Digital Twin-assisted Multi-Layer Network: Resource Optimization for Low-Latency and Energy-Efficiency

by

Muhammad Adnan Qadir

A thesis presented to the Lakehead University in partial fulfillment of the thesis requirement for the degree of

Master of Science

in

Electrical and Computer Engineering

Lakehead University

Thunder Bay, Ontario, Canada

April 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis.

Supervisor:	Dr. Waleed Ejaz Associate Professor, Dept. of Electrical & Computer Engineering, Lakehead University
Examiner:	Dr. Salama Ikki Professor, Dept. of Electrical & Computer Engineering, Lakehead University
Examiner:	Dr. Farhan Ghaffar Assistant Professor, Dept. of Electrical & Computer Engineering, Lakehead University

Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The sixth-generation (6G) wireless networks are expected to provide ubiquitous connectivity, high data rate, low latency, energy efficiency, and edge intelligence for Internet of Things (IoT) applications. However, effective resource scheduling and network configuration in 6G is challenging due to the resource-constrained devices, high quality-of-service (QoS) requirement, and high density of heterogeneous devices. Multi-layer networks are potential candidates for addressing the challenge of resource-constrained devices to meet their tasks' QoS requirements. Still, there is the challenge of resource scheduling and management of multi-layer networks. Digital twin technology is a promising solution to enable multi-layer wireless networks that incorporate IoT devices on the ground, unmanned aerial vehicles (UAVs) as mobile edge computing (MEC) servers, and cloud servers. Multi-layer processing can handle time-sensitive and computationally intensive tasks from IoT devices. In this thesis, we propose a digital twin-assisted multi-layer network for low-latency and energy-efficient communication and computation. We mathematically formulate an optimization problem to minimize task latency and energy consumption of IoT devices by optimizing their association with the UAV-MECs, computation resources, communication resources, and offloading portions of tasks. The formulated problem is a non-linear and non-convex optimization problem. We propose a two-stage scheme based on the K-means method and the deep neural network approach to solve the above optimization problem. The K-means method is utilized for the optimal placement of UAV-MECs in the first stage, and then we associate the IoT devices with UAV-MECs for offloading tasks. In the second stage, the deep learning architecture is utilized to optimize network resources. We compare the proposed two-stage scheme with existing schemes to highlight the scalability of the proposed solution. We perform extensive simulations by varying the number of UAV-MECs and IoT devices in the network to look at the impact on task latency and energy consumption by IoT devices. Fixed offloading portioning is compared with optimized offloading portioning to highlight the usefulness of optimization in terms of latency and energy minimization. Simulation results demonstrate the usefulness of the multi-layer network in achieving low latency and energy-efficient computation and communication.

Acknowledgements

I would like to thank all the people who made this thesis possible. I want to thank my instructors for their shared knowledge, my wireless communication network (WCN) research group fellows for a collaborative learning experience, and my friends and family, especially my mother and brother, Nouman. Special thanks to Dr. Waleed Ejaz for his supervision and ongoing mentoring.

Dedication

This is dedicated to my father, the late Muhammad Abdul Qadir Khan, who encouraged me beyond his means and supported me in my goal of pursuing higher education in engineering. Unfortunately, he could not live to see the day I completed it.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Acronyms	Description
5G	Fifth-generation
6G	Sixth-generation
IoT	Internet of Things
IIoT	Industrial internet of things
IoV	Internet of Vehicles
IPM	Interior point method
LoS	Line of sight
DDPG	Deep deterministic policy gradient
DT	Digital twin model of network
MEC	Mobile edge computing
VEC	Vehicular edge computing
ML	Machine learning
AI	Artificial intelligence
DL	Deep learning
RL	Reinforcement learning
DRL	Deep reinforcement learning
AO	Alternating optimization
QoS	Quality-of-service
QoE	Quality-of-experience
UAV	Unmanned aerial vehicle
UAV-MECs	Unmanned aerial vehicle mobile edge computing servers
SDN	Software-defined network
THz	Terahertz
URLLC	Ultra-reliable and low-latency communication
KIPM	K-means assisted interior point method
KOA	K-means assisted outer approximation
GNN	Graphical neural network
RNN	Recurrent neural network

GAN	Generative adversarial network
WCN	Wireless communication network

List of Symbols

Symbol	Description
U	Number of IoT devices
N	Number of UAV-MECs
p_u	*u-th* IoT transmission power
h_{un}	Channel coefficient between *u-th* IoT device and *n-th* UAV-MECs
μ_u^L	Average task arrival rate of *u-th* IoT device
γ_u	SINR of *u-th* IoT device
$\bar{U}, \bar{N}, \bar{C}$	Digital twin models for IoT device, UAV-MECs, cloud server
B	System bandwidth
E_u^{MAX}	Maximum energy consumption by the *u-th* IoT device
R_u	Date rate of *u-th* IoT device
P_u^{MAX}	Maximum transmission power budget for *u-th* IoT device
M^{MAX}	Maximum number of IoT devices associated with UAV-MECs
T_u^{MAX}	Maximum allowable latency of *u-th* IoT device' task
F_u^{MAX}	Maximum local processing rate of *u-th* IoT device
R_u^{MIN}	Minimum transmission rate assigned to *u-th* IoT device
f_u^L	Estimated processing rate of *u-th* IoT device
f_n^E	Estimated processing rate of *n-th* UAV-MEC
f^C	Estimated cloud server processing rate
β_u	Portion of *uth* IoT device task offloaded to UAV-MECs
$\frac{\theta_u}{2}$	Average activity factor of *u-th* IoT device
α_{un}	Portion of task offloaded from UAV-MECs to CS
t_u	Total task latency of *uth* IoT device
x_{un}	Association matrix between IoT devices and UAV-MECs
t^C	Latency of computation at cloud server
D_u	Task size of *u-th* IoT device

t^E	Latency of computation at UAV-MECs
C_u	Computational cycles required for *u-th* IoT device' task
t_u^L	Local latency of computation at the *u-th* IoT device
c	Speed of light
f_c	Carrier frequency
d_{un}	Distance between the *u-th* IoT device and *n-th* UAV-MECs
h_n	Height of *n-th* UAV-MECs

Chapter 1

Introduction

The demand for low latency and energy-efficient communication has increased due to the growing number of Internet of Things (IoT) applications. These requirements are critical in areas such as industry automation, driverless cars, and healthcare automation, where real-time data transmission, computation efficiency, and decision-making are crucial. These applications have computationally complex and stringent latency requirements tasks, which require a low latency and computationally efficient network is needed. The sixth-generation (6G) wireless networks are envisioned to provide extreme connectivity, high data rate, low latency, energy efficiency, and intelligent edge computing for IoT applications [1].

1.1 Preliminaries of Digital Twin-assisted Offloading in Multi-layer Network

This section will discuss the preliminaries of digital twin-assisted offloading in a multi-layer network. This includes technologies such as digital twins, unmanned aerial vehicles (UAVs) based mobile edge computing servers (MECs), multi-layer networks, and digital twin-assisted computation offloading.

1.1.1 Digital Twin

Digital twin technology creates copies of the physical environment in a virtual environment, which includes the physical entities, virtual models, and a two-way data flow between them. The bi-directional data flow allows for control of the physical entities through their virtual models. The effective creation of a digital replica can provide intelligence and power to achieve the desired function [2]. This process consists of three key components: (i) a physical space for capturing the physical world, (ii) a virtual space that creates the virtual replicas of the physical world, and (iii) a link that facilitates communication and control between the physical and virtual worlds [3]. Digital twin enables the real-time replicas of physical systems. In the context of 6G wireless networks, digital twin networks can be used for design, simulation, conducting what-if-analysis, and optimizing networks through machine learning and artificial intelligence algorithms [4]. Digital twin can increase operational efficiency by generating virtual replicas of physical systems and processes.

The digital twin is an effective technology that enhances the performance of a network by equipping it with intelligence. The digital twin processes information from the network configuration and infrastructure, along with the continuous feedback from the network as input to replicate and optimize the network, as Fig. 1.1 shows. Fig. 1.1 is an exemplary situation where the digital twin is utilized for the effective management of disaster through the digital twin. Digital twin technology enables the selection of optimal configuration and relief actors tailored to the requirements of the disaster situation. Fig. 1.1 shows a model structure of digital twin and multipurpose UAVs network along with three disaster scenarios: earthquakes, floods, and fires. The digital twin center is responsible for having all the virtual instances of UAVs and users in the affected areas. There is a two-way information sharing for the digital twin, enabling the replication and control of the physical network.

Digital twin is a crucial tool to achieve control of physical objects through system modeling, real-time data processing, edge computing, and cloud computing [5]. In multi-layer networks, digital twin technology can generate the virtual copy of the networks that can be utilized for the adaptive edge association for users, enabling the path to 6G networks [6]. By utilizing the digital

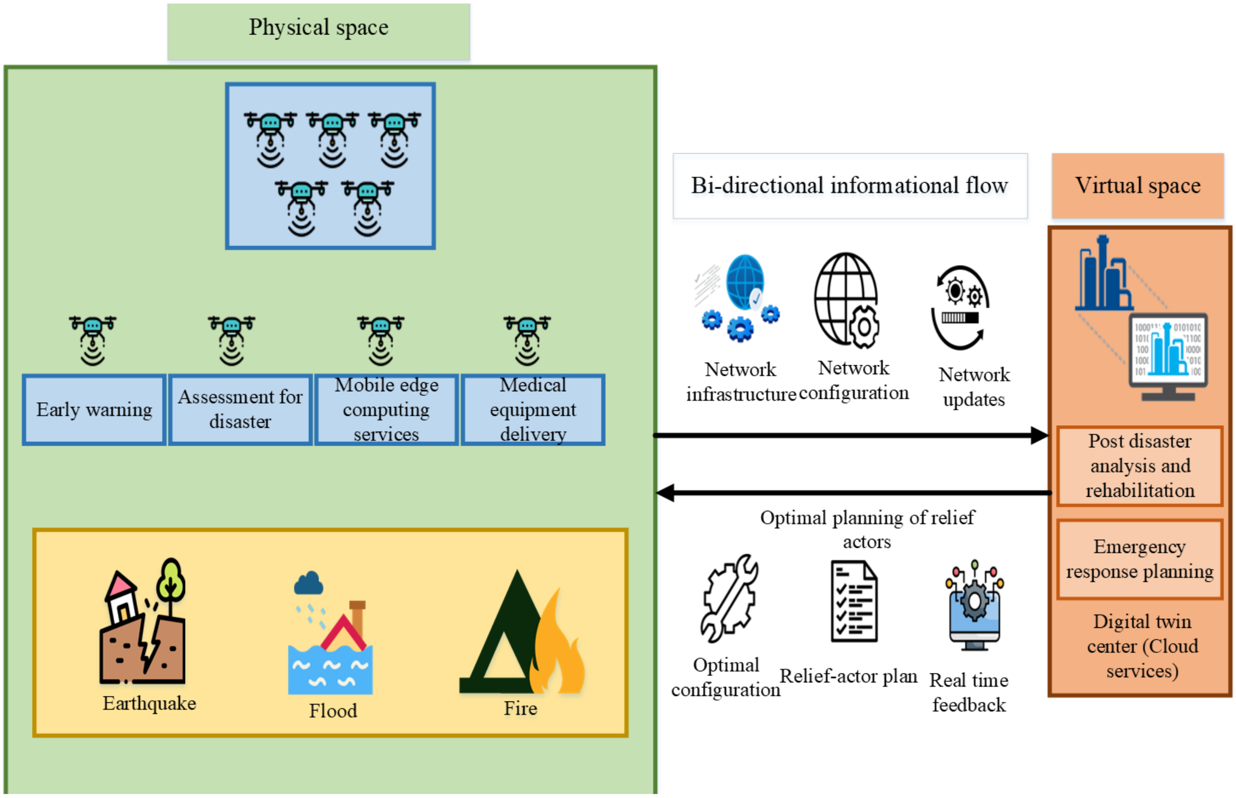


Figure 1.1: The digital twin-enabled network of UAVs framework for disaster management.

twin concept, the efficiency of UAVs to act as MEC server can be significantly enhanced in terms of storage and computation resources [7]. This is achieved by optimizing the UAVs placement and motion control using deep reinforcement learning algorithms [8]. Additionally, by using Q-learning through digital twins, we can reduce energy consumption and latency of UAVs services while optimizing transmit power for connection with the base station of UAVs [9]. The energy consumption is reduced in UAVs by avoiding excessive moves during placement. Furthermore, digital twins are also useful in optimizing service chain placement for task offloading to UAV-MECs.

1.1.2 UAV-MECs

UAVs are highly mobile and can access remote locations, which makes them a good choice as servers for remote location devices. Moreover, UAVs can establish a temporary communication network by acting as flying relays connecting users to the communication network [10].

UAVs combined with computing capabilities are known as UAV-MECs, functional to reduce latency and load of cloud data centers [11]. UAV-MECs are a potential technology for IoT devices to achieve low end-to-end latency by offloading their computation tasks to nearby servers with high processing resources. The coverage for the ground users is enhanced through using UAV-MECs for connectivity services [12]. Also, it is utilized for fast computing the users' time-sensitive and intensive computational tasks by providing the computation resources in the users' proximity. In [13], a survey is presented for different configurations of edge computation and discussed resource scheduling in edge computation to highlight the usefulness of UAV-MECs. The UAV-MECs assisted communication enables the task computation with stringent latency and intensive computation resources requirement [14].

The performance of UAV-MECs can be enhanced by utilizing the digital twin in the network. The digital twin-enabled multi-UAVs network uses the deep reinforcement learning algorithm for optimal task offloading from the users to UAVs for computation to achieve efficient resource utilization [15]. Furthermore, digital twins are crucial in optimizing service chain placement. The real-time updates to the digital twin of MEC servers enable task sharing among the available UAVs. This task sharing among the UAVs is called the service chain placement, which allows the UAVs to serve as MEC servers [16]. The digital twin-enabled UAV-MECs can achieve intelligent task offloading by optimizing the UAV's trajectory, mobile user association with UAVs, resource allocation to UAVs, and the transmission power of mobile users for offloading to minimize the energy consumption by UAV-MECs.

1.1.3 Multi-layer Network

A multi-layer network is useful for computationally complex tasks that involve multiple layers of UAV-MEC servers and cloud servers. These layers work together to process concurrently using computation resources available at different layers. UAVs, when combined with computing capabilities, can reduce latency and load of cloud data centers [11]. UAV-MEC is a potential technology for IoT devices to achieve low end-to-end latency by offloading their computation tasks to the nearby server with high processing resources. However, a bottleneck exists in UAV-MEC resources due to the high rate of computational tasks arriving at UAVs offloaded from users [17].

Cloud computing is crucial in multi-layer networks to facilitate extremely high computational tasks. In cloud computing, there is limited computational resource constraint, which makes it an essential component of multi-layer networks. A cloud server is necessary to avoid resource scarcity, which can cause a bottleneck in offloaded computation to UAV-MECs [17]. In a multi-layer network, resource distribution is necessary to utilize the resources efficiently. Therefore, a decision-making technology is needed to handle these distribution tasks. Digital twin is a promising solution for efficient resource management and optimal network configuration in multi-layer networks [18].

In [19], the authors proposed a joint optimization in a multi-layer network with optimal offloading and resource allocation to reduce energy consumption with the constraint on service latency. For ultra-reliable low latency communication (URLLC) and edge computing, the extreme value theory is applied in [20] by imposing the constraints for edge association to minimize the energy consumption by the users. To minimize energy consumption by the dense multi-device system, a joint optimization problem is solved in [21] through optimizing the device association with the UAV-MECs, task offloading, and resource allocation to devices and servers.

1.1.4 Digital Twin-assisted Computation Offloading

Digital twin plays an important role in the computation offloading from the resource-constrained devices in the system. Digital twin-assisted UAVs are more efficient in providing computation offloading services as we can have the optimal aerial base station placement using reinforcement learning at the digital twin [9]. The digital twin virtual replicas of UAVs provide the reinforcement learning datasets and enable optimal placement by the digital twin control on the physical system. Additionally, using Q-learning in the digital twin's virtual space can achieve optimal placement for aerial base stations to provide computation services. Iterations of Q-learning training in the virtual domain of digital twin reduced energy consumption by UAVs. The optimal placement through Q-learning with the help of digital twin results in a higher data rate for users and reduces the total base station's transmission power [9].

A digital twin technology framework with UAVs can control the UAVs' flight operations [22]. The digital twin-controlled UAVs were helpful in quickly and accurately to remote locations. The digital twin technology with a deep learning algorithm controls the UAV's flight operation to reduce their arrival time in remote locations by optimizing the UAV's placement and motion control [8]. Digital twin addresses the resource limitations of UAVs as aerial base stations by performing reinforcement learning iterations at the digital twin to avoid frequent moves of the aerial base station and enable time-sensitive and energy-efficient UAV base station services [23].

The digital twin technology generates a virtual replica of UAVs for managing storage and computation resources to enhance the performance of the UAVs [7]. A digital twin is utilized in UAVs' networks to optimize resource allocation. For delay-sensitive tasks, this is achieved by analyzing the availability and usage of resources of UAVs through digital twins [10]. A deep reinforcement learning-based resource allocation approach applied through digital twin maximizes resource utilization of UAVs to provide communication services for dense environments to achieve low-overhead communication services [10]. A digital twin-driven deep-Q learning algorithm can be utilized for a dynamic resource allocation scheme that minimizes the communication overhead for UAV-MECs in a dense, high-traffic environment [24]. The algorithm controls the physical system through bidirectional information flow between the digital twin and the physical system. This approach enables efficient dynamic resource allocation for UAVs. A digital twin-enabled multi-UAVs network that uses the deep reinforcement learning algorithm is useful for optimal task offloading from the users to UAVs for computation to achieve efficient resource utilization [15]. The digital twin determines the intelligent computation offloading schemes to maximize resource utilization [16].

From the above discussion, we observed that the digital twin plays a key role in computation offloading by assisting in optimal UAVs placement, controlling the UAVs, and optimizing the resources of aerial base stations, resulting in efficient resource utilization. Through digital twin, the performance of UAVs acting as edge servers for computation offloading can be enhanced.

1.2 Motivation

To efficiently handle computationally complex and time-sensitive services, we need a combination of digital twin, UAV-MECs, and cloud computing architecture with low latency and energy consumption. Therefore, we propose a digital twin-assisted multi-layer network framework to minimize the task latency and energy consumption of IoT devices. This framework incorporates several key elements, including IoT device association with UAV-MECs, task offloading partitioning, transmission power allocation for IoT devices, and their processing rates. Using this framework, we can efficiently reduce IoT devices' energy consumption and task latency.

1.3 Thesis Objective

The main objective of this thesis is to develop a digital twin-assisted multi-layer network framework to minimize latency and enhance energy efficiency. We aim to optimize the association of IoT devices, task offloading, communication, and computational resource utilization in digital twin-assisted multi-layer networks to minimize both latency and energy consumption. This results in a non-linear and non-convex optimization problem. To solve this problem, we propose a two-stage scheme based on the K-means method and deep learning architecture for latency and energy minimization. We compare our proposed solution with two existing schemes and present simulation results to evaluate our multi-layer network's energy efficiency and latency minimization performance.

1.4 Thesis Contributions

The main contributions of this thesis can be summarized as follows:

- We mathematically formulate a non-linear and non-convex optimization problem to minimize latency and energy consumption of IoT devices in the digital twin-assisted multi-layer network by optimizing the IoT devices' association, transmission power, offloading portioning, and computation resources of IoT devices.

- We propose a two-stage algorithm to solve the optimization problem based on the K-means method and a deep learning architecture to solve the optimization problem with low computational complexity. We divided the problem into two subproblems: optimal association depending on the best channel condition after the K-means-assisted UAV-MECs placement and optimizing communication and computation resources through the deep learning method.

- We solve the optimization problem with two existing schemes: (i) based on K-means and outer-approximation (KOA) and (ii) K-means and interior point method (KIPM) for comparison with our proposed two-stage scheme in terms of performance.

- We present the extensive simulation results to evaluate the performance of our proposed multi-layer network in terms of latency minimization and reducing energy consumption. Also, we compare the performance of the proposed two-stage scheme with two existing schemes in terms of the scalability of the solution.

1.5 Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 covers the research on edge association and resource scheduling, network configuration optimization, network redundancy by the digital twin, and machine learning (ML) roles in digital-twin-assisted networks. Chapter 3 discusses the system model adopted to formulate the low latency and energy efficiency optimization problem in multi-layer network. Chapter 4 presents the proposed two-stage scheme to solve the optimization problem, followed by the simulation results and their analysis. Finally, Chapter 5 concludes the thesis and presents the future research directions.

Chapter 2

Literature Review

This chapter presents existing studies on digital twins, multi-stage networks, and edge processing technologies. We focus mainly on existing literature on adaptive edge association, resource scheduling, network configuration optimization, digital twin-enabled Industry 4.0, network redundancy, and the role of machine learning in digital twin-assisted networks. Fig. 2.1 shows the flow of the literature review by highlighting the roles of the digital twins in networks and describing the benefits of utilizing the digital twin in networks. We highlight how our proposed work differs from existing research by utilizing multiple technologies such as digital twin and multi-stage networks, applying different solutions to optimization problems, and adding value to existing research work.

2.1 Edge Association

Computationally constrained devices are used to offload their tasks for computation to deal with highly complex tasks. The devices are associated with the edge server for offloading the tasks, which is known as edge association. In this section, we discussed the edge association's benefits for resource-constrained users' task offloading and highlights the earlier research done. In [13], Luo et al. discussed different configurations of edge computation along with resource scheduling in edge computation to highlight the usefulness of UAV-MECs. The researchers formulated different optimization problems involving the edge association to achieve better performance of the network.

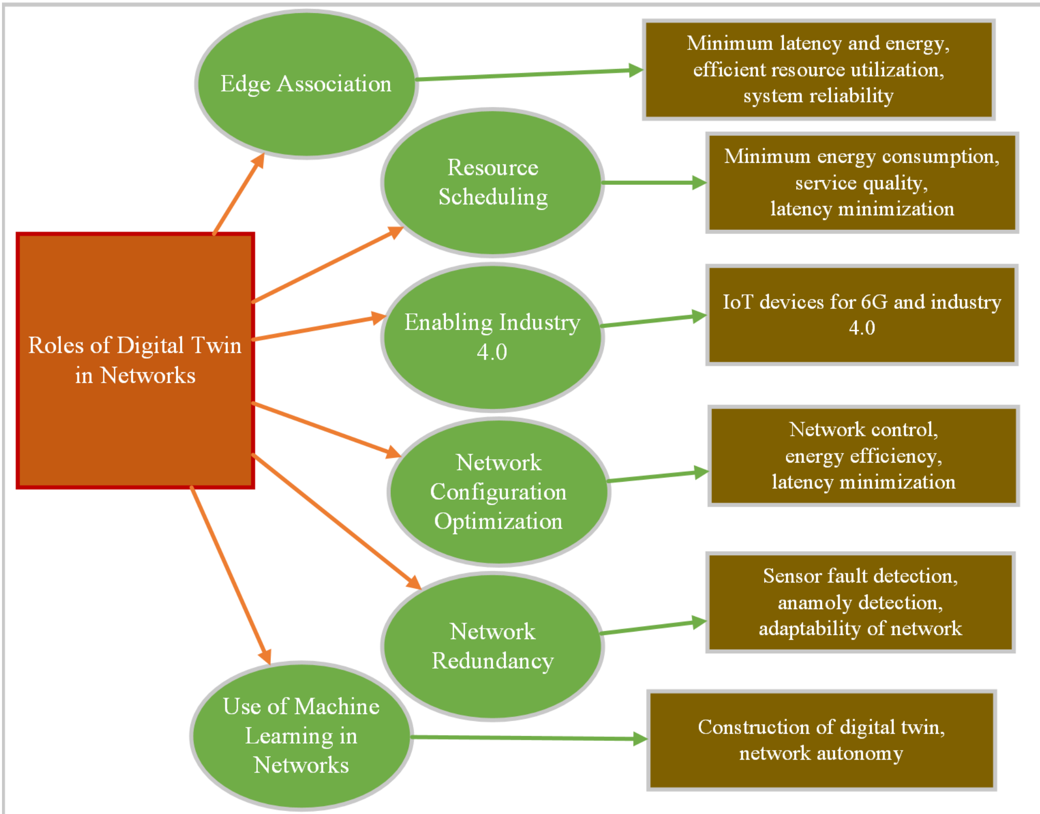


Figure 2.1: Roles of digital twin in wireless networks.

Chen et al. proposed the energy minimization problem in [25] for a digital twin-empowered MEC architecture through dependency-aware task offloading. An optimization problem is formulated to optimize the offloading decisions and resource allocation to minimize energy consumption through an asynchronous advantage actor-critic solution method. Similarly, in [26], Tan et al. proposed an adaptive caching scheme in a digital twin-assisted multi-stage network to minimize the energy for the heterogeneous IoT network. They applied evolutionary stability strategies to optimize energy and delay within the constraints of the caching capacity of a multi-stage network. A digital twin-empowered integrated sensing, communication, and computation network is presented in [27] by Li et al. to minimize the offloading energy consumption and beampattern performance of multi-input and multi-output systems. They utilized the UAVs to act as MECs. They solved the optimization problem by optimizing the multi-agent proximal policy with service delay constraints. A digital twin-assisted mobile edge computing architecture is presented by Duy et al. in [28] for IoT devices to minimize the end-to-end latency while having the constraints of computation resources and energy utilized. The stochastic offloading in the digital twin-assisted network for industrial Internet of Things (IIoT) devices is carried out by Dai et al. in [29] with utilizing deep learning to reduce energy consumption. Another deep learning training was carried out by Dong et al. in [30] through digital twin for associating the user with MECs to reduce the normalized energy consumption while ensuring the quality of services (QoS) for users.

Huynh et al. in [31] presented a latency minimization problem in an edge-cloud multi-layer network with ultra-reliable and low-latency communication (URLLC) protocol. The URLLC protocol ensures connectivity and low latency in future wireless networks in which multi-layer networks are used for parallel processing to reduce the user's latency. In [32], Huynh et al. considered the digital twin-assisted UAV-MECs network for latency minimization by optimizing the association, resource allocation, and task offloading. In a multi-tier computing network with a cloud layer in addition to the MECs layer, a latency minimization objective is achieved through optimizing the offloading policies, processing rates, and user association policies [17]. In [33], Lu et al. presented the wireless digital twin edge network for optimal edge association problem. They considered the constraints of dynamic network states and varying network topologies to gain the lowest system cost and better latency performance by solving the optimization problem with deep reinforcement learning. In [34] Sun et al. solved the offloading optimization through the training by the deep reinforcement learning (DRL) from the digital twin in the MEC network to minimize the offloading latency. By utilizing the intelligent reflecting services aided offloading in [35], Dai et al. utilized the federated reinforcement learning to minimize the system latency by considering the caching capacity of edge network and communication resources constraints. In the internet of vehicles (IoV), the digital twin-assisted MEC in [36] Yuan et al. utilized for minimizing the overall latency through deep deterministic policy gradient (DDPG) while considering the communication resources limitation. By utilizing the adaptive partial offloading, another latency minimization is solved through convex optimization by Mitsiou et al. in [37].

In [38], Yao et al. considered an intelligent, cooperative task offloading and service caching achieved through digital twin to maximize service-based system utility quality. A graph-based multi-agent reinforcement learning algorithm solution is presented with the constraint of computation and communication resources of the system. Another resource utilization maximization is achieved by Tang et al. in [39] through a deep reinforcement learning approach while maintaining the task latency deadline. An intelligent offloading framework by digital twin in the MEC network is presented by Zhang et al. in [40] for maximizing the utility of MEC computational resources in terms of services. The system reliability is also enhanced by the digital twin in the MEC network through digital twin, like in [41] Van et al. utilized convex optimization. A digital twin-enabled computation offloading is utilized by Wang et al. in [42] to enhance the system reliability by optimizing the offloading decision under uncertainty.

Table 2.1 summarizes the above discussion and highlights the benefits of edge association in terms of latency minimization, energy minimization, maximizing resource utilization, and enhancing system reliability by optimizing the offloading decision in the digital twin-assisted edge network, describing the constraints and solution algorithm. The edge association can be done without the utilization of digital twin. However, with digital twin, we have the real-time update of the environment, making it adaptive and more efficient in resource utilization. Edge association is necessary for the multi-layer network and helps users to compute computationally complex tasks through the digital twin.

Table 2.1: Summary of edge association related work.

Ref.	**Research Problem**	**Technologies**	**Objective**	**Constraints**	**Solution**
[25]	Digital twin empowered MEC server architecture for offloading	Digital twin and MEC servers	Minimize energy consumption	Delay requirement of tasks and computation resources constraint	Asynchronous advantage actor-critic method

[26]	Adaptive caching scheme for multi-layer network	Digital twin and multi-layer network	Optimize the delay and energy for heterogeneous IoT devices	Caching capacity constraint	Evolutionary stability strategies
[27]	Digital twin empowered integrated sensing, commmunication and computation network	Digital twin and UAV-MECs	Minimize energy consumption and beam-pattern performance	Computation resources constraint of UAVs and users	Multi-agent proximal policy optimization
[28]	Digital twin assisted MEC architecture	Digital twin, IoT, and MEC server	Minimizing the end-to-end latency	Energy consumption and computation resources	Distributed solution with alternate optimization
[29]	Stochastic offloading in digital twin network of IIoT	Digital twin, IIoT, and MEC server	Reducing energy consumption	Data processing efficiency	Lyapunov optimization and actor critic algorithm
[30]	Digital twin trained deep learning for MEC server	Digital twin and MEC server	Minimized the normalized energy consumption	QoS requirements	DL neural network
[33]	Wireless digital twin edge network	Digital twin and MEC server	Optimal edge association	Dynamic network states and network topology	DRL
[34]	Digital twin edge networks for offloading dicision training	Digital twin and MEC server	Minimizing the offloading latency	Migration service cost constraint for user mobility	Actor-critic DRL
[35]	Intelligent reflecting services aided task offloading	Digital twin edge network	Minimizing the system latency	Service caching capacity and communication resources constraints	Federated DRL

[36]	Digital twin-driven vehicular task offloading and intelligent reflecting services configuration framework	Digital twin, MEC server, and IoV	Minimizing the overall delay and energy consumption	Communication resources constraint	DDPG
[37]	Digital twin-aided grant free random access adaptive partial offloading	Digital twin and MEC server	Minimizing the average delay of partial offloading	Communication resources and computation resources constraints	Convex optimization
[38]	Intelligent co-operative task offloading and caching	Digital twin and MEC server	Maximizing the quality of service-based system utility	Service caching capacity of MEC server and communication resources of users	Graph-based multi-agent RL algorithm
[39]	Digital twin-assisted task assignment	Digital twin and UAV-MECs	Maximizing the resources intensive utilization	Task time constraints	DRL approach
[40]	Digital twin-driven intelligent task offloading framework	Digital twin and MEC server	Maximizing the MECs sytem computing services	Computation resources limitation	DRL
[41]	MEC-based URLLC architecture	Digital twin and MEC server	Reliablity and latency for networked metaverse	Communication and computation resources constraints	Convex optimization algorithm
[42]	Digital twin-enabled computation offloading	Digital twin and UAV-MECs	Optimizing the offloading decision under uncertainity	Energy consumption limit	Upper confidence bound-based stable matching algorithm

2.2 Resource Scheduling

In wireless networks, optimal resource scheduling allocation is essential to maximize network resource utilization. The digital twin is also useful for resource scheduling in the MEC network. Here, we discussed the recent work related to resource scheduling and highlighted the benefits of resource scheduling in networks.

Duong et al. in [14] proposed a digital twin-assisted edge-computing network with MECs. They solved an optimization problem for low latency of IoT devices by optimizing the communication and computation resources of the network. Similarly, in [43], Huynh et al. went for latency minimization for industrial IoT environment in the two-layer network assisted with digital twin by optimizing the communication and computation resources. Duy et al. carried out a digital twin-aided MEC network in [18] by optimizing the user association, processing rates of industrial IoT devices, and intelligent task offloading to get low end-to-end latency. In [5], Li et al. proposed a digital twin framework for the IoT network with the UAVs to act as MECs for mission-critical services to reduce the latency through optimal offloading to UAV-MECs. Huynh et al. considered a multi-layer network in [44] to optimize communication and computation resources for low computation latency.

The digital twin is utilized for scheduling heterogeneous edge network resources by Xu et al. in [45]. They utilized multi-agent DRL in wireless multi-stage networks to minimize user task completion time. A digital twin-assisted wireless network framework is presented by Yang et al. in [46] for managing the network resources to achieve the reduced transmission delay of the network for users. They utilized alternating optimization (AO) for the iterative solution of each variable in optimization while considering the limits on energy consumption and model accuracy. Digital twin plays an essential role in network resources management, like by Gong et al., a resources management by digital twin in [47] is presented for the vehicular edge computing (VEC) network. They utilized the DDPG algorithm to minimize the overall response time while considering the mobility of vehicles and time-varying environments for the network. In [48], Zhang et al. proposed an architecture of wireless computing power networks-empowered digital twin for efficient resource utilization. They utilized the Shapley value and double auction scheme to achieve low latency and digital twin error by considering the heterogeneous nature of computing nodes. In [19], Vu et al. proposed a joint optimization in a three-layered architecture with optimal offloading and resource allocation to reduce energy consumption with the constraint on service latency. For the URLLC and edge computing Liu et al. applied the extreme value theory in [20] by imposing the constraints for edge association to minimize the energy consumption by the users. To minimize energy consumption in a dense multi-device system, a joint optimization problem is solved by Zhou et al. in [21] by optimizing the device association with the MEC server, task offloading, and resource allocation to devices and servers. The digital twin wireless control and resource optimization is carried out by Li et al. in [49] for beamforming and transmission power optimization in the network. They utilized the two-stage, convex, and iterative optimization to minimize user transmission power.

In [50] Dai et al. presented the digital twin-assisted MEC network for the service placement to maximize the MEC servers' services by optimally utilizing the computational resources. They utilized the Merkle tree to optimize while ensuring the quality of service for users. The QoS is ensured by the digital twin in the virtual reality network by Feng et al. in [51] through the greedy-style heuristic algorithm for fairness of resource allocation. They maximized the quality of experience (QoE) of the worst-case head-mounted display users.

A summary of the discussion is given in Table 2.2; the advantages of resource scheduling by digital twin, which minimizes latency, reduces energy consumption and power and enhances service quality in digital twin-assisted MEC networks. Also, the algorithm utilized for resource scheduling is presented with optimization constraints. The digital twin enhances the resource scheduling in the edge network.

Table 2.2: Summary of resource scheduling related work.

Ref.	**Research Problem**	**Technologies**	**Objective**	**Constraints**	**Solution**

[45]	Digital twin-driven edge-end collaborative scheduling of heterogeneous resources	Digital twin and wireless multi-layer networks	Task completion minimization	Task deadline	Multi-agent DRL
[46]	A framework for digital twin-based wireless network	Digital twin, wireless network	Reduced the transmission delay of system	Total energy and model accuracy	AO for iterative solution
[47]	Digital twin model for the network management	Digital twin and VEC	Minimize the overall response time	Mobility of vehicles and time varying environment	DDPG
[48]	An architecture of wireless computing power networks-empowered digital twin	Digital twin and wireless computing power networks	Efficient resource utilization, low latency and digital twin error	Heterogeneous nature of computing nodes	Shapley value and double auction scheme
[50]	Digital twin-assisted MEC architecture	Digital twin and MEC servers	Maximizing the services by the MEC servers	QoS from wireless device	Merkle tree
[51]	Digital twin-enabled QoE optimal problem for wireless virtual reality system	Digital twin virtual reality	Maximize the QoE of the worst-case for head-mounted display users	Computational and communication resources constraints	Greedy-style heuristic algorithm
[49]	Digital twin model for the wireless control and resources optimization	Digital twin and wireless networks	Minimize the transmission power	Limited range of beamforming power	Convex approximation and iterative optimization

2.3 Digital Twin Enabling Industry 4.0

Industry 4.0 refers to the fourth industrial revolution, which is enabled by advanced communication technologies, digitalization, machine learning, and artificial intelligence. Digital twin technology enables the environment of Industry 4.0, and the earlier research work is presented here.

In [52], Han et al. presented the survey on a 6G-empowered industrial digital twin network. They discussed the potential application of industrial digital twin like monitoring, simulation, and network control, which is essential for the industry 4.0 environment. They also highlighted the technologies that enable the 6G-empowered industrial digital twin network, like artificial intelligence for communication and computation co-design. This proposed ecosystem powers connecting the humans, machines, and data infrastructure to enable numerous novel applications. In [53], Luan et al. proposed an intelligent industrial system based on digital twin and artificial intelligence (AI) to improve network performance. They proposed a Dijkstra's algorithm for reducing the load delay of the system.

A construction of digital twin-empowered industrial Internet of things (IIoT) is presented by Xiang et al. in [54]. The digital twin-driven IIoT can make intelligent decisions during run-time. They proposed the unique digital twin construction by utilizing credibility-weighted swarm learning to eliminate the need for a central server for the system's digital twin generation to enhance the system's reliability and minimize energy consumption while considering the limitation of computation resources.

Efficient integration of digital twins for industry 4.0 is carried out by Kherbache et al. in [55] to optimize the performance of industrial systems. Real-time network management is achieved by replicating IIoT sensors, actuators, and communication infrastructure in the digital twin domain. They achieved optimal resource distribution, predictive maintenance, and diagnosis in industrial systems. They addressed the communication challenges between the network nodes and implemented the prototype through the software-defined controller (SDN) in [55].

A summary of the discussion is given in Table 2.3 highlighting the digital twin gains in the IoT devices that pave the path for Industry 4.0. In the presented work, the digital twin improves the latency and the reliability of IoT device networks.

Table 2.3: Summary of digital twin for industry 4.0 related work.

Ref.	Research Problem	Technologies	Objective	Constraints	Solution
[52]	A survey on 6G-empowered industrial digital twin system	Digital twin, IoT, and industry 4.0	Potential application of industrial digital twin in 6G like monitoring, simulation, and controlling	Challenges of digital twin in IIoT	Enabling technologies like artificial intelligence and communication and computation co-design
[53]	An intelligent industrial system based on digital twin and AI	Digital twin, AI, and SDN	Improved network performance and reduced the delay	Allowable load delay	Dijkstra's algorithm
[54]	Credibility weighted swam learning to construct digital twin models	Digital twin and IIoT	Enhancing the system reliability and minimizing energy consumption	Computation and communication resources limitation	Swarm learning
[55]	Efficient integration of digital twin in industry 4.0	Digital twin and IIoT	Predictive maintenance, network diagnosis and resources allocation	Scheduling the communication between nodes	Prototype implementation by SDN controller

2.4 Network Configuration Optimization

This section will discuss how digital twins help optimize network configuration to enhance the network's performance. In [56], Xie et al. presented the digital twin-assisted UAVs network. They used mmWave radar imaging for UAVs to characterize their radio frequency to accomplish channel modeling through three-dimensional (3D) ray tracing. They achieved the smart operation and administration of the UAVs network. A digital twin-based terahertz (THz) signal guidance framework is proposed in [57] by Pengnoo et al. to utilize the metasurface reflector to model, predict, and control signal propagation characteristics. They applied the cone selection by ensuring the line of sight between the sender and receiver. A micro-services-based digital twin framework is designed and implemented in [58] by Lombardo et al. for network management and control. They applied the Dijkstra algorithm to augment the capabilities of the network managers.

An integrated data and energy transfer in a cell-free network with the help of a digital twin is proposed in [59] by Shui et al. to achieve the energy sustainability of the network. They employed the double-parameterized deep Q-network for data and energy transfer to enhance the network's energy sustainability. A digital twin-assisted area-controlled mobile ad-hoc networking is proposed by Ono et al. in [60]. They achieved the efficiency of the network by reducing the traffic volumes while ensuring the packet arrival rates by employing a relay range restriction algorithm. In [61], active noise control is proposed for reduced frequency noise in network communication. They employed a reference least mean square algorithm with a digital twin filter to avoid the high-cost processors for active noise control.

Yaqoob et al. presented the digital twin-based approach to deploy the beyond fifth generation (5G) network core function by employing the network slicing in [62]. They observed the traffic over the deployed slices and analyzed the round-trip time observed in the network by designing the digital twin on Open5GS.

Table 2.4 summarizes the digital twin-enabled network configuration; we observed the digital twin adds latency minimization, energy efficiency, operation, and network control gains through the optimal network configuration.

Table 2.4: Summary of network configuration optimization related work.

Ref.	Research Problem	Technologies	Objective	Constraints	Solution

[56]	A framework for digital twin-based UAVs application	Digital twin, UAVs, and mmWave radar imaging	Smart operating and administration of UAVs network	Channel modelling of UAVs	3D ray-tracing
[57]	A digital twin based THz signal guidance	Digital twin, metasurface reflector and THz communication	Model, predict and control signal propagation characteristics	Line of sight (LoS) between receiver and sender	Digtal twin based cone selection
[58]	A microservices-based digital twin framework for network management and control	Digital twin and artificial intelligence (AI)	Augmenting the capabilities of network manager	Network adaptation for different applications	Dijkstra algorithm
[59]	An integrated data and energy transfer in cell-free network	Digital twin and cell-free network	Energy sustainability	Guaranteed optimal solution	Double parameterized deep Q-network
[60]	An area controlled mobile ad-hoc networking	Digital twin and mobile ad-hoc networks	Reduced traffic volumes	Packet arrival rates	A relay range restriction algorithm
[61]	Active noise control based on digital twin architecture	Digital twin and active noise controller	Reduced frequency noise	High cost processors	Digital twin filtered-reference least mean squares algorithm
[62]	A novel digital twin based approach to deploy the beyond5G network core functions	Digital twin and beyond5G networks	Improved round trip time in all scenarios	Packet loss rate	Open5GS

2.4.1 Network Redundancy

The digital twin can add network redundancy and make it adaptive for different circumstances. In [63] Yu et al. presented a digital twin-driven self-healing mechanism for 6G edge networks. They utilized the graph neural network (GNN) for service re-deployment during abnormal conditions. A framework of a real-world IoT mobile virtual network operator is proposed in [64] by Geibler et al. to enhance the network survivability under unforeseen conditions by utilizing the overload control mechanism.

The digital twin makes the network redundant by timely detection of the faults in the network. In [65], a digital twin architecture for sensor fault detection is proposed by Darvishi et al. to achieve the timely detection of false alarms and classification of faulty sensors through a neural network simulator and estimator. In [66], Hasan et al. proposed a digital twin-inspired approach for sensor fault detection by utilizing the Wasserstein generative adversarial network (GAN). A digital twin is utilized for sensor fault detection, isolation, and accommodation in [67] by Darvishi et al. They proposed a multi-layer perceptron neural network for anomaly detection in sensors' measurements. The wireless digital twin platform for wireless software applications evaluation is proposed in [68] by Lai et al. for real-time evaluation through the Kalman filter and recurrent neural network (RNN). A security framework for cloud-assisted body area networks through digital twin is proposed by Sama et al. in [69]. They achieved the low-cost cyber-physical threat prediction earlier to configure the network accordingly. In [70] Wang et al. proposed the digital twin cyber platform based on network function virtualization for network security and management from external threats.

A summary in Table 2.5 shows the research problem, objective, constraints, and proposed solution for a digital twin-assisted network. The research discussed above-enabled network adaptability and timely fault detection through digital twins in the networks.

Table 2.5: Summary of network redundancy related work.

Ref.	Research Problem	Technologies	Objective	Constraints	Solution

[63]	Digital twin-driven service self-healing mechanism for 6G edge networks	Digital twin and edge networks	Service re-deployment during abnormal conditions	Resources and location constraints	Graph neural network (GNN)
[64]	A framework of a real-world IoT mobile virtual network operator	Digital twin and IoT	Network survivability under unforeseen conditions	IoT traffic	Overload control mechanism
[65]	Sensor fault detection architecture for digital twin	Digital twin and IoT sensors	Timely detection of false alarms and classification of faulty sensors	Number of layers and nodes in layers	Neural network simulator and estimator
[66]	Digital twin-inspired fault detection approach	Digital twin and IoT	Sensor fault detection	Fault types limitation	Wasserstein GAN
[67]	Sensor fault detection, isolation, and accommodation through digital twin	Digital twin and industry 4.0	Anomaly detection in measurements of sensors	Different levels of faults	Multi-layer perceptron neural network
[68]	Wireless digital twin platform for wireless software applications evaluation	Digital twin and wireless software applications	Wireless software applications evaluation	Real-time evaluation	Kalman filter and RNN
[69]	An integrated security framework for cloud-assisted body area networks	Digital twin and body area networks	Cyber-physical security optimization at low cost and time	Digital twin for cloud-assisted body area network	Digital twin environment prediction of threats
[70]	Digital twin cyber platform based on network function virtualization	Digital twin and network function virtualization	Network security and management	Design and functions optimization	A prototype of proposed network designed

2.4.2 Role of Machine Learning in Digital Twin-assisted Network

Here, we will highlight the benefits of machine learning in digital twin-assisted networks and the earlier work done in this domain is discussed. The network can be optimized in real-time through machine learning and artificial intelligence algorithms for future-generation networks using digital twin technology [4, 6]. A survey presented by Wu et al. in [71] highlights the definition, enabling technologies, issues, and usefulness of digital twin in different domains. The digital twin enables the control of the physical network through the bi-directional communication link [3, 5]. Digital twin in wireless networks generates the virtual copy of wireless networks, which can be utilized for the adaptive edge association for the users, enabling the path to future generation networks [6]. Sun et al. solved the movement and unpredictability of users in MEC networks by utilizing the digital twin to reduce the offloading latency in [1].

In [72], Vilas et al. utilized the untrained neural network and conditional GAN for the channel state information through digital twin. They improved the performance of the network by providing low-overhead channel state information. An iterative optimization through particle swarm optimization and DDPG is utilized in [73] by Cui et al. to enhance the transmission rate of users. They managed the cell-free system with reconfigurable intelligent surfaces through digital twin. In [74], Xu et al. presented the digital twin-empowered wireless body area networks to enhance the network's performance by maximizing the energy efficiency of sensors. A random graph-inspired DDPG algorithm is utilized. The network's performance regarding users' mean transmission rate is enhanced using the GNN in [75]. Zhang et al. in [75] utilized the digital twin with machine learning to manage the THz communication resources.

Machine learning also helps to construct a digital twin; in [76] Qian et al. proposed a federated learning-based IoT digital twin network. They minimized the energy consumption during the construction of the digital twin. Lian et al. proposed a lightweight digital twin-empowered air-ground network architecture in [77]. They utilized federated learning with a distributive incentive scheme to construct digital with reduced energy consumption and enhanced model accuracy of the digital twin. Deng et al. presented the implementation of the digital twin approach for the 6G wireless network in [78]. They utilized the knowledge graph and GNN for digital twin construction to achieve high autonomy for the network. To deal with the uncertainty of the digital twin, a Bayesian framework is proposed by Ruah et al. in [79]. They utilized the multi-agent RL algorithm to reduce the model uncertainty of the digital twins.

Federated transfer learning is employed in a communication-assisted-sensing scenario in the digital twin-empowered network by Mu et al. in [80]. They enhanced the communication efficiency of the network by ensuring data safety. A digital twin network with asynchronous federated learning and DDPG is utilized for data privacy and security by He et al. in [81]. They enhanced the efficiency and accuracy of intrusion detection in networks. A digital twin-assisted wireless network for edge-processing proposed by Lu et al. in [82]. They proposed federated learning to enhance the system's reliability and security in addition to the low latency of edge processing.

Table 2.6 highlights the summary of the roles of machine learning in the digital twin. Machine learning improves the digital twin-assisted network's performance, efficiently constructing the digital twin of the network and improving the network's autonomy.

Table 2.6: Summary of the roles of machine learning in digital-twin assisted networks related work.

Ref.	**Research Problem**	**Technologies**	**Objective**	**Constraints**	**Solution**
[72]	Channel state information based on digital twin	Digital twin and machine learning	Low overhead channel state information	Location of users and base stations	Untrained neural network and conditional GAN
[73]	Reconfigurable intelligent surfaces-assisted user-centric cell-free system managed by digital twin	Digital twin and reconfigurable intelligent surfaces	Enhances the sum-rate for users	Large and complex solution space	Particle swarm optimization and DDPG for iterative optimization
[74]	A digital-twin-empowered wireless body area networks	Digital twin and wireless body networks	Maximizing the energy efficiency of sensors	Reliability for emergency critical services	Random graph-inspired DDPG

[80]	A communication-assisted-sensing scenario with federated learning in digital twin empowered mobile network	Digital twin and mobile network	Improved the communication efficiency of mobile network	Data safety	Federated transfer learning
[81]	A federated continuous learning framework based on digital twin network	Digital twin, UAVs, and IoT	Higher efficiency and accuracy of intrusion detection system	Data privacy and security	Asynchronous federated learning and DDPG
[82]	A digital twin wireless networks for edge processing	Digital twin and 6G networks	Enhancing the reliability and security of system	Communication resources and training data limitation	Blockchain empowered federated learning framework and multi-agent RL

2.5 Summary

The Tables 2.1, 2.2, 2.3, 2.4, 2.5, 2.6 summarize the existing works related to the multi-layer network, digital twin, machine learning and edge association is highlighted with the advantages they have achieved in terms of latency, energy, reliability and adaptability of the network. So, previous work highlights the importance of edge association, resource scheduling, and network configuration to improve the network's performance. Therefore, this thesis is considered a digital twin-assisted three-layer network that optimizes edge association, resource scheduling, and network configuration for a low-latency and energy-efficient multi-layer network. We propose a two-stage scheme to solve the optimization problem and enhance the performance of the network in terms of latency minimization and energy efficiency.

Chapter 3

Digital Twin-assisted Multi-Layer Network: Resource Optimization for Low-Latency and Energy-Efficiency

This chapter introduces the framework of a multi-layer network system model that utilizes digital twin technology. The proposed framework minimizes latency and energy consumption through a nonlinear and nonconvex optimization problem. In this problem, we optimize the allocation of edge devices, power transmission of IoT devices, the offloading process, and the computation resources of IoT devices. Although this optimization problem is defined in this chapter, it is nonlinear and nonconvex due to the objective function and some constraints.

3.1 System Model and Problem Formulation

We consider a digital twin-assisted offloading in a multi-layer network with U number of IoT devices and N number of UAV-MECs and a cloud server C for low-latency and energy-efficient communication and computation, as shown in Fig. 3.1. We assume that IoT devices are resource-constrained and may be unable to perform their computation tasks locally. A task of *u-th* IoT device

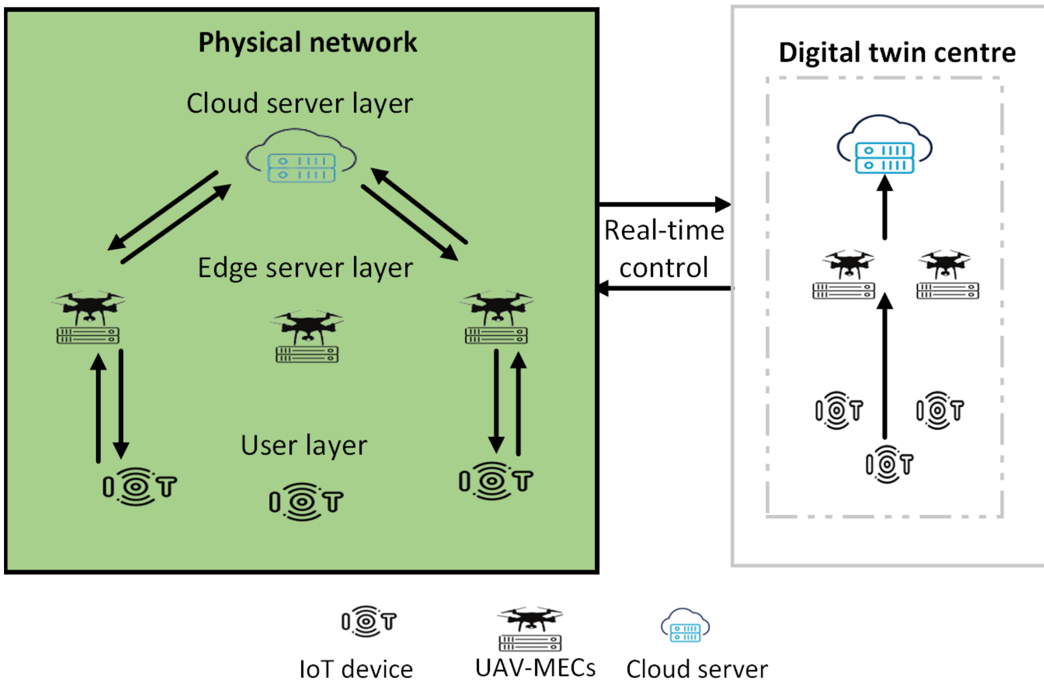


Figure 3.1: System model for digital twin-assisted multi-layer network.

can be represented by the tuple $J_u = [D_u, C_u, T_u^{MAX}]$, where the D_u represent the task size, C_u the computational CPU cycles required for the task, and T_u^{MAX} is the maximum allowable latency for the task. The task from *u-th* IoT device can be computed locally with the processing rate f_u^L and also can be offloaded to UAV-MECs with the distribution factor $\beta_u \in [0, 1]$. The binary variable for the association of IoT devices with UAV-MECs can be represented as $x_{u,n}$, where $x_{u,n} = 1$ if *u-th* IoT device is associated with *n-th* UAV-MEC and $x_{u,n} = 0$ otherwise.

The multi-layer network consists of the following layers:

Cloud layer: In the cloud layer, a cloud server is available with higher data processing rates f^C cycles/sec to deal with highly complex computational tasks.

Edge layer: The edge layer consists of N number of UAV-MECs, which are responsible for processing the offloaded tasks of IoT devices. The processing rate of *n-th* UAV-MECs is denoted by f_n^E cycles/sec. To minimize the tasks' latency from IoT devices, the task portion can be partially offloaded through sharing factor $\alpha_{un} \in [0, 1]$ to the cloud server.

User layer: The user layer consists of U number of IoT devices. The task of *u-th* IoT devices can be locally computed with the processing rate f_u^L and also can be offloaded to UAV-MECs with the distribution factor $\beta_u \in [0, 1]$. We assume that the tasks of IoT devices are granular for offloading to achieve low-latency communications [32].

Channel Model

The channel between the UAV-MECs and the IoT devices is based on the line of sight (LOS) propagation, and there exist effects of attenuation, blockage, and shadowing [5]. We assumed UAVs and IoT devices have direct LOS [11]. The LOS model for the link between the *u-th* IoT device and *n-th* UAV-MEC is given as [83]:

$$h_{un} = \beta_o d_{un}^{-\zeta}, \tag{3.1}$$

where β_o denotes the reference channel gain at the reference distance of 1m and $\zeta \geq 2$ denotes the path loss exponent [83]. d_{un} represents the Euclidean distance between the *n-th* UAV-MECs and *u-th* IoT device: $d_{un} = \sqrt{e_{un}^2 + l_k^2}$, where e_{un} denotes the horizontal distance between the *u-th* IoT device and *n-th* UAV-MECs, and l_n^2 represents height of the *n-th* UAV-MECs from IoT devices.

The signal-to-noise ratio of the *u-th* IoT device connected with *n-th* UAV-MEC can be written as [83]:

$$\gamma_{un}(p, x) = \frac{\sum_{n \in N} x_{un} p_u ||h_{un}||^2}{N_o}, \tag{3.2}$$

where p_u represents the transmission power and N_o denotes the noise floor power. The data rate of the *u-th* IoT device connected with *n-th* UAV-MEC can be approximated as [32]:

$$R_u(p, x_u) = B \log_2[1 + \gamma_{un}(p, x_u)](bits/sec), \tag{3.3}$$

where B represents the bandwidth available.

Digital Twin-based Computation Model

The digital twin model of the proposed multi-layer network can be represented as:

$$DT = \{\bar{U}, \bar{N}, \bar{C}\}, \tag{3.4}$$

where $\bar{U}$, $\bar{N}$, and $\bar{C}$ denote the virtual replicas of IoT devices, UAV-MECs, and a cloud server respectively. We can control and manage the physical system through the digital twin by exchanging the data between the model and the digital twin instance [18]. Different tools are available to implement the digital twin concept like the Modelica, DELMIA, FlexSim, etc [32].

After processing the computation task, the data size is typically negligible, and the UAV-MECs can transmit it with high power as compared to the IoT devices' transmission power budget. Thus, we assumed that the downlink transmission rate is higher and the downlink latency after processing can be ignored [32].

Local processing latency

The digital twin model of the *u-th* IoT device is denoted by DT_u^L which is given as:

$$DT_u^L = (f_u^L, \bar{f}_u^L), \tag{3.5}$$

where f_u^L denotes the estimated processing rate at digital twin and $\bar{f}_u^L$ represents the error in the approximation. The *u-th* IoT device can partially offload its task's portion β_u to the UAV-MECs with the μ_u^L mean task arrival rate at users. The latency to process the task locally at *u-th* IoT device is given as [17]:

$$\bar{t}_u^L(\beta_u, f_u^L) = \frac{(1-\beta_u)C_u\mu_u^L}{f_u^L}. \tag{3.6}$$

Assuming that the digital twin approximation error is known earlier, the error in latencies can be computed as:

$$\triangle t_u^L(\beta_u, f_u^L) = \frac{(1-\beta_u)C_u\mu_u^L\bar{f}_u^L}{f_u^L(f_u^L - \bar{f}_u^L)}. \tag{3.7}$$

Thus, the actual latency of local computation can be written as [17]:

$$t_u^L(\beta_u, f_u^L) = \bar{t}_u^L(\beta_u, f_u^L) + \triangle t_u^L(\beta_u, f_u^L). \tag{3.8}$$

Offloading latency

The latency to offload the β_u task portion of *u-th* IoT device's can be written as [17]:

$$t_u^O(\beta_u, p_u, x_u) = \frac{\beta_u D_u}{R_u(p, x_u)}. \tag{3.9}$$

UAV-MECs processing latency

The digital twin model for the *n-th* UAV-MECs is denoted by DT_n^E and given as:

$$DT_n^E = (f_n^E, \bar{f}_n^E), \tag{3.10}$$

where f_n^E represents the estimated processing rate and $\bar{f}_n^E$ as the approximation error. As the task sharing variable $\alpha_{un} \in [0, 1]$ is responsible for offloading of the task to the cloud server from UAV-MECs, the estimated processing latency at UAV-MECs can be written as [17]:

$$\bar{t}^E(\beta, \alpha, x, f_n^E) = \frac{\sum_{u \in U} x_{un} \beta_u \mu_u^L (1 - \alpha_{un}) C_u}{f_n^E}. \tag{3.11}$$

The error in latency value and its digital twin estimation can be calculated as:

$$\triangle t^E(\beta, \alpha, x, f_n^E) = \frac{\sum_{u \in U} x_{un} \beta_u \mu_u^L (1 - \alpha_{un}) C_u \bar{f}_n^E}{f_n^E (f_n^E - \bar{f}_n^E)}. \tag{3.12}$$

As a result, the actual digital twin latency to compute the task at UAV-MECs can be expressed as: $t^E = \triangle t^E + \bar{t}^E$.

Offloading latency UAV-MECs to Cloud

Each UAV-MECs offloads the tasks portion specified by $\alpha_{un} = [0, 1]$ to the cloud for parallel processing. The offloading latency between the UAV-MECs and the cloud is given as [17]:

$$t^F(\beta, \alpha, x) = \sum_{u \in U} x_{un} \beta_u \alpha_{un} \mu_u^L \frac{D_u}{R^F}, \tag{3.13}$$

where R^F represents the data rate of offloading the task from UAV-MECs to the cloud server.

Cloud processing latency

The digital twin model for the cloud is denoted by DT^C which is given as:

$$DT^C = (f^C, \bar{f}^C), \tag{3.14}$$

where f^C denotes the estimated processing rate, and $\bar{f}^C$ represents the approximation error. Thus, the processing latency estimated by the digital twin is given as [17]:

$$\bar{t}^C(\beta, \alpha, x) = \frac{\sum_{u \in U} \sum_{n \in N} x_{un} \beta_u \alpha_{un} C_u \mu_u^L}{f^C}. \tag{3.15}$$

The error in latency estimation is calculated as:

$$\triangle t^C(\beta, \alpha, x) = \frac{\sum_{u \in U} \sum_{n \in N} x_{un} \beta_u \alpha_{un} \mu_u^L C_u \bar{f}^C}{f^C(f^C - \bar{f}^C)}. \tag{3.16}$$

Thus, the actual latency of the cloud processing given as:

$$t^C(\beta, \alpha, x) = \triangle t^C(\beta, \alpha, x) + \bar{t}^C(\beta, \alpha, x). \tag{3.17}$$

From the above (3.6) – (3.17), the overall latency for the task of *u-th* IoT device can be written as:

$$t_u(f_u^L, p, \beta, \alpha, x) = t_u^L(\beta_u, f_u^L) + t_u^C(\beta_u, p, x) + t^F(\beta, \alpha, x) + t^E(\beta, \alpha, x, f_n^E) + t^C(\beta, \alpha, x) + t_u^S. \tag{3.18}$$

where t_u^S denotes the synchronization latency to share network information from between physical to digital twin. The total energy consumed by the *u-th* IoT device for the local processing and offloading can be computed as [17]:

$$E_u(f_u^L, p, \beta, x) = (1 - \beta_u)\frac{\theta_u}{2} C_u (f_u^L)^2 + p_u \frac{\beta_u D_u}{R_u(p, x)} + (1 - \beta_u)\frac{\theta_u}{2} C_u \frac{(\bar{f}_u^L)^2}{(f_u^L - \bar{f}_u^L)^2}, \tag{3.19}$$

where the constant $\frac{\theta_u}{2}$ is the average activity factor of the *u-th* IoT device [32]. There is also an error term in energy consumption due to the approximation error of digital twin.

3.1.1 Problem Formulation

In this thesis, we focus on minimizing task latency and the energy consumption of IoT devices in multi-layer network. We mathematically formulate the optimization problem to jointly optimize IoT devices' association with UAV-MECs, computational capacity, task portioning between the layers, and transmission powers of IoT devices for task offloading and communication.

Input

The input values which we give to this problem are:

- Number of IoT devices (U)
- Channel gains for reference distance (β_o)
- Number of UAV-MECs (N)
- System bandwidth (B)
- Location coordination of IoT devices and UAV-MECs ($\{x_u, y_u\}$ & $\{x_n, y_n, h_n\}$)
- All tasks latency deadline (T_u^{MAX})
- Maximum capacity of UAV-MECs association (M^{MAX})
- Total computational capacities of all three layers (F_u^{MAX})
- Minimum data rate requirement of *u-th* IoT device (R_u^{MIN})
- Maximum energy consumption by IoT devices (E_u^{MAX})
- Maximum transmission power available (P_u^{MAX})

Optimization Variables

The variables which will be optimized in this problem are:

- Task portioning (α_{un} & β_u)
- IoT devices association (x_{un})
- Computational capacity of IoT devices (f_u)
- Communication power (p_u)

Output

In the optimization problem, we minimized the latency and energy consumption by jointly optimizing the network's computational and communication resources and association of IoT devices. The output will be the optimized variables mentioned above, resulting in minimized task latency and energy consumption.

3.1.2 Optimization Constraints

We consider different constraints under which to achieve the objective:

- **C1:** The total task computation and offloading latency in (3.18) should be less than the task's deadline T_u^{MAX}.

$$t_u^L(\beta_u, f_u^L) + t_u^O(\beta_u, p, x) + t^F(\beta, \alpha, x) + t^E(\beta, \alpha, x, f_n^E) + t^C(\beta, \alpha, x) + t_u^S \leq T_u^{MAX}, \forall u, n. \tag{3.20}$$

- **C2:** Energy consumption in the computation by IoT devices in (3.19) should be less than the maximum allowable energy consumption limit denoted by E_u^{MAX}. The maximum energy consumption specified can be found by IoT devices' mean task generation rate and the power budget specified for the computation.

$$E_u(f_u^L, p, \beta, x) \leq E_u^{MAX}, \forall u. \tag{3.21}$$

- **C3:** Quality of service ensures that the data rate of *u-th* IoT device in (3.3) should be greater than the minimum data rate represented by R_u^{MIN}.

$$R_u(p, x) \geq R_u^{MIN}, \forall u. \tag{3.22}$$

- **C4:** Range of task sharing factor between *u-th* IoT device β_u to UAV-MECs should be between 0 and 1.

$$0 \leq \beta_u \leq 1, \ \forall u. \tag{3.23}$$

- **C5:** Range of task sharing factor of *u-th* IoT device between *n-th* UAV and cloud should be between 0 and 1.

$$0 \leq \alpha_{un} \leq 1, \; \forall u, n.$$

- **C6:** Maximum transmission power budget of *u-th* IoT device should be less than the maximum available transmission power budget denoted by P_u^{MAX}.

$$0 \leq p_u \leq P_u^{MAX}, \; \forall u. \tag{3.24}$$

- **C7:** Computational capacity bound of *u-th* IoT devices is bounded by maximum capacity denoted by F_u^{MAX}.

$$0 \leq f_u^L \leq F_u^{MAX}, \; \forall u. \tag{3.25}$$

- **C8:** The *u-th* IoT device can only be associated with one UAV-MEC at a given time.

$$\sum_{n \in N} x_{un} \leq 1, \forall \, u. \tag{3.26}$$

- **C9:** The number of IoT devices that can be associated with the *m-th* UAV-MEC is bounded by the maximum association capacity of UAV-MECs represented by M^{MAX}.

$$\sum_{u \in U} x_{un} \leq M^{MAX}, \forall \, n. \tag{3.27}$$

- **C10:** Binary association variable can be 0 or 1.

$$x_{un} = \{0, 1\}, \forall \, u, n.$$

3.1.3 Mathematical Formulation

We formulate the optimization problem to minimize the task latency and energy consumption for IoT devices as follows:

$$\min_{\beta,\alpha,x,p,f} : \sum_{u} (\omega t_u(f_u^L, p, \beta, \alpha, x) + (1-\omega) E_u(f_u^L, p, \beta, x)) \tag{3.28}$$

Subject to: $C1 - 10$,

where ω denotes the weight associated with the objective function. The weight prioritizes task's latency and energy consumption of IoT devices. The value of ω ranges from 0 to 1, where 0 means that the optimization will focus only on energy consumption, and 1 means that the objective is to minimize latency. The optimization problem in (3.28) is a mixed integer non-linear and non-convex problem because of the constraints of $C2, C4$ and $C8$ and the non-linear multi-objectives in terms of optimization variables.

3.2 Summary

This chapter introduces the system model for a digital twin-assisted multi-layer network. We present a mathematical formulation for the task latency and energy consumption of IoT devices in the network. We also define an optimization problem to minimize task latency and energy consumption while optimizing the network's association, communication, and computation resources. We explain the objective function, inputs, optimization variables, and constraints considered in the formulation.

Chapter 4

Proposed Scheme and Simulation Results

This chapter introduces a two-stage scheme that utilizes the K-means method and deep learning architecture. The proposed scheme is compared with two existing schemes, and the simulation environment and results are presented in terms of task latency and energy consumption.

4.1 Solution Approach

In the proposed multi-layer network, IoT devices and UAV-MECs are placed in a 2-dimensional frame with UAV-MECs positioned at a specific height. The optimization problem in (3.28) can be solved using an exhaustive search method to find a global solution. However, this approach could be computationally inefficient for large-scale networks. Therefore, we propose a two-stage scheme described in Algorithm 1 based on the K-means method and deep learning architecture approach to reduce the computation complexity. In the first stage, we apply the K-means learning algorithm [84] to cluster IoT devices based on the number of UAV-MECs available in the system. The centroid of these clusters is then assigned as the location of UAV-MECs to ensure a high connectivity range for offloading tasks of the IoT devices. After the placement, we make the association for offloading based on the best channel condition with the UAV-MECs satisfying constraints $C8 - C10$. Once the association is made, the problem changes to (4.1) with the optimal association variable (x^*).

$$\min_{\beta,\alpha,p,f} : \sum_{u} \left(\omega t_u(f_u^L, p, \beta, \alpha, x^*) \, + (1-\omega) E_u(f_u^L, p, \beta, x^*)\right) \tag{4.1}$$

Subject to: $C1 - C7$.

In the second stage, we utilize a deep learning architecture to solve the optimization problem in (4.1). The deep neural learning architecture is consists of the following layers: (i) the input layer, which takes input of the initial values of (p, f, β, α) variables that we want to optimize. The size of input layer depends on the length of the array containing these variables; (ii) hidden layers, which are middle layers with different numbers of neurons to capture the insights of optimization during training. For linear optimization with linear constraints, a neural network with one hidden layer is useful [85].

However, since we are dealing with a non-linear and non-convex optimization pattern, we opted for a deep learning architecture with multiple hidden layers to achieve better performance; (iii) the output layer, which has same length as input layer, provides the optimized values $(p^*, f^*, \alpha^*, \beta^*)$. The activation function for the neurons in adjacent deep learning architecture layers can impact optimization performance [85]. Thus, we select relative linear unit (RelU) as activation function. The elementary task for deep learning architecture is training, which is carried out on previous optimization datasets to train the weights and bases of neurons in deep learning architecture. We trained the proposed two-stage algorithm based on dataset obtained using K-means and interior point method (KIPM).

We compare the proposed two-stage scheme with the K-means and interior point method (KIPM) and K-means and outer approximation (KOA). Similar to proposed two-stage scheme, we divide the problem into two sub-problems: optimizing the association of IoT devices with the UAV-MECs placement using the K-means algorithm and optimizing network resources using the interior point and outer approximation algorithms.

Algorithm 1 : Proposed Two-Stage Scheme

Specifying the environment
Initialization
Step 1:Initializing all variables
Step 2:Placement of IoT devices
Output: Coordinates of IoT devices
Stage 1: K-means UAV-MECs placement
Step 1: Clustering of IoT devices
Step 2: Placement of UAV-MECs
Step 3: **User association** (x^*)
repeat
for $u \leftarrow 1$ to U **do**
  distance matrix $d(u, n)$ for each n
  Assign *u-th* IoT device to specific UAV-MECs
  Set $x_{un} \leftarrow 1$
  if Total connections of UAV-MECs u $> M_{max}$ **then**
    Set $x_{un} \leftarrow 0$
  else
    continue
  end if
end for
Stage 2: A deep neural network
Hidden layers specified according to optimization
Training of deep neural network
Initialize the optimization variable p, x, α, β
Training with previous optimization results
Testing of deep neural network
Input the initial values p, x, α, β to neural network
Computing the results
Output: Optimized variables ($p^*, f^*, \alpha^*, \beta^*$)

4.1.1 K-means assisted Interior Point Method (KIPM)

Algorithm 2 describes the KIPM scheme. First, the K-means method [84] is utilized for the optimal UAV-MECs placement to achieve a high connectivity range for the IoT devices offloading. The optimized association variable (x^*) got through making the association with the best channel conditions while considering the constraints $C8 - C10$. In the final stage, the interior point algorithm [86] is applied to solve the optimization problem in (4.1). The slack variables are defined for all the constraints, and the search region is specified where slack variables are positive and within the bounds of optimization variables.

Algorithm 2 :KIPM

Stage 1: Similar to Algorithm 1
Stage 2: Interior point method
Initialization
Initialize $x0$, A, B, A_{eq}, b_{eq}, lb, ub
Step 1: Slack variables for non-linear constraints
Define objective function and constraints
Step 2: Direct step
repeat
for $count \leftarrow 1$ until the algorithm gives minimum objective function **do**
    if Convergence criteria is satisfied **then**
        continue
    end for
end if
Output: Optimized variables

4.1.2 K-means assisted Outer Approximation (KOA)

Algorithm 3 outlines the KOA scheme. Similar to proposed scheme and KIPM, we solved optimal UAV-MECs placement using the K-means method. After placement, we associated IoT devices with the best channel conditions (x^*) to achieve a higher data rate for task offloading. Then, to solve the (4.1), we applied the outer approximation in the second stage to get the optimized energy and latency minimization variables. In outer approximation, the non-linear optimization problem is solved by linearizing the non-linear inequalities by making the outer approximation to these non-linearities. Then, by satisfying the constraints, the optimization area is specified. After that, the upper and lower bound of solutions found for the optimization problem are carried out until the difference between them is less than the specified criteria to find the minimum points of each optimization variable [87].

4.1.3 Computation Complexity Analysis

The computation complexity of the proposed two-stage scheme depends on both the K-means method and the deep learning approach. The K-means placement method has the computational complexity of $O(UN)$ [84]. Meanwhile, the deep learning approach's complexity is determined

Algorithm 3 :KOA

Stage 1: Similar to Algorithm 1
Stage 2: Outer approximation optimization
Initialization
Initialize $x0$, A, B, A_{eq}, b_{eq}, lb, ub
Step 1: Region of optimization specified
Approximate the linearized problem and constraints
A region of optimization specified
repeat
for $count \leftarrow 1$ until the algorithm gives minimum objective function **do**
    Optimization variables updated
    if Convergence criteria is satisfied **then**
        continue
    end for
end if
Output: Optimized variables

by the number of trainable parameters related to the number of IoT devices and UAV-MECs in the multi-layer network [85]. The computational complexity of the deep learning architecture is also $O(UN)$, which results in our proposed two-stage scheme complexity being $O(UN)$. The proposed two-stage scheme efficiently computes the optimization problem for large-scale networks. In comparison, the computational complexity of the sub-optimal KIPM scheme also depends on the complexity of K-means $O(UN)$ and interior point method complexity, which is $O(U^3N^3)$ [86]. This cubic complexity of KIPM makes this scheme inefficient for larger networks. In KOA, the accuracy of the results is compromised by using a linear approximation solution for our non-linear problem, and certain epsilon convergence criteria are specified. The computational complexity of KOA is $O(U^{3.5}N^{3.5}log(1/\epsilon))$ [11]. The proposed two-stage scheme is significantly lower in computation complexity than KOA and KIPM, making it a candidate for large-scale networks.

4.2 Performance Evaluation

We first compare the K-means placement with random placement of UAV-MECs to highlight the impact of optimal placement. Additionally, we evaluate the performance of our proposed two-stage scheme by comparing it with two existing schemes (KIPM and KOA) in terms of latency

Table 4.1: Simulation Parameters.

Parameters	Values
Number of IoT devices, U	80
Number of UAV-MECs, N	5
Task size, D_u	100 KBytes
Computational cycles requirement of task, C_u	800 Megacycles
Task computation completion deadline, T_u^{MAX}	2 sec
Energy consumption maximum limit, E_u^{MAX}	2 J
Minimum data rate requirement, R_u^{MIN}	1 Mbps
Maximum possible processing rate of IoT device, f_L^{MAX}	3 GHz
Maximum allowable transmission power for IoT device, P_u^{MAX}	23 dBm
Noise floor power, N_o	-174 dBm
Tasks arrival rate from IoT devices, γ_u^L	10 tasks/sec
Maximum associated devices with UAV-MECs, M^{MAX}	15
Digital twin approximation error, $\hat{f}$	3 %
Speed of light, c	3×10^8 m/sec
Bandwidth available, B	10 MHz

and energy consumption. Furthermore, we consider different scenarios of empirical weights and optimization of task portioning to highlight the usefulness of the proposed multi-layer network.

4.2.1 Simulation Parameters and Environment

We conduct the simulations on MATLAB software. A multi-layer network is considered where the UAV-MECs, IoT devices, and a cloud server are present. We consider N= 5 UAV-MECs and U= 80 number of randomly distributed IoT devices for the simulation scenario until specified further. The task size generated by the *u-th* IoT device is D_u= 100 KBytes, and the required computational resources are considered C_u= 800 Megacycles with maximum allowable latency T_u^{MAX}= 2 sec. The task generation rate at the IoT devices is γ_u^L = 10 $tasks/sec$. The maximum energy consumption limit is considered to be E_u^{MAX}= 2 J, and the minimum data rate requirement of $R_u^{MIN} = 1$ Mbps. The noise floor's power is N_o= -174 dBm, and digital twin error is considered 3 %. The simulation parameters considered are similar to [32] and presented in Table 4.1.

4.2.2 Simulation results

Fig. 4.1 shows the data rate (bps) versus the number of IoT devices for the random placement and the optimal K-means placement of UAV-MECs. In this scenario, we consider five UAV-MECs (N = 5), computational cycles requirement of task from *u-th* IoT device (C_u = 800 Megacycles), and task size (D_u = 100 KBytes). As the number of IoT devices increases, the data rate decreases because of interference caused by IoT devices associated with the same UAV-MECs and their association capacity limit. Optimal K-means placement yields a better data rate than random placement since it considers better channel conditions for IoT devices with optimal placement. Thus, it is crucial to have an optimal placement of UAV-MECs to achieve higher data rates for IoT devices, leading to a better end-user experience.

Task dropping can occur when an IoT device is too far away from any available UAV-MECs in the system or when there is an outage of association capacity. Fig. 4.2 shows the percentage of tasks dropping versus the number of IoT devices for fixed number UAV-MECs ($N = 5$), computational cycles requirement from *u-th* IoT device (C_u = 800 Megacycles), and task size (D_u = 100 KBytes). The percentage of tasks dropping increases as IoT devices increase due to a fixed number of UAV-MECs in the network. The maximum number of IoT devices that can be associated with the five UAV-MECs is 75, thus the 25 IoT devices will dropped from the association in case of 100 IoT devices. The task drop rate can be improved by adding more UAV-MECs in the system. The task drop is lower in the K-means UAV-MECs placement compared to the random placement. The K-means algorithm considers the optimal location of UAV-MECs to maximize the association of IoT devices for task offloading. Thus, the optimal placement is essential for accommodating a maximum number of IoT devices for offloading in the system.

Fig. 4.3 shows the impact of the number of IoT devices on energy consumption of IoT devices and task latency. We consider a scenario with five UAV-MECs (N = 5), computational cycles requirement of task (C_u = 800 Megacycles), and task size (D_u= 100 KBytes). We also compared our proposed two-stage scheme with two existing schemes, KIPM and KOA. The result indicates that energy consumption and latency also increase as the number of IoT devices increases. This

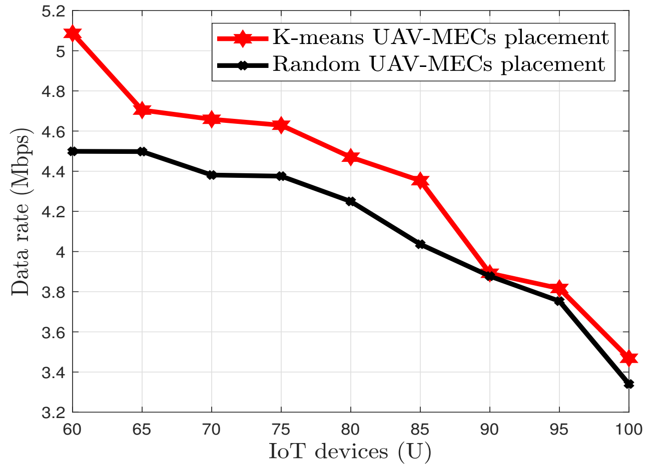


Figure 4.1: Data Rate comparison with the random user association and K-means UAV-MECs placement association.

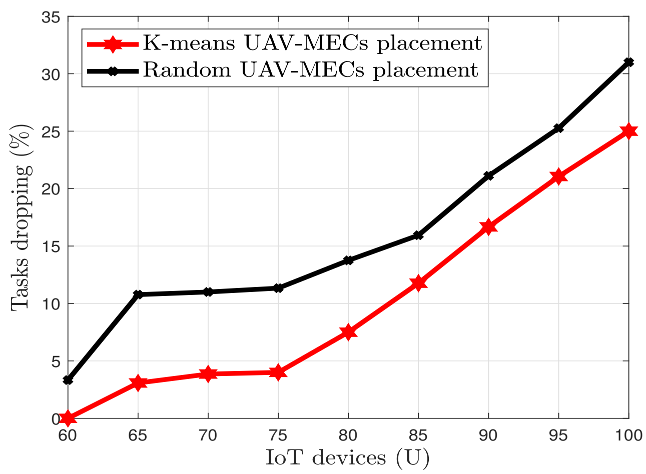
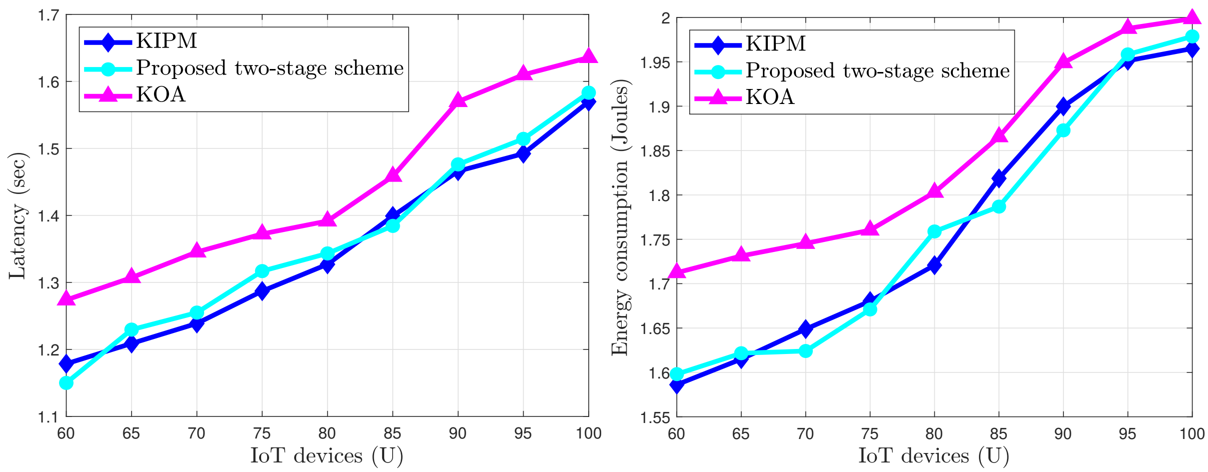


Figure 4.2: Percentage of tasks drop for IoT devices with random user association and K-means UAV-MECs placement association.

is because more tasks are generated, leading to longer computation time and increased computation energy. The proposed two-stage scheme performs well as compared to KOA and comparable to KIPM with less complexity. All three schemes' follow a similar trend. KOA has the highest latency and energy consumption due to the upper approximation of non-linearities in the optimization problem. It is essential to consider the network's capacity and the number of users using the available resources to ensure that the system meets the strict allowable latency and energy consumption requirements.

Fig. 4.4 illustrates the performance in terms of task latency and energy consumption of IoT



(a) (b)

Figure 4.3: Performance evaluation in terms of the impact of number of IoT devices when UAV-MECs (N = 5), computational cycles requirement (C_u = 800 Megacycles), and task size (D_u= 100 KBytes on (a) task latency and (b) energy consumption of IoT devices.

devices versus the number of UAV-MECs, ranging from 3 to 10. We consider IoT devices (U = 80), computational cycles requirement (C_u = 800 Megacycles), and task size (D_u = 100 KBytes). Fig. 4.4(a) shows the task latency for the proposed two-stage scheme and compares it with KOA and KIPM. On the other hand, Fig. 4.4(b) shows the energy consumption of IoT devices versus the number of UAV-MECs. The proposed two-stage scheme gives close results to the existing scheme of KIPM with reduced complexity. The number of UAV-MECs determines the resources available in the edge layer. As the number of UAV-MECs in the system increases, more tasks can be offloaded to the edge layer for faster computation, resulting in lower latency and lower energy consumption by IoT devices. However, after N = 5, the impact on latency and energy consumption reduction becomes insignificant as the essential requirements for the IoT devices can be fulfilled with five UAV-MECs. Thus, choosing the number of UAV-MECs based on IoT devices' traffic is vital to ensure optimal latency and energy requirements and a cost-efficient network design.

Fig. 4.5 illustrates the impact of computational cycles required for the tasks generated by IoT devices on task latency and energy consumption of IoT devices. We consider a range of computational cycles requirement (C_u = 800 - 960) Megacycles, with UAV-MECs (N = 5), IoT devices (U = 80), and task size (D_u = 100 KBytes). The task latency and energy consumption

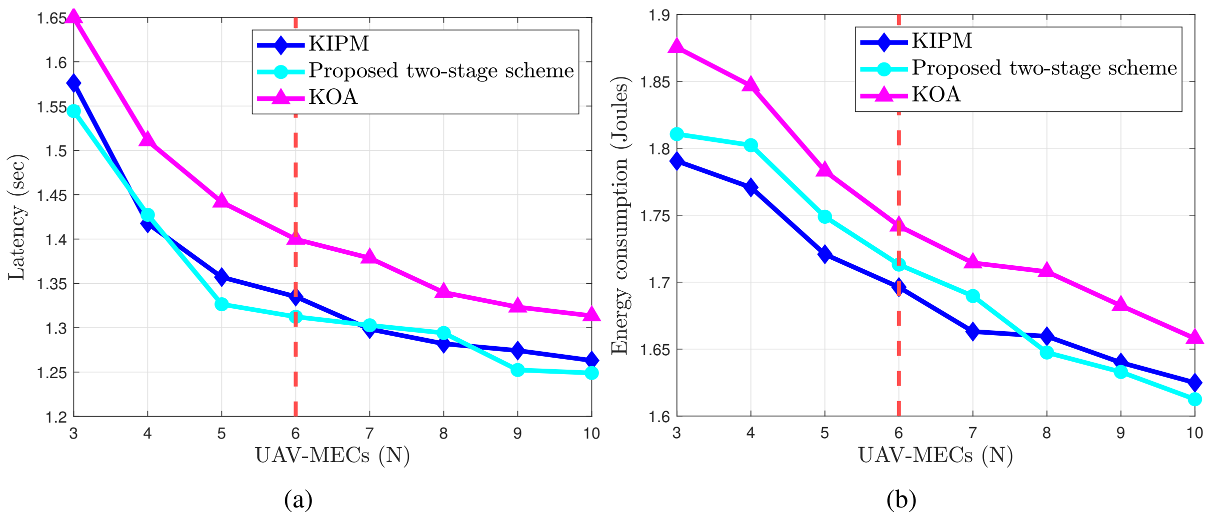
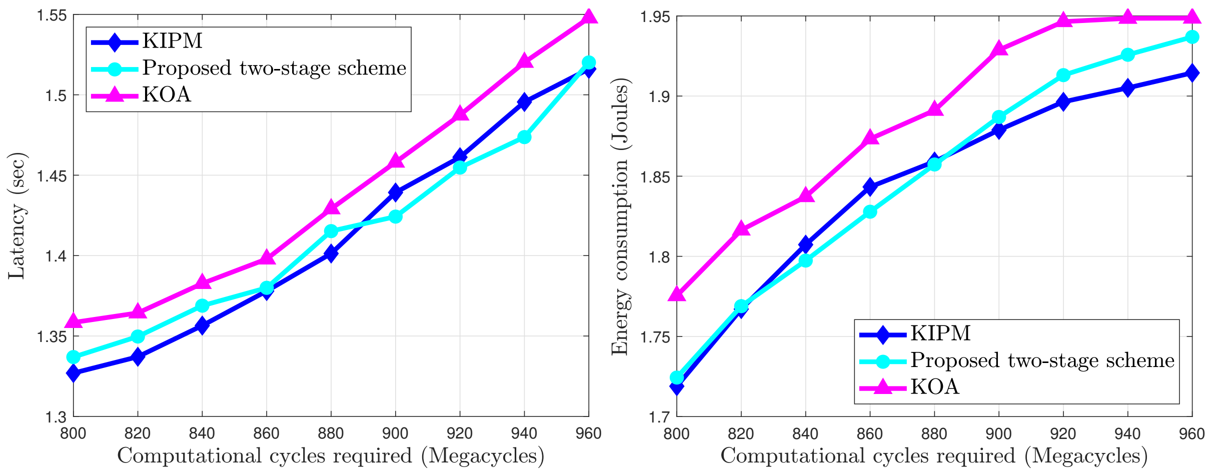


Figure 4.4: Performance evaluation in terms of the impact of the number of UAV-MECs when IoT devices (U = 80), computational cycles requirement (C_u = 800 Megacycles), and task size (D_u= 100 KBytes on (a) task latency and (b) energy consumption of IoT devices.

of the proposed scheme, KIPM, and KOA are plotted in Figs. 4.5(a) and 4.5(b), respectively. As the computational cycle requirement increases, tasks require more processing time and energy, increasing task latency and energy consumption. The proposed two-stage scheme performs better than the KOA but is nearly as close to KIPM, with less complexity. These curves show that tasks' computational complexity directly impacts task latency and energy consumption.

Fig. 4.6 shows the task latency and energy consumption of IoT devices versus digital twin approximation error, ranging from 0-12%, with IoT devices (U = 80), UAV-MECs (N = 5), task size (D_u = 100 KBytes), and computational cycles (C_u $=$ 800 Megacycles). The task latency results of IoT devices from the proposed two-stage scheme, KOA, and KIPM presented in Fig. 4.6(a), which increase with the increment of error in the approximation of the digital twin as per (3.7). Thus, the digital twin approximation error should be limited to reduce the latency of IoT devices' tasks. Fig. 4.6(b) shows the impact of digital twin approximation error on energy consumption. The energy consumption follows an increasing trend as the approximation error increases. We presented the result of the proposed two-stage scheme with the results of KOA and KIPM. The proposed scheme gave close results to KIPM with low computational complexity. The results in Fig. 4.6 demonstrate that the digital twin error directly affects the performance of our



(a) (b)

Figure 4.5: Performance evaluation in terms of the impact of computation cycles required when IoT devices (U = 80), UAV-MECs (N = 5), and task size (D_u= 100 KBytes on (a) task latency and (b) energy consumption of IoT devices.

proposed network. Therefore, it is crucial to limit the error to an acceptable bound and consistently update the digital twin from the physical world within a specific time interval.

The empirical weight factor ω in (3.28) plays a crucial role in the objective function as it determines the extent to which latency and energy consumption should be minimized. A higher value of ω will prioritize minimizing task latency, while a lower value will focus on reducing energy consumption. Fig. 4.7 shows IoT devices' task latency and energy consumption on five different weight scenarios. We consider IoT devices (U = 80), UAV-MECs (N = 5), computational cycles requirement (C_u = 800 Megacycles), and task size (D_u = 100 KBytes). For comparison, we have plotted the results for the proposed two-stage scheme, KOA and KIPM. Fig. 4.7(a) shows that the higher empirical weight reduces the latency, and this trend follows in all three schemes. The lowest latency values are for the weight values of 1 (ω = 1); there is no significant reduction from 0.5 to 1 values of ω. Fig. 4.7(b) shows the energy consumption of IoT devices for the proposed two-stage, KIPM, and KOA schemes versus different scenarios of weights. We observed that energy consumption increases with higher weight values. The minimum energy consumption is with the $\omega = 0$. The scenarios show that the weight ω from 0.5 to 1 has no significant impact on energy reduction. Moreover, the results show that the proposed two-stage scheme gives results

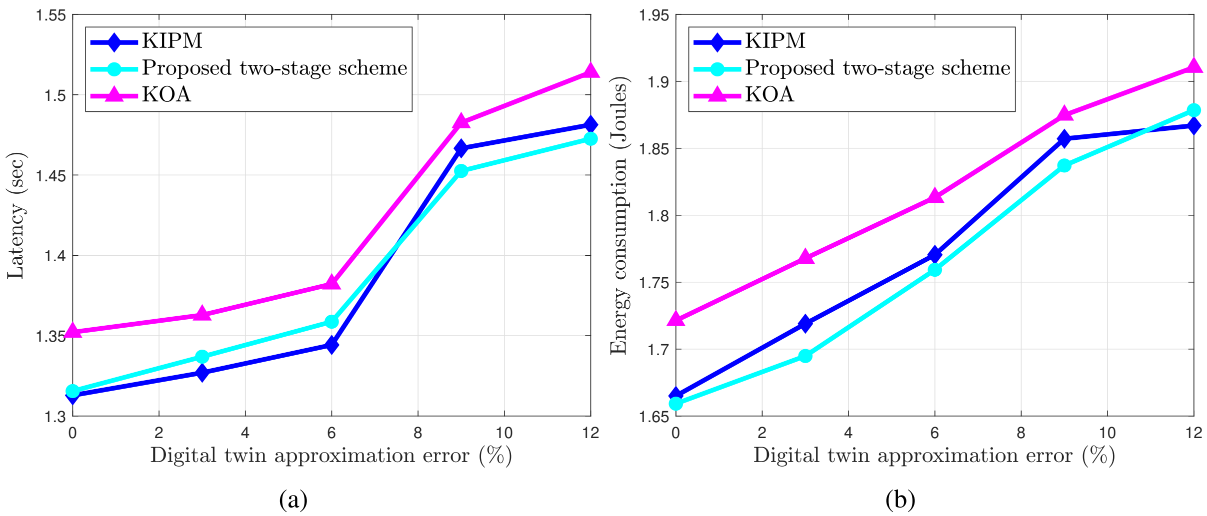


Figure 4.6: Performance evaluation in terms of the impact of digital twin approximation error when IoT devices (U = 80), UAV-MECs (N = 5), computational cycles requirement (C_u = 800 Megacycles), and task size (D_u= 100 KBytes on (a) task latency and (b) energy consumption of IoT devices.

nearly identical to KIPM results but with lower computational complexity.

The task partitioning factors determine the percentage of tasks that are offloaded to other layers of the network. By optimizing these factors, we can achieve the minimum possible latency. To illustrate the usefulness of optimizing these factors, we consider four scenarios with varying α & β values, IoT devices (U = 80), UAV-MECs (N = 5), computational cycles requirement (C_u = 800 Megacycles), and task size (D_u = 100 KBytes). We compare the results of the proposed two-stage scheme with the KIPM and KOA schemes. Fig. 4.8(a) shows task latency for different scenarios of α & β, and the minimum latency is observed when optimized values of α & β. The remaining scenarios with maximum offloading to 75% result in minimum latency. However, task latency is still higher than the optimized scenarios. This emphasizes the importance of finding the optimal distribution of tasks in the network layers rather than a fixed portion of offloading. Fig. 4.8(b) presents the energy consumption for the different scenarios of α & β, and the minimum energy consumption can be observed in the case of optimized α & β. The trend of decreasing energy consumption is observed with increasing the offloading portioning of tasks from IoT devices. However, the energy consumption is still higher than the optimized scenarios. Therefore, optimal

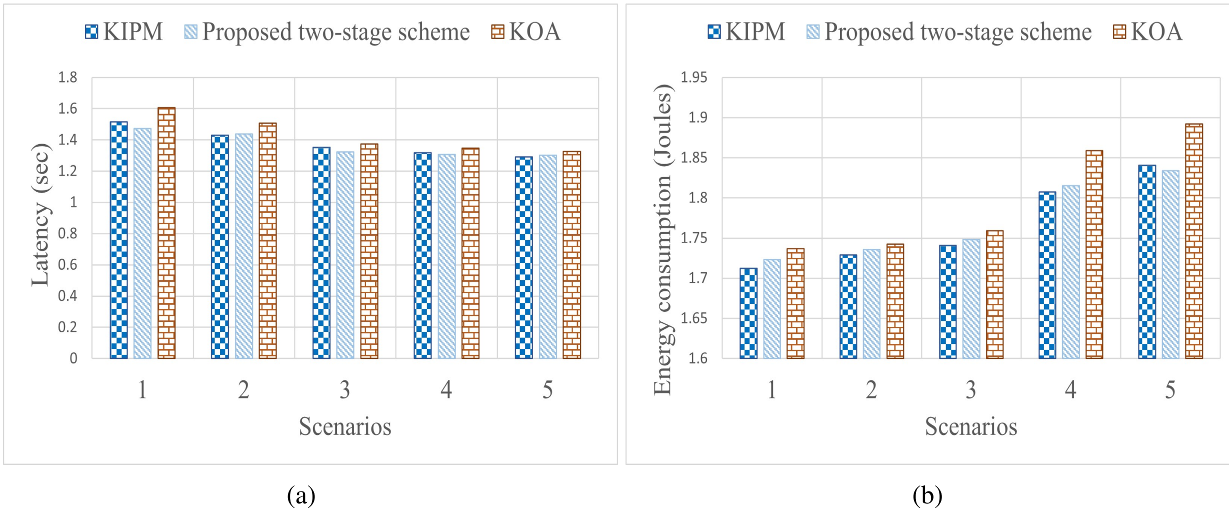
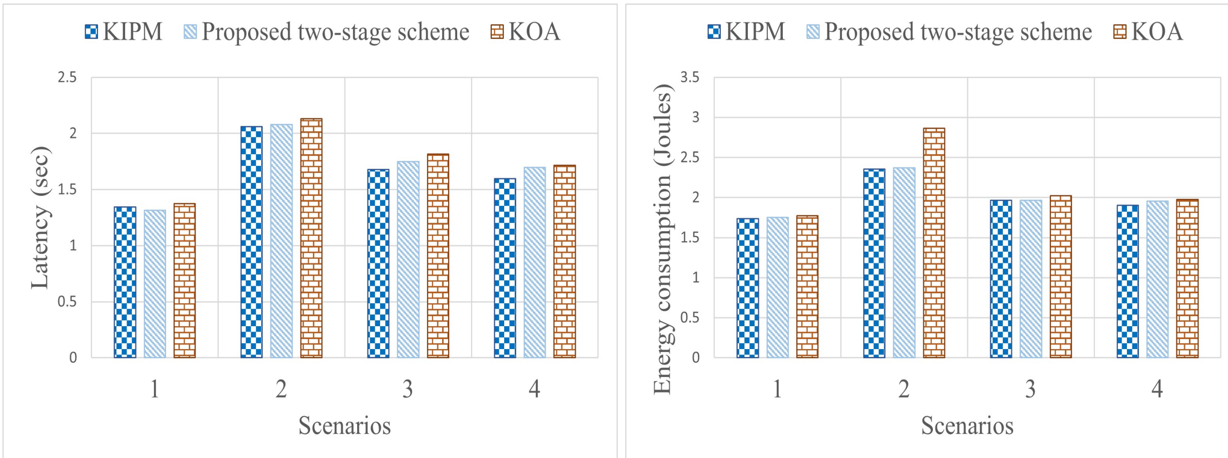


Figure 4.7: Performance evaluation in terms weight ω when IoT devices (U = 80) and UAV-MECs (N = 5) on (a) scenario 1: $\{w = 0\}$, (b) scenario 2: $\{w = 0.25\}$, (c) scenario 3: $\{w = 0.5\}$, (d) scenario 4: $\{w = 0.75\}$, and (e) scenario 5: $\{w = 1\}$.

task distribution is necessary for better energy and latency performance in multi-layer networks. Fig. 4.8 shows that the proposed two-stage scheme results are closer to both existing schemes with low computational complexity.

4.3 Summary

In this chapter, we discussed the proposed two-stage solution, followed by a discussion of simulation results. The proposed solution involves a two-stage scheme to solve the optimization problem. In the first stage, the K-means method is applied to cluster the IoT devices present, and the center of clusters is selected for the UAV-MECs location. After UAV-MECs placement, the association was made with the IoT devices. In the second stage, the communication and computation resources are optimized with the deep neural network to achieve low latency and energy efficiency in a multi-layer network. Two existing schemes are utilized to solve the optimization problem and compare the results with the proposed two-stage scheme. The energy and latency increase with the network's increased number of IoT devices but decrease with the increase of UAV-MECs. Optimized offloading partitioning of the multi-layer network reduces task latency and enhances energy



(a) (b)

Figure 4.8: Performance evaluation in terms of task partitioning factors (α & β) when IoT devices (U = 80) and UAV-MECs (N = 5) on (a) scenario 1: α & β optimized, (b) scenario 2: α & β fixed to 0.25, (c) scenario 3: α & β fixed to 0.5, and (d) scenario 4: α & β fixed to 0.75.

efficiency. The digital twin-assisted multi-layer network is useful for reducing task latency and energy consumption by IoT devices.

Chapter 5

Conclusion and Future Work

5.1 Conclusions

In this thesis, we formulated the optimization problem to minimize task latency and energy consumption of IoT devices in a digital twin-assisted multi-layer network. The optimization problem considered the communication and computation resources that need to be optimized to achieve an energy-efficient network with minimum task latency. The minimum latency and energy-efficient multi-layer network is achieved by optimizing the association, offloading portioning, transmit powers allocation, and processing rates of users. To achieve this, we proposed a two-stage scheme based on the K-means and deep learning architecture to solve the optimization problem. We compared the performance of the proposed two-stage scheme with two existing schemes. Compared to existing schemes, the proposed two-stage scheme proved computationally efficient for optimizing the resources of digital twin-assisted multi-layer networks. We presented simulation results to demonstrate the usefulness of the proposed multi-layer network and the impacts of different factors. For future research, we plan to explore the convexification of the optimization problem for larger-sized multi-layer networks.

5.2 Future Research Directions

Based on the presented framework for low latency and energy-efficient multi-layer networks, several open research issues and future research directions exist.

- **Dynamic digital twin network:** In this thesis, we chose the static digital twin for our multi-layer network due to the limitations of our environment and resources. Nonetheless, we can increase adaptability and make the network more resilient by utilizing dynamic digital twin network technology.

- **Convexification of problem:** In this thesis, we opted for the two sub-optimal solutions due to non-linear and non-convex optimization problems. For a very low-complexity optimal solution, we can convexify the optimization problem and solve it through convex optimization.

- **Data handling:** The concept of a digital twin involves creating a virtual replica of a real network's infrastructure. However, handling the large amount of data required for large-scale networks presents a significant challenge, especially regarding data analytics. We need to gain insights into the data stored in the digital twin to make accurate predictions. This is where AI algorithms can prove helpful, as they can help to identify patterns and trends in the data, which can then be used to make better predictions and inform network updates. However, the storage of large datasets remains an issue that must be addressed. To tackle this problem, a dynamic resource allocation procedure can be introduced to manage the storage and update the virtual structure of the digital twin instances.

- **Coordination of UAVs network:** The thesis assumes that placing UAV-MECs in a collision-free environment is feasible. However, for multipurpose UAVs with high mobility and the ability to access remote locations, managing a network of UAVs can be complex and restrict their usage. To address this problem, a decision-making framework based on machine learning can be developed to manage the entire UAV-MECs cycle and make real-time decisions while considering constraints related to storage and computation resources.

Bibliography

[1] W. Sun, H. Zhang, R. Wang, and Y. Zhang, “Reducing offloading latency for digital twin edge networks in 6G,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12240–12251, Oct. 2020.

[2] A. Agrawal, V. Singh, and M. Fischer, “A new perspective on digital twins: Imparting intelligence and agency to entities,” *IEEE Journal of Radio Frequency Identification*, vol. 6, pp. 871–875, Nov. 2022.

[3] N. Apostolakis, L. E. Chatzieleftheriou, D. Bega, M. Gramaglia, and A. Banchs, “Digital twins for next-generation mobile networks: Applications and solutions,” *IEEE Communications Magazine*, vol. 61, no. 11, pp. 80–86, May 2023.

[4] X. Lin, L. Kundu, C. Dick, E. Obiodu, T. Mostak, and M. Flaxman, “6g digital twin networks: From theory to practice,” *IEEE Communications Magazine*, vol. 61, no. 11, pp. 72–78, Jun. 2023.

[5] Y. Li, D. V. Huynh, T. Do-Duy, E. Garcia-Palacios, and T. Q. Duong, “Unmanned aerial vehicle aided edge networks with ultra-reliable low latency communications:a digital twin approach,” *IET Signal Processing*, vol. 16, no. 8, pp. 897–908, Apr. 2022.

[6] Y. Lu, S. Maharjan, and Y. Zhang, “Adaptive edge association for wireless digital twin networks in 6G,” *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16219–16230, Nov. 2021.

[7] L. Lei, G. Shen, L. Zhang, and Z. Li, “Toward intelligent cooperation of UAV swarms: When machine learning meets digital twin,” *IEEE Network*, vol. 35, no. 1, pp. 386–392, Feb. 2021.

[8] G. Shen, L. Lei, Z. Li, S. Cai, L. Zhang, P. Cao, and X. Liu, “Deep reinforcement learning for flocking motion of multi-UAV systems: Learn from a digital twin,” *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11141–11153, Jul. 2022.

[9] T. N. Guo, “Robust Q-learning for fast and optimal flying base station placement aided by digital twin for emergency use,” pp. 1–6, Jan. 2023.

[10] B. Hazarika, K. Singh, C.-P. Li, A. Schmeink, and K. F. Tsang, “RADiT: Resource allocation in digital twin-driven UAV-aided internet of vehicle networks,” *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3369–3385, Aug. 2023.

[11] Z. Shah, U. Javed, M. Naeem, S. Zeadally, and W. Ejaz, "Mobile edge computing (MEC)-enabled UAV placement and computation efficiency maximization in disaster scenario," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 10, pp. 13406–13416, May 2023.

[12] R. Masroor, M. Naeem, A. Tallha, A. M. Almasoud, and W. Ejaz, "Optimal stratified placement of balloons and UAVs to support users' coverage," *Elsevier Internet of Things*, vol. 23, no. 100865, Oct. 2023.

[13] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, "Resource scheduling in edge computing: A survey," *IEEE Communications Surveys Tutorials*, vol. 23, no. 4, pp. 2131–2165, Aug. 2021.

[14] T. Q. Duong, D. V. Huynh, Y. Li, E. Garcia-Palacios, and K. Sun, "Digital twin-enabled 6G aerial edge computing with ultra-reliable and low-latency communications," *2022 1st International Conference on 6G Networking (6GNet)*, pp. 1–5, Jul. 2022.

[15] X. Tang, X. Li, R. Yu, Y. Wu, J. Ye, F. Tang, and Q. Chen, "Digital-twin-assisted task assignment in multi-UAV systems: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 15362–15375, Mar. 2023.

[16] B. Li, Y. Liu, L. Tan, H. Pan, and Y. Zhang, "Digital twin assisted task offloading for aerial edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 10863–10877, Oct. 2022.

[17] D. V. Huynh, V.-D. Nguyen, S. Chatzinotas, S. R. Khosravirad, H. V. Poor, and T. Q. Duong, "Joint communication and computation offloading for ultra-reliable and low-latency with multi-tier computing," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 2, pp. 521–537, Feb. 2023.

[18] T. Do-Duy, D. V. Huynh, O. A. Dobre, B. Canberk, and T. Q. Duong, "Digital twin-aided intelligent offloading with edge selection in mobile edge computing," *IEEE Wireless Communications Letters*, vol. 11, no. 4, pp. 806–810, Jan. 2022.

[19] T. T. Vu, D. N. Nguyen, D. T. Hoang, E. Dutkiewicz, and T. V. Nguyen, "Optimal energy efficiency with delay constraints for multi-layer cooperative fog computing networks," *IEEE Transactions on Communications*, vol. 69, no. 6, pp. 3911–3929, Jun. 2021.

[20] C.-F. Liu, M. Bennis, M. Debbah, and H. V. Poor, "Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4132–4150, Jun. 2019.

[21] T. Zhou, Y. Yue, D. Qin, X. Nie, X. Li, and C. Li, "Joint device association, resource allocation, and computation offloading in ultradense multidevice and multitask IoT networks," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18695–18709, Oct. 2022.

[22] Z. Lv, D. Chen, H. Feng, H. Zhu, and H. Lv, "Digital twins in unmanned aerial vehicles for rapid medical resource delivery in epidemics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25106–25114, Dec. 2022.

[23] H. Guo, X. Zhou, J. Wang, J. Liu, and A. Benslimane, "Intelligent task offloading and resource allocation in digital twin based aerial computing networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3095–3110, Aug. 2023.

[24] Q. Guo, F. Tang, and N. Kato, "Resource allocation for aerial assisted digital twin edge mobile network," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3070–3079, Aug. 2023.

[25] L. Chen, Q. Gu, K. Jiang, and L. Zhao, "A3c-based and dependency-aware computation offloading and service caching in digital twin edge networks," *IEEE Access*, vol. 11, pp. 57564–57573, Jun. 2023.

[26] J. Tan, F. Tang, M. Zhao, and Y. Zhu, "Adaptive caching scheme for jointly optimizing delay and energy consumption in heterogeneous digital twin iot," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 6, pp. 4020–4032, May 2023.

[27] B. Li, W. Liu, W. Xie, N. Zhang, and Y. Zhang, "Adaptive digital twin for uav-assisted integrated sensing, communication, and computation networks," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 4, pp. 1996–2009, Dec. 2023.

[28] T. Do-Duy, D. Van Huynh, O. A. Dobre, B. Canberk, and T. Q. Duong, "Digital twin-aided intelligent offloading with edge selection in mobile edge computing," *IEEE Wireless Communications Letters*, vol. 11, no. 4, pp. 806–810, Apr. 2022.

[29] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4968–4977, Jul. 2021.

[30] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5g services in mobile edge computing systems: Learn from a digital twin," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, Oct. 2019.

[31] D. V. Huynh, V.-D. Nguyen, S. R. Khosravirad, and T. Q. Duong, "Minimising offloading latency for edge-cloud systems with ultra-reliable and low-latency communications," *ICC 2022 - IEEE International Conference on Communications*, pp. 5122–5127, May 2022.

[32] D. V. Huynh, V. D. Nguyen, S. R. Khosravirad, V. Sharma, O. A. Dobre, H. Shin, and T. Q. Duong, "URLLC edge networks with joint optimal user association, task Offloading and resource allocation: A digital twin approach," *IEEE Transactions on Communications*, vol. 70, no. 11, pp. 7669–7682, Sept. 2022.

[33] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6g," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16219–16230, Nov. 2021.

[34] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6g," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12240–12251, Oct. 2020.

[35] Y. Dai, J. Wu, J. Zhao, B. Gong, and Y. Lu, "Intelligent reflecting surfaces aided task offloading in digital twin edge networks," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, pp. 1–5, Dec. 2023.

[36] X. Yuan, J. Chen, N. Zhang, J. Ni, F. R. Yu, and V. C. M. Leung, "Digital twin-driven vehicular task offloading and irs configuration in the internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24290–24304, Dec. 2022.

[37] N. A. Mitsiou, V. K. Papanikolaou, P. D. Diamantoulakis, T. Q. Duong, and G. K. Karagiannidis, "Digital twin-aided orchestration of mobile edge computing with grant-free access," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 841–853, Mar. 2023.

[38] Z. Yao, S. Xia, Y. Li, and G. Wu, "Cooperative task offloading and service caching for digital twin edge networks: A graph attention multi-agent reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3401–3413, Nov. 2023.

[39] X. Tang, X. Li, R. Yu, Y. Wu, J. Ye, F. Tang, and Q. Chen, "Digital-twin-assisted task assignment in multi-uav systems: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 10, no. 17, pp. 15362–15375, Sep. 2023.

[40] Y. Zhang, J. Hu, and G. Min, "Digital twin-driven intelligent task offloading for collaborative mobile edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3034–3045, Oct. 2023.

[41] D. Van Huynh, S. R. Khosravirad, A. Masaracchia, O. A. Dobre, and T. Q. Duong, "Edge intelligence-based ultra-reliable and low-latency communications for digital twin-enabled metaverse," *IEEE Wireless Communications Letters*, vol. 11, no. 8, pp. 1733–1737, Aug. 2022.

[42] B. Wang, Y. Sun, H. Jung, L. D. Nguyen, N.-S. Vo, and T. Q. Duong, "Digital twin-enabled computation offloading in uav-assisted mec emergency networks," *IEEE Wireless Communications Letters*, vol. 12, no. 9, pp. 1588–1592, Sep. 2023.

[43] D. V. Huynh, V.-D. Nguyen, V. Sharma, O. A. Dobre, and T. Q. Duong, "Digital twin empowered ultra-reliable and low-latency communications-based edge networks in industrial IoT environment," in *IEEE International Conference on Communications (IEEE ICC)*, pp. 5651–5656, Aug. 2022.

[44] D. V. Huynh, S. R. Khosravirad, A. Masaracchia, O. A. Dobre, and T. Q. Duong, "Edge intelligence-based ultra-reliable and low-latency communications for digital twin-enabled metaverse," *IEEE Wireless Communications Letters*, vol. 11, no. 8, pp. 1733–1737, Aug. 2022.

[45] C. Xu, Z. Tang, H. Yu, P. Zeng, and L. Kong, "Digital twin-driven collaborative scheduling for heterogeneous task and edge-end resource via multi-agent deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3056–3069, Aug. 2023.

[46] Z. Yang, M. Chen, Y. Liu, and Z. Zhang, "Optimizing synchronization delay for digital twin over wireless networks," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 9106–9110, Mar. 2024.

[47] Y. Gong, Y. Wei, Z. Feng, F. R. Yu, and Y. Zhang, "Resource allocation for integrated sensing and communication in digital twin enabled internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 4, pp. 4510–4524, Apr. 2023.

[48] Y. Zhang, H. Zhang, Y. Lu, W. Sun, L. Wei, Y. Zhang, and B. Wang, "Adaptive digital twin placement and transfer in wireless computing power network," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 10924–10936, Mar. 2024.

[49] L. Li, X. Xu, B. Bian, H. Yang, W. Li, Y. Zhao, H. Zhang, and J.-B. Wang, "Joint optimization of beamforming and transmission power based on digital twins control system," in *2022 IEEE Smartworld, Ubiquitous Intelligence Computing, Scalable Computing Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*, pp. 2231–2237, Jul. 2023.

[50] C. Dai, K. Yang, and C. Deng, "A service placement algorithm based on merkle tree in mec systems assisted by digital twin networks," in *2022 IEEE 21st International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*, pp. 37–43, Apr. 2023.

[51] J. Feng, L. Liu, X. Hou, Q. Pei, and C. Wu, "Qoe fairness resource allocation in digital twin-enabled wireless virtual reality systems," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3355–3368, Nov. 2023.

[52] B. Han, M. A. Habibi, B. Richerzhagen, K. Schindhelm, F. Zeiger, F. Lamberti, F. G. Prattico, K. Upadhya, C. Korovesis, I.-P. Belikaidis, P. Demestichas, S. Yuan, and H. D. Schotten, "Digital twins for industry 4.0 in the 6g era," *IEEE Open Journal of Vehicular Technology*, vol. 4, pp. 820–835, Oct. 2023.

[53] F. Luan, J. Yang, H. Zhang, Z. Zhao, and L. Yuan, "Optimization of load-balancing strategy by self-powered sensor and digital twins in software-defined networks," *IEEE Sensors Journal*, vol. 23, no. 18, pp. 20782–20793, Sept. 2023.

[54] W. Xiang, J. Li, Y. Zhou, P. Cheng, J. Jin, and K. Yu, "Digital twin empowered industrial iot based on credibility-weighted swarm learning," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 1, pp. 775–784, Jan. 2024.

[55] M. Kherbache, M. Maimour, and E. Rondeau, "Network digital twin for the industrial internet of things," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 573–578, Aug. 2022.

[56] W. Xie, F. Qi, L. Liu, and Q. Liu, "Radar imaging based uav digital twin for wireless channel modeling in mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3702–3710, Aug. 2023.

[57] M. Pengnoo, M. T. Barros, L. Wuttisittikulkij, B. Butler, A. Davy, and S. Balasubramaniam, "Digital twin for metasurface reflector management in 6g terahertz communications," *IEEE Access*, vol. 8, pp. 114580–114596, Jun. 2020.

[58] A. Lombardo, G. Morabito, S. Quattropani, and C. Ricci, "Design, implementation, and testing of a microservices-based digital twins framework for network management and control," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 590–595, Aug. 2022.

[59] T. Shui, J. Hu, K. Yang, H. Kang, H. Rui, and B. Wang, "Cell-free networking for integrated data and energy transfer: Digital twin based double parameterized dqn for energy sustainability," *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 8035–8049, Nov. 2023.

[60] S. Ono, T. Yamazaki, T. Miyoshi, A. Taya, Y. Nishiyama, and K. Sezaki, "Amond: Area-controlled mobile ad-hoc networking with digital twin," *IEEE Access*, vol. 11, pp. 85224–85236, Aug. 2023.

[61] C. Shi, F. Du, and Q. Wu, "A digital twin architecture for wireless networked adaptive active noise control," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, Aug. 2022.

[62] M. Yaqoob, R. Trestian, and H. X. Nguyen, "Leveraging digital twin approach for network slicing in b5g network," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 242–247, Oct. 2023.

[63] P. Yu, J. Zhang, H. Fang, W. Li, L. Feng, F. Zhou, P. Xiao, and S. Guo, "Digital twin driven service self-healing with graph neural networks in 6g edge networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 11, pp. 3607–3623, Nov. 2023.

[64] S. Geibler, F. Wamser, W. Bauer, S. Gebert, S. Kounev, and T. Hobfeld, "Mvnocoresim: A digital twin for virtualized iot-centric mobile core networks," *IEEE Internet of Things Journal*, vol. 10, no. 15, pp. 13974–13987, Aug. 2023.

[65] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. S. Rossi, "Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827–4838, Feb. 2021.

[66] M. N. Hasan, S. U. Jan, and I. Koo, "Wasserstein gan-based digital twin-inspired model for early drift fault detection in wireless sensor networks," *IEEE Sensors Journal*, vol. 23, no. 12, pp. 13327–13339, May 2023.

[67] H. Darvishi, D. Ciuonzo, and P. S. Rossi, "A machine-learning architecture for sensor fault detection, isolation, and accommodation in digital twins," *IEEE Sensors Journal*, vol. 23, no. 3, pp. 2522–2538, Feb. 2023.

[68] Z. Lai, D. Yuan, H. Chen, Y. Zhang, and W. Bao, "Wirelessdt: A digital twin platform for real-time evaluation of wireless software applications," in *2023 IEEE/ACM 45th International*

Conference on Software Engineering: Companion Proceedings (ICSE-Companion), pp. 146–150, Jul. 2023.

[69] N. U. Sama, K. Zen, A. Ud Din, N. Azim, and A. Ur Rahman, "Security of cloud-assisted bans using digital twin," in *2023 13th International Conference on Information Technology in Asia (CITA)*, pp. 37–42, Sep. 2023.

[70] X. Wang, Y. Gao, L. Deng, and M. Chen, "Dtcpn: A digital twin cyber platform based on nfv," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 579–583, Aug. 2022.

[71] Y. Wu, K. Zhang, and Y. Zhang, "Digital twin networks: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13789–13804, Sep. 2021.

[72] B. Vilas Boas, W. Zirwas, and M. Haardt, "Machine learning for csi recreation in the digital twin based on prior knowledge," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1578–1591, Sep. 2022.

[73] Y. Cui, T. Lv, W. Ni, and A. Jamalipour, "Digital twin-aided learning for managing reconfigurable intelligent surface-assisted, uplink, user-centric cell-free systems," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3175–3190, Oct. 2023.

[74] Y.-H. Xu, L. Suo, W. Zhou, and G. Yu, "A graph-learning-inspired resource optimization for digital-twin-empowered wireless body area networks," *IEEE Sensors Letters*, vol. 7, no. 11, pp. 1–4, Nov. 2023.

[75] H. Zhang, X. Ma, X. Liu, L. Li, and K. Sun, "Gnn-based power allocation and user association in digital twin network for the terahertz band," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3111–3121, Oct. 2023.

[76] L. P. Qian, M. Li, P. Ye, Q. Wang, B. Lin, Y. Wu, and X. Yang, "Secrecy-driven energy minimization in federated-learning-assisted marine digital twin networks," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 5155–5168, Feb. 2024.

[77] S. Lian, H. Zhang, W. Sun, and Y. Zhang, "Lightweight digital twin and federated learning with distributed incentive in air-ground 6g networks," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1–5, Aug. 2022.

[78] J. Deng, L. Yue, H. Yang, and G. Liu, "A digital twin network approach for 6g wireless network autonomy," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 415–420, Oct. 2023.

[79] C. Ruah, O. Simeone, and B. M. Al-Hashimi, "A bayesian framework for digital twin-based control, monitoring, and data collection in wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3146–3160, Oct. 2023.

[80] J. Mu, W. Ouyang, T. Hong, W. Yuan, Y. Cui, and Z. Jing, "Digital twins-enabled federated learning in mobile networks: From the perspective of communication-assisted sensing," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, pp. 3230–3241, Oct. 2023.

[81] X. He, Q. Chen, L. Tang, W. Wang, T. Liu, L. Li, Q. Liu, and J. Luo, "Federated continuous learning based on stacked broad learning system assisted by digital twin networks: An incremental learning approach for intrusion detection in uav networks," *IEEE Internet of Things Journal*, vol. 10, no. 22, pp. 19825–19838, Nov. 2023.

[82] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6g networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5098–5107, Jul. 2021.

[83] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 729–743, Feb. 2020.

[84] M. K. Pakhira, "A linear time-complexity k-means algorithm using cluster shifting," in *2014 International Conference on Computational Intelligence and Communication Networks*, pp. 1047–1051, Mar. 2015.

[85] X. Gao and L.-Z. Liao, "A new one-layer neural network for linear and quadratic programming," *IEEE Transactions on Neural Networks*, vol. 21, no. 6, pp. 918–929, Apr. 2010.

[86] I. Pólik and T. Terlaky, *"Interior point methods for nonlinear optimization"*, pp. 215–276. Berlin, Heidelberg: Springer Berlin Heidelberg, Jan. 2010.

[87] M. R. Ramzan, M. Naeem, O. Chugtai, W. Ejaz, and M. Altaf, "Radio resource management in energy harvesting cooperative cognitive uav assisted iot networks: A multi objective approach," *Digital Communications and Networks*, Jan. 2023.