Sustainable Edge Deployment of Consumer-Centric Digital Twins for Healthcare 5.0: A Resource-Efficient Optimization Framework

Jialin Li, Rahul Kumar, Sachin Kumar, *Member, IEEE*, Keqin Li, *Fellow, IEEE*, and Jianhui Lv, *Senior Member, IEEE*

***Abstract*—Healthcare 5.0 emerges as a transformative paradigm positioning patients as active participants through consumer-centric digital twin technologies that enable proactive health management and personalized interventions. Consumer-centric digital twins (CCDTs) function as high-fidelity virtual patient replicas continuously processing physiological data streams from wearable sensors and medical devices to predict health trajectory deviations and recommend timely corrective actions. While patient CCDTs exhibit heterogeneous requirements for processing electrocardiograms, glucose monitors, and mobility data. This study formulates sustainable CCDT deployment as a binary integer programming problem minimizing total system latency under computational capacity, storage limits, synchronization requirements, and collaboration needs, proving its NP-hardness. The proposed sustainable patient-centric edge deployment (SPCED) algorithm combines branch-and-bound initialization through linear relaxation with matching-theory refinement using move and swap operations, achieving bilateral exchange stability. Simulation results demonstrate that SPCED outperforms six baseline algorithms, including state-of-the-art decentralized federated learning approaches, achieving 99.81% computational efficiency improvement while maintaining superior latency performance across varying patient populations, resource configurations, and collaboration patterns, validating its effectiveness for sustainable healthcare 5.0 CCDT ecosystems.**

***Index Terms*—Healthcare 5.0, consumer-centric digital twins, edge computing, matching theory.**

I. Introduction

THERE is still a strain on the healthcare sector of the world due to the increased ageing, the upsurge in the prevalence of chronic illnesses, and the escalation of medical bills [1]. Existing models of reactive medicine, which intervene when the symptoms are clear, are insufficient in managing such problems. Healthcare 5.0 is another trend that has received an opportunity to reconsider the proactive form of health management strategy in contrast to the disease treatment, with the prospect of successful connectivity between artificial intelligence, Internet of Medical Things (IoMT), and big data analytics and robotics [2], [3]. In contrast to previous versions of healthcare, healthcare 5.0 makes the patient the focal point of their patient journeys with real-time physiological data and predictive analytics that provide personalized interventions before health-damaging events get underway [4]. Such a consumerist notion only reinvents the doctor-patient relationship, turning passive consumers into active stakeholders who work together with experts in their respective fields of healthcare by undergoing continuous digitalization [5].

The consumer-centric digital twins (CCDTs) are the backbone technology of actualizing the vision of healthcare 5.0 [6], [7]. A patient CCDT is a digital copy of its real-world analog, and perceives a stream of wearable sensor biosignals, including heart rate, lung and skin temperatures, medical confinement, and world monitors in real-time [8]. In addition to direct data aggregation, CCDTs model the physiological dynamics with strong computational algorithms to estimate health trajectory deviations and provide custom requirements that are prescribed. In the instance of a patient with diabetes, the CCDT is constantly processing the glucose sensor data, nutrition records, exercise, and analyzes the stress records, predicts the glycemic events before their occurrence, thereby allowing corrective action to be taken in time. Equally, CCDTs assist cardiac patients by offering the potential to identify the antecedents of arrhythmia through the identification of the insidious shifts in the recorded waveforms of the electrocardiogram, the alterations in the heart rate, and the blood pressure pattern [9]. Such predictable abilities make healthcare a sequence of ongoing association whereby the groups of medical professionals will observe the CCDTs of the patients to determine possible threats and prepare in advance.

Latencies bottlenecks created by the utilization of patient CCDTs of cloud infrastructure degrade time-sensitive healthcare applications [10]. In situations when the patient is receiving such emergency symptoms as ectoderm pain or drastic hypoglycemia, the CCDT must be capable of rapid processing of arriving physiological information, allowing more complicated diagnostic routines and relaying surgical instructions to the emergency room or family [11]. Placing everything on clouds requires this data to go over long

Received 15 October 2025; revised 29 December 2025; accepted 12 February 2026. Date of publication 2 March 2026; date of current version 2 June 2026. *(Corresponding author: Jianhui Lv.)*

Jialin Li is with the Department of Pathology, The First Affiliated Hospital of Jinzhou Medical University, Jinzhou 121001, China (e-mail: lijialin@stu.jzmu.edu.cn).

Rahul Kumar is with the Department of Mathematics, S.S.V P.G. College, Hapur 245101, India (e-mail: ujjwalrahul@gmail.com).

Sachin Kumar is with the Department of Data Science, Galgotias College of Engineering and Technology, Greater Noida 201310, India (e-mail: imsachingupta@rediffmail.com).

Keqin Li is with the College of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Jianhui Lv is with the Multi-Modal Data Fusion and Precision Medicine Laboratory, The First Affiliated Hospital of Jinzhou Medical University, Jinzhou 121001, China (e-mail: lvjh@jzmu.edu.cn).

Digital Object Identifier 10.1109/TCE.2026.3669402

distances on network paths over congested backhaul connections, and this may take hundreds of milliseconds or even seconds to respond [12]. This may not be accepted in such application domains as remote surgery and robots, where the surgeon can only order the patient to act in milliseconds, or emergency surgery, where the patient is in a critical situation and needs to be taken to the correct facility, depending on real-time consideration of the patient's condition [13]. The limitations discussed above can be circumvented in the network computing architectures, which spread out the computational capacities into network peripheries residing near hospitals, clinics, and patient homes [14]. CCDTs placed on healthcare edge servers reduce patient-CCDT synchronization delay by hundreds of milliseconds to tens of milliseconds and support real-time operation of high-level runtime applications without compromising patient data privacy by local processing [15].

The conditions demanded to operate healthcare edge servers are highly demanding, making optimization of the application of CCDT very difficult. In the majority of cases, financial capacity and physical products of the clinical environment make the edge infrastructure of the medical institutions less powerful in computational tasks and data storage than the hyperscale cloud facilities. Their resources needed by CCDTs patients are uneven, based on the underlying health disease, and the execution of monitoring needs. During the postoperative care, a cardiac surgery patient also needs a CCDT that has tremendous computing ability to facilitate a high-frequency electrocardiogram, arterial pressure, and ventilating parameters data processing with a high degree of computationally intensive algorithms to identify surgical complications. On the other hand, a weak elderly customer threatened by falls and needing to be tracked must have a lighter-weight CCDT, which can measure accelerator force and GPS positions and deduce comparatively small computing energy [16]. In the meantime, CCDTs have diverse requirements regarding the medical data repository to hold medical history, medication history, radiological studies, and genomic profiles [17]. Such multi-dimensional resource needs pose a challenge to the deployment algorithms to provide appropriate matching of patient CCDTs and edge servers based on CPU, memory, storage, and energy capitals [18].

The implementation of CCDT is even harder by virtue of inter-patient dynamics that occur in a joint healthcare environment [19]. In the modern medical practice, the patient outcomes are increasingly coming to be regarded not merely as dependent upon the physiological condition of the individual characters, but also through the support groups and shared care organization [20]. When the elderly family members in CCDTs are overseeing their own, answering the questions of how well all the issues of medication adherence are met, mobility issues are addressed. Mental abilities are discussed among themselves, and a plan is worked out; it is an advantageous case. Being a part of the support circles of chronic conditions, the patients interpret the worth in the anonymized treatment rates and advice on how to cope with the symptoms provided by their CCDTs [21]. Numerous patients with related conditions can be provisioned in clinical care teams whose patient CCDTs are employed on parallel-edge servers and can quickly interact with one another without traversing the network infrastructure [22]. All these interactions imply that the algorithms employed to implement CCDT will not be able to ignore inter-patient CCDT communication latency, but rather that of patient-CCDT synchronization [23]. Placing the groups of patients CCDTs of the group on the same edge servers close to each other decreases the latency of the interaction between the CCDTs and improves their performance in coordinating the care [24]. The achievement of such patient-CCDT synchronization demands interaction, and co-work interaction implies demand and multi-faceted resource requirements, which would necessitate sophisticated methods in algorithm development [25].

The main contributions of this study are as follows.

1) We design a hierarchical healthcare 5.0 edge architecture deploying patient CCDTs near medical facilities while accounting for patient-CCDT synchronization latency, inter-patient collaboration requirements, and heterogeneous multi-dimensional resource demands.
2) We formulate sustainable CCDT deployment as binary integer programming under computational capacity, medical data storage, and maximum synchronization latency constraints, prove its NP-hardness, and analyze structural properties guiding algorithm design.
3) We propose the sustainable patient-centric edge deployment (SPCED) algorithm, combining branch-and-bound initialization through linear relaxation with matching-theory refinement using move and swap operations, achieving bilateral exchange stability.

The remaining of this article is structured as follows. Section II establishes the system model and problem formulation. Section III describes the proposed SPCED algorithm addressing the NP-hard deployment problem through a two-phase approach. Section IV evaluates SPCED performance through simulations. Section V concludes the paper with key findings and future research directions.

II. System Model and Problem Formulation

A. Healthcare Edge Network Architecture

Fig. 1 denotes a healthcare 5.0 edge network architecture that we consider because the architecture comprises two hierarchical layers. Wearable electrocardiogram devices, continuous glucose levels, pulse oximeters, activity trackers, and smart medication dispensers are a few examples of different types of medical sensors and IoMT devices, which fall under the lower physical layer. This kind of patients' lives within the coverage areas served by edge infrastructure that is either physically located in hospitals or community health facilities, or are edge facilities that are located and geographically close to their homestead. They are literally cloned in the upper layer of a CCDT, and all the CCDTs are constantly communicating with the entering health-associated data in order to maintain a computerized representation of the patient in check.

The healthcare edge network encompasses $\mathrm{E} = \{1, 2, \ldots, E\}$ edge servers, each collocated with wireless access points that

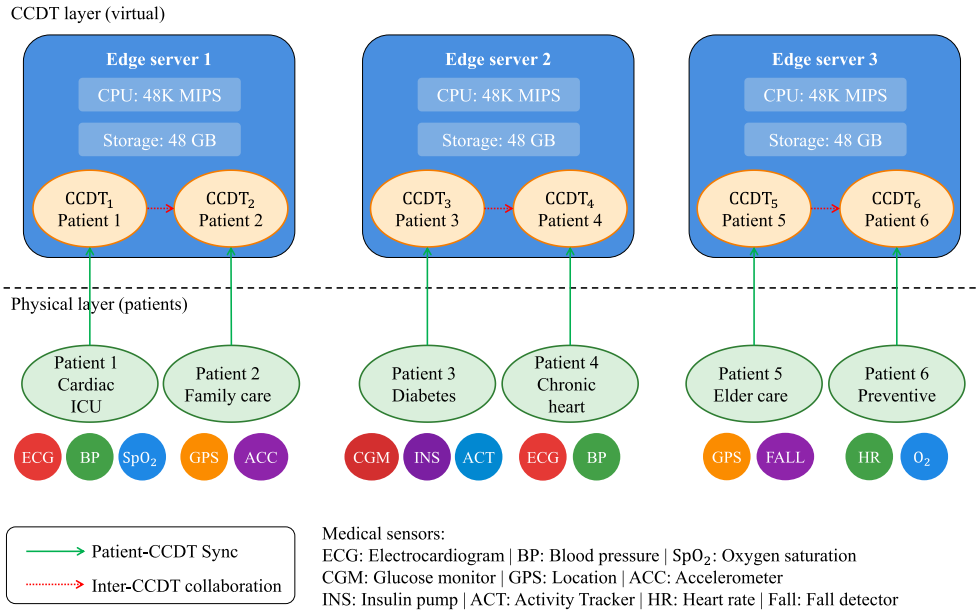

Fig. 1. Healthcare 5.0 CCDT edge network architecture.

provide connectivity to patients within their coverage zones. Edge server e possesses finite computational capacity measured in processing cycles per second and limited medical data storage measured in gigabytes. Within the coverage regions, $\mathrm{P} = \{1, 2, \ldots, P\}$ denotes the set of patients, each requiring a CCDT deployed on exactly one edge server. Patients remain geographically stationary during the deployment horizon, representing scenarios such as inpatient hospital monitoring, residential elder care, or chronic disease management, where patients maintain stable locations.

The stationary patient assumption represents deployment epochs between mobility events rather than perpetual fixed locations. Patients transitioning between hospital departments or returning home after discharge trigger CCDT redeployment, where the algorithm executes with updated patient-edge proximities. Mobile health monitoring scenarios require migration mechanisms transferring CCDT state between edge servers as patients move across coverage zones. Migration latency and bandwidth consumption represent additional optimization objectives beyond initial deployment. Frequent recomputation addresses mobility by treating patient location updates as triggering conditions for incremental redeployment rather than full population rescheduling, where only affected patients undergo reassignment.

The structure is shown in Fig. 1 as two layers. At the physical layer, patients are connected to edge infrastructure by a type of medical sensors that relay healthy information to patients via wireless connections. The CCDT layer would signal the virtual patients, which are currently executing in edge servers that interpret the streaming data received to ensure patient models are brought in line. The patient-CCDT synchronization data flow and inter-patient CCDT collaboration data flow, as illustrated by solid or dashed arrows respectively, are depicted in the figure below.

We define binary variable ϕ_{ij} to indicate collaborative relationships:

$$\phi_{ij} = \begin{cases} 1 & \text{if patient } i \text{ and patient } j \\ & \text{require CCDT collaboration} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

CCDT implementation is done in stages. In wireless uplink channels, the patients fix the stored health information to the edge infrastructure and then broadcast it. Base stations convert this medical data into wired backhaul connections to allocated edge servers. Policies build complex algorithms on the received data, such as machine learning models to forecast diseases, physiological simulation engines to forecast health courses, and clinical decision support systems to recommend interventions. Edge servers then compile and maintain patient CCDTs. Beyond performance optimization, CCDT deployments must satisfy regulatory frameworks governing medical data protection. Edge-based processing inherently reduces privacy risks by maintaining patient information within local healthcare facilities rather than transmitting it to distant cloud servers. Deployment decisions can incorporate privacy zones where certain patient groups with heightened confidentiality requirements receive preferential placement on dedicated edge infrastructure. Such privacy-aware deployment extends traditional latency minimization by weighting edge server selection according to institutional data governance policies and patient consent pReferences. The received products, health risk assessment, prescription modification, and lifestyle recommendations are annotated back to the patients via the downlink wireless messages. These personalized insights assist patients in changing their own behavior, in interacting with caregivers, and with care providers.

The wireless channel capacity between patient p and edge server e follows:

$$\gamma_{pe} = \Omega \log_2 \left(1 + \frac{\Theta_p \varsigma_{pe}}{\Omega \Psi_0}\right). \tag{2}$$

where Ω represents the allocated wireless channel bandwidth, Θ_p denotes patient p's transmission power, Ψ_0 indicates the noise power spectral density, and ς_{pe} characterizes the wireless channel power gain between patient p and edge server e. The channel gain typically follows path loss models $\varsigma_{pe} = \delta_{pe}^{-\nu}$, where δ_{pe} measures the physical distance and ν represents the path loss exponent that varies with environmental propagation characteristics. Urban hospital settings exhibit higher path loss exponents due to building structures and medical equipment interference, while rural health facilities experience lower values.

Practical wireless channels experience random fading and interference, requiring reliability mechanisms. Medical sensors employ automatic repeat request protocols retransmitting lost packets until acknowledgment, where effective transmission latency increases with channel error rates. Forward error correction adds redundancy to transmitted data, reducing retransmission frequency at the cost of increased payload size. The path loss model represents average channel conditions, while instantaneous capacity fluctuates around this mean. Priority queuing at wireless access points gives medical data precedence over non-critical traffic, maintaining low latency despite network congestion. Channel quality indicators allow patient devices to adjust transmission power dynamically, improving reliability during poor propagation conditions.

Medical data transmission employs encryption protocols that protect patient confidentiality during wireless communication. Security mechanisms introduce overhead where encrypted payload sizes exceed raw medical data volumes, reducing effective channel capacity. Authentication handshakes between patient devices and edge infrastructure add latency to initial connection establishment. The wireless channel model implicitly accounts for security overhead through effective bandwidth Ω, which represents available capacity after protocol overhead. Lightweight encryption algorithms suitable for resource-constrained medical sensors balance confidentiality requirements with computational limitations. Edge infrastructure maintains certificate authorities validating patient device identities before accepting physiological data streams.

Given medical data volume Λ_p accumulated by patient p between synchronization intervals, the uplink transmission latency to edge server e becomes:

$$\mathrm{T}_{pe}^{\mathrm{sync}} = \frac{\Lambda_p}{\gamma_{pe}}. \tag{3}$$

Following successful data transmission, edge server e must allocate computational resources to execute patient p's CCDT algorithms. These algorithms vary substantially across patient populations and health conditions. Intensive care patients require CCDTs that process high-resolution electrocardiogram waveforms through deep learning arrhythmia detection models, which demand substantial computational cycles per data sample. Chronic disease patients need CCDTs running pharmacokinetic models and treatment optimization algorithms with moderate computational intensity. Healthy elderly patients monitored for fall risks utilize lightweight CCDTs with modest processing requirements. We denote μ_p as the computational intensity measured in CPU cycles required to process each bit of patient p's medical data. Computational intensity varies with patient health status across monitoring periods. Diabetic patients experiencing glucose instability require frequent prediction model execution, while stable glycemic control reduces computational frequency. Post-surgical patients progress from intensive monitoring with high-frequency vital sign processing toward lighter surveillance as recovery advances. CCDT deployment optimization operates over temporal horizons where computational requirements remain relatively stable, typically ranging from hours to days. Significant health status changes trigger redeployment evaluation, where algorithms reassess patient-server assignments based on updated resource requirements. Time-varying demands motivate periodic reoptimization rather than assuming perpetual static resource needs throughout patient monitoring. The CCDT computation latency becomes:

$$\mathrm{T}_{pe}^{\mathrm{proc}} = \frac{\mu_p \Lambda_p}{\rho_{pe}}. \tag{4}$$

where ρ_{pe} represents the computational resources allocated to patient p's CCDT on edge server e, measured in processing cycles per second.

Collaborative care scenarios introduce additional latency from inter-patient CCDT communication. When patient p's CCDT hosted on edge server e must exchange medical insights with patient q's CCDT on edge server f, the collaboration latency depends on the network distance. CCDTs deployed on the same edge server communicate through fast shared memory without network traversal, while CCDTs on different servers require data transmission through inter-edge network links. We model inter-edge communication latency proportional to physical infrastructure distance:

$$\mathrm{T}_{peqf}^{\mathrm{collab}} = z_{pe} z_{qf} \phi_{pq} \Xi_{ef}. \tag{5}$$

where $\Xi_{ef} = \kappa \cdot \delta(e, f)$ represents latency between edge servers e and f, with κ as the latency-distance mapping coefficient and $\delta(e, f)$ measuring physical separation between edge facility locations. The binary deployment variable z_{pe} indicates CCDT placement:

$$z_{pe} = \begin{cases} 1 & \text{if patient } p\text{'s CCDT deploys on edge server } e \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Inter-edge communication experiences variable latency beyond static distance-based models. Network congestion during peak hours increases packet delays and reduces effective bandwidth between edge servers. Alternative routing paths activated during link failures exhibit different latency characteristics than primary connections. QoS mechanisms can prioritize medical traffic, maintaining low latency for CCDT collaboration despite background network loads. Dynamic latency estimation through periodic probing between edge infrastructure enables deployment decisions reflecting current network conditions rather than idealized distance calculations. Robust deployment considers worst-case latency bounds,

ensuring collaboration requirements remain satisfied despite normal network variability.

Notably, when collaborating patients' CCDTs deploy on identical edge servers ($e = f$), $\Xi_{ee} = 0$ reflects negligible intra-server communication latency. Combined patient-CCDT synchronization latency becomes:

$$\tau_{pe} = \mathrm{T}^{\mathrm{sync}}_{pe} + \mathrm{T}^{\mathrm{proc}}_{pe}. \tag{7}$$

The total healthcare 5.0 CCDT ecosystem latency aggregates all synchronization and collaboration components:

$$\mathrm{T}_{\mathrm{total}} = \sum_{p\in\mathbf{p}}\sum_{e\in\mathrm{E}} z_{pe}(\mathrm{T}^{\mathrm{sync}}_{pe} + \mathrm{T}^{\mathrm{proc}}_{pe}) + \sum_{p\in\mathbf{p}}\sum_{e\in\mathrm{E}}\sum_{q\in\mathbf{p}}\sum_{f\in\mathrm{E}} \mathrm{T}^{\mathrm{collab}}_{peqf}. \tag{8}$$

B. Problem Formulation for Sustainable CCDT Deployment

The following framework of sustainable CCDT deployment aims to reduce the overall ecosystem delay due to the optimal location of patient CCDTs on healthcare edge servers, in diverse resource constraints and quality-of-service targets. CCDT requires heterogeneous computational and storage services for every patient, reflective of the complexity of health monitoring. ICU patients demand CCDTs whose processing capabilities are large enough to compute electrocardiograms, arterial waveforms, and ventilator settings through computationally intensive algorithms, and also to store high-resolution time-series data.

We denote α_p as the computational resources required by patient p's CCDT, measured in processing cycles per second, and β_p as the medical data storage requirement, measured in gigabytes. Storage allocations represent active CCDT working sets rather than complete electronic health records. CCDTs maintain recent physiological measurements spanning monitoring windows of days to weeks, sufficient for real-time health trajectory prediction. High-resolution electrocardiograms sampled at 500 Hz generate approximately 100 MB daily for continuous monitoring. Glucose measurements every fifteen minutes accumulate less than 10 MB weekly. Historical medical records, radiology archives, and genomic data reside in hospital data repositories, accessed by CCDTs through queries rather than local replication. Active storage focuses on recent data, enabling immediate prediction, while archival information remains in centralized databases, reducing edge storage requirements to operationally relevant datasets. Edge server e possesses total computational capacity $\mathrm{C}^{\mathrm{cpu}}_e$ and storage capacity $\mathrm{C}^{\mathrm{stor}}_e$. Sustainable edge infrastructure must account for electrical power consumption alongside computational and storage resources. CCDTs executing intensive algorithms for arrhythmia detection consume substantially more energy than lightweight fall monitoring applications. Edge servers possess maximum power budgets determined by electrical infrastructure capacity and cooling system limitations. We denote energy consumption rate per MIPS as φ, where total power draw from patient deployments must satisfy constraint $\sum_{p\in\mathrm{P}} z_{pe}\alpha_p\varphi \le E^{\mathrm{power}}_e$ for each edge server $e \in \mathrm{E}$. Energy-aware deployment balances computational performance with environmental impact and operational costs, particularly relevant for battery-powered mobile edge nodes in ambulatory care settings. Sustainable healthcare 5.0 operation requires that aggregate resource demands from all deployed CCDTs never exceed available edge server capacities, preventing resource contention that would degrade medical monitoring quality. The sustainable CCDT deployment optimization problem becomes:

$$\begin{aligned}
\mathbf{P1}:\ & \min_{\mathbf{z}} \mathrm{T}_{\mathrm{total}} \\
\text{s.t. } \mathbf{C1}:\ & \tau_{pe} \le \mathrm{L}_{\max},\ \forall p \in \mathrm{P}, \forall e \in \mathrm{E} \\
\mathbf{C2}:\ & \sum_{p=1}^{P} z_{pe}\alpha_p \le \mathrm{C}^{\mathrm{cpu}}_e,\ \forall e \in \mathrm{E} \\
\mathbf{C3}:\ & \sum_{p=1}^{P} z_{pe}\beta_p \le \mathrm{C}^{\mathrm{stor}}_e,\ \forall e \in \mathrm{E} \\
\mathbf{C4}:\ & \sum_{e\in\mathrm{E}} z_{pe} = 1,\ \forall p \in \mathrm{P} \\
\mathbf{C5}:\ & z_{pe}, z_{qf} \in \{0,1\},\ \forall p,q \in \mathrm{P}, \forall e,f \in \mathrm{E}.
\end{aligned} \tag{9}$$

At constraint C1, the fix to patient-CCDT is guaranteed with the latency of the response to the sensitive healthcare applications. C2 constraint will not saturate recalculations, and the total CPU requirement of deployed CCDTs will not burden the edge server. Constraint C3 also requires storage capacity such that the edge servers will possess the necessary storage capacity to ensure that all the medical information of patients and CCDT models do not exceed the constraints. Fair CCDT deployment ensures equitable service quality across patient diversity rather than optimizing aggregate metrics alone. Intensive care patients with high resource demands should not experience systematically higher latencies compared to preventive care cases. Fairness constraints can enforce maximum latency ratios between patient groups, preventing resource-efficient deployments that disadvantage clinically vulnerable populations. Alternative formulations minimize maximum individual latency rather than total system latency, prioritizing worst-case patient experiences. Fairness-aware deployment balances utilitarian efficiency with egalitarian principles, reflecting healthcare ethics emphasizing patient equity. Healthcare institutions may weigh patient priorities based on medical urgency, where life-threatening conditions receive preferential resource allocation. Constraint C4: Every given patient C4 specifies that each given patient CCDT is deployed on at most one edge server and thus has no split deployments and thus no split-up medical records. Binary decision variables are provided in constraint C5.

Patient-centric Healthcare 5.0 can incorporate individual preferences within deployment optimization. Some patients prefer CCDTs hosted at primary care facilities where established provider relationships exist rather than distant edge infrastructure. Family caregivers may request colocation on common edge servers, facilitating coordinated monitoring. Privacy-conscious patients might specify deployment constraints limiting CCDT hosting to particular institutional affiliations. Preference-aware deployment extends latency minimization by adding soft constraints or preference penalties within the objective function. Patient preference weights balance individual desires with system-wide efficiency, where

strong preferences override marginal latency improvements while weak preferences defer to optimal technical placement.

Multi-objective formulations extend single-objective latency minimization by incorporating additional performance dimensions. Energy-aware deployment minimizes computational power consumption alongside latency, where Pareto-optimal solutions reveal trade-off curves between response time and sustainability. Cost objectives account for heterogeneous edge server operational expenses, where premium infrastructure with lower latency incurs higher deployment costs. Fairness objectives prevent worst-case patient experiences from degrading excessively despite efficient aggregate performance. Weighted scalarization combines objectives through preference coefficients reflecting institutional priorities. Healthcare decision-makers specify weight parameters balancing competing goals based on organizational values, patient population needs, and resource constraints.

Problem P1 constitutes a binary integer programming formulation that proves NP-hard through reduction from the multi-dimensional knapsack problem [26]. Healthcare edge servers act as knapsacks with multiple capacity dimensions (CPU, storage), while patient CCDTs represent items with heterogeneous multi-dimensional resource demands and collaboration-dependent values. The collaboration terms in Eq. (8) introduce quadratic coupling between deployment variables, further complicating solution approaches. Exact optimization methods using exhaustive enumeration require evaluating E^P possible deployment configurations, becoming computationally prohibitive even for modest healthcare networks with tens of patients and edge servers [27]. Commercial integer programming solvers achieve optimality for small problem instances but exhibit exponential runtime growth that renders them impractical for realistic healthcare 5.0 deployments [28]. These computational challenges motivate the development of efficient heuristic algorithms that obtain high-quality deployment solutions within practical timeframes [29].

III. Sustainable Patient-Centric Edge Deployment Algorithm

A. Branch-and-Bound Initialization for CCDT Deployment

Branch-and-bound algorithms perform well in what the literature terms mixed integer programs that have restricted integer programming problems with a problem space of less scale, zealously exploring the solution space systematically in a methodical way, through partitioning and pruning. Nonetheless, a very large space of searches is comprised of any large-scale healthcare 5.0 network with a high population of patients and edge servers. To solve this numerical difficulty, we are solving the relaxed formulation of a linear program that provides preliminary advice, and we are searching with limited use of branch-and-bound to obtain a satisfactory integer solution.

The collaboration latency terms in Eq. (5) exhibit quadratic structure due to products of binary deployment variables $z_{pe}z_{qf}$. This nonlinearity prevents direct application of linear programming solvers. Initial solution quality depends on linear programming relaxation tightness and branching heuristics. Relaxation solutions with minimal fractional variables provide better integer solutions upon rounding compared to highly fractional relaxations. Constraint tightness influences relaxation quality, where additional valid inequalities strengthen linear bounds. Branching variable selection, prioritizing patient-server pairs with the largest fractional values, often improves convergence speed. Warm-start techniques, initializing branch-and-bound with proximity-based solutions, can accelerate convergence when geographic structure provides good starting points. Understanding these factors enables adaptive initialization, selecting appropriate strategies based on problem characteristics, such as patient density or collaboration intensity. We introduce auxiliary binary variables η_{peqf} that linearize the problem:

$$\eta_{peqf} = \begin{cases} 1 & \text{if } z_{pe} = 1 \text{ and } z_{qf} = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Substituting η_{peqf} for the product term, Eq. (5) becomes:

$$\mathrm{T}^{\mathrm{collab}}_{peqf} = \eta_{peqf}\mathrm{Y}_{peqf}. \tag{11}$$

where $\mathrm{Y}_{peqf} = \phi_{pq}\Xi_{ef}$ consolidates the collaboration indicator and inter-edge latency. The reformulated total latency becomes:

$$\begin{aligned} \mathrm{T}_{\mathrm{total}} = &\sum_{p\in\mathrm{P}}\sum_{e\in\mathrm{E}} z_{pe}(\mathrm{T}^{\mathrm{sync}}_{pe} + \mathrm{T}^{\mathrm{proc}}_{pe}) \\ &+ \sum_{p\in\mathrm{P}}\sum_{e\in\mathrm{E}}\sum_{q\in\mathrm{P}}\sum_{f\in\mathrm{E}} \eta_{peqf}\mathrm{Y}_{peqf}. \end{aligned} \tag{12}$$

The auxiliary variables must satisfy logical constraints ensuring consistency with deployment variables:

$$\begin{aligned} &\eta_{peqf} \le z_{pe}, \ \forall p,q \in \mathrm{P}, \forall e,f \in \mathrm{E} \\ &\eta_{peqf} \le z_{qf}, \ \forall p,q \in \mathrm{P}, \forall e,f \in \mathrm{E} \\ &\eta_{peqf} \ge z_{pe} + z_{qf} - 1, \ \forall p,q \in \mathrm{P}, \forall e,f \in \mathrm{E}. \end{aligned} \tag{13}$$

These constraints enforce that η_{peqf} equals one only when both z_{pe} and z_{qf} equal one. Problem P1 reformulates as:

$$\begin{aligned} \mathbf{P2}: \ &\min_{\mathbf{z},\boldsymbol{\eta}} \mathrm{T}_{\mathrm{total}} \\ &\text{s.t. } \mathbf{C1} \sim \mathbf{C5} \\ &\quad \mathbf{C6}: \ \eta_{peqf} \le z_{pe}, \ \forall p,q \in \mathrm{P}, \forall e,f \in \mathrm{E} \\ &\quad \mathbf{C7}: \ \eta_{peqf} \le z_{qf}, \ \forall p,q \in \mathrm{P}, \forall e,f \in \mathrm{E} \\ &\quad \mathbf{C8}: \ \eta_{peqf} \in \{0,1\}, \ \forall p,q \in \mathrm{P}, \forall e,f \in \mathrm{E}. \end{aligned} \tag{14}$$

Constraints C6 and C7 establish relationships between auxiliary and deployment variables, while constraint C8 maintains binary variable domains.

Branch-and-bound search begins by solving a linear programming relaxation where binary variables become continuous over $[0, 1]$ intervals:

$$\begin{aligned} \mathbf{P}3: \ &\min_{\mathbf{z},\boldsymbol{\eta}} \mathrm{T}_{\mathrm{total}} \\ &\text{s.t. } \mathbf{C1} \sim \mathbf{C}4 \\ &\quad \mathbf{C5}: \ z_{pe}, z_{qf} \in [0,1], \ \forall p,q \in \mathrm{P}, \forall e,f \in \mathrm{E} \\ &\quad \mathbf{C6}: \ \eta_{peqf} \le z_{pe}, \ \forall p,q \in \mathrm{P}, \forall e,f \in \mathrm{E} \end{aligned}$$

$$\mathbf{C7}: \ \eta_{peqf} \le z_{qf}, \ \forall p, q \in \mathrm{P}, \forall e, f \in \mathrm{E}$$
$$\mathbf{C8}: \ \eta_{peqf} \in \{0, 1\}, \ \forall p, q \in \mathrm{P}, \forall e, f \in \mathrm{E}. \quad (15)$$

Slack problem P3 can be transformed into a standard linear problem that is comfortable to solve using simplex or interior-point to obtain an optimal solution with fractions of the variables. This relaxation gives a perfect aim to the original problem.

B. Matching-Theory-Based CCDT Deployment Refinement

Problem P1 will be a many-to-one matching model whereby patient CCDTs are matched to edge servers with multiple patients relating to one server due to capacity concerns. The peer effect of the collaborative needs of interdependence also forms peer effects in which a patient's CCDT utility of collaborating with other patients is conditional on the location of the peers. Matching theory allows us to optimize the initial deployment, running local search operations that are aware of such interdependencies.

Before presenting the refinement algorithm, we establish matching model notation and definitions.

Many-to-one matching with capacity constraints: A many-to-one matching function Ω maps from set $\mathrm{P} \cup \mathrm{E} \cup \{\emptyset\}$ to itself, satisfying for all $p \in \mathrm{P}$ and $e \in \mathrm{E}$:

1) $|\Omega(p)| = 1$ for all $p \in \mathrm{P}$, and if $\Omega(p) \notin \mathrm{E}$ then $\Omega(p) = \emptyset$.
2) $|\Omega(e)| \le q_{\max}$ for all $e \in \mathrm{E}$, where $q_{\max}$ represents maximum patients assignable to edge server e given resource constraints, and if p is not matched to any e then $\Omega(e) = \emptyset$.
3) $\Omega(p) = e$ if and only if $p \in \Omega(e)$.

Condition 1 ensures each patient CCDT deploys on at most one edge server. Condition 2 enforces edge server capacity limits, where $q_{\max}$ derives from multi-dimensional resource constraints (CPU, storage). Condition 3 establishes matching symmetry and transitivity.

The optimization objective seeks to minimize total healthcare 5.0 ecosystem latency, defined as the utility function:

$$\mathrm{U}(\Omega) = \sum_{p \in \mathrm{P}} \sum_{e \in \mathrm{E}} (\mathrm{T}^{\mathrm{sync}}_{pe} + \mathrm{T}^{\mathrm{proc}}_{pe}) + \sum_{p \in \mathrm{P}} \sum_{e \in \mathrm{E}} \sum_{q \in \mathrm{P}} \sum_{f \in \mathrm{E}} \mathrm{T}^{\mathrm{collab}}_{peqf}. \quad (16)$$

Patient p's CCDT initially deployed on edge server e may achieve lower latency by relocating to alternative edge server f. Fig. 2(a) illustrates the move operation. If edge server f ranks higher in patient p's preference list and possesses sufficient residual resources, we evaluate the system utility change: $\mathrm{U}(\Omega') < \mathrm{U}(\Omega)$, where Ω' represents the post-move matching. If the move reduces total latency while satisfying edge server f's resource constraints C2, C3 and patient p's synchronization latency constraint C1, we execute the move:

$$\Omega' = \{\Omega \setminus \{(p, e)\}\} \cup \{(p, f)\}. \quad (17)$$

Fig. 2 shows two deployment refinement operations. Panel (a) depicts a move operation where patient p's CCDT relocates from edge server e to server f to reduce latency. Panel (b) illustrates a swap operation where patient p's CCDT on server e and patient q's CCDT on server f exchange positions to improve both patients' utilities.

Patients p and q with CCDTs initially deployed on edge servers e and f, respectively, may both benefit from exchanging positions. Fig. 2(b) shows the swap operation. If both patients rank the opposite server higher in their preference lists, we evaluate the post-swap utility: $\mathrm{U}(\Omega'') < \mathrm{U}(\Omega)$.

If the swap reduces total latency while both edge servers satisfy resource constraints C2, C3, and both patients satisfy synchronization constraints C1, we execute the swap:

$$\Omega'' = \{\Omega \setminus \{(p, e), (q, f)\}\} \cup \{(p, f), (q, e)\}. \quad (18)$$

where Ω'' denotes the post-swap matching.

Swap operations become beneficial when patients mutually prefer opposite servers despite individual moves being infeasible. Patient groups with strong collaboration relationships benefit from exchanging positions when one patient has high synchronization latency while the other experiences excessive collaboration delay. Swaps enable deployment changes that preserve resource constraints when individual moves would violate capacity limitations. Necessary conditions require that both patients improve their individual utilities post-swap while maintaining aggregate latency reduction. Swap evaluation complexity scales quadratically with patient populations, motivating heuristics restricting swap consideration to patients experiencing high latencies or sharing collaboration relationships. Practical implementations evaluate moves before swaps, attempting simpler operations before complex exchanges.

A matching Ω achieves bilateral exchange stability if no move or swap operation exists that reduces system utility while satisfying all constraints.

While load balancing prevents edge server saturation by distributing patient deployments evenly across the infrastructure. Balanced loads maintain spare capacity on each server, enabling rapid patient admission without triggering redeployment. Variance minimization objectives penalize uneven distributions where the standard deviation of server utilization provides load balance metrics. Balanced deployments increase robustness, where individual edge failures impact fewer patients compared to concentrated placements. However, load balancing may conflict with latency optimization, where patient-server proximity creates natural load concentrations. Healthcare institutions adjust balance preference based on network dynamics, favoring a stronger balance in volatile environments with frequent patient changes, while accepting imbalance when populations remain stable.

Matching-theory refinement converges to locally optimal solutions satisfying bilateral exchange stability. Each move and swap operation strictly decreases total system latency, creating a monotonic improvement sequence. A finite solution space with binary deployment variables ensures convergence within bounded iterations. Bilateral stability provides a stopping criterion where no beneficial exchanges exist among patient-server pairs. While global optimality remains unguaranteed in polynomial time due to NP-hardness, bilateral stable matchings demonstrate near-optimal performance empirically. Convergence analysis shows algorithms terminate when marginal latency improvements fall below threshold values, typically

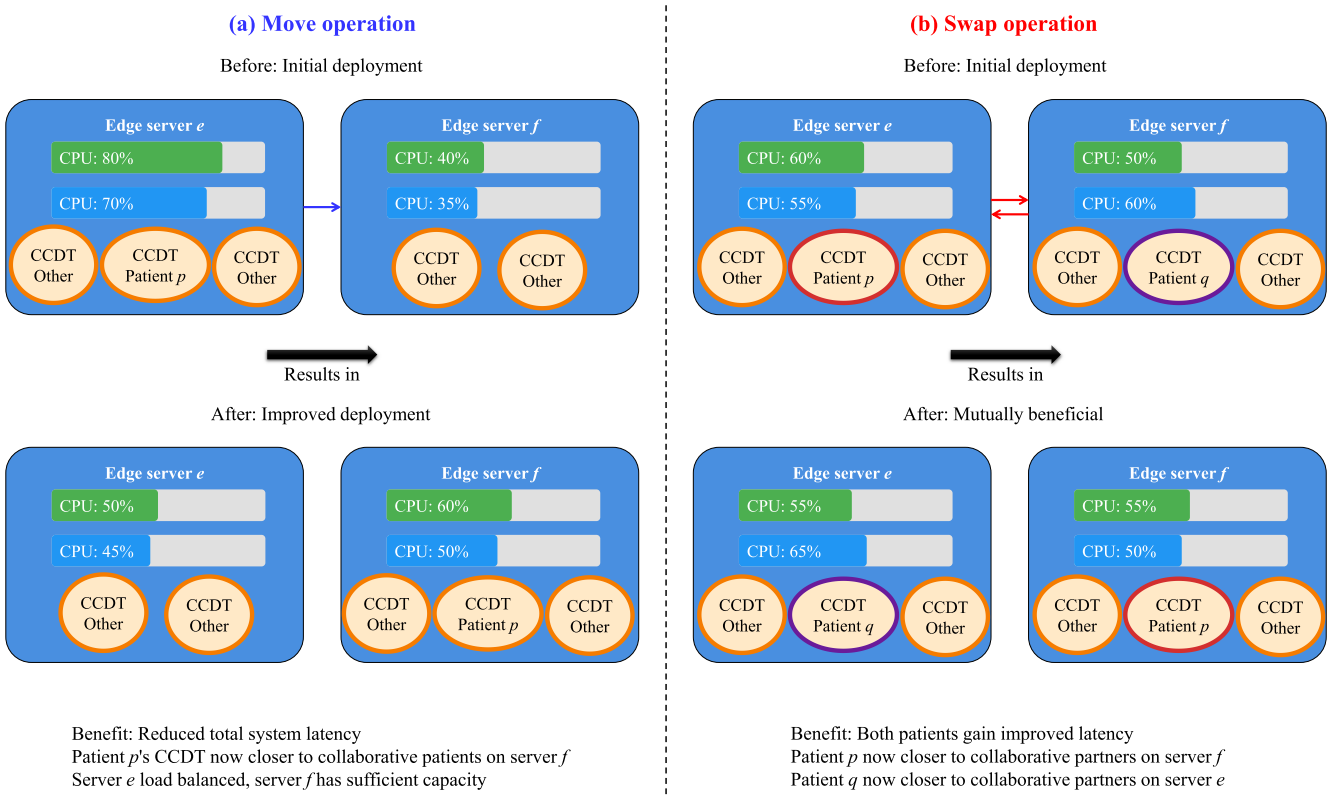

Fig. 2. Illustration of move and swap operations for CCDT deployment refinement.

within five to ten refinement iterations for healthcare networks studied.

Branch-and-bound initialization solves linear programming relaxation in polynomial time, with complexity depending on constraint matrix dimensions scaling as $O(|P| \cdot |E|)$ for patient-server assignments. Linear program solvers employ interior-point methods, achieving convergence within dozens of iterations regardless of problem size. Matching-theory refinement explores local neighborhoods through move and swap operations, where each iteration evaluates $O(|P| \cdot |E|)$ potential moves and $O(|P|^2 \cdot |E|^2)$ possible swaps. Bilateral stability typically requires fewer than ten iterations empirically, yielding practical complexity approximating $O(10 \cdot |P|^2 \cdot |E|^2)$. Combined complexity remains polynomial in problem dimensions, contrasting exponential $O(|E|^{|P|})$ exhaustive enumeration, enabling real-time deployment computation for practical healthcare networks.

Edge infrastructure reliability requires fault tolerance mechanisms protecting patient monitoring continuity. Redundant deployment maintains backup CCDT instances on secondary edge servers, enabling rapid failover when primary servers fail. Periodic state synchronization between primary and backup CCDTs minimizes recovery time, though synchronization overhead consumes network bandwidth and computational resources. Failure detection through heartbeat protocols triggers automatic CCDT migration to alternative edge infrastructure within seconds. Load redistribution algorithms reassign patients from failed servers to surviving nodes while respecting resource constraints. Healthcare institutions provision excess edge capacity specifically for failure scenarios, accepting reduced utilization during normal operation to ensure monitoring continuity during outages.

Dynamic healthcare environments require adaptive redeployment responding to runtime changes. Patient admissions trigger incremental optimization, evaluating whether new patients fit existing edge servers or necessitate reshuffling current deployments. Condition deterioration, altering computational requirements, may prompt individual patient migration when resource demands exceed current server capacity. Collaboration relationship changes as patients join support groups require evaluating whether colocation with collaborating patients reduces overall latency. Periodic reoptimization scheduled during low-activity hours maintains deployment quality despite gradual environmental drift. Incremental algorithms modifying existing deployments rather than full recomputation reduce computational overhead, enabling frequent adaptation responding to evolving healthcare network conditions.

IV. Simulation Results and Performance Analysis

A. Simulation Configuration

To test the performance of the proposed SPCED algorithm, one will have to conduct sophisticated simulations that would indicate the conditions of the actual deployment of healthcare 5.0. The simulation environment is an example of a healthcare edge network with a size of 3 kilometers by 3 kilometers, which is the district of urban healthcare, and has many hospitals, outpatient clinics, and residential settlements

where the patients receive the remote-monitoring services. The locations where edge computing infrastructure is implemented are strategic locations like the hospital data center, community health centers, and edge nodes, which are specifically placed to provide as much wireless coverage as possible to the population of patients. Many of the locations covered by this coverage are representative of the difference, as valid to the fluctuation in the care environments, between the intensive care units in need of continual monitoring of the vital signs of patients and the home-based care in patient care of chronic diseases and preventative wellness among populations of the elderly.

The healthcare edge network is a six-server network of edge servers that have constrained computational and storage capabilities, which are typical of medical institution infrastructures. A single edge server has 48 million instructions per second (MIPS) of processing power and 48 gigabytes of medical data storage, real-world hardware capabilities of healthcare edge facilities, when limited by cost and physical space concerns. These servers are connected to a 10-megahertz bandwidth with 200 milliwatts of patients on channels with exponent four path losses on transactions that signify urban healthcare conditions of building structure and medical device interference. The noise power spectral density is at -174 dBm/hertz, which falls within the indoor clinical environment.

Edge server dimensioning must align computational capacity with patient loads. With six servers and one hundred patients, average loads approach seventeen patients per server, though deployment optimization creates uneven distributions. Profile 1 intensive care patients consuming 2000 MIPS each would saturate 48,000 MIPS servers at twenty-four patients, while Profile 3 preventive care patients at 1000 MIPS enable forty-eight deployments per server. Mixed patient populations enable higher densities through load diversity, where heterogeneous resource demands facilitate efficient packing. Healthcare institutions provision edge capacity based on patient acuity distributions and monitoring requirements rather than uniform assumptions about computational needs.

The number of patients that a given case of simulation covers ranges between 40 and 100 people; this is a sign of healthcare networks that meet the needs of communities with a diverse population. All patients present healthcare monitoring heterogeneity data, which must undergo processing by CCDT, and three various resource requirement profiles are outlined. The intensive care patients in Profile 1 during the post-cardiac surgery period require 2000 MIPS of computer power and 0.85 GB of memory to process the high-frequency electrocardiograms and to process arterial waveforms. Profile 2 is the category of patients with chronic illnesses like diabetes or heart conditions who need 2500 MIPS, 375 GB of storage to access huge volumes of historical health records, and execute algorithms to optimize treatments. Profile 3 is the preventive care in healthy aging patients under fall risk supervision, taking 1000 MIPS and 1.70 GB in storage to monitor lightweight activities and location tracking. The patient-to-patient relationship in a collaborative model works in a random and varying level of density fashion, which emulates family-based networks of caregiving, peer support groups, and coordinated care groups that exchange medical information due to CCDT interaction.

It has a constant latency-distance mapping coefficient of 3.3 milliseconds per kilometer of inter-edge communication, which is fiber-optic backhaul between healthcare institutions. Latency of patient-CCDT synchronization is maximally tolerable and varies in experiments between 5.0 and 7.0 milliseconds, and is determined by the emergency response time requirements or benign chronic disease surveillance of the healthcare application. The simulations each use the SPCED algorithm in Python on top of CVXPY to do operations in the realms of linear programming and NumPy to do operations in the realms of numerical operations.

SPCED's performance evaluation compares against six established baseline algorithms representing the state-of-the-art in digital twin deployment and healthcare computing.

1) STIN-DT [30]: A satellite-terrestrial integrated network (STIN) model, alleviating the problem of insufficient single-layer deployment flexibility of the traditional edge network by deploying DTs in multi-layer nodes in a STIN.
2) VFA-DT [31]: An optimized offloading decision - making framework that combines value function approximation utilizing digital twin -based deep learning and reinforcement learning methodologies.
3) IoTwins [32]: Implementing distributed and hybrid digital twins in industrial manufacturing and facility management settings.
4) KTWIN [33]: A serverless Kubernetes-based DT platform.
5) KubeTwin [34]: A DT framework for Kubernetes deployments at scale.
6) DSFL [35]: A decentralized SplitFed learning approach for healthcare consumers in the metaverse.

These six algorithms span from STIN-DT as the weakest baseline to DSFL as the strongest competitor, providing comprehensive performance benchmarking across multiple dimensions.

Topology sensitivity analysis evaluates algorithm performance across diverse spatial configurations. Clustered topology concentrates edge servers within hospital districts separated by gaps representing residential zones, creating high-density and low-density regions. Linear topology arranges edge infrastructure along transportation corridors, modeling highway medical facilities. Random topology distributes servers according to spatial point processes reflecting organic healthcare facility growth. Performance metrics vary across topologies, where clustered configurations reduce average patient-server distances compared to dispersed arrangements. SPCED maintains relative performance advantages across all tested topologies, though absolute latency values increase with infrastructure dispersion. Topology-aware deployment recognizes that spatial structure influences achievable performance bounds.

Table I shows the simulation parameters for healthcare 5.0 CCDT edge network.

TABLE I
SIMULATION PARAMETERS FOR HEALTHCARE 5.0 CCDT EDGE NETWORK

Parameter	Value	Description
Wireless bandwidth Ω	10 MHz	Channel bandwidth for patient-edge communication
Patient transmission power Θ_p	200 mW	Uplink power from medical sensors
Path loss exponent ν	4	Urban healthcare environment propagation
Noise power spectral density Ψ_0	-174 dBm/Hz	Indoor clinical setting noise floor
Number of edge servers E	6	Healthcare edge infrastructure nodes
Patient population P	[40, 100]	Range of monitored patients
Edge CPU capacity C_e^{cpu}	48,000 MIPS	Computational resources per server
Edge storage capacity C_e^{stor}	48 GB	Medical data storage per server
Latency-distance coefficient κ	3.3 ms/km	Inter-edge network latency
Profile 1 (Intensive care)	2000 MIPS / 0.85 GB	Post-surgical cardiac monitoring
Profile 2 (Chronic disease)	2500 MIPS / 3.75 GB	Diabetes and heart disease management
Profile 3 (Preventive care)	1000 MIPS / 1.70 GB	Elderly fall risk monitoring
Maximum sync latency L_{max}	[5.0, 7.0] ms	Application-specific latency thresholds
Coverage area	3 km × 3 km	Urban healthcare district dimensions

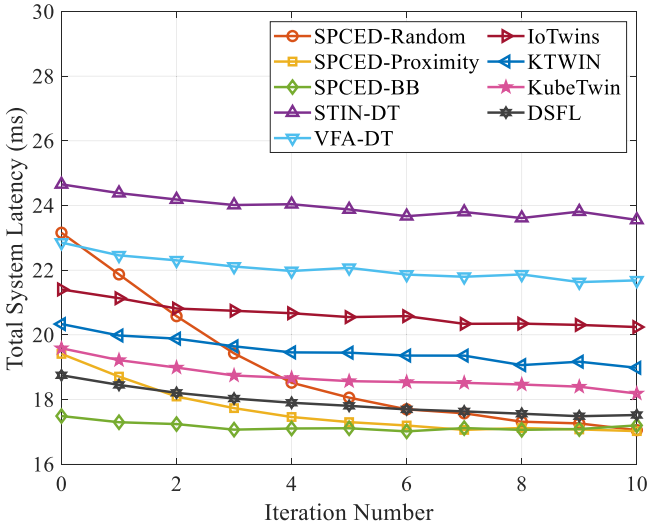

Fig. 3. Impact of initial deployment strategy.

B. Impact of Initial Deployment Strategy

Fig. 3 illustrates the convergence behavior and ultimate latency performance of the composite of numerous methods of initializing the model and the relevant matching-theory refinement stage of SPCED, given 100 random patients who generated collaborative relations. The experiment considers three initialization methods, including random assignment in which patient CCDTs are scattered on random edge servers without considering any proximity and resource constraint, proximity-based implementation in which a patient will be deployed on the closest edge module with exclusive consideration on the distance between the patients and the initiation resources, and the proposed branch-and-bound initialization.

The convergence curves show the relevance of the quality of the initializations on the general deployment performance in 5.0 CCDT ecosystems in healthcare. Branch-and-bound pioneering SPCED initialisation minimises the total system latency in just three refinement steps of 17.1 milliseconds, displaying rapid convergence of a well-initialized, high-quality starting position. The proximity-based initialization of SPCED has the same optimal latency and requires five iterations, with proximity heuristic values set to 19.3 milliseconds, which implies that the collaborative relationships are ignored and that the patient CCDTs are located on an independent basis of geographic distance.

C. Resource Requirement Heterogeneity Analysis

Fig. 4 shows overall system latency, as well as the average inter-patient collaboration latency, in the four resource requirement conditions of the diverse and variable population of 5.0 in healthcare 50 when five patients are present, and the connection between them on average. In Profile 1, all patients are used and would be intensive care cases that require 2000 MIPS and 0.85 GB of storage, which is a model of post-surgical cardiac monitoring with high-frequency vital sign processing. The reason is that profile 2 assigns all patients 2500 MIPS and 3.75 GB of need, which identifies diabetes and heart failure management with massive historical data storage. The preventive care specifications of 1000 MIPS and 1.70 GB are provided to all patients in Profile 3, which outlines healthy geriatric populations under observation of fall risk.

The natural heterogeneity of resource demands defines the dynamics of CCDT implementation and the potential latency performance in healthcare 5.0 edge networks. In the lightweight resource footprints that allow edge servers to serve more patient CCDTs on a node, and collaboration groups collaborate, profile three preventative care patients achieve the lowest total system latency of 17.3 milliseconds and collaboration latency of 148 milliseconds among all other algorithmic strategies. Chronic disease in 2017 profile patients, whose storage model is significant, produce the highest latencies of 22.8 milliseconds total, and 215 milliseconds collaboration, because the heavy resource demand necessitates the allocation

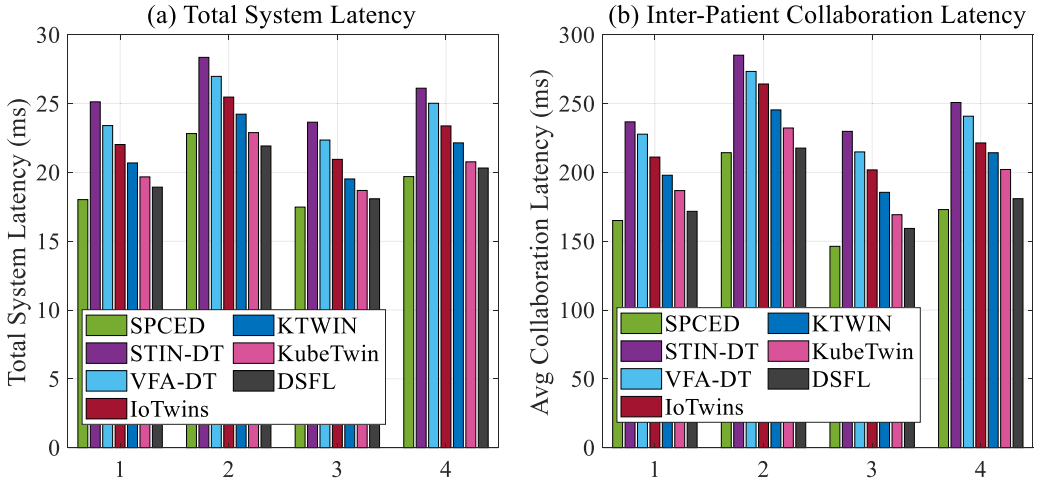

Fig. 4. Resource requirement heterogeneity analysis.

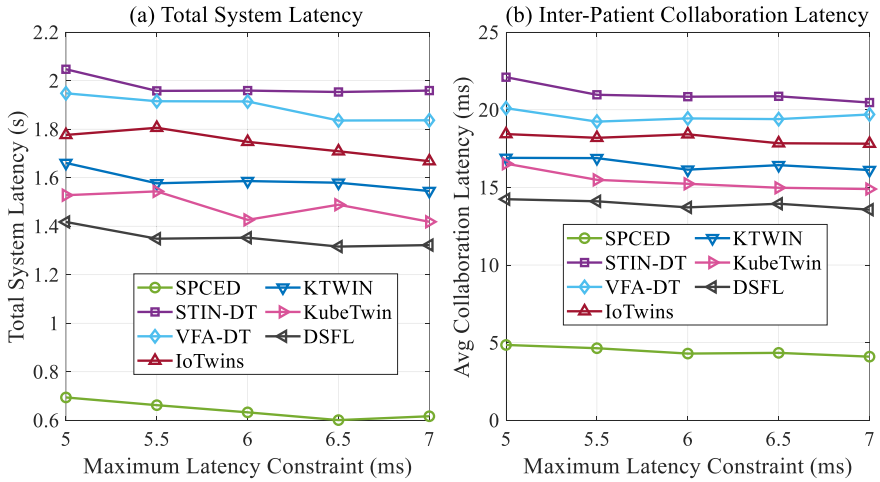

Fig. 5. Maximum latency constraint sensitivity analysis.

of patient CCDT to many edge servers, which causes the inter-edge network movements to coordinate patient care.

D. Maximum Latency Constraint Sensitivity

Fig. 5 illustrates the algorithm's behavioral sensitivity to maximum synchronization latency limits in 80 patients with an average collaborative connection among five individuals, testing five constraint values sequentially, and separated by 0.5 milliseconds, between 5.0 and 7.0 milliseconds. Stricter limits, with values under 5.0 milliseconds, are time-sensitive applications like robotic surgery, which are remotely controlled and require a surgeon's control command to reach patient CCDTs during a sub-10-millisecond time range, and emergency triage systems, where patient-acuity scores need to be generated within seconds and routed into an appropriate route. The 6.0 -7.0 milliseconds moderate limits serve sustained cardiac arrhythmia monitoring of electrocardiogram waveforms processed and the subsequent segments. Weaker constraints in the range of 7.0 milliseconds are suitable where daily glucose readings are habitually monitored (where the patient completes the measurements) and medicine preferences are modified in minutes, not milliseconds.

The upper limits of synchronization latency are relatively low in the overall system latency in the healthcare 5.0 deployment algorithms because the spatial proximity of patients and edge servers imposes synchronization latencies at critical levels within smaller deployments, 3-kilometer areas. STLCED is characterized by a total latency of 0.62- 0.68 seconds in all constraint conditions, and reduces only by a slight percentage (8.8) as the constraints are moved between 5.0 and 7.0 milliseconds. By relaxing constraints, deployment algorithms are allowed to be more flexible to optimize collaboration at the expense of patient-CCDT proximity being significantly slack, such that CCDTs experience a minor increase in cooperation relative to the opportunity cost.

E. Scalability With Patient Population Growth

The scalability of the algorithms is depicted in Fig. 6, where the patient populations were increased in 10 patient steps with healthcare 5.0 edge networks supporting an average of ten patients each, with the healthcare institutions expanding monitoring services to new patient groups at each step without altering the intensity of care coordination. A smaller patient population of 40 patients describes more specialized care units, such as cardiac intensive care wards or post-operative recovery units, with limited patient groups having close care monitoring needs and strong care team cooperation. Community health

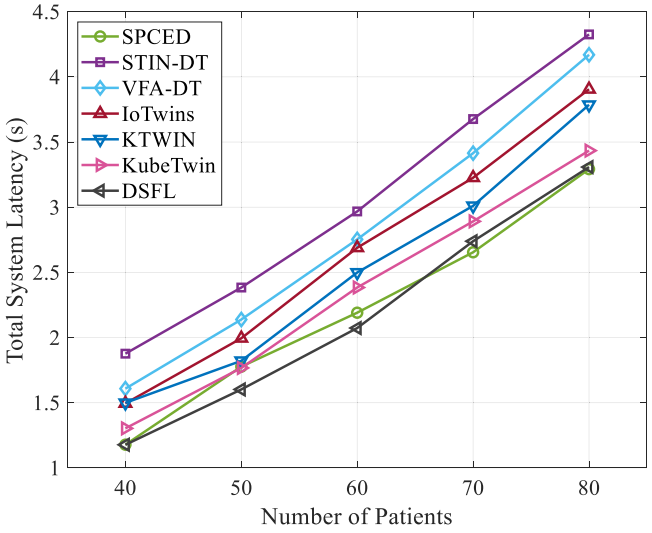

Fig. 6. Scalability with patient population growth.

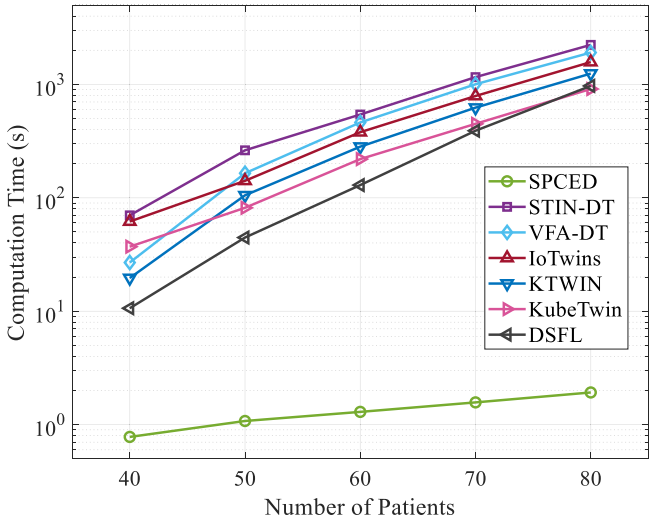

Fig. 7. Computational efficiency analysis.

centers that include cohorts of chronic diseases (diabetes, hypertension, and heart failure) with peer support networks, but moderate 60-patient populations, represent the population.

The high scalability properties of SPCED are evident because patient populations in healthcare 5.0 remain consistently scaled to characteristics despite varying population sizes and show nearly linear increase in latency with increasing population size, which relates to the underlying complexity of problems instead of inefficiency in algorithms. At 40 patients, SPCED has 1.25 seconds total latency over 1.17 seconds of DSFL, which is a small difference of 6.4 percent, showing a near-optimal performance with the tractability of the problem being exhaustively explored. A 2x population of 80 patients also doubles SPCED latency to 3.24 seconds, which indicates a 159 percent growth in terms of quadratic collaboration relationship expansion between 400 and 6400 possible patient pairs that CCDT must coordinate. Nonetheless, SPCED has the relative advantage of 3.28 seconds (80 patients) with DSFL and 4.38 seconds (35 percent lower performance) with STIN-DT.

F. Computational Efficiency Analysis

Fig. 7 presents the most revealing performance analysis of the algorithms: computational time requirements as the healthcare 5.0 patient populations grow between 40 and 80 patients with average collaborative ties of 10 patients each, and the execution time is measured between the problem initiation and the final solution deployment on the same set of hardware infrastructure. Computational efficiency becomes essential due to realistic implementations of healthcare 5.0 since hospitals are often required to recompute CCDT placements due to changing patient populations with admission and discharge, updating monitoring needs with disease evolution, forming and breaking relationships with support groups through support group dynamics, and maintaining supporting infrastructure with corresponding temporary CCDT migrations. Algorithms with exponential computational complexity can be impractical even at small pilot deployments, and algorithms with a computational complexity that is of polynomial time allow responsive real-time deployment adjustments in support of dynamic healthcare situations where patient condition and care coordination requirements evolve continuously across treatment trajectories.

SPCED realizes dramatic computational efficiency benefits over any baseline algorithm, and is fast enough to run in under 2 seconds with any population size, while competitors show exponential scaling that makes them practically impossible to run in practice in healthcare 5.0 deployments. In 40 patients, SPCED takes 0.82 seconds versus 12 seconds of DSFL, which is a 93.2 percent efficiency improvement with the aid of the matching-theory refinement with poly-time match instead of the exhaustive optimization. The increase in population to 80 patients doubles the time of SPCED execution to only 1.87 seconds. It doubles the time of DSFL to 980 seconds, indicating a 99.81 percent gain in efficiency as exponential complexity penalties are highly experienced in a base direction. STIN-DT executes the slowest, with 2245 seconds, with 80 patients, because of saturation of the satellite network paths, which is 1200 times slower than SPCED and makes the algorithm entirely infeasible in dynamic healthcare settings. Branch-and-bound initialized. Branch-and-bound produces high-quality initial solutions quickly, by using linear programming relaxation that is solvable in polynomial time, and matching-theory refinement search explores local deployment neighborhoods, efficiently investigating local move and swap operations, instead of considering full exponentially sized solution spaces. The advantage of having the computational efficiency of SPCED allows healthcare institutions to re-optimize the deployment of a system in real-time as the patient populations change, as do the relationships between collaborators, and the edge infrastructure undergoes a maintenance incident that prompts the CCDT migration to provide high-quality care continuously.

V. Conclusion

This study addressed the sustainable CCDT deployment problem for healthcare 5.0 edge networks under multi-dimensional resource constraints. We established a hierarchical network architecture encompassing patient populations

equipped with medical sensors and edge servers hosting CCDT virtual replicas. The simulations showed that the proposed SPCED algorithm attained 99.81% computational efficiency gain over six baseline algorithms, including state-of-the-art decentralized federated learning methods, with better latency performance across different patient populations, resource settings, and collaboration patterns.

Limitations were fixed locations of patients and fixed collaborative relationships, whereas dynamic healthcare environments are characterized by patient mobility and changing networks of care coordination. Future research directions include the expansion of SPCED to mobile patients who need real-time CCDT migration mechanisms, the use of energy consumption models to optimize sustainable ecosystems between performance and environmental impact, the formulation of multi-objective formulations that allow addressing latency, energy and cost in a single formulation, the use of federated learning to train collaborative models to maintain medical data privacy, and the experimental validation of its applicability in practice through real healthcare edge testbeds illustrating application in clinical settings.

REFERENCES

[1] J. Chen, M. Zhao, R. Zhou, W. Ou, and P. Yao, "How heavy is the medical expense burden among the older adults and what are the contributing factors? A literature review and problem-based analysis," *Frontiers Public Health*, vol. 11, Jun. 2023, Art. no. 1165381.

[2] M. Wazid, J. Singh, A. K. Das, and J. J. P. C. Rodrigues, "An ensemble-based machine learning-envisioned intrusion detection in Industry 5.0-driven healthcare applications," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1903–1912, Feb. 2024.

[3] Y. Cao, C. S. Ku, R. Kumar, and A. Khan, "Privacy and trust in blockchain-federated intrusion detection systems: Taxonomy, challenges and perspectives," *J. Reliable Secure Comput.*, vol. 1, no. 1, pp. 4–24, Nov. 2025.

[4] M. Wazid, A. K. Das, N. Mohd, and Y. Park, "Healthcare 5.0 security framework: Applications, issues and future research directions," *IEEE Access*, vol. 10, pp. 129429–129442, 2022.

[5] H. R. Chi, M. de Fátima Domingues, H. Zhu, C. Li, K. Kojima, and A. Radwan, "Healthcare 5.0: In the perspective of consumer Internet-of-Things-Based fog/cloud computing," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 745–755, Nov. 2023.

[6] A. De Benedictis, N. Mazzocca, A. Somma, and C. Strigaro, "Digital twins in healthcare: An architectural proposal and its application in a social distancing case study," *IEEE J. Biomed. Health Informat.*, vol. 27, no. 10, pp. 5143–5154, Oct. 2023.

[7] C. Wang, Z. Cai, and Y. Li, "Human activity recognition in mobile edge computing: A low-cost and high-fidelity digital twin approach with deep reinforcement learning," *IEEE Trans. Consum. Electron.*, vol. 71, no. 2, pp. 6451–6459, May 2025.

[8] T. M. Machado and F. T. Berssaneti, "Literature review of digital twin in healthcare," *Heliyon*, vol. 9, no. 9, Sep. 2023, Art. no. e19390.

[9] L. Sciullo, A. De Marchi, A. Trotta, F. Montori, L. Bononi, and M. Di Felice, "Relativistic digital twin: Bringing the IoT to the future," *Future Gener. Comput. Syst.*, vol. 153, pp. 521–536, Apr. 2024.

[10] A. I. Awad, M. M. Fouda, M. M. Khashaba, E. R. Mohamed, and K. M. Hosny, "Utilization of mobile edge computing on the Internet of Medical Things: A survey," *ICT Exp.*, vol. 9, no. 3, pp. 473–485, Jun. 2023.

[11] K. R. De Guzman, C. L. Snoswell, M. L. Taylor, L. C. Gray, and L. J. Caffery, "Economic evaluations of remote patient monitoring for chronic disease: A systematic review," *Value Health*, vol. 25, no. 6, pp. 897–913, Jun. 2022.

[12] A. Alboksmaty et al., "Effectiveness and safety of pulse oximetry in remote patient monitoring of patients with COVID-19: A systematic review," *Lancet Digit. Health*, vol. 4, no. 4, pp. e279–e289, Apr. 2022.

[13] N. Taimoor and S. Rehman, "Reliable and resilient AI and IoT-based personalised healthcare services: A survey," *IEEE Access*, vol. 10, pp. 535–563, 2022.

[14] X. Lv, S. Rani, S. Manimurugan, A. Slowik, and Y. Feng, "Quantum-inspired sensitive data measurement and secure transmission in 5G-enabled healthcare systems," *Tsinghua Sci. Technol.*, vol. 30, no. 1, pp. 456–478, Feb. 2025.

[15] A. Rauniyar et al., "Federated learning for medical applications: A taxonomy, current trends, challenges, and future research directions," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 7374–7398, Mar. 2024.

[16] P. K. Mall et al., "A comprehensive review of deep neural networks for medical image processing: Recent developments and future opportunities," *Healthcare Analytics*, vol. 4, Dec. 2023, Art. no. 100216.

[17] S. S. Tripathy et al., "FedHealthFog: A federated learning-enabled approach towards healthcare analytics over fog computing platform," *Heliyon*, vol. 10, no. 5, Mar. 2024, Art. no. e26416.

[18] X. Yuan, W. Ni, M. Ding, K. Wei, J. Li, and H. V. Poor, "Amplitude-varying perturbation for balancing privacy and utility in federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1884–1897, 2023.

[19] P. N. Bathula and M. Sreenivasulu, "An integrated blockchain framework for secure data sharing in IoT fog computing," *Tsinghua Sci. Technol.*, vol. 30, no. 3, pp. 957–977, Jun. 2025.

[20] N. Kumar and R. Ali, "A smart contract-based robotic surgery authentication system for healthcare using 6G-tactile internet," *Comput. Netw.*, vol. 238, Jan. 2024, Art. no. 110133.

[21] J. Lv, B.-G. Kim, B. D. Parameshachari, A. Slowik, and K. Li, "Large model-driven hyperscale healthcare data fusion analysis in complex multi-sensors," *Inf. Fusion*, vol. 115, Mar. 2025, Art. no. 102780.

[22] J. Li et al., "Digital twin-enabled service provisioning in edge computing via continual learning," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 7335–7350, Jun. 2024.

[23] B. Xu, H. Zhao, H. Cao, S. Garg, G. Kaddoum, and M. M. Hassan, "Edge aggregation placement for semi-decentralized federated learning in industrial Internet of Things," *Future Gener. Comput. Syst.*, vol. 150, pp. 160–170, Jan. 2024.

[24] X. Ju, S. Su, C. Xu, and H. Wang, "Computation offloading and tasks scheduling for the Internet of Vehicles in edge computing: A deep reinforcement learning-based pointer network approach," *Comput. Netw.*, vol. 223, Mar. 2023, Art. no. 109572.

[25] Z. Xu et al., "HierFedML: Aggregator placement and UE assignment for hierarchical federated learning in mobile edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 1, pp. 328–345, Jan. 2023.

[26] T. Nixon, R. M. Curry, and B. P. Allaissem, "Mixed-integer programming models and heuristic algorithms for the maximum value dynamic network flow scheduling problem," *Comput. Oper. Res.*, vol. 175, Mar. 2025, Art. no. 106897.

[27] L. Yin, J. Sun, and Z. Wu, "An evolutionary computation framework for task off-and-downloading scheduling in mobile edge computing," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 22399–22412, Jun. 2024.

[28] M. Hussain, L.-F. Wei, F. Abbas, A. Rehman, M. Ali, and A. Lakhan, "A multi-objective quantum-inspired genetic algorithm for workflow healthcare application scheduling with hard and soft deadline constraints in hybrid clouds," *Appl. Soft Comput.*, vol. 128, Oct. 2022, Art. no. 109440.

[29] S. Lu, C. Ma, M. Kong, Z. Zhou, and X. Liu, "Solving a stochastic hierarchical scheduling problem by VNS-based metaheuristic with locally assisted algorithms," *Appl. Soft Comput.*, vol. 130, Nov. 2022, Art. no. 109719.

[30] Y. Tao, B. Lei, H. Shi, J. Chen, and X. Zhang, "Adaptive multi-layer deployment for a digital-twin-empowered satellite-terrestrial integrated network," *Frontiers Inf. Technol. Electron. Eng.*, vol. 26, no. 2, pp. 246–259, Feb. 2025.

[31] N. B. Ha and Y. S. Jeong, "Fusion of digital twin and blockchain for secure and efficient IoV networks," *Hum.-Centric Comput. Inf. Sci.*, vol. 14, p. 37, Jun. 2024.

[32] P. Bellavista and G. Di Modica, "IoTwins: Implementing distributed and hybrid digital twins in industrial manufacturing and facility management settings," *Future Internet*, vol. 16, no. 2, p. 65, Feb. 2024.

[33] A. G. Wermann and J. A. Wickboldt, "KTWIN: A serverless Kubernetes-based digital twin platform," *Comput. Netw.*, vol. 259, Mar. 2025, Art. no. 111095.

[34] D. Borsatti et al., "KubeTwin: A digital twin framework for Kubernetes deployments at scale," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 4, pp. 3889–3903, Aug. 2024.

[35] V. Stephanie, I. Khalil, and M. Atiquzzaman, "DSFL: A decentralized splitfed learning approach for healthcare consumers in the metaverse," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2107–2115, Feb. 2024.