and trainer evaluations	Training performance prediction, behavioral advising, and coach support	Physical fitness level, training performance, and well-being indicators	Interactions between team members; no interaction with external DTs or service providers
Shen et al. [18]	QoE-driven 6G network management	QoS metrics, user behavior, environmental complexity, and application context	Dynamic personalized QoE modeling, resource demand prediction, and network slicing	User satisfaction, behavioral dynamics, and resource demand	Multi-level hierarchical interactions (edge–cloud) between the HDT and the network service providers
Ours	QoE-driven network management and resources allocation	Facial expressions, speech features, body movements, EEG, ECG and EDA	Collaborative quality, emotions and behavior modeling	User satisfaction, emotional state, and behavioral patterns	Interaction with other HDTs, network service providers and devices

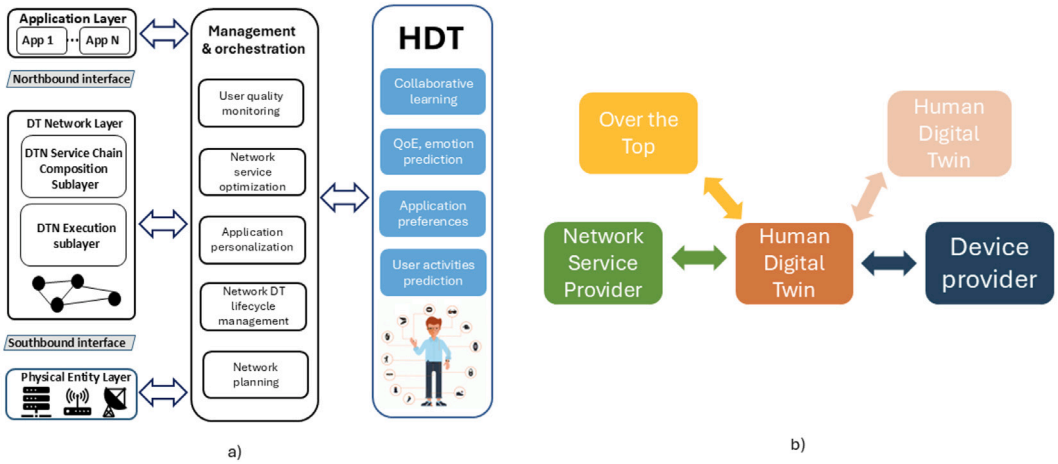

Fig. 1. Role of the HDT in service management: (a) functionalities and relationships with the network management and orchestration modules; (b) relationship and communication with the other service providers.

these models used for service optimization, but these are also used for real-time quality prediction. The knowledge on the current quality level allows for taking corrective actions when low levels are detected or an over-provisioning of resources is found.

3.3. Personalization of applications

The knowledge of the QoE and emotions perceived by the users in different settings and when using different applications constitutes unique values for the creation of applications also in a personalized way. The application's interface and content can be designed according to the preference of the specific user. For instance, content characteristics, such as color schemes, content category, and sound intensity, could directly affect a user's emotional state. Indeed, bright and vibrant visuals may evoke excitement or positivity, while darker tones may elicit calmness or introspection. These emotional responses are measurable through physiological signals such as facial muscle movements and eye gaze variability. The HDT leverages these insights to identify patterns in user emotions and their perceived QoE. Based on this understanding, the HDT can propose content to the user that is tailored to provoke positive emotions while maintaining high QoE levels, fostering a more engaging and satisfying experience. This is becoming more and more useful in the context of semantic communications where the generative AI is used for the production of synthetic content which may be driven by the users preferences [20].

3.4. Prediction of user behavior for network what if analysis

The HDT collects the user's data related to the utilization of network resources during the day, day by day, with the aim of creating a user

profile including these data. Some network utilization patterns would likely be derived from these data based on the user habits, such as limited network utilization when the user is out of home for working, scheduled sport activities, etc. By creating these network utilization profiles for each user (family) of an entire neighborhood, it would be possible to predict day by day the amount and kind of network resources required to satisfy their preferred services. This process would scale for entire cities by enabling network providers to intelligently accommodate network resources based on the predicted consumption. Additionally, the observation of the user mobility by the HDT allows for predicting short-term mobility patterns which can be fed to the network management for optimizing the allocation of resources to the different mobile users [21].

4. Proposed HDT solution

In the following sections, we firstly analyze the major addressed requirements, then we describe the proposed HDT architecture.

4.1. Addressed requirements

To achieve the objectives described in the previous section, it is important to define the role of HDT in the ecosystem of providers of devices, services and applications that revolves around the users of multimedia communications. Indeed, we believe that the HDT should provide services to third-party application and service providers that may leverage information on the users status and preferences as sketched on the right side of Fig. 1. The HDT will be implemented and provided by a service provider which may also be the NSP (Network Service Provider) or the OTT (Over The Top) service provider. For the sake of generalization, we consider this service to be implemented by an additional provider, other than them. The HDT will then interface with the OTT and NSP for a mutual exchange of data and services; the former provides predictions data on the user's status and behavior, whereas the latter makes information on the value of factors influencing the users available, e.g., statistics on packet flows, buffer status streaming applications, content coding parameters. An important interaction is also with the providers of the devices that are used for the multimedia services. This is the case of the user with a wearable display where an application is running, which is also collecting action units (AUs) of user's face and may also get data from a body movement. All these data are important for the HDT, for the current user status estimation, but also for training relevant prediction models. The other important scenario is where user's devices accept to install some HDT plugins which provide data on the user status while also sending this data to the application/service providers. Clearly, it should be done in any case with the consent of the user. Note that the HDT will also interact with other HDTs to implement the collaborative prediction learning described in the previous section.

For these reasons, the collection and sharing of user-related data by the HDT must comply with existing regulatory frameworks, such as the European General Data Protection Regulation (GDPR). As explored in [2], HDTs are increasingly adopted in personalized medicine and clinical decision support, where highly sensitive physiological and diagnostic data are processed to assist medical decisions. In such contexts, hospitals, laboratories, and application providers that exchange information, must act under strict GDPR compliance. A comparable situation is that presented in [16], where the HDT is used to improve personal training, by representing a continuously updated digital counterpart of an individual's body and performance. Multimodal data streams from wearables, smart equipment, and contextual sensors are shared among service providers and device manufacturers, within regulated privacy frameworks. These data are of different types, such as: physiological data (heart rate, oxygen saturation, muscle activation, fatigue indicators), biomechanical data (posture, gait, joint angles, movement quality), behavioral data (training frequency, adherence, sleep, stress) and the contextual data (environment, equipment used, training load history). Similarly, in the proposed scenario, the HDT processes user-related data such as facial AUs and speech-related features, as well as service usage descriptors. In this context, principles such as explicit consent, data minimization, and purpose limitation align with the proposed architecture, ensuring that only service-relevant information is processed and shared among stakeholders.

The integration of cloud–edge computing with HDTs offers numerous advantages, such as providing timely and actionable insights, improving decision-making processes, and allowing for scalable solutions where computational resources can be dynamically allocated based on demand requirements. This is particularly useful for handling varying workloads efficiently. It is then important to design a solution where the processing of data is done at the edge, whenever possible, to limit the amount of data that needs to be sent to the cloud, and then minimize latency and save bandwidth. Inference modules and preprocessing of large amount of data, before being sent to the cloud, should then be done at the edge, whereas major training should usually be performed at the cloud. Additionally, offloading data management to the cloud storage solutions may reduce the need for specialized developer expertise, minimizes development risks, and ensures more efficient and reliable performance for both users and administrators [22,23]. The decision on what should be performed in the cloud and in the edge needs to be taken dynamically sometimes, depending on the requirements of the applications the HDT interfaces with. For instance, it may happen that the HDT is needed for real-time prediction of the user perceived quality with the input data for the prediction provided in the cloud by the third-party application. In this case, it is better the prediction to be executed in the cloud. There may be cases where the sensors used by the users are locally available and the optimization should be performed locally so that the predictor should then perform at the edge. It is also important to highlight that the HDT module should follow the users as they move around so that the reactive modules are hosted close-by.

It is worth noting that, although the HDT is conceived as a comprehensive representation of the user, it is not necessarily meant to continuously operate on an unbounded set of human attributes. Instead, the HDT might rely on a dynamic, context-aware ontological organization, where only subsets of influence factors are activated, depending on the target service or application. In this context, the full-view virtually represents the set of all available attributes, while service-specific sub-ontologies allow lighter, partial HDT instances to operate on a limited number of relevant features. This design helps limit computational and communication load, and supports scalable and interpretable collaborative HDT interactions.

Finally, while the HDT architecture proposed here is intentionally defined at a general level to adjust to heterogeneous deployment scenarios, the practical use case discussed in Section 6 represents a concrete application of the framework. In this setting, the HDT operates over well-defined device types and learning models for QoE prediction, demonstrating the effectiveness of the proposed approach under realistic and specific conditions.

4.2. The HDT architecture

The proposed HDT architecture, illustrated in Fig. 2, defines the main components, functionalities, and communication interfaces with the External Systems (ESs) which are those of the NSP, OTTs and other HDT implementations. The role of the ESs encompasses any entity interested in utilizing the services developed in the proposed HDT architecture. The HDT architecture consists of three main layers, namely the device layer, the edge layer, and the cloud layer, which can exchange data employing RESTful APIs and Publish&Subscribe interfaces. More specifically, the device layer generates raw data and extracts features related to the user status, which are forwarded to the edge layer for inference and model updating. Aggregated estimation results, user profiles, and model parameters may be

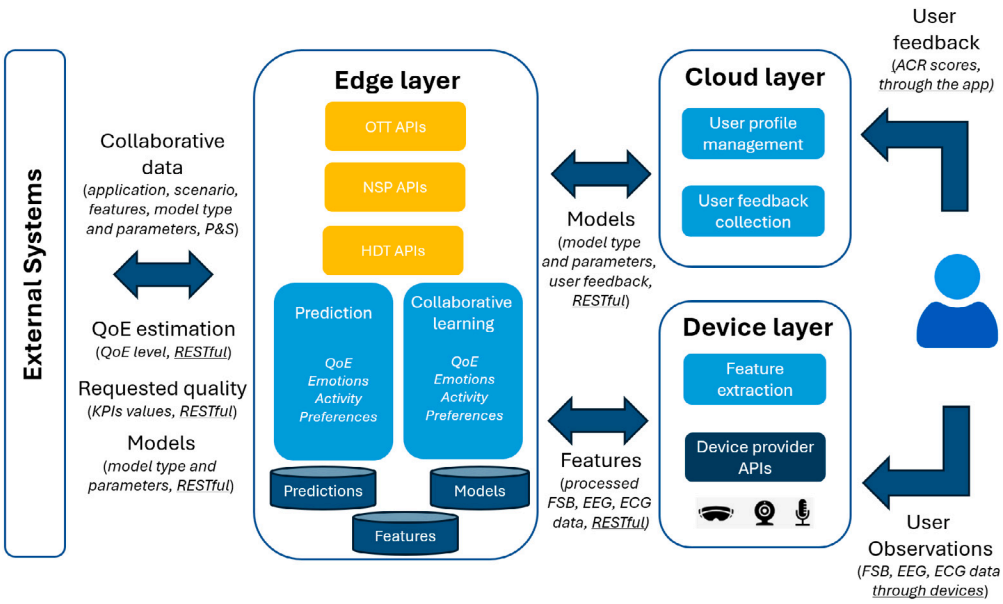

Fig. 2. The proposed HDT architecture. The text at the bottom of the arrows mentions the exchanged data and the types of APIs used (the latter are highlighted with underlined text).

exchanged with the cloud layer depending on the considered functionality. Publish&Subscribe mechanisms are primarily adopted to support collaborative interactions among HDTs and the dissemination of model-related information, whereas RESTful APIs are mainly used to enable on-demand interactions between the HDT and external service and network providers.

4.2.1. Device layer

This layer is responsible for collecting and pre-processing user data collected through different types of devices, such as: Facial expressions, Speech characteristics, and Body movements data (FSB) as well as body temperature, EEG (Electroencephalography), ECG (electrocardiography), EDA (electrodermal activity) and others. The most common devices are wearable displays, video cameras, microphone, LIDAR sensors as well as smartwatches. The integration of physical devices and equipment within the device layer allows for capturing and extracting user characteristics to construct a digital twin, which represents the digital counterpart of the user's behavior and interactions. As a result, when the physical state of the process changes, the digital representation will also automatically update. The collected data are processed, and the relevant features are extracted and transmitted to the edge layer via appropriate communication schemes, where they are stored in the database and utilized by the estimation models. Note that the data from the devices are obtained with the mentioned APIs with the device provider.

4.2.2. Edge layer

User features data are transmitted to the edge layer from the data flow generated by the devices. This data is used to continuously improve the learning and update the relevant models for quality, emotions and behavior estimation. The learning needs user feedback that are collected from the user through the cloud layer application. This learning phase is performed in an anonymous way and collaboratively with other HDTs as explained in the following section. The device layer is also devoted to predicting the status of the user as requested by ESs and exploiting the most updated models. The updated model parameters are continuously sent to the edge to be used for the user status estimation. The ESs can access the estimation results within the edge layer to make decisions regarding the services provided through the HDT architecture based on the user's emotions and QoE perceptions. This encompasses overseeing model behavior, ensuring efficient performance, and making necessary adjustments to align with system objectives.

4.2.3. Cloud layer

Personal information collected from users at the device layer, along with the emotional states and QoE estimation results produced by the models at the edge layer, are sent to the cloud layer for storage in the user profile within the user module. Each user has a unique profile created in the cloud layer within the user module, enabling persistent management of users' activities and behaviors, as well as providing valuable historical records. These data can be used for reporting, trend analysis, and evaluating service effectiveness, as well as identifying areas for improvement and guiding future development decisions.

User feedback is requested time by time by the cloud application to collect ground-truth data that is vital for the training activities.

Another operation in this layer involves transmitting the user's personal information to the cloud layer via proper interfaces to create a unique profile for each user, which can help to have consistency management in providing services to the users.

4.3. Communications with external systems

The proposed HDT architecture comprises different Publish& Subscribe and RESTful API interfaces as described in the following.

4.3.1. HDT publish&subscribe

The objective of this interface is to identify the HDTs of users that are involved in the same type of application and are willing to share relevant knowledge so as to improve HDT performance and finally service management quality. Accordingly, a MQTT (Message Queuing Telemetry Transport) broker has been deployed to which each HDT willing to collaborate registers and receives request to join a collaborative learning session. Each topic represents an application session described with the following format:

/application_type/application_scenario/features

where *features* lists all the influence factors separated by &. In this first implementation we have not defined a common complete vocabulary that should be used by the HDT but this is necessary to be done for an effective implementation so as to make it possible to define correct collaborative sessions.

At first, each HDT downloads the list of existing topics, where each one is used to describe all the active sessions the HDTs are or have been involved in. If none of them is of interest for the new registering HDT, it generates a new topic describe the application session

of interest. Whenever a new session of this type is created and the relevant estimation model is updated, the HDT publishes the relevant parameters which can be used by the other HDTs for collaborative learning. The collaborative model updating process is described in the following Section 5.

4.3.2. NSP/OTT APIs

The NSPs and OTTs are interested in knowing the status of the user in terms of emotions, quality of experience and predicted short/medium-term behavior. This information is obtained with RESTful APIs that the providers make available to the HDT so that at a given desired frequency the estimated level of quality is conveyed to the NSP. Complementary APIs have also been developed to make the information available by the HDT so that the query is triggered by the service provider whenever it is needed. Another level of cooperation is obtained with APIs that the HDT uses to communicate to the providers about the need of additional resources expressed in terms of KPIs (latency, packet losses and bandwidth) when quality degradations occurrences are detected by the HDTs. Clearly, this trigger is activated after an agreement has been achieved among the two parties about the desired level of quality.

Finally, a third level of interaction is obtained by the APIs through which the HDT provides the service provider with a prediction model that can be used to estimate the quality and that can be used to perform cause-root analysis in case quality degradations have been identified and to perform resource optimization.

5. Collaborative HDT model learning

There are different scenarios that could be encountered with reference to HDT-collaborative learning for QoE and emotion estimation. The scenario that we believe is the most common is the case where the collaborative users are using the same application (e.g., immersive visit of a remote site through immersive technologies) but different sets of influence factors are observed (which could partially overlap or not). This happens often as the users may use different sensors that collect data on the environment; some may share information about the end-devices and some may not; some users may allow for using human factors data and some users may not. To allow for the preservation of user's privacy, and to avoid sharing personal data, we propose a solution based on multi-view (MV) learning.

As a practical example, let us consider user HDT_1 and user HDT_2 monitoring the perceived QoE and emotions of video streaming services consumed by their respective users. Due to the specific limitations, HDT_1 creates its own model that estimates the QoE and emotions of the video streaming service as a function of the end-to-end delay and packet losses. Differently, the HDT_2 has its own model that performs the estimations during video streaming service as a function of the playout buffering, only. Since QoE and emotions depend on all these influence factors, these two models may improve their performance by leveraging past observations of both HDTs. However, even if they are willing to share the knowledge on the models, they might not be keen to share relevant user data, i.e., past observations of the user status.

To address this necessity, we leverage on the multi-view learning techniques to propose an approach that can potentially enable the implementation of estimation models based on data originally collected by different HDTs. With MV techniques, it is possible to integrate knowledge from multiple datasets or "views" to predict a precise target. By taking advantage of multiple views, MV-based predictors can capture different aspects of the data and produce more accurate and reliable predictions. For these reasons, the MV learning approach fosters the reuse and integration of subjective datasets collected by different entities, aiming to develop enhanced prediction models. Moreover, MV learning preserves data privacy by integrating the information from the neural network's (NN) hidden layers trained on the separated datasets. Thus, there is no need to share raw data between diverse entities (which are often unwilling to share collected data), but each model can potentially enhance its prediction performance by integrating information learned from other predictors trained on different views.

For the sake of simplicity, in the following section we refer to the QoE estimation task only, but the same approach applies to emotion recognition as well, without loss of generality. We thus define K_i as the vector encoding all the influence factors (IFs) which are considered by HDT_i (HDT of user i) and that are used to predict user's QoE through model $Q_i(K_i; D_i)$:

$$Q_i^{est} = Q_i(K_i; D_i); 1 \leq Q_i^{est} \leq 5, \tag{1}$$

where we assume that the quality model has been created to estimate the QoE which could be measured using the 5-level Absolute Category Rating (ACR) scale [24], and through appropriate training from the past observations D_i relative to the ith user. Note that the IFs are collected by the Device Layer and the prediction is performed by the relevant component in the device layer of Fig. 2. The objective is to build a more accurate QoE model $\hat{Q}_i(K_i; D_i, O_f)$ through the integration of *information* embedded in the fusion layer output O_f, generated thanks to the Q_j models trained on datasets D_j, for $j = \{1, \ldots, J\}$ and $j \neq i$, where J is the number of entities that collaborate to create a joint model. This improvement is performed by the collaborative model located in the device layer (see Fig. 2).

Let us consider the specific case for the sake of simplicity which is depicted in Fig. 3, where the 2 models $Q_1(K_1; D_1)$ and $Q_2(K_2; D_2)$ estimate the quality for the same service as a function of K_1 and K_2 IFs for two different users through he relevant HDTs, respectively, so that:

$$Q_1^{est} = Q_1(K_1; D_1), \tag{2}$$

$$Q_2^{est} = Q_2(K_2; D_2). \tag{3}$$

We aim to enhance the prediction performance of these models by sharing information from their learning function:

$$\hat{Q}_1^{est} = \hat{Q}_1(K_1; D_1, O_f); 1 \leq \hat{Q}_1^{est} \leq 5, \tag{4}$$

$$\hat{Q}_2^{est} = \hat{Q}_2(K_2; D_2, O_f); 1 \leq \hat{Q}_1^{est} \leq 5, \tag{5}$$

where $\hat{Q}_x(K_x; D_x, O_f)$ is the model that predicts the QoE from the K_x IFs (with $x = \{1, 2\}$); it was trained with the D_x dataset and with the fusion layer output O_f.

To better illustrate the implementation and the performance assessment of the proposed collaborative HDT multi-view based prediction model and the alternative ones, herein we summarize the considered different approaches:

1. Partial view (PV): each $Q_i(K_i; D_i)$ model is trained on the dataset D_i, which includes the values of the K_i IFs and the corresponding QoE. This represents the case where each entity builds its model independently, with Q_i^{est} as output.
2. Multi-View (MV): each $\hat{Q}_x(K_x; D_x, O_f)$ model is trained on the dataset D_x (including the values of the K_x IFs and the corresponding QoE) and with the support of the fusion layer output O_f. This is the proposed model whose output is $\hat{Q}_x^{est}$.
3. Full view (FV): a single QoE model, $Q(K; D)$ is trained on the dataset D, which includes all the $K = \cup_i K_i$ IFs and the corresponding QoE. This is the case where the different HDTs share the raw data, and is introduced here for comparison purposes (it is a very uncommon scenario).

The PV corresponds to the case where each HDT does not collaborate in learning. The FV corresponds to the case where the HDT share all the observations. The MV is the solution we proposed to enhance the prediction, while preserving the privacy of the user.

It is crucial to acknowledge that the effectiveness of collaborative HDT learning largely depends on the quality and reliability of

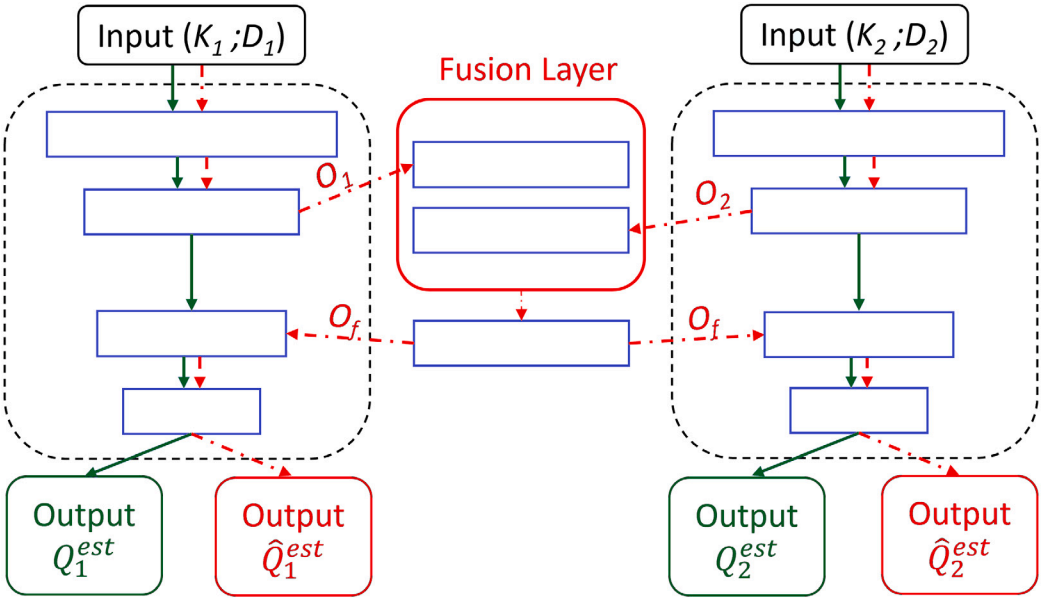

Fig. 3. PV approach: the green solid lines indicate the $Q_x(K_x; D_x)$ models, whose outputs are Q_x^{est} $(x = 1, 2)$. MV approach: the red dashed lines indicate the $\hat{Q}_x(K_x; D_x, O_f)$ models, which are trained with the support of the fusion layer output O_f, and whose outputs are $\hat{Q}_x^{est}$ $(x = 1, 2)$; O_x is the output of one of the hidden layers of $\hat{Q}_x$ models $(x = 1, 2)$.

the collected data. In practical deployments, user data may be incomplete, noisy, or biased due to limited user engagement, inaccurate self-reported information, or hardware and software faults at the sensing level. In such conditions, errors in the local HDT representations may propagate throughout the training process, hindering the accuracy of the resulting predictions. However, this effect is more pronounced in full-view and partial-view approaches, since multi-view learning can partially mitigate the contribution of unreliable modalities. Nonetheless, robustness to biased or faulty data remains an important topic, and a relevant direction for future improvements of collaborative HDT-based QoE estimation and service personalization.

6. Experimental performance analysis

The following use case illustrates the applicability of the proposed collaborative HDT learning exploiting the multi-view learning approach for predicting perceived QoE in WebRTC communications, leveraging facial and speech features, as well as web-level internal metrics.

In the first subsection we describe the application scenario we refer to with the relevant datasets and the results obtained with the HDT collaborative learning. In the second subsection we show how the prediction models can be used to monitor the user perceived quality. The last third subsection we show this developed collaboratively enhanced models can be used to improve the performance of resource allocation algorithms.

6.1. Application scenario and dataset

The data set used was originally collected in the authors' previous work, in the context of [9] and [25], and is thoroughly described in [26]. Their experimental setup comprised a series of 15 two-minute "Who am I?" WebRTC interactions between 20 acquaintances paired in couples (although two participants were discarded, since their data were identified as outliers). Each conversation was subject to different degrees of three types of impairment. The participants in the conversation were then asked to assign a vote on the ACR scale (1–5) to the perceived quality of the call. Over the course of these 300 sessions, facial and speech features were recorded, alongside internal web metrics. The most significant among them, in terms of their representation of the perceived QoE, were then identified via a one-way ANOVA analysis; these include 111 speech-related features, 18 facial features, and 168 web-internal features. The details of the dataset are summarized in Table 2.

In our study, machine learning models are developed and trained to predict the perceived QoE, using the speech-related features, facial features and web-internal features from [26] as input, and the ACRs as target values. However, the ACRs collected during the webRTC sessions were affected by a substantial disproportion in favor of the intermediate values, which might bias our model. To address class imbalance, we augmented the data set using the ADAptive SYNthetic (ADASYN) algorithm. Our experimental design involves three distinct observers (hereafter referred to as "views") that independently monitor the same communication sessions. Each view records a separate modality, among facial expressions, speech signals, or web-internals. Accordingly, we have three sets of influence factors (each one collected by a different HDT). Based on this framework, we have trained different models following the three approaches mentioned in the previous section, i.e., the PV, the FV and the MV models.

To implement these models, we developed two Fully Connected Deep Neural Network (FCDNN) architectures. On the one hand, for both PV and FV setups, a single-branch FCDNN processes either partial or complete sets of IFs, to generate a unified QoE prediction. On the other hand, the MV architecture can be seen as a three-headed generalization of the structure depicted in Fig. 3. It begins with three initially independent branches, each dedicated to one modality; these branches are the same ones that constitute the PV architectures. The intermediate fusion layer is then introduced to enable cross-view interaction, through the exchange of pre-processed data. The output of this fusion layer is further reintegrated into the original branches, which continue processing to produce three parallel predictions. Henceforth, the notation "$MV_{a\ (b,c)}$" will indicate the branch that begins with modality *a*, and then integrates information from modalities *b* and *c*, after fusion. As an example, "$MV_{sp(fac,wi)}$" refers to the prediction deriving from the branch initially processing speech-related features (*sp*), and later augmented with facial (*fac*) and web-internal (*wi*) information.

All considered models begin with a *MinMaxScaler* function, necessary to normalize the input features. The size of the initial hidden layer (i.e. the number of neural units) varies depending on the configuration.

Table 2
Properties of the used dataset. For the prediction of the QoE, three different sets of features have been considered as listed in the table.

Subjects # (Outliers #)	Application	Influencing factors	Feedback	Sessions #	Speech features (sp)	Facial features (fac)	Web internal features (wi)
20 (2)	WebRTC conversation	Delay, Jitter, Packageloss	ACR scale	300	111	18	168

Table 3
Hyperparameters used to train the MV and FV models. MV configurations match those of the corresponding PVs, except for the depth: the base PV depth is indicated outside parentheses, with three additional hidden layers per branch shown in parentheses.

Approach	Depth	Hidden dim. 1	Batch size	lr	Dropout	Weight decay	Sched. factor	Sched. patience
MV sp(fac,wi)	2(+3)	2048	16	5e−4	0.20	1e−4	0.86	2
MV fac(sp,wi)	3(+3)	2048	16	5e−4	0.20	1e−4	0.86	2
MV wi(sp,fac)	2(+3)	1024	16	5e−4	0.20	1e−4	0.86	2
FV	11	32,768	256	1e−4	0.20	1e−4	0.93	2

However, in every configuration all the hidden layers contain half as many neurons as their predecessor. Each hidden layer is activated by a Rectified Linear Unit (*ReLU*) activation function, with the exception of the output layers, which are activated by a *Softmax* function. To improve training stability, each hidden layer is followed by a normalization layer and dropout. The "weight" and "bias" parameters are refined using the *Adam* optimizer. Although the initial learning rates (*lr*) differ across configurations, their decay during training is regulated by the *ReduceLROnPlateau* learning rate scheduler. Finally, for model training and validation, we apply a 70%/30% data split. All the other, model-specific hyperparameters are selected through iterative refinement to maximize predictive performance, and are listed in Table 3. The hyperparameters used to train the MV-related neural networks are the same ones used to train the corresponding PVs, with the only difference that three extra hidden layers are added to each branch of the MV, after the fusion layer.

Prediction performances of the different approaches are assessed in terms of their *accuracy*, *precision* and *F1-score*. To ensure statistically meaningful results, each model is trained 100 times. The results of the training, averaged over the 100 iterations, are presented in Table 4. The MV configurations outperform both PV and FV setups across almost all considered metrics. In particular, the $MV_{sp(fac,wi)}$ branch provides the highest values in all three considered evaluation metrics, averaged across all the ACRs. With an average accuracy of 80.4%, this configuration surpasses the accuracy of the FV by 3.7%. On the other hand, the highest relative gain compared to its corresponding PV counterpart in terms of accuracy is achieved by the $MV_{fac(sp,wi)}$ branch. It is important to note that this is an exploratory study, and the absolute performance of our models may seem to fall short of current state-of-the-art benchmarks (for instance, in [9] an accuracy of 93% was obtained, with the application of various data fusion techniques); this is most likely imputable to the limited size of our sample. This effect is also reflected in the general difficulty of all the approaches to correctly classify samples belonging to class 3, although this fact might also be linked to the ambiguous nature of this category. Nevertheless, all MV approaches show enhanced performances in classifying objects belonging to class 3 with respect to all other considered configurations, with the $MV_{sp(fac,wi)}$ configuration reaching a top accuracy of 46.5%.

Recently, many studies have investigated collaborative and multimodal learning approaches for QoE prediction. For instance, [27] proposes DeepQoE, a multimodal deep learning framework that combines visual, audio, and contextual information for video QoE estimation. The different modalities are mapped into a shared representation, allowing the model to capture perceptual relationships and improve prediction robustness across different content conditions. Moreover, clustered federated learning (FL) strategies have been introduced in [28] to address user heterogeneity in web-based QoE prediction through decentralized model aggregation. In this approach, users with similar perception patterns are grouped together, and specialized models are trained for each group to better reflect individual reactions to the same stimuli.

Due to differences in datasets, feature spaces, and evaluation protocols, a direct quantitative comparison with these approaches is hardly feasible. Nevertheless, these works provide a useful reference to contextualize our results, as they similarly demonstrate the benefits of accounting for heterogeneous information sources and user diversity. In this context, the proposed HDT-based multi-view learning approach offers a complementary solution, where collaboration is enabled through view separation and partial information sharing, yielding measurable performance gains while preserving architectural flexibility. Although a direct comparison is challenging, a preliminary comparison between the prediction performance of the MV approach and of an alternative collaborative learning baseline is here presented. To this end, we implemented a FL baseline, operating on the same dataset and feature sets of the MV and FV approaches. This comparison allows us to evaluate whether the performance of the proposed MV framework derive from cross-view cooperation, or from parameter aggregation alone. In the FL configuration, each view locally trains a model using its own modality (either speech-, facial-, or web internals-related features), and periodically shares model parameters with a central backbone that computes a global model through an averaging procedure (*FedAvg* [29]). This setting emulates a collaborative scenario where raw data remain decentralized, and only model updates are exchanged. To adjust to the heterogeneous dimensionality of the input sets of features, each independent branch is preceded by a modality-specific linear adaptation layer that projects the original features into a common latent representation. The adapter is trained locally together with the classifier, whereas only the backbone parameters are involved in the aggregation process. This structure ensures that collaboration occurs in a shared decision space, while modality-specific encodings are preserved. The shared backbone employs the same architecture and training procedure adopted for the web-internals-related PV configuration (elected as the best-performing one, after testing): two normalized hidden layers, with 1024 and 512 neural units, respectively; the first one is *ReLU*-activated, while the second one produces its classification through a *Softmax* function. The hyperparameters used during training are exactly the same ones listed in Table 3. Classification performance of the FL approach, reported in the bottom row of Table 4, shows per-class trends similar to those observed for the other considered configurations: on the one hand, objects belonging to class 5 are consistently the easiest to identify, and are associated to an accuracy of approximately 86.9%; on the other hand, class 3 remains the most challenging, due to its intrinsic ambiguity, leading to an average accuracy of 36.5%. In terms of overall accuracy, the FL approach performs worse than two standalone PV models out of three, with a rate of success of 62.8%

Table 4
QoE estimation performance of the proposed MV models, as well as those of the PV, FV and FL baselines, in terms of accuracy, precision, and F1-score. For each approach, metrics are reported both as class-wise values across ACR levels and as overall averages.

Approach	Metric	ACR 1	ACR 2	ACR 3	ACR 4	ACR 5	Average
MV sp(fac,wi)	Accuracy	0.898	0.847	0.465	0.823	0.971	0.804
	Precision	0.919	0.680	0.795	0.768	0.876	0.807
	F1-Score	0.909	0.755	0.587	0.794	0.921	0.793
MV fac(sp,wi)	Accuracy	0.898	0.811	0.461	0.836	0.971	0.798
	Precision	0.899	0.670	0.789	0.767	0.876	0.800
	F1-Score	0.898	0.734	0.582	0.800	0.921	0.787
MV wi(sp,fac)	Accuracy	0.900	0.778	0.448	0.836	0.969	0.789
	Precision	0.893	0.658	0.746	0.761	0.878	0.787
	F1-Score	0.896	0.713	0.560	0.797	0.922	0.778
PV_{sp}	Accuracy	0.805	0.689	0.356	0.576	0.927	0.677
	Precision	0.665	0.606	0.542	0.682	0.826	0.664
	F1-Score	0.728	0.644	0.430	0.625	0.873	0.660
PV_{fac}	Accuracy	0.704	0.419	0.230	0.685	0.887	0.588
	Precision	0.560	0.544	0.394	0.594	0.718	0.562
	F1-Score	0.624	0.473	0.291	0.637	0.793	0.563
PV_{wi}	Accuracy	0.932	0.566	0.377	0.724	0.928	0.708
	Precision	0.699	0.720	0.658	0.662	0.767	0.701
	F1-Score	0.800	0.634	0.479	0.692	0.840	0.689
FV	Accuracy	0.900	0.699	0.440	0.789	0.991	0.767
	Precision	0.810	0.631	0.717	0.801	0.852	0.762
	F1-Score	0.853	0.663	0.546	0.795	0.917	0.755
FL	Accuracy	0.797	0.570	0.365	0.520	0.869	0.628
	Precision	0.645	0.536	0.493	0.648	0.768	0.619
	F1-Score	0.713	0.553	0.419	0.577	0.815	0.617

(classification performance of the PV_{fac} approach being the worst one, with a 58.8% accuracy). This behavior is to be attributed to the shared backbone, which operates on aggregated parameter updates, and therefore struggles to capture the characteristics of individual, decentralized modalities.

In conclusion, while the HDT collaborative learning MV approach still presents wide room for possible improvement, it also clearly demonstrates its efficacy and potential in enhancing the PV, FV and FL baselines.

6.2. HDT for quality monitoring

In order to obtain additional confirmation of the robustness of the classification performance achieved by the MV learning approach, we tested it on the data of one randomly selected user (User 12), after removing this user's data from the training set (only for this specific stage). We thus compared the QoE predicted by each approach, and for every test condition, against the actual ACR scores provided by user 12 during the subjective test. All MV branches correctly predicted 12 out of 15 ACRs, confirming a classification accuracy of approximately 80%, whereas the FV model correctly identified 8 ACRs. By contrast, the PV models achieved at most 4 correct predictions (PV_{wi}), with the lowest performance being 1 correct prediction (PV_{sp}). The discrepancy between the accuracies obtained by the PV approaches during training and those observed in this validation step remains an open issue requiring further investigation. A graphical representation of these results is provided in Fig. 4, where the gap between the MV and PV accuracies is evident. For clarity purposes, MV and FV predictions are shown in the top panel, while PV results are displayed in the bottom panel. The actual ACRs are shown for comparison in both diagrams.

While accuracy provides a useful measure of classification performance, it does not capture the magnitude of the difference between incorrect predictions and the actual values. To address this, we also compared the MOS values produced by our models for each TC with the empirical MOS, and the corresponding standard deviations. Fig. 5 presents the MOS for all approaches and for each TC, with the same top/bottom panel split and color coding as in Fig. 4. From the comparison of the two panels, it is evident that the MV approaches, as well as the FV model, track the empirical MOS trends much more closely across all TCs. In contrast, the PV results appear almost entirely uncorrelated with the actual values. In fact, the difference between MV predictions and the empirical MOS is approximately 0.14, on average, across all three branches (0.15 for the FV). In contrast, the PV approaches exhibit differences greater than 1, with the largest error reaching 1.12 for PV_{wi}. Similarly, Fig. 6 shows that the standard deviations of the PV approaches are systematically higher than those of all other approaches. This confirms that the MV approaches not only achieve greater accuracy than the PV ones, but also that their misclassifications tend to remain closer to the actual values. However, in certain TCs (for example, TC 5 and 8), the standard deviations of the MV models exceed those of the ground truth. This is due to the strong concentration of actual ratings around 3 and 4 in these conditions, which amplifies the effect of even small misclassifications on the predicted variance.

6.3. HDT for quality control

Among the possible applications of QoE modeling, this section presents a practical demonstration of how a NSP can exploit the predicted QoE values to dynamically optimize the resources allocated to a given service (and thus its costs) while maintaining an acceptable quality level. In this example, the decision of the NSP to allocate more, fewer, or equal resources to the system is driven by the so-called *utility functions*. Let us consider a generic system operating in N discrete states, where each *i-th* state corresponds to a perceived QoE level (QoE_i) and a specific amount of allocated resources r_i, such that $r_i > r_{i+1}$. We define utility functions (U_i)—as linear combinations of QoE and resources, which is used by the NSP to decide at each instant whether it is more convenient to increase (+), maintain (=), or reduce (–) the

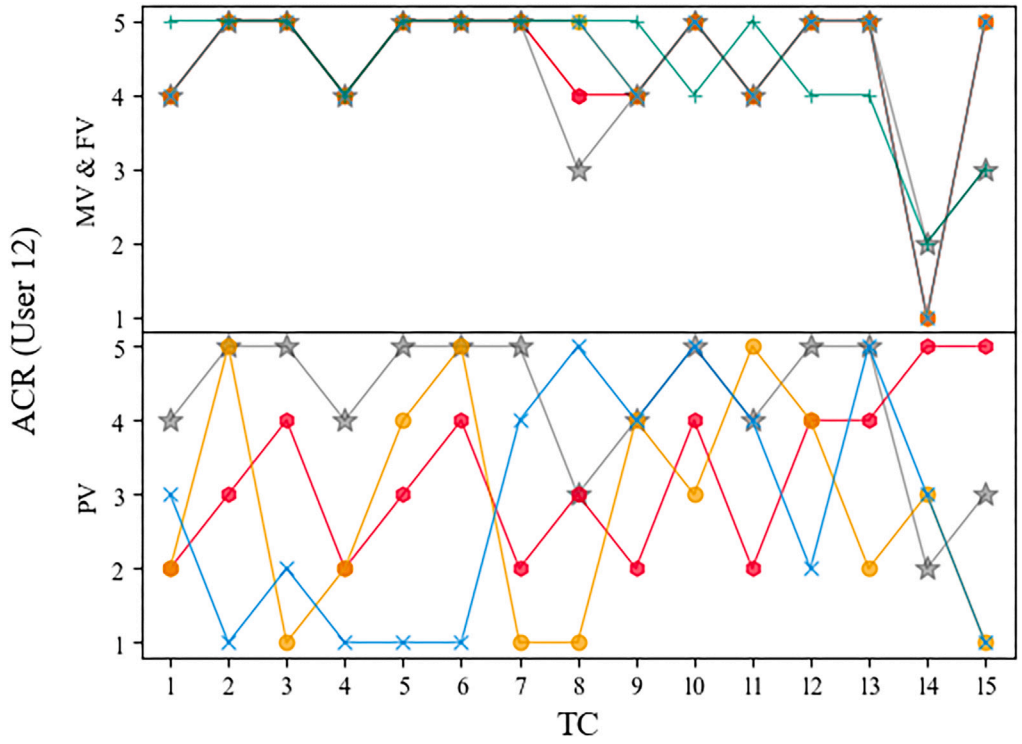

Fig. 4. Predictions of the various approaches, relative to User 12: the red hexagons represent the data relative to the $MV_{sp(fac,wi)}$ branch, while the yellow circles and the light-blue tilted crosses correspond to the $MV_{fac(sp,wi)}$ and $MV_{wi(sp,fac)}$ branches, respectively; for consistency, the results of each PV branch are displayed using the same color scheme as the corresponding MV branch from which it originates; the predictions of the FV model are represented by the green crosses, while the actual ACRs are indicated by the gray stars.

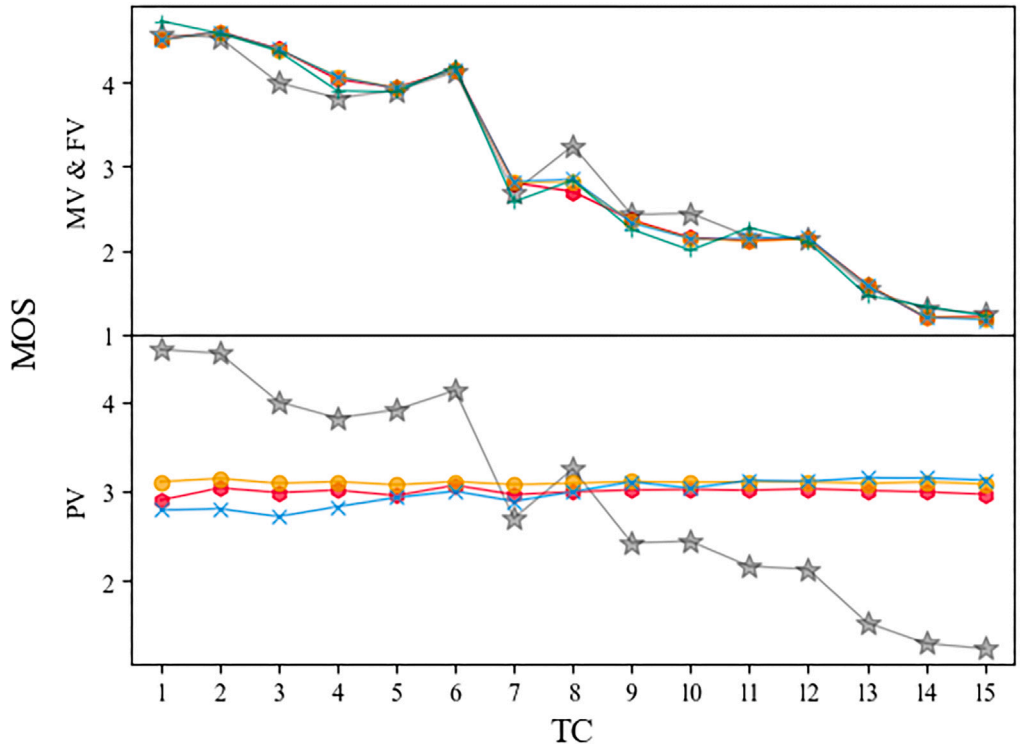

Fig. 5. Predictions of our models, averaged (Mean Opinion Scores - MOS) over each test condition. The same panel split and color-code of Fig. 4 are applied.

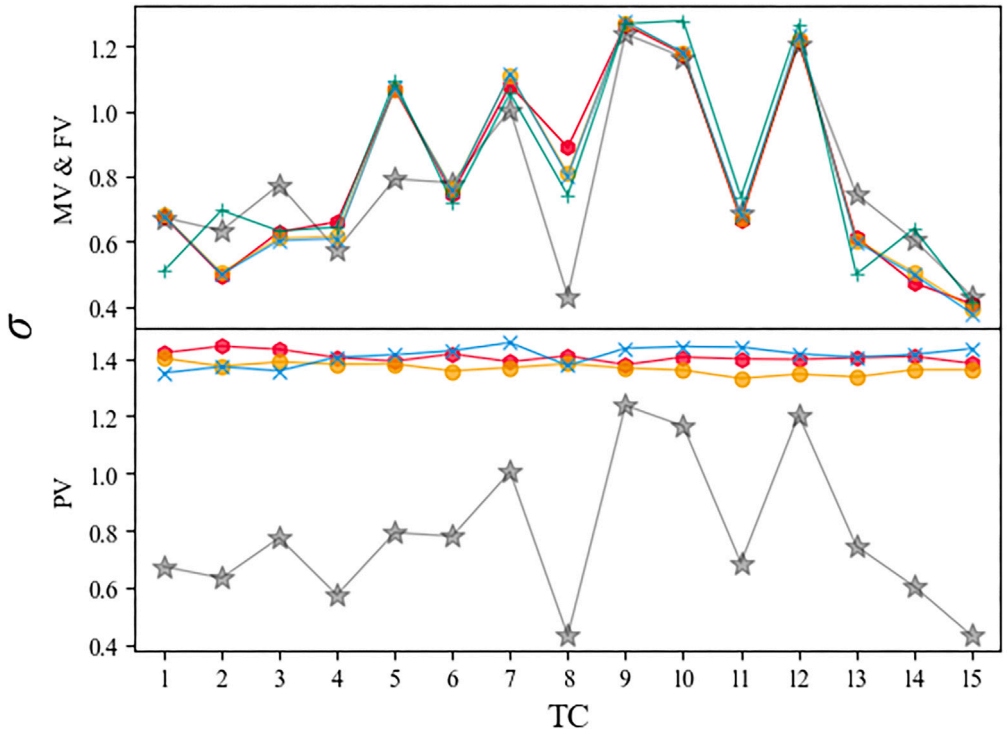

Fig. 6. Standard deviations of the predictions of our models. The same panel split and color-code of Figs. 4 and 5 are applied.

amount of allocated resources:

$$
\begin{aligned}
U_{i,+} &= \alpha(P_{highest} \cdot QoE_{i-1} + P_{intermediate} \cdot QoE_i + \\
&\quad + P_{lowest} \cdot QoE_{i+1}) - \beta \cdot r_{i-1} \\
U_{i,-} &= \alpha(P_{highest} \cdot QoE_{i+1} + P_{intermediate} \cdot QoE_i + \\
&\quad + P_{lowest} \cdot QoE_{i-1}) - \beta \cdot r_{i+1} \\
U_{i,=} &= \alpha(P_{highest} \cdot QoE_i + \frac{1 - P_{highest}}{2} \cdot QoE_{i-1} + \\
&\quad + \frac{1 - P_{highest}}{2} \cdot QoE_{i+1}) - \beta \cdot r_i.
\end{aligned}
\tag{6}
$$

Here, α and β represent the relative weights of QoE and resources, respectively, in the decision process. The system maintains a non-zero probability of not following the direction suggested by the utility function with the highest value; for instance, if the utility functions indicate that resources should be increased, the system will: migrate to a more resource-demanding state with the highest probability ($P_{highest}$), remain in the current state with an intermediate probability ($P_{intermediate}$), or move to a lower-resource state with the lowest probability (P_{lowest}). These probabilities act as weights within the definitions of the utility functions and govern the stochastic evolution of the system.

We applied this mechanism to our WebRTC communication use case to illustrate its evolution in a realistic scenario, and we compared it with the evolution based on the subjective-test data. Instead of the individual QoE scores, we used their values averaged (the MOSes) across the different system states (i.e., the TCs), and we adopted the following assumptions:

- The impacts of MOS and resources on the utility functions are considered equal ($\alpha = \beta = 0.5$).
- During every iteration, $P_{highest}$ assumes a random value between 0.85 and 0.95, while $P_{intermediate}$ and P_{lowest} vary between 0.01 and 0.14.
- The required resources decrease linearly with the state of the system.[1]
- Resources vary discretely, meaning that only a fixed amount Δr can be added or removed from the system at any given time.
- At each stage, the system can either remain in the current TC or migrate only to adjacent ones.

We observed that the utility functions computed using the ground-truth MOS values agree with those derived from $MV_{fac(sp,wi)}$, $MV_{wi(sp,fac)}$, and FV approaches for 80% of the TCs, while the agreement rate decreases to 73.3% when using MOS values from $MV_{sp(fac,wi)}$. On the other hand, the utility functions obtained from the PV approaches achieve an agreement rate between 26.7% and 40% with the ground-truth utilities. This is consistent with the fact that MOSes predicted by the PV approaches are not correlated with the ground-truth values (see the bottom panel of Fig. 5). Moreover, since the MOS values obtained by the PV approaches are approximately constant across all the TCs, their influence on the utility functions is the same, and the allocated resources become the discriminating factor in evaluating the largest utility function. Since, by assumption, the resources decrease linearly with the TC, U_- is almost always the largest utility function for the PV approaches. The decisions based on the utility functions per TC and per learning approach are summarized in Table 5.

Based on the obtained utility functions, we simulated the temporal evolution of the system starting from an initial TC. For each iteration, a random number uniformly distributed between 0 and 1 was generated to determine whether the system would follow the direction indicated by the utility function with the maximum value for that TC. To observe the system's evolution over time, this process was repeated for 200 iterations, updating the initial TC to the one decided in the previous iteration after each step. The same procedure was applied for all learning approaches, as well as for the ground-truth data. Fig. 7 illustrates an example of the evolution of the system driven by the utility-function-based decisions, across the various learning configurations.

[1] Although we acknowledge that this assumption does not always hold, a more realistic resource model lies beyond the scope of this paper. Moreover, this simplification does not compromise the validity of the proposed approach.

Table 5
Summary of the decisions based on the highest utility functions obtained for each learning approach and TC. Abbreviations: "*red.*" (reduce), "*maint.*" (maintain), "*incr.*" (increase) resources; "*GT*" (Ground-Truth).

TC	MV sp(fac,wi)	MV fac(sp,wi)	MV wi(sp,fac)	PV sp	PV fac	PV wi	FV	GT
1	red	red	red	red	red	red	incr	red
2	maint	maint	maint	red	red	maint	incr	maint
3	incr	incr	incr	red	red	red	maint	incr
4	incr	incr	incr	red	red	red	incr	red
5	red	red	red	red	red	red	red	red
6	maint	maint	maint	maint	red	maint	maint	maint
7	incr	incr	incr	red	red	red	incr	incr
8	incr	maint	maint	red	red	red	red	maint
9	incr	incr	incr	red	red	red	maint	incr
10	incr	incr	incr	red	red	red	red	maint
11	red	red	red	red	red	red	maint	incr
12	maint	maint	maint	red	red	red	red	maint
13	incr	incr	incr	red	red	red	red	incr
14	incr	incr	incr	red	red	red	maint	incr
15	maint	maint	maint	maint	maint	maint	incr	maint

Table 6
Normalized average cumulative Ground-Truth (GT) utility over 200 iterations for each learning approach, computed across 5 simulation runs. The "*Appr*" columns show the cumulative GT utility evaluated using the GT MOSes, based on the decisions of the learning approach; the "*GT*" columns show the same metric, but based on the GT decisions, at each step of the iterations. These values were calculated for each initial TC, as well as overall.

Initial TC	MV sp(fac,wi)		MV fac(sp,wi)		MV wi(sp,fac)	
	Appr	GT	Appr	GT	Appr	GT
1	0.364	0.364	0.364	0.364	0.363	0.363
2	0.364	0.364	0.364	0.364	0.363	0.364
3	0.364	0.364	0.364	0.364	0.363	0.363
4	0.362	0.362	0.363	0.363	0.363	0.363
5	0.346	0.346	0.344	0.344	0.343	0.343
6	0.344	0.344	0.344	0.344	0.343	0.343
7	0.345	0.346	0.341	0.341	0.341	0.341
8	0.344	0.344	0.333	0.333	0.335	0.335
9	0.343	0.344	0.331	0.331	0.334	0.334
10	0.337	0.337	0.330	0.330	0.330	0.330
11	0.202	0.203	0.211	0.212	0.209	0.210
12	0.210	0.211	0.215	0.216	0.206	0.207
13	0.208	0.209	0.213	0.213	0.199	0.200
14	0.211	0.212	0.204	0.205	0.209	0.210
15	0.194	0.195	0.197	0.197	0.200	0.200
Overall	**0.303**	**0.303**	**0.301**	**0.301**	**0.300**	**0.300**

Initial TC	PV sp		PV fac		PV wi		FV	
	Appr	GT	Appr	GT	Appr	GT	Appr	GT
1	0.153	0.156	0.129	0.133	0.183	0.186	0.356	0.366
2	0.150	0.153	0.128	0.131	0.176	0.179	0.357	0.366
3	0.149	0.152	0.127	0.130	0.151	0.154	0.347	0.360
4	0.147	0.149	0.125	0.129	0.154	0.156	0.352	0.364
5	0.147	0.149	0.124	0.128	0.155	0.158	0.320	0.326
6	0.145	0.148	0.123	0.126	0.150	0.153	0.317	0.323
7	0.123	0.125	0.122	0.125	0.123	0.126	0.299	0.306
8	0.121	0.124	0.121	0.124	0.121	0.124	0.157	0.177
9	0.121	0.123	0.121	0.123	0.121	0.123	0.143	0.163
10	0.120	0.122	0.120	0.122	0.120	0.122	0.131	0.147
11	0.120	0.121	0.120	0.122	0.120	0.121	0.128	0.144
12	0.120	0.121	0.120	0.121	0.120	0.121	0.120	0.135
13	0.119	0.121	0.119	0.121	0.119	0.121	0.119	0.135
14	0.119	0.120	0.119	0.120	0.119	0.120	0.119	0.135
15	0.119	0.120	0.119	0.120	0.119	0.120	0.119	0.134
Overall	**0.132**	**0.134**	**0.123**	**0.125**	**0.137**	**0.139**	**0.226**	**0.239**

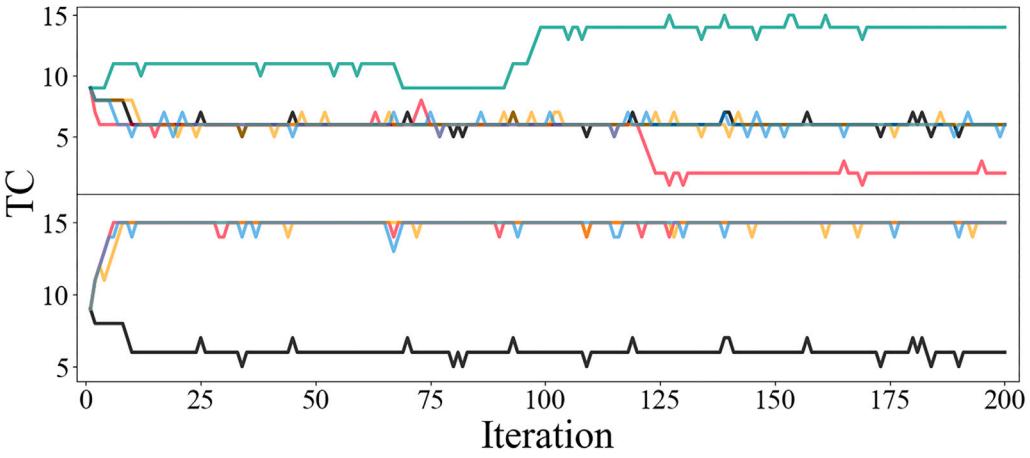

Fig. 7. Example of simulated temporal evolution of the system, driven by the *utility functions*. In both panels, the possible evolutions of the system based on the different learning approaches are compared to that relative to the ground-truth data. The same color-code of Fig. 4 is applied.

As noticeable, the trajectories of the system eventually diverge and evolve independently, unless the highest utility functions for the considered approach and for the ground-truth coincide for every TC.[2]

To assess how efficiently the proposed technique manages resource allocation during the evolution of the system, we simulated the process 100 times for each of the 15 possible initial TCs and for every learning approach, resulting in a total of $15 \times 100 \times 7 = 10{,}500$ simulations. For each simulation, the sum of the ground-truth (GT) utility functions corresponding to the decisions taken by each learning approach was compared with the sum of the maximum achievable GT utility functions. A smaller deviation between these two sums indicates a higher efficiency of the proposed mechanism. The resulting normalized cumulative utility metrics — obtained by dividing by both the highest possible MOS and the total number of steps per iteration (i.e., 5×200) — are summarized in Table 6, reported as averages per initial TC and as overall means across all TCs. The cumulative utilities tend to decrease for higher initial TCs. This behavior arises because the utilities associated with higher TCs are generally smaller, given that the resources are modeled to decrease linearly with TC, while the MOS values deteriorate more rapidly.

It is evident that the MV-based approaches consistently outperform both PV and FV methods across all initial conditions as well as in the overall averages. Among these, $\mathrm{MV}_{\mathrm{fac(sp,wi)}}$ achieves the best performance, with only a 0.07% deviation between its overall average cumulative utility and the ground-truth benchmark. Conversely, the FV approach yields the lowest performance, with a 5.5% difference, while the PV approaches exhibit intermediate behavior, showing percentual deviations ranging between 1.5% and 1.9%.

Finally, we emphasize that these specific results are highly dependent on the chosen shape of the utility-functions and probability values, and may vary under different configurations. Moreover, we are confident that the reliability of the proposed monitoring technique would benefit from a wider training set for the considered NNs.

7. Conclusion

This study introduced a comprehensive HDT architecture designed to continuously monitor and adapt to users' emotional states and perceived QoE, intending to enhance user experience in HDT-enabled environments. The proposed solution is intended to foster interoperability between the HDT and the service providers for the benefit of the overall service quality. The HDT is also designed to collaborate with other peers so that better prediction models can be developed. We have also proposed a multi-view training model that allows HDTs involved in the same applications but with different views to improve the estimation accuracy. We have demonstrated the benefits of this solution in two cases: user quality monitoring and resource optimization. Through extensive simulations we have shown that in the first case the developed approach allows for obtaining an estimation accuracy of 0.804 with respect to 0.767 achievable in the full view case, whereas in the second case the solution allows for improving the overall utility by 27% with respect to the scenarios without HDT collaboration.

Future research will focus on refining the model's adaptability, exploring additional modalities, and further evaluating its performance in diverse application scenarios. This work contributes to the advancement of HDT systems, underscoring their potential as a core component in the next generation of personalized HDT services.

CRediT authorship contribution statement

Matteo Fratta: Writing – review & editing, Writing – original draft, Validation. **Alessandro Floris:** Supervision, Methodology, Conceptualization. **Simone Porcu:** Writing – review & editing, Supervision. **Luigi Atzori:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Simone Porcu reports financial support was provided by European Union. Alessandro Floris reports was provided by European Union. Luigi Atzori reports financial support was provided by European Union. Matteo Fratta reports financial support was provided by European Union. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUP C29J24000300004, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART") and by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, "Sustainable Mobility Center" Centro Nazionale per la Mobilità Sostenibile, CNMS, CN_00000023.

[2] However, even in this ideal case, there remains a probability $(P_{\mathrm{intermediate}} + P_{\mathrm{lowest}})^2$ that the evolutions will still diverge.

Data availability

Data will be made available on request.

References

[1] J. Chen, C. Yi, S.D. Okegbile, J. Cai, X. Shen, Networking Architecture and Key Supporting Technologies for Human Digital Twin in Personalized Healthcare: A Comprehensive Survey, IEEE Commun. Surv. Tutor. 26 (1) (2024) 706–746, http://dx.doi.org/10.1109/COMST.2023.3308717.

[2] S.D. Okegbile, J. Cai, D. Niyato, C. Yi, Human digital twin for personalized healthcare: Vision, architecture and future directions, IEEE Netw. 37 (2023) 262–269, http://dx.doi.org/10.1109/MNET.118.2200071.

[3] M.W. Lauer-Schmaltz, P. Cash, J.P. Hansen, A. Maier, Designing human digital twins for behaviour-changing therapy and rehabilitation: A systematic review, Proc. Des. Soc. 2 (2022) 1303–1312, http://dx.doi.org/10.1017/pds.2022.132.

[4] A. Löcklin, T. Jung, N. Jazdi, T. Ruppert, M. Weyrich, Architecture of a human-digital twin as common interface for operator 4.0 applications, Procedia CIRP 104 (2021) 458–463, http://dx.doi.org/10.1016/j.procir.2021.11.077, 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0. URL https://www.sciencedirect.com/science/article/pii/S2212827121009756.

[5] M.W. Lauer-Schmaltz, P. Cash, J.P. Hansen, A. Maier, Designing human digital twins for behaviour-changing therapy and rehabilitation: a systematic review, Proc. Des. Soc. 2 (2022) 1303–1312.

[6] P. Zhang, Y. Su, J. Wang, C. Jiang, C.-H. Hsu, S. Shen, Reinforcement learning assisted bandwidth aware virtual network resource allocation, IEEE Trans. Netw. Serv. Manag. 19 (4) (2022) 4111–4123, http://dx.doi.org/10.1109/TNSM.2022.3199471.

[7] A. Bhattacharya, W. Wu, Z. Yang, Quality of experience evaluation of voice communication: an affect-based approach, "Human-Centric Comput. Inf. Sciences" 2 (1) (2012).

[8] S. Porcu, S. Uhrig, J.-N. Voigt-Antons, S. Möller, L. Atzori, Emotional impact of video quality: Self-assessment and facial expression recognition, in: 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX), 2019, pp. 1–6, http://dx.doi.org/10.1109/QoMEX.2019.8743186.

[9] G. Bingöl, S. Porcu, A. Floris, L. Atzori, Qoe estimation of webrtc-based audiovisual conversations from facial and speech features, ACM Trans. Multimed. Comput. Commun. Appl. 20 (5) (2024) http://dx.doi.org/10.1145/3638251.

[10] Y.F. Yeznabad, M. Helfert, G.-M. Muntean, Qoe-driven cross-layer bitrate allocation approach for MEC-supported adaptive video streaming, IEEE Trans. Netw. Serv. Manag. 21 (6) (2024) 6857–6874, http://dx.doi.org/10.1109/TNSM.2024.3453992.

[11] S. Ravindran, S. Chaudhuri, J. Bapat, D. Das, Novel adaptive multi-user multi-services scheduling to enhance throughput in 5G-advanced and beyond, IEEE Trans. Netw. Serv. Manag. 21 (2) (2024) 2323–2338, http://dx.doi.org/10.1109/TNSM.2024.3351669.

[12] S. Wang, J. Zhang, P. Wang, J. Law, R. Calinescu, L. Mihaylova, A deep learning-enhanced digital twin framework for improving safety and reliability in human–robot collaborative manufacturing, Robot. Comput.-Integr. Manuf. 85 (2024) 102608.

[13] A. Fawkes, D. Burden, Digital human twins and the military metaverse: opportunities and challenges, AI SOCIETY (2025) 1–13.

[14] M. Lu, Z. Hu, Digital twin-enhanced programming education: An empirical study on learning engagement and skill acquisition, Computers 14 (8) (2025) 322.

[15] A. Greco, M. Caterino, M. Fera, S. Gerbino, Digital twin for monitoring ergonomics during manufacturing production, Appl. Sci. 10 (21) (2020) 7758.

[16] B.R. Barricelli, E. Casiraghi, J. Gliozzo, A. Petrini, S. Valtolina, Human digital twin for fitness management, IEEE Access 8 (2020) 26637–26664.

[17] L. Zhao, S. Ni, D. Wu, L. Zhou, Cloud-edge-client collaborative learning in digital twin empowered mobile networks, IEEE J. Sel. Areas Commun. 41 (11) (2023) 3491–3503, http://dx.doi.org/10.1109/JSAC.2023.3310060.

[18] X.S. Shen, X. Huang, J. Xue, C. Zhou, X. Shi, W. Zhuang, Revolutionizing qoe-driven network management with digital agents in 6G, IEEE Commun. Mag. (2025) 1–8, http://dx.doi.org/10.1109/MCOM.001.2400679.

[19] S. Subramanian, K. De Moor, M. Fiedler, K. Koniuch, L. Janowski, Towards enhancing ecological validity in user studies: a systematic review of guidelines and implications for QoE research, Qual. User Exp. 8 (2023) http://dx.doi.org/10.1007/s41233-023-00059-2.

[20] L. Xia, Y. Sun, C. Liang, L. Zhang, M.A. Imran, D. Niyato, Generative AI for semantic communication: Architecture, challenges, and outlook, IEEE Wirel. Commun. 32 (1) (2025) 132–140, http://dx.doi.org/10.1109/MWC.003.2300351.

[21] L.B. Makai, P. Varga, Predicting mobility management demands of cellular networks based on user behavior, in: NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, 2023, pp. 1–6, http://dx.doi.org/10.1109/NOMS56928.2023.10154215.

[22] A.J.H. Redelinghuys, A.H. Basson, K. Kruger, A six-layer architecture for the digital twin: a manufacturing case study implementation, J. Intell. Manuf. 31 (6) (2020) 1383–1402, http://dx.doi.org/10.1007/s10845-019-01516-6.

[23] M.C. Hlophe, B.T. Maharaj, From cyber–physical convergence to digital twins: A review on edge computing use case designs, Appl. Sci. 13 (24) (2023) http://dx.doi.org/10.3390/app132413262, URL https://www.mdpi.com/2076-3417/13/24/13262.

[24] ITU, Methods for Subjective Determination of Transmission Quality, Recommendation ITU-T P.800, 1996.

[25] M. Hamidi, G. Bingöl, A. Floris, S. Porcu, L. Atzori, Analysis of application-layer data to estimate the QoE of webrtc-based audiovisual conversations, in: 2023 IEEE Globecom Workshops (GC Wkshps), 2023, pp. 365–370, http://dx.doi.org/10.1109/GCWkshps58843.2023.10464821.

[26] G. Bingol, S. Porcu, A. Floris, L. Atzori, Webrtc-qoe: A dataset of QoE assessment of subjective scores, network impairments, and facial & speech features, Comput. Netw. 244 (2024) 110356, http://dx.doi.org/10.1016/j.comnet.2024.110356.

[27] H. Zhang, L. Dong, G. Gao, H. Hu, Y. Wen, K. Guan, DeepQoE: A multimodal learning framework for video quality of experience (QoE) prediction, IEEE Trans. Multimed. 22 (12) (2020) 3210–3223, http://dx.doi.org/10.1109/TMM.2020.2973828.

[28] S. Porcu, A. Floris, L. Atzori, A clustered federated learning approach for estimating the quality of experience of web users, in: 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops, ICASSPW, IEEE, 2023, pp. 1–5.

[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, Artificial intelligence and statistics (2017) 1273–1282.