Received 8 March 2024; accepted 22 March 2024. Date of publication 28 March 2024; date of current version 12 April 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3382265

Artificial Intelligence-Defined Wireless Networking for Computational Offloading and Resource Allocation in Edge Computing Networks

SYED DANIAL ALI SHAH[1], MARK A. GREGORY[2] (Senior Member, IEEE), FAYCAL BOUHAFS[1] (Senior Member, IEEE), AND FRANK DEN HARTOG[1,3] (Senior Member, IEEE)

[1]School of Systems and Computing, The University of New South Wales, Canberra, ACT 2610, Australia

[2]School of Engineering, Royal Melbourne Institute of Technology, Melbourne, VIC 3001, Australia

[3]Network Engineering and Cyber Security, University of Canberra, Canberra, ACT 2617, Australia

CORRESPONDING AUTHOR: F. BOUHAFS (e-mail: F.Bouhafs@unsw.edu.au)

This work was supported by the School of Systems and Computing, University of New South Wales, Canberra, Australia.

ABSTRACT The advent of the Internet of Everything and new Ultra-Reliable Low-Latency Communication (URLLC) services has resulted in an exponential growth in data demands at the network's edge. To meet the stringent performance requirements of evolving 5G (and beyond) applications, deploying dedicated resources closer to mobile users is essential. Multi-Access Edge Computing (MEC) is a promising technology for bringing computational resources closer to users. However, the distributed and limited MEC resources must be effectively optimized to maximize the number of mobile users benefiting from low-latency MEC services at each time slot in highly congested, large-scale, and dynamic wireless network scenarios. In this research, we propose and evaluate a novel Artificial Intelligence-Defined Wireless Networking (AIDWN) approach that builds on conventional Software-Defined Networking (SDN), implementing a new AI-defined application plane for computational offloading and resource allocation in MEC-enabled wireless networks. The AIDWN approach implements a deep reinforcement learning framework and deep neural networks that dynamically adapt optimal computational offloading and wireless resource allocation decisions while considering the handover, mobility, and coordinated resource allocation challenges in highly dynamic and mobile multi-MEC server environments. Compared to recent state-of-the-art proposals, the proposed AIDWN demonstrates a substantial performance improvement, utilizing more than 90% of MEC resources per time slot across all MEC servers. It also accommodates significantly more mobile users in highly congested wireless network scenarios. We identified various future research directions highlighting the potential of the AIDWN approach in simplifying the management of next-generation wireless networks.

INDEX TERMS Edge computing, resource management, AI, wireless networks, SDN, task offloading, 5G and beyond.

I. INTRODUCTION

THE GROWTH in data demands for Ultra-Reliable and Low-Latency Communication (URLLC) services, e.g., Vehicle-to-Everything (V2X) and virtual reality/augmented (VR/AR) reality, has led to a significant need for the deployment of dedicated network resources closer to the users [1], [2]. Multi-access Edge Computing (MEC) is a promising technology standardized to bring the computational resources closer to the users [3]. Computational offloading is a method that allows power-constrained mobile users to offload their low-latency and computation-intensive tasks to nearby MEC servers [4]. However, owing to the limited computational capacities of the MEC servers, the distributed and limited MEC resources must be effectively managed to maximize the number of mobile users offloading the low-latency computational tasks to the MEC servers at each time slot in highly congested, large-scale, and dynamic wireless network scenarios.

Computational offloading is a challenging problem, where the users frequently relocate, resulting in constant variations in communication links, channel quality, and signal strength [5]. This poses a major challenge in wireless networks as it becomes essential to optimally and dynamically adapt computational offloading and resource allocation decisions to the time-varying wireless channel conditions and resources available on the MEC servers in real-time [6]. Therefore, designing effective and optimal methods and approaches for computational offloading and resource allocation in dynamic MEC-enabled next-generation wireless networks demands the development of procedures to quickly solve complex combinatorial optimization problems within the channel coherence time. This is hard to achieve with conventional numerical optimization methods [7].

The computational offloading problem could be modeled as a Mixed-Integer Nonlinear Programming (MINLP) Problem [8], which is a complex optimization problem involving continuous and discrete variables. The continuous variables include resource allocation parameters such as CPU frequency, transmit power, and bandwidth, while the discrete variables include the computational offloading decisions. The MINLP problem becomes challenging to solve because of the exponential growth of the action space with the problem size, i.e., the number of mobile users in the network [9]. A potential solution for the computational offloading and resource allocation problem in dynamic and mobile scenarios such as MEC-enabled vehicular networks is the use of AI-based approaches, e.g., deep reinforcement learning approach, where optimal policy is learned through training that maximizes the overall MEC system utility and reduces the computational complexity of optimization problems.

The existing AI-based approaches for computational offloading and resource allocation only focus on optimizing the MEC system utility of individual systems, i.e., computational rate, processing delay, latency, and energy consumption [10], [11], [12], [13], and do not take into account the challenges pertaining to multi-MEC server environments, e.g., handover and mobility management, adaptive and coordinated resource allocation, and load balancing. This is insufficient for highly dynamic and mobile wireless network scenarios where mobile users frequently relocate, resulting in frequent network topology changes.

In this paper, the complexity arising from using AI on a distributed system as edge computing is addressed by leveraging the flexibility and centralized programmability offered by Software-Defined Networking (SDN). We propose an Artificial Intelligence-Defined Wireless Networking (AIDWN) approach that employs an AI-defined application plane that dynamically adapts to the time-varying wireless channel conditions and resources available on multiple MEC servers, allowing for more efficient and effective resource management and coordination. The AIDWN approach uses artificial intelligence and machine learning techniques in the AI-defined application plane, integrating behavioral models and reasoning processes tailored to automate decision-making and network configurations in SDN-based wireless networks. The SDN controller utilizes the global network topology information, i.e., network analytics, to make informed and intelligent control decisions, and learn optimal applications and use case-specific control policies.

The remainder of this paper is organized as follows. The following Section II reviews the related works, research gaps, and motivations and contributions are described. The challenges and design considerations for the proposed AIDWN approach are provided in Section III. Section IV introduces the proposed AIDWN architecture and its application in the MEC-enabled next-generation wireless networks domain. The novel AI-defined application plane is presented in Section V for efficient computational offloading and resource allocation in MEC. The final two sections discuss our numerical results, conclusions, and potential use cases.

II. STATE OF THE ART

A. RELATED WORKS

Several related works implement AI-based approaches for computational offloading and resource allocation with different optimization objectives. The authors in [14] modeled the computational offloading policy as a Markov decision process to maximize the long-term MEC system utility performance based on the task and energy queue states and the channel conditions of the wireless users. The authors in [15], [16] proposed a value iteration-based reinforcement learning approach and deep reinforcement learning approach for joint optimization of computational offloading and resource allocation in a multi-user MEC system environment to optimize the energy consumption of the MEC system, respectively. In [17], authors proposed a deep reinforcement learning-based joint secure offloading and resource allocation approach that applies physical layer security and spectrum sharing techniques to improve vehicular edge computing networks' secrecy performance and resource efficiency. The authors aim to minimize the overall MEC system processing delay and secure the offloading process through physical layer security techniques.

The correlation between the service caching and computation offloading strategies for MEC has been discussed in [18]. The authors proposed a deep reinforcement learning approach to jointly optimize the service caching and computational offloading allocations in a vehicular edge computing scenario with time-varying task requests for minimal average task processing delay. In [19], the authors proposed a deep reinforcement learning approach to maximize the number of computational tasks offloaded to the MEC server with limited computational capacity. In [20], authors proposed an asynchronous deep reinforcement algorithm to jointly optimize the problem of distributed task offloading and multi-resource management. The authors investigated a collaborative computing framework where vehicle mobile users collaborate their resources to maximize the system's utility. In [21], authors proposed a deep reinforcement learning algorithm to optimize the utility of a resource-constrained MEC server by maximizing the distribution of tasks between the MEC server and other computing groups in the network to minimize the overall task processing latency in the system.

The authors in [22] proposed a multi-agent soft-actor-critic-discrete based URLLC-constrained task offloading and resource allocation to optimize throughput and reduce power consumption at the remote end, taking into account extended URLLC constraints. The approach offers superior performance in delay and other characteristics related to URLLC. In [23], authors proposed a double deep Q-network-based algorithm and a dynamic offloading model to optimize offloading decisions for compute-intensive tasks in a congested scenario, to minimize the total delay and wait times of mobile users, e.g., vehicles. A deep meta-reinforcement learning-based offloading algorithm is proposed in [24] to tackle the complexity of the optimal offloading-decision-making optimization problem that leverages multiple parallel DNNs and Q-learning to enable precise offloading decisions within dynamic environments. In [25], authors proposed a deep deterministic policy gradient-based mobility-aware computation offloading and task migration approach based on trajectory and resource prediction to minimize the task turnaround time and system energy consumption and optimize the offloading decisions in mobility scenarios. The authors in [26] proposed a cost-efficient computation offloading framework that utilizes a two-stage optimization algorithm balancing cost and latency in edge computing environments. A meta-reinforcement learning approach for computational task offloading and power control among users in a resource-limited MEC network is proposed in [27] that aims to enhance user computation efficiency by minimizing power consumption during local computing and uplink transmission within the MEC network environment. In [28], a deep deterministic policy gradient algorithm based on a greedy strategy is proposed that jointly optimizes scheduling, device association, and task allocation of unmanned aerial vehicles, aiming to minimize the weighted sum of total system energy consumption and time delay.

The past decades saw a shift in the way data traffic and spectrum resources are managed in telecommunication and data networks. The introduction of novel softwarization techniques such as SDN [29] and Network Function Virtualisation (NFV) [30] has enabled more scalable and efficient management of these networks. Although initially limited to wired networks, such techniques are now also used for wireless communications. The rise of Software Defined Wireless Networking (SDWN) [31] represents an extension of SDN specifically in this context. One application domain of SDWN could be 5G and MEC, as suggested by various researchers (see, e.g., [32] and [33]), but this is not part of the current 5G standard and will more likely be integrated with 6G [32].

Some recent works have proposed SDN-based edge computing networks adopting AI and machine learning techniques for optimal computational offloading and resource allocation decisions. In [34], authors proposed an SDN-based multi-agent system to optimize computational offloading in MEC networks, where the SDN controller provides the global network view to allocate computational resources to network devices optimally. The authors proposed a stochastic game model that allows each user to maximize its benefits, e.g., system delay and energy consumption, by considering the constraints of the MEC server in the multi-user environment. Knowledge-defined edge computing networks are proposed in [35] to optimize computational offloading and resource allocation strategy. The architecture allows the collection of network information, self-learning, and making decisions or recommendations for large-scale network systems to jointly optimize the computation offloading and resource allocation decisions and maximize the long-term utility of the system.

In summary, the existing literature does not take into account the challenges arising from the multi-MEC server and highly varying wireless network environment, e.g., handover and mobility management, resource coordination between different MEC servers, and load balancing. Instead, our proposed AIDWN approach is based on SDN-based intelligent architecture that learns an optimal policy maximizing the overall MEC system utility, i.e., the number of mobile users benefiting from low-latency MEC services at each time slot in a highly congested and dynamic wireless network environment. The AIDWN approach then utilizes the advantages of a centralized SDN control plane for effectively managing the mobility, handover, and resource coordination between the MEC servers by performing dynamic flow updates, user associations, and traffic routing between the users and the MEC servers, and allocating network resources. The SDN control plane dynamically installs and updates flow in the data plane, allocating the optimal computational offloading, mobility, and network resource allocation decisions as received from the AI-defined application plane. The controller also operates the Extended Forwarding Modules (EFM) as presented in our previous research [33] to collect the wireless network topology, i.e., wireless network association, channel conditions, and resources availability on MEC servers, and feeds the information to the AI-defined application plane, where the network information is used to train the deep neural networks and receive optimal control actions.

B. MOTIVATIONS AND CONTRIBUTIONS

1) MOTIVATIONS

The existing AI-based approaches for computational offloading and resource allocation only focus on optimizing the MEC system utility, i.e., computational rate, processing delay, latency, and energy consumption [10], [11], [12], [13], and do not take into account the challenges pertaining to multi-MEC server environments, e.g., handover and mobility management, adaptive and coordinated resource allocation, and load balancing. In a highly dynamic wireless network environment with frequent topology changes, the computational offloading decisions received at each time slot may change for each user depending upon the various parameters, e.g., channel conditions, load on MEC servers, and the number of tasks to be processed by each user. In addition, in the case of a multi-MEC server environment, the computational offloading requests from the mobile users may be distributed at different MEC servers at each time slot. Therefore, developing effective handover and mobility management techniques for computational offloading and resource allocation in a multi-MEC server environment is essential.

However, the existing deep reinforcement learning approaches do not implement the computational offloading decisions in real time. I.e., the current approaches only achieve optimized computational offloading actions at each time slot without translating those actions into the network environment by the dynamic management of traffic flows and traffic routing establishing low-latency communication paths between the mobile users and their respective MEC servers. In addition, in a resource-constrained multi-MEC server environment, actively monitoring the available resources on MEC servers and coordinating the resources to maximize the overall MEC system utility is essential to increase the number of mobile users benefiting from computational offloading at each time slot. This is particularly important in highly congested, large-scale, and dynamic wireless network scenarios, where the demand for MEC resources is always higher than the availability.

Within the context of the dynamic wireless and mobile environment pertaining to multi-MEC server scenarios, there remain unaddressed research challenges and gaps. They serve as a compelling motivation for introducing the novel concept of AIDWN, using intelligence supported by the SDN controller. The extant research challenges and gaps may be categorized as follows:

- *Handover and Mobility Management:* The presence of multiple MEC servers for computational task offloading within the system gives rise to a set of challenges concerning handover and mobility management. As a result, the mobility aspects of the system, encompassing dynamic handovers and connections between mobile users and MEC servers, necessitate careful consideration. The need for effective handover and mobility management schemes is particularly important in dynamic wireless network environments, where mobile users frequently relocate, leading to dynamic changes in network topology. In this context, the proposed concept of AI-defined wireless networking emerges as a potential solution to address this research challenge. It uses centralized intelligence and provides a global network topology view to enhance handover and mobility management capabilities.
- *Adaptive Resource Allocation:* In a highly dynamic wireless and mobile environment, the demand for MEC resources may vary significantly in each time slot, i.e., the overall number of computational tasks requested for computational offloading at each time slot will vary. Consequently, predicting the varying demands for MEC resources at each time slot and optimizing the number of tasks to be offloaded at each time slot becomes essential. Adopting an aggressive approach whereby computational tasks are extensively offloaded to MEC servers could result in the over-utilization of MEC resource capabilities. Such over-utilization may potentially violate the stringent low-latency QoS requirements expected by mobile users. In contrast, the conservative approach where fewer computational tasks are offloaded to the MEC server, may result in the under-utilization of scarce MEC resources. Therefore, striking a balance between the two cases, as mentioned above, is essential. The introduction of the proposed AIDWN paradigm, defining the AI-defined application plane endowed with centralized intelligence and a comprehensive global view of network topology, including the ability to monitor resource availability across multiple MEC servers consistently, holds significance in efficiently optimizing resource allocation while adhering to resource constraints in this dynamic context.
- *Load Balancing for multi-MEC Scenario:* In a multi-MEC scenario, the imperative of load management across MEC servers is underscored, necessitating the efficient, fair, and balanced allocation of computational tasks while considering the resource capacity of each MEC server. The concept of AIDWN is pivotal in this context. This paradigm enables the real-time collection of pertinent network data, encompassing insights into the workload and resource availability across various MEC servers, access points, and switches. Accordingly, the controller orchestrates AI-defined control actions, implementing a policy-driven approach to load balancing through the southbound interface. This entails the dynamic selection of optimal paths and load-aware routing strategies to ensure a fair and balanced distribution of computational tasks across different MEC servers.

2) CONTRIBUTION AND VISION

Motivated by the research questions and challenges identified above, the proposed AIDWN approach addresses the complexity arising from the highly dynamic and varying wireless network environment, e.g., multi-MEC server environment, by leveraging the flexibility and centralized programmability offered by SDN, and the intelligence, optimization, and automation provided by the AI and machine learning techniques in the AI-defined application plane. The main contributions of this article are summarized as follows:

- We investigated and introduced design considerations for the AIDWN paradigm, addressing the challenges within the dynamic wireless and mobile environment pertaining to multi-MEC server scenarios as identified through a state-of-the-art literature review.

- We proposed a novel AIDWN approach that builds on the conventional SDN architecture and implements an AI-defined application plane integrating application-specific AI and machine learning models to automate decision-making and network configurations in SDN-based wireless networks.
- We evaluated the application of the proposed AIDWN approach in MEC-enabled dynamic wireless networks to maximize the overall utility of MEC system resources while considering the handover, mobility, and resource coordination challenges in multi-MEC server environments. The AIDWN approach uses a deep reinforcement learning framework implementing deep neural networks that dynamically learn the optimal binary computational offloading and resource allocation decisions from past experience and collected network information through the SDN management plane, eliminating the need to solve complex combinatorial optimization problems at each time frame.
- In contrast to the existing AI-based approaches for computational offloading and resource allocation in MEC-enabled networks that focus on optimizing MEC system utility in terms of computational rate, processing delay, latency, and energy consumption, our research integrates AI with SDN to optimize the utility of MEC system resources in a multi-MEC server environment. The proposed approach maximizes the number of mobile users benefiting from the MEC resources in highly congested and demanding network scenarios at each time frame. Moreover, potential use cases and applications of the proposed AIDWN approach are discussed in terms of simplifying and automating the management of next-generation wireless networks.

III. CHALLENGES AND DESIGN CONSIDERATIONS FOR AI-DEFINED WIRELESS NETWORKING

Given the related works and research challenges, it is apparent that the conventional machine learning approaches, e.g., deep reinforcement learning, are not enough to deal with the complex computational offloading and resource management optimization problem in highly dynamic and MEC-enabled next-generation wireless network scenarios. Integrating AI with SDN can play a significant role in dynamically adapting the resource allocations to the time-varying wireless channel conditions and automating real-time management and control of next-generation wireless networks. In our proposed AIDWN approach, we include the following design considerations, focusing on addressing the aforementioned challenges within the context of the dynamic wireless and mobile environment pertaining to multi-MEC server scenarios.

A. COMPUTATIONAL COMPLEXITY

The optimization methods adapted in the literature for computational offloading and resource allocation in highly dynamic and MEC-enabled next-generation wireless network scenarios are computationally hard, i.e., NP-hard optimization problems are generally considered computationally challenging to achieve an optimal solution. In the context of highly dynamic and time-varying wireless network conditions, especially in low-latency 5G and beyond wireless networks where rapid and swift decision-making is essential, solving NP-hard optimization problems in real time can be a challenging task. This is particularly applicable when the optimization problem involves complex variables, as the time required for computation may lead to increased latency, potentially affecting the performance requirements of MEC, i.e., ultra-low latency requirement of up to 1 ms latency [36].

Integrating AI with SDN could be a potential solution to address the challenges posed by computationally complex optimization problems in highly dynamic wireless network conditions, especially in low latency 5G and beyond networks. The programmability and centralized control offered by SDN allow for real-time network adaptability. Integrating it with AI and machine learning could facilitate the analysis of historical network data, e.g., network topology information, and make predictions about time-varying wireless channel conditions, where the SDN controllers can learn optimal network management policies through model training. SDN controllers can use this information to reconfigure the network proactively, optimizing resource allocation, implementing policies, and ensuring low-latency services even in the presence of computational complexity.

B. RESOURCE MONITORING IN MULTI-MEC SCENARIOS

Integrating AI with SDN for computational offloading and resource allocation represents a paradigm that excels beyond conventional deep reinforcement learning approaches. The key role played by SDN in facilitating efficient network orchestration is one of the most important aspects of this AI paradigm. The integration empowers real-time monitoring and dynamic allocation of available resources and enables a flexible distribution of computational tasks to MEC servers. In contrast, the traditional deep reinforcement learning methods adapted for computational offloading and resource allocation in edge computing are often based on a relatively static and less responsive network configuration.

Moreover, the integration facilitates intelligent decision-making and optimizes the QoS requirements. Dynamic adaptability and control over network resources could be achieved by constantly monitoring wireless network conditions and assessing the specific performance requirements of computational tasks. The collaboration enhances user experiences and resource utilization, thus significantly raising the system's utility. In addition, SDN is pivotal in achieving efficient traffic management within the network infrastructure. The centralized control and programmability of network elements enabled by the SDN optimize the routing of computational offloading traffic, preemptively mitigate congestion, and avert network bottlenecks. Unlike the conventional machine learning approaches adapted in the literature for computational offloading and resource allocation in edge computing that usually operate within more fundamental traffic management paradigms, AI combined with SDN offers a fine level of granular traffic control.

C. MONITORING WIRELESS ASSOCIATIONS

Integrating AI with SDN is instrumental in improving computational offloading and resource allocation in MEC. SDN plays a significant role in monitoring and gathering the status and associations of wireless connections, feeding them into the AI models running in the application plane to achieve informed and intelligent AI-defined controls. The informed decisions enable dynamic network path adjustments based on wireless connectivity status and ensure that computational tasks are efficiently routed through the most stable and available connections, considering the network congestion and resource availability.

D. DYNAMIC DATA FLOW MANAGEMENT

SDN enhanced with AI can play a crucial role in managing dynamic traffic control for computational offloading and resource allocation in edge computing scenarios involving multiple MEC servers. AI can provide intelligent decision-making capabilities to optimize data flow management. In this context, AI is used to analyze real-time wireless network information, application-specific requirements, and the capabilities of MEC servers (collected through the SDN management plane), and make informed decisions about computational offloading and resource allocation, taking into account factors such as latency, server load, and energy efficiency. In particular, when mobile users frequently relocate, they may be scheduled to offload their computational tasks to different MEC servers at different time slots depending on resource availability and wireless channel conditions. Therefore, dynamic routing of the computational tasks to the most optimal MEC server at each time slot is essential, ensuring efficient resource utilization and minimal latency for compute-intensive tasks. Additionally, AI can predict traffic patterns and anticipate when and where computational offloading will be needed, allowing SDN controllers to allocate resources and establish low-latency connections proactively. The level of automation and intelligence enhances overall dynamic data flow management, enabling edge computing networks to be more responsive and capable of meeting the diverse performance requirements of MEC services.

IV. THE PROPOSED AI-DEFINED WIRELESS NETWORKING ARCHITECTURE

A. AI-DEFINED SDN APPLICATION PLANE

The proposed AIDWN approach uses artificial intelligence and machine learning techniques at the application plane of SDN to optimize network performance and decision-making. The AIDWN approach evolves the conventional SDN application plane to an AI-defined application plane, as shown in Fig. 1. The architectural layers of the proposed AIDWN approach are defined as follows:

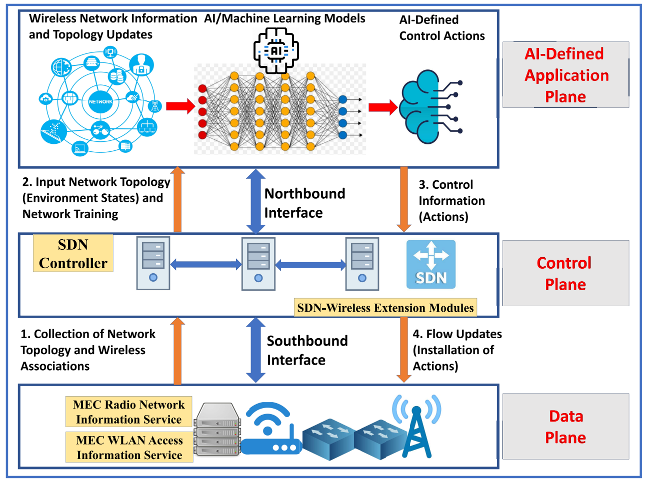


FIGURE 1. The proposed AI-defined wireless networking architecture.

- *AI-defined Application Plane:* A major component of the proposed AIDWN approach is the AI-defined application plane. The AI-defined plane integrates behavioral models and reasoning processes tailored to automate decision-making and network configurations in SDN-based wireless networks. The AI-defined plane uses the control and management planes to achieve a global view of the wireless network topology. The fundamental working process of AIDWN involves the ability of the AI-defined application plane to process the network analytics, e.g., network topology information collected through the management plane (EFM modules: SDN-wireless extension modules), and transforming it into AI via machine learning models. The AI is used to automate wireless network management decisions. The AI-defined application plane could learn different models and network management policies from the analytics platform's historical network data and real-time wireless network information, e.g., channel conditions, signal strength, and user location updates, collected from the radio network information service and WLAN access information service offered by MEC systems, as shown in Fig. 1. More discussion on the EFM modules is provided in the later sections. Different machine learning and deep learning approaches could be implemented in the AI-defined application plane to learn from the network behavior, including supervised, unsupervised, and deep reinforcement learning. For example, in the reinforcement learning approach, the network administrator can set a target policy to optimize the system utility of MEC by learning optimal computational offloading and resource allocation decisions; then, the reinforcement learning agent acts on the SDN controller by applying all the different computational offloading decisions and resource allocation configurations and

receiving a reward for each action, that increases as the agent learns the optimal target policy. In this context, the optimal policy could be learned through online or offline training.
- *Control Plane:* The SDN control plane uses the AI acquired to make informed real-time decisions, such as dynamic flow updates, traffic routing, and resource allocation, without solving complex combinatorial optimization problems at each iteration in highly latency-sensitive, dynamic, and varying wireless network environments. This allows for more efficient and effective management and utilization of constrained network resources in highly dynamic and rapidly changing environments, e.g., MEC-enabled resource-constrained environments. The control plane uses AI to automate network configuration and decision-making, dynamically adapting to varying channel conditions, enforcing control policies, and ensuring optimal data transmission by dynamically adjusting network parameters based on real-time network state analysis.
- *Data Plane:* The data plane constitutes the operational layer responsible for directly handling and forwarding network traffic. The actions and decisions received from the AI-defined application plane and the centralized control plane are implemented in the data plane. Specifically, the data plane executes the transmission and reception of data packets, performing functions such as packet forwarding, resource allocation, and routing based on decisions orchestrated by the AI-defined application and control planes. The data plane is realized through software or programmable hardware that physically handles the traffic based on the network configurations. This abstraction enhances adaptability to dynamic network conditions, flexibility, and efficient resource utilization.

B. AIDWN APPLICATION: COMPUTATIONAL OFFLOADING AND RESOURCE ALLOCATION IN EDGE COMPUTING

Fig. 2 depicts an application and use case of the proposed AIDWN approach that learns an optimal policy to optimize computational offloading and resource allocation, maximizing the utility of limited MEC resources in a highly congested, large-scale, and dynamic wireless network scenario. The data plane consists of base stations and multiple MEC servers M, where each MEC server j has its own dedicated computational resources and is connected to its respective base station/access point. We assume that there are U mobile users in the network, where each user u requests the offloading of computational tasks L, and execution of each computational task l requires computational resources from the MEC server at each time frame.

In practical applications, the dynamic and constantly varying MEC environment demands mobile users to make effective and real-time computational offloading decisions considering the uncertainty and unpredictability of the future channel conditions and task arrivals [37]. However, this presents two major challenges. Firstly, the rapidly changing wireless environment demands frequent re-solving of complex optimization problems in real-time, which can be computationally infeasible when the problem size is large, or the service request has ultra-low-latency requirements. Secondly, it is difficult for online computational offloading decisions to meet long-term system constraints, such as power and bandwidth consumption, and to dynamically adapt the optimal number of users selected for computational offloading as each mobile user may have varying numbers of low-latency tasks to be offloaded at each instance.

In this research, we propose a novel AIDWN approach for computational offloading and resource allocation in MEC. The SDN controller (control plane) in this approach consistently tracks the wireless network information, e.g., topology, channel conditions, signal strength, and user location updates, from the Radio Access Network (RAN) and feeds this information to the AI-defined application plane. In an emulator such as Mininet-WiFi [38], this can be done using EFM, as previously proposed in [33]. The EFM module provides the WLAN/RAN information service the SDN controller uses to optimize the wireless network management. This is achieved using *hostapd_cli*, a tool in Linux that runs in the background and scans and collects access network information, and a Python *Scapy* module that monitors mobile user, e.g., wireless channel information such as Received Signal Strength Indicator (RSSI). The collected information is then sent to OpenFlow-enabled Access Points (APs) on a specified source destination port. The APs use the EFM module to forward the wireless network information, i.e., network topology updates to the SDN controller by generating a *packet_in* message. The SDN controller then extracts the network information, achieving a global view of the underlying networks. More detailed information on the SDN extension for wireless networks and the EFM module can be found in [33] and the official documentation of Mininet-WiFi [38].

The network topology information is then fed into an AI-defined application layer that uses deep neural networks to provide real-time and optimal binary computational offloading, inter-MEC mobility and handover, and resource coordination and allocation decisions, as explained in more detail further on in this paper. The SDN controller then installs and updates new flows to route the mobile user traffic towards the optimal MEC server and allocates the MEC resources based on the optimal actions received from the AI-defined application plane.

V. AI-DEFINED APPLICATION PLANE FOR RESOURCE MANAGEMENT IN EDGE COMPUTING

The AI-defined application plane implements a deep reinforcement learning approach to learn from the network information and effectively apply AI-defined decisions in the data plane. The AI-defined application plane consists of

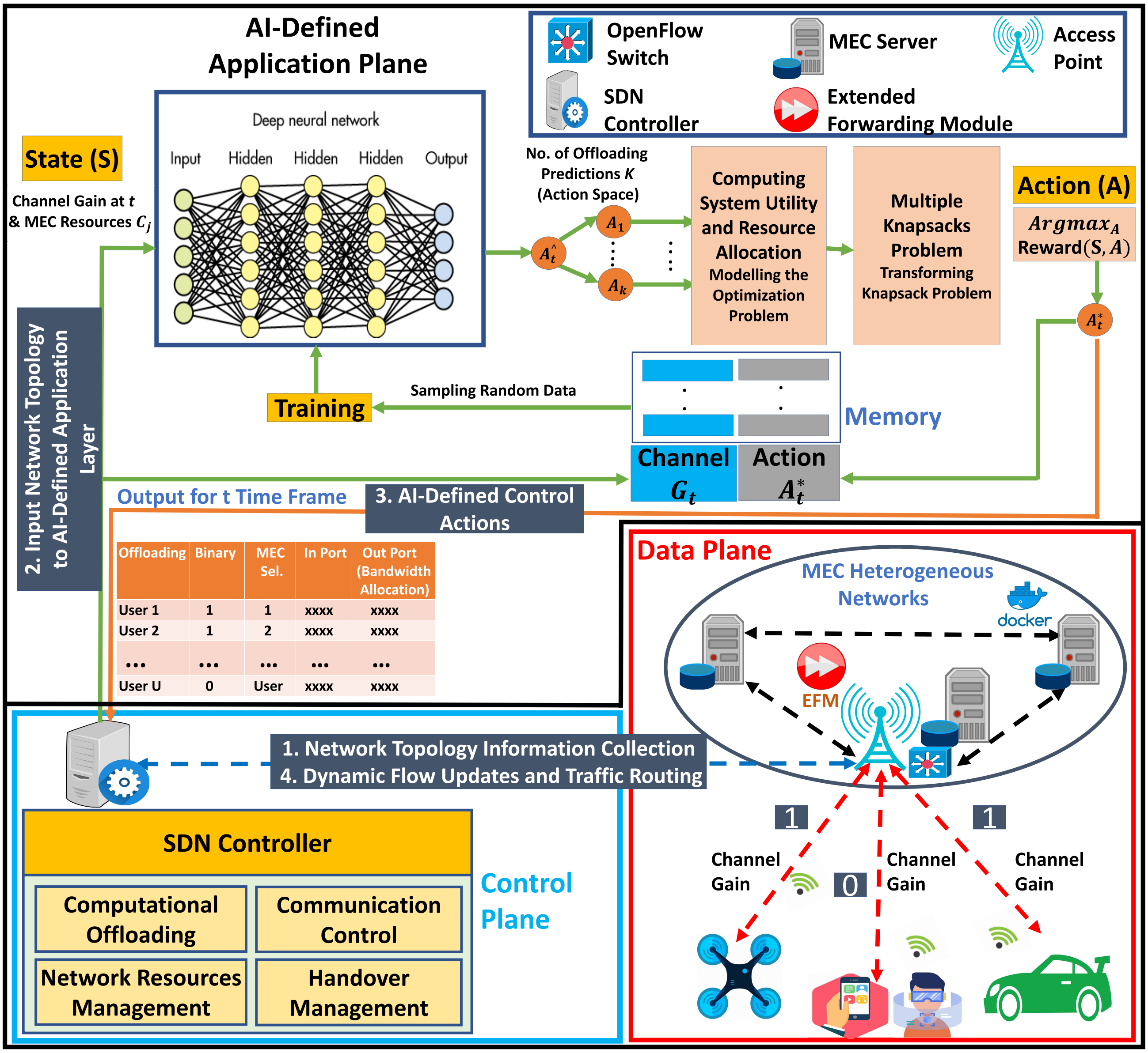


FIGURE 2. AI-defined wireless networking approach for computational offloading and resource allocation in MEC.

the following key components: the state and environment, actions, rewards, and system utility. We define these components in our systems as follows:

A. STATE AND ENVIRONMENT

The state S is represented as the collection of all environmental states, defined as the channel conditions, e.g., channel gain for each mobile user at any time slot t, and the corresponding number of binary computational offloading combinations as predicted by the deep neural networks in the AI-defined application plane. The SDN controller collects the global topology information, e.g., wireless channel conditions, association, and RSSI information, and feeds that into the deep neural networks in the application plane. The deep neural networks predict all the possible combinations of binary computational offloading decisions. The state space varies based on the number of mobile users in the network. In dynamic and highly congested scenarios such as vehicular networks where a large number of mobile users demand access to the limited resources of the MEC server, the state space could be significantly ample. The large state and action space make the conventional combinatorial optimization problems inefficient, as they are computationally intensive and lack the utilization of historical data. The conventional methods need to recalculate each time there are changes in the parameters of the wireless system, hindering real-time implementation. This makes the conventional optimization methods inefficient for highly dynamic and mobile scenarios such as vehicular networks. We decomposed the original optimization problem into two sub-problems, i.e., the resource allocation sub-problem and the offloading decision sub-problem, thus, avoiding the curse of dimensionality problem and not requiring the discretization of wireless channel gains. The decomposition approaches are known to effectively reduce the computational complexity of an optimization problem by decomposing a large-size problem into smaller parallel sub-problems [11].

B. ACTION AND SYSTEM UTILITY

The action A at any time slot t is defined as the resource allocation and computational offloading decisions received by the SDN controller, i.e., agent. The SDN controllers feed the wireless channel information, e.g., channel gain, to the AI-defined application plane at each time slot t that generates several binary computational offloading combinations predictions K. We modeled an optimization problem to select the most optimal computational offloading combination that maximizes the system utility, i.e., data rate, in a Gaussian channel. The optimization method is also used to determine the optimal allocation of power and bandwidth resources to each user selected for computational offloading to the MEC server in any combination of binary computational offloading prediction A_k.

The system utility S can be defined as:

$$S = \sum_{u=1}^{N} R_u, \tag{1}$$

where N represents the number of users predicted for computational offloading in any A_k, and R_u represents the achievable bit rate of each user u in a receiver channel in *bps*.

When the mobile user u is selected to offload its task to the MEC server, the task is first uploaded to the associated base station. A fraction of bandwidth and power resources are assigned for the transmission from the user u to the base station [39], [40].

Similar to the communication model in [10], [33], [39], [40], [41], R_u can be defined as:

$$R_u = W_{uj} \log_2\left(1 + \frac{G_{uj} P^j_{lu}}{N_o W_{uj}}\right), \tag{2}$$

where P^j_{lu} is the fraction of power assigned to task l by user u for the transmission from user u to the base station and its associated MEC server j [39], [41], W_{uj} is the fraction of total bandwidth provided by the base station and its associated MEC server j allocated to each user u, and N_o is the receiver noise power [10], [39], [40], [42], [43]. Here G_{uj} is a known positive representing the channel gain between the user u and the base station and its associated MEC server j. The objective is to select the most optimal computational offloading combination, i.e., action A^s_k, that maximizes the MEC system utility subject to the resource constraints.

The optimization problem is formulated as follows:

$$\begin{aligned}
S_{max} = \text{Max} \quad & \sum_{u=1}^{N}\sum_{j=1}^{M} R_u x_{uj} \\
\text{Subject to:} & \\
\text{CO1} \quad & \sum_{u=1}^{N} W_u x_{uj} \leq W_j^{\text{total}}, \quad \forall j \in M \\
\text{CO2} \quad & \sum_{j=1}^{M} x_{uj} = 1, \quad \forall u \in N \\
\text{CO3} \quad & x_{uj} \in \{0, 1\}, \quad \forall u \in N, j \in M
\end{aligned} \tag{3}$$

Here, CO1 guarantees that the combined bandwidth allocated to the N users by MEC server j does not exceed the maximum capacity of MEC server j. CO2 ensures that each mobile user u is handed over to no more than one MEC server at any given time slot. Thus, it takes care that tasks from user u are offloaded to at most one MEC server j. CO3 represents the binary decision constraint, i.e., x_{uj} can only be 0 or 1, which indicates whether a user u is offloaded to MEC server j. In addition, as (3) restricts each mobile user to be handed over to a maximum of one MEC server at a time, we assumed that each computational task l for MEC server j by user u requires equal transmit power, such that the cumulative transmit power allocated for the transmission of computational tasks L to MEC server j remains within the specified maximum for user u, i.e.,

$$\sum_{l=1}^{L} P^j_{lu} \leq P_u^{\text{total}}. \tag{4}$$

The complexity of the optimization problem (3) can be reduced if the bandwidth resources allocated to each user selected for computational offloading, i.e., W_u, are known. In our system model, the SDN controller accumulates and coordinates the available resources among all the MEC servers in the network and acts as a central intelligent entity that optimally allocates bandwidth resources to each user, i.e., W_u. The allocation is solved using the convex optimization solver CVXPY [44], where the objective is to maximize S subject to

$$\sum_{u=1}^{N} W_u \leq W_M^{total}. \tag{5}$$

Here, W_M^{total} is the accumulated bandwidth resources of all the MEC servers M in the network coordinated by the SDN controller.

The optimization problem (3) is an NP-hard problem. To establish the NP-hardness of (3), we elucidate a polynomial-time reduction from the 0-1 Multiple Knapsack Problem (01-MKP) to (3) [45]. The reduction involves transforming instances of the 01-MKP into instances of (3) so that solving (3) implies an efficient solution to the 01-MKP, a known NP-hard problem [45]. In the 01-MKP, we have a set of items I with values v_i and weights w_i, and a set of knapsacks kn

with capacities C_{kn}. To map these elements to (3), we define each item i in the 01-MKP corresponding to a user-MEC pair (u, j), the value v_i of item i represents the achievable bit rate R_u of user u, and the weight w_i of item i is transformed to represent the bandwidth requirement of offloading tasks from user u. Additionally, each knapsack kn in the 01-MKP corresponds to a MEC server j in (3), with the capacity C_{kn} mapped to the total available resources W_j^{total} of MEC server j. This transformation is represented as:

$$v_i \to R_u, \quad w_i \to W_u, \quad C_{kn} \to W_j^{\text{total}}$$

Through this correspondence, solving (3) yields an optimal solution to the 01-MKP, and vice versa, establishing the NP-hardness of (3).

The current solutions in the literature proposing deep reinforcement learning approaches for computational offloading in a MEC environment mainly consider the achievable system utility, e.g., computation rates and processing delay, as the reward function where the A_k^s is selected that maximizes the system utility [10], [11], [12]. The existing proposals prioritize selecting users for computational offloading with the highest channel gain, which could lead to a higher system utility. The solutions do not take into account the number of MEC servers and their corresponding resource availability, leading to inefficient use of vital and scarce MEC resources. Therefore, in our proposed solution, in addition to the data rates, we also aim to maximize the number of mobile users benefiting from computational offloading to fully use the limited and varying MEC resources available at each time slot t. Although in most use-case scenarios, some mobile users may prefer to process a simple or delay-insensitive task using their own computational resources, we assumed that each mobile user has delay-sensitive computational tasks to be offloaded to the MEC server at each time slot t. This assumption is made to evaluate the performance of the proposed approach in a practical, complex, and highly congested wireless network environment where the demand for MEC resources is always higher than the availability. Therefore, we define the overall system utility function (initial reward) U_o as:

$$U_o = \rho_s S + \beta N_p, \tag{6}$$

where N_p is the ratio of the number of mobile users predicted for computational offloading in any A_k to the total number of mobile users in the network, i.e., $N_P = \frac{N}{U}$. Here ρ_s is a scaling factor introduced to adjust the magnitudes of S to achieve a balanced contribution of all the terms and a balanced learning process [46], [47]. The parameter β is a control parameter used to define the priority of each component in this utility function. The SDN controller then compares each combination of binary computational offloading predictions received from the deep neural networks and selects the action A_k^s that has the highest overall system utility (initial reward). A list of the used notations in the paper is provided in Table 1.

TABLE 1. Notation table.

U	Total number of users in the network
M	Total number of MEC servers in the network
L	A set of computational tasks
u	A user u in the network
j	A MEC server j
l	A single computational task l
K	No. of binary offloading combinations predictions
A_k	Any combination of binary offloading prediction
N	No. of users predicted for offloading in any A_k
A_k^s	Selected binary offloading prediction
A_t^*	Optimal computational offloading action at time slot t
R_u	Bit rate for user u in a receiver channel
W_{uj}	A fraction of total bandwidth resources of MEC j allocated to user u
P_{lu}^j	A fraction of power assigned to task l by user u for its transmission from user u to MEC j
N_o	Receiver noise power
G_{uj}	Channel gain between user u and MEC j
S	System utility
U_o	Overall system utility (initial reward)
W_M^{total}	Accumulated bandwidth resources of MEC servers M coordinated by the SDN controller
x_{uj}	Binary decision variable for offloading user u to MEC server j
V_u	Data rate for user u if selected for offloading
z_u	Computing resources requested for offloading by user u
C_j	Maximum compute capacity of MEC server j

C. REWARD

Maximizing the number of mobile users selected for computational offloading also poses a significant challenge. The selection of A_k^s having the highest overall system utility could result in over-utilizing the limited MEC resources as the SDN controller would always tend to select the action with the maximum number of users predicted for computational offloading. Given the limited and varying MEC resource availability at any time slot t, this could lead to over-utilized MEC resources and, as a result, cause significant performance degradation in the quality of service achieved by mobile users. Therefore, we formulated a 01-MKP which is an optimization problem that aims to select users for computational offloading and assigns them to MEC servers, i.e., knapsacks, that maximize the overall system value, without exceeding the capacity of the MEC servers.

The 01-MKP is a combinatorial optimization problem where a set of items are packed into a fixed number of knapsacks, i.e., bins, while maximizing the overall system profit and staying within the constraints of limited knapsack capacity. Each item has a specific size, and each knapsack has a limited capacity. The 01-MKP is an NP-hard problem that is computationally intensive and infeasible to find an optimal and exact solution within a reasonable time. We define the fixed number of knapsacks as the number of MEC servers, each with limited resource capacity, whereas the item is defined as a computational task of a specific size requested for computational offloading by the mobile user.

Let x_{uj} be a binary decision variable that indicates whether the computational task requested for offloading by user u is handed over to the MEC server j. It is defined as:

$$x_{uj} = \begin{cases} 1, & \text{if user } u \text{ is handed over to MEC server } j, \\ 0, & \text{otherwise.} \end{cases}$$

The objective is to maximize the overall MEC system value V, i.e., aggregated sum of achievable data rate in *Mbps* by optimizing the selection of mobile users selected for computational offloading, subject to resource constraints. The objective function is defined as:

$$\begin{aligned} V_{max} = \max \quad & \sum_{u=1}^{N}\sum_{j=1}^{M} V_u \cdot x_{uj} \\ \text{s.t. } C1 : & \sum_{j=1}^{M} x_{uj} = 1 \quad \text{for } u = 1, 2, \ldots, N, \\ C2 : & \sum_{u=1}^{N} z_u \cdot x_{uj} \leq C_j \quad \text{for } j = 1, 2, \ldots, M. \end{aligned} \tag{7}$$

Here, V_u is the individual value of user u in *Mbps*, representing the data rate that can be achieved if selected for computational offloading. Constraint C1 represents that each mobile user must be handed over to a maximum of one MEC server at a particular time slot, whereas C2 represents the total computing resources required to compute the tasks from the mobile users selected for computational offloading should not exceed the maximum capacity of MEC server j. z_u represents the number of processing units, i.e., virtual CPUs (vCPUs), required to compute the task(s) requested for offloading by user u, and C_j represents the maximum computing resources, i.e., vCPUs, available in MEC server j.

Given the highly dynamic nature of mobile networks, e.g., UAV and V2X networks, with frequent topology changes, system parameter variations, and stringent performance requirements, finding optimal computational offloading and resource allocation decisions with ultra-low latency is essential. Therefore, we formulated a reward function to train the deep neural networks to maximize the expected reward over time and learn an optimal policy. The AI-defined application plane selects the mobile users from the selected computational offloading combination, i.e., action A_k^s having the highest overall system utility in (6), that maximizes the overall system value, i.e., aggregated sum of achievable data rate, while following the resource constraints of the MEC servers using (8), producing the optimal computational offloading action A_t^*, at each time slot t. The proposed approach takes into account the varying resource demands for MEC resources, i.e., the number of tasks offloading requests at each time slot t, and adapts the optimal number of users selected for computational offloading while maximizing the overall system value and meeting the MEC resources constraints. The reward function RW can be defined as:

$$\begin{aligned} RW = & \sum_{u=1}^{N}\sum_{j=1}^{M}\left(\tau_v \cdot V_u \cdot x_{uj}\right) - A_p \cdot \sum_{j=1}^{M}\left(\sum_{u=1}^{N}(z_u \cdot x_{uj}) - C_j\right) \\ & - B_p \cdot \sum_{u=1}^{N}\left(1 - \sum_{j=1}^{M} x_{uj}\right) \end{aligned} \tag{8}$$

Here, A_p is a constant penalty factor that controls the trade-off between optimizing the overall system value and adhering to MEC resource constraints, with higher values resulting in amplifying penalties for exceeding resource limits. B_p is a constant penalty factor that influences the balance between maximizing the overall system value and enforcing the constraint that each mobile user is assigned to only one MEC server. τ_v is a scaling factor introduced to adjust the units and magnitudes of V_u to achieve a balanced contribution of all the terms to the overall reward and a balanced learning process [46], [47]. The process is shown in Fig. 2.

The pseudo-code of the working of the proposed AIDWN approach for computational offloading and resource allocation in a highly dynamic and mobile wireless networking environment is provided in Algorithm 1. In addition, the overall learning process of the proposed solution is summarized as described in the following paragraph.

The system utility evaluates the maximum achievable data rate for different binary computational offloading combinations predicted by deep reinforcement learning agents in the AI-defined application plane. Meanwhile, the overall system utility consolidates multiple objectives that encourage behaviors that maximize achievable data rates and increase the number of mobile users who successfully offload computational tasks to the MEC servers and effectively utilize scarce MEC resources. To manage resource constraints and avoid overestimating the number of mobile users selected for computational offloading and thus violating the QoS requirements, a formulated reward function selects actions that maximize overall system value, ensuring efficient utilization of computational resources while maximizing data rates. Together, system utility, overall system utility, and reward provide critical feedback mechanisms that guide agent decisions in the AI-defined application plane, facilitating the optimization of computational offloading strategies in MEC environments.

VI. NUMERICAL RESULTS AND DISCUSSIONS

A. SIMULATION SETUP

The simulation results are presented to exhibit the performance gain achieved by using the novel AIDWN approach in terms of maximizing the overall system value, the number of mobile users successfully offloading the tasks to the MEC, and ensuring effective utilization of limited MEC resources. We also provided numerical comparisons with other similar approaches in the literature that apply a deep reinforcement learning framework for computational

Algorithm 1 AIDWN for Computational Offloading and Resource Allocation in MEC

Input: Wireless channel information, e.g., channel gain, total MEC resources available **(SDN Northbound Interface)**
Output: AI-defined control actions: optimal computational offloading actions, inter-MEC mobility and handover decisions, resource coordination and allocation decisions, and SDN flow, control, and routing updates **(SDN Southbound Interface)**
Begin
Initialize deep neural network and an empty memory
Setting time iterations T=10000 and a training interval
for *for t=1,2,3,4,......T* **do**
    Generate binary offloading combinations predictions K
    Computing system utility S and resource allocations P_{lu}^{j} and W_{uj} for each generated combination
    Computing overall system utility U_o (initial reward) for each generated combination
    Select the action A_k^s that maximizes the U_o
    Calculate the Reward *RW* at time t
    Select the optimal computational offloading action A_t^* that maximizes the reward using (8)
    Updating the memory with wireless channel gain (Input) and the corresponding A_t^* (Action)
    if t mod *training interval* == 0 **then**
        Uniform sampling of data set from the memory
        Train the deep neural network
    end if
end for

offloading in MEC. The proposed approach is compared to the following other approaches:

- The Deep Reinforcement Learning for Online Computation Offloading (DROO): The DROO approach presented in [10], [11], [48] aims to maximize wireless users' computation rates by achieving optimal computational offloading decisions. However, as with most other work in the literature, the algorithm tends to always select the computational offloading decision as an optimal action where the mobile users have the strongest wireless channel gain, without taking into account the varying resource availability on MEC servers per time slot, leading to inefficient use of limited MEC resources.
- The Conventional Proximity-based Offloading (CPBO) approach as detailed in [49], [50] where the mobile users are selected to offload/handover tasks for computation to the nearest MEC servers based on the channel signal strength, e.g., RSSI-based computational offloading. The conventional approach lacks resource coordination and collaboration among the MEC servers enabled by the proposed AIDWN approach. This resource-unaware strategy can lead to inefficient utilization of MEC resources, wherein one MEC server may experience overload while another in the same network remains underutilized.

In an extensive and congested wireless network with a large number of mobile users, several candidate binary combinations of computational offloading decisions are possible. Prioritizing the combination where only a few users have a very high wireless channel gain could lead to a severe under-utilization of scarce MEC resources. For example, in a network of 10 mobile users, a binary combination [1, 0, 0, 0, 0, 0, 1, 0, 0, 0] (i.e., offload users 1 and 7) may become preferred over [1, 1, 0, 0, 1, 0, 1, 0, 1, 0], as the former combination has two users with significantly higher channel conditions, resulting in higher system utility, irrespective of it maybe resulting in significant under-utilization of MEC resources. Therefore, our proposed solution leverages the centralized intelligence and programmability offered by the SDN controller to consistently monitor and coordinate the resources available on multiple MEC servers, and it fully uses the available resources to maximize the system utility in a highly congested wireless network scenario.

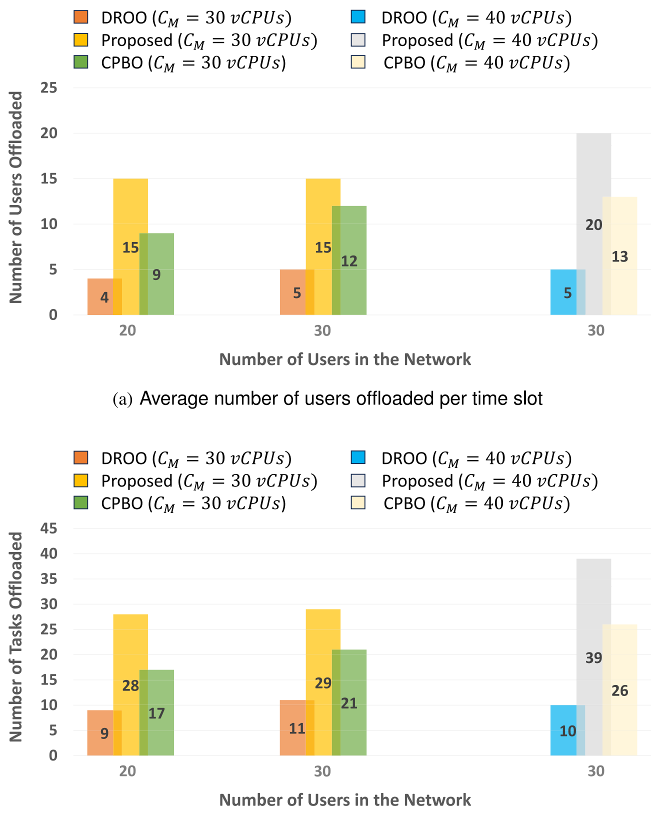
We considered a fully connected deep neural network with 1 input layer, 2 hidden layers, and 1 output layer, with the first and second hidden layers having 120 and 80 hidden neurons, respectively. The proposed AIDWN is implemented in Python using TensorFlow 2.0 with a training interval set to 10, training batch size equal to 128, memory size of 1024, and learning rate for Adam optimizer as 0.01. These parameters of the deep neural network align closely with those of the DROO approach [10] for a fair comparison with our proposed solution. We considered 3 MEC servers, i.e., M=3, available in the network with a total bandwidth of $W_M^{total} = 50MHz$, and receiver noise power N_o=10^{-10} [10]. We evaluated our proposal for varied capacities by varying the total number of computing resources available in the MEC, i.e., C_M=30 vCPUs and 40 vCPUs, where a MEC resource (vCPU) is assigned to compute only one of the tasks l requested for offloading by the user u, and each time slot t is 1 second [51], [52]. A task is a specific set of computations, i.e., a computation job, that can be offloaded to the MEC servers and performed by 1 vCPU during a single time slot. C_M is the aggregated and coordinated capacity of all the MEC servers in the network that the SDN controller could use to offload and distribute the computational tasks from the mobile users. We also varied the computing resources of each MEC server j, i.e., C_j, and used different combinations, e.g., (C_1, C_2, C_3)=(20,5,5) vCPUs and (C_1, C_2, C_3)=(20,10,10) vCPUs.

As with most existing research, as discussed in the related works, the DROO approach [10], [11], [48] only explores a scenario involving a single MEC server. In contrast, our proposed AIDWN approach considers a more comprehensive network with multiple MEC servers. To facilitate a fair comparison, we assumed that the maximum capacity of the single MEC server is equivalent to the aggregated capacity of multiple MEC servers in our network, enabled by the SDN controller. This ensures that both approaches under evaluation have the same maximum MEC capacity for mobile users' computational offloading.

We also evaluated the proposed solution for varying numbers of mobile users in the network, i.e., U=20 and 30, where the users are randomly distributed across the coverage areas of the MEC servers in the network [53]. We assumed that each mobile user has a varying number of computational tasks to be offloaded to the MEC servers at each time slot t, where the samples are uniformly distributed, i.e., each mobile user can offload 1 to 3 tasks of the same size at any time interval t, i.e., z_u=1–3 vCPUs. This assumption is made to evaluate the performance of the proposed AIDWN approach in complex, highly congested, and resource-demanding wireless network scenarios, where full utilization of scarce MEC resources is essential. Considering the parameters specified above, we explored various simulation scenarios to assess the viability of our proposal, focusing on highly congested environments where the demand for MEC resources is higher than the available capacity. We considered different combinations including: (1) U=20, C_M=30 vCPUS, (2) U=30, C_M=30 vCPUS, and (3) U=30, C_M=40 vCPUS.

We used Mininet-WiFi [38], [54], a widely accepted fork of Mininet that allows the simulation of SDN-capable APs, wireless stations, and docker containerized services and applications. Using Mininet-WiFi, mobile users were simulated as wireless hosts capable of connecting with the APs and their corresponding MEC servers. Three OpenFlow-supported APs were simulated, each connected to their respective MEC server. Each MEC server is modeled as a virtual host with computational resources. We considered a network deployment scenario where all the APs have overlapping coverage areas to facilitate a cooperative scenario where the SDN controller can make use of the AI-defined application plane to efficiently enable coordination and distribute computational tasks among different APs and their corresponding MEC servers depending upon the availability of the resources and system utility. We chose RYU for the SDN controller for our experimental evaluation. A detailed description of the Mininet-WiFi components used in this research and their working is provided in [49]. The proposed solution is radio access technology agnostic and can be applied to both the Wi-Fi and 5G networks, a requirement defined by the European Telecommunications Standards Institute (ETSI) in [55]. Therefore, the proposed approach can be applied to both the cellular and vehicular networks.

Unlike the existing research, the AI-defined control actions, e.g., computational offloading decisions and resource allocation configurations, are implemented in real-time as the SDN controller installs dynamic flows, routing, and control updates in the data plane devices. The mobility updates redirect and hand-over the traffic from the mobile users selected for computational offloading to the corresponding MEC servers.



(a) Average number of users offloaded per time slot

(b) Average number of tasks offloaded per time slot

FIGURE 3. **Average number of mobile users and tasks offloaded at each time frame.**

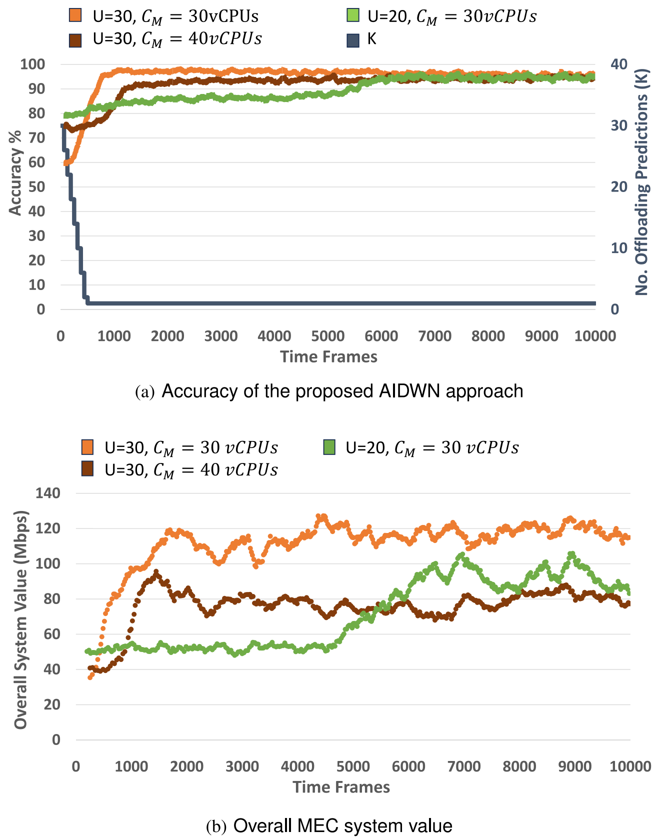
B. NUMERICAL RESULTS

The numerical results exhibit the performance gains of the proposed AIDWN approach in terms of maximizing the usage of limited MEC resources, the number of users benefiting from the MEC computational capabilities, and the overall system value, i.e., aggregated sum of data rate, in highly dynamic and mobile scenarios. We also highlighted the potential of our proposed approach in learning from the past and significantly reducing the number of binary offloading combinations predictions K, i.e., action space, required to obtain an optimal computational offloading decision for all the mobile users in the network. In a highly congested and dynamic wireless network environment, the action space becomes enormous because of the large number of mobile users in the network. Therefore, to reach an optimal computational offloading and resource allocation action, the optimization algorithms must be executed many times at each time slot t, i.e., solving the optimization problem for each action, which adds significant computational complexity to the network. In our proposed approach, we reduced the number of actions the deep neural network generates as it learns to adapt and predict the most optimal computational offloading action during the training process, as shown further on in this paper.

Fig. 3 illustrates the performance in terms of the number of mobile users benefiting from offloading computational tasks to the MEC server and the total number of tasks offloaded to the MEC servers at each time slot. Fig. 3 also compares the proposed AIDWN with DROO and the CPBO approach. The results show that, for a typical network with 3 MEC servers with a total of 30 or 40 available vCPUs, and with 20-30 users offloading 1-3 tasks, the proposed AIDWN outperforms DROO and the CPBO approach in terms of maximizing the use of limited MEC resources, as shown in Fig. 3. A significantly larger number of mobile users could benefit from computational offloading to MEC servers at any time slot while staying within the limits of maximum MEC capacity C_M, as seen in Fig. 3(a). The proposed approach can fully use the resources available at the MEC servers, where the average number of tasks successfully offloaded per time slot is very close to the maximum MEC capacity C_M, as shown in Fig. 3(b). As mentioned above, the reason for this phenomenon is that DROO selects the computational offloading actions where the users have the highest channel gain. It does not take into account the resource availability and capabilities of MEC servers, resulting in the under-utilization of MEC resources. In the CPBO approach, uneven and random distribution of mobile users leads to a bias toward frequently offloading tasks to nearby MEC servers, potentially causing overloading and neglecting less loaded neighboring MEC servers, resulting in the inefficient utility of MEC resources. Maximizing the number of mobile users benefiting from the MEC capabilities at any time slot is essential in highly congested and dynamic wireless network scenarios where the demand for MEC resources is significantly higher than the availability.

Fig. 4(a) shows the accuracy of the proposed approach in terms of predicting the varying demands for MEC resources, i.e., the number of computational tasks requested for offloading by each mobile user, and selecting an optimal number of mobile users for computational offloading at each time slot. The prediction accuracy is the ratio of the number of mobile users successfully offloading the computational tasks to the MEC and the number of mobile users predicted for computational offloading while staying within the resource constraints of the MEC at each time slot. The users successfully offloading tasks consistently end up being fewer or equal to the initially predicted set, resulting in an accuracy metric consistently ≤ 1. This is because of the early-stage strategy that the agent adopts, overestimating the number of mobile users predicted for offloading to ensure robust utilization of MEC resources. Subsequently, through a combination of the multiple knapsack optimization problem and a carefully designed reward function, the agent consistently improves its prediction accuracy to optimally select users for offloading, maximizing rewards within resource constraints. The agent learns to predict optimal computational offloading decisions throughout training, leading to reduced predictions and algorithmic executions. The accuracy Acc per time slot can be defined as:

$$Acc = \frac{\text{Users offloading}}{N} \times 100 \tag{9}$$



(a) Accuracy of the proposed AIDWN approach

(b) Overall MEC system value

FIGURE 4. Performance of the proposed AIDWN approach and dynamic tuning of action space.

The results show that the prediction accuracy gradually increases and finally converges, due to the adaptability of the proposed approach in dynamically learning the optimal computational offloading actions and reducing the number of actions required to find the optimal one during the training process. This significantly reduces the processing time and latency associated with achieving the optimal computational offloading decisions, as it reduces the number of times K (dark blue curve) the optimization problem needs to be solved. The same phenomenon can be seen in Fig. 4(b) where the overall system value, i.e., V_{max} as defined in (7), gradually increases and converges during the training process, whereas the fluctuations observed are because of the rapid variations in the channel conditions of the mobile users at each time slot affecting the aggregated sum of data rate.

Fig. 5 illustrates the ability of the proposed AIDWN approach to efficiently manage mobility and distribute computational offloading tasks among different MEC servers, with the main objective of optimizing overall MEC system utility while adhering to the individual constraints of each MEC server. The AIDWN framework facilitates the real-time collection of pertinent network data, providing valuable insights into workload patterns and resource availability across various MEC servers, and it allocates computational

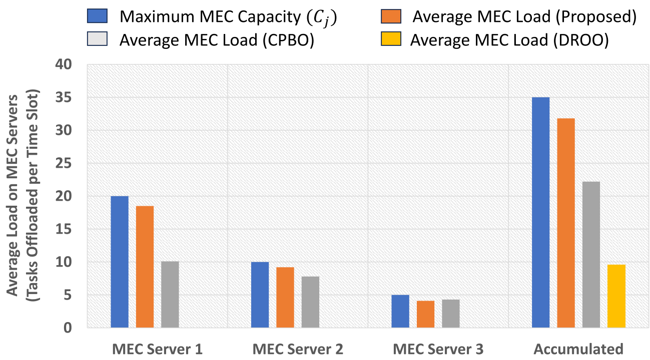


FIGURE 5. Average number of tasks offloaded/handover to MEC servers per time slot.

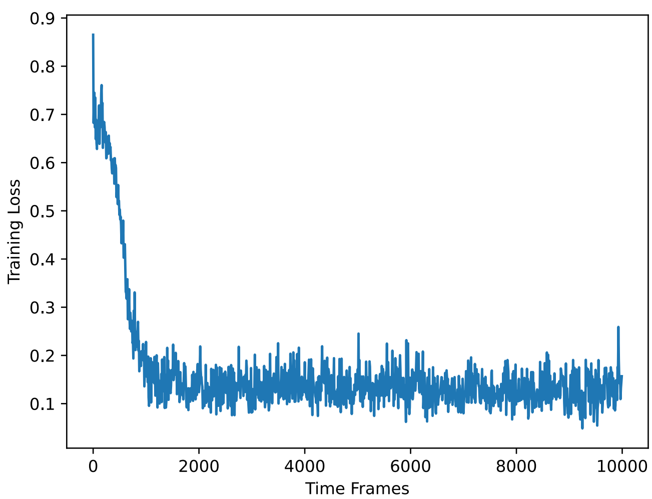


FIGURE 6. Training loss.

offloading tasks to different MEC servers based on available resources. Based on these allocations, the SDN controller performs the traffic handover from the mobile users selected for computational offloading to their respective MEC servers. In Fig. 5, we present a scenario featuring multiple MEC servers within the system, each characterized by varying maximum resource capacity denoted as C_j, i.e., (C_1, C_2, C_3)=(20,10,5) vCPUs. As observed in Fig. 5, the AIDWN approach effectively optimizes mobility and workload distribution among these MEC servers, thereby maximizing the resource utilization of each server. Fig. 5 also compares the effectiveness of the proposed AIDWN approach with that of DROO and the CPBO approach in terms of optimizing MEC resource utility. As DROO is designed for a single MEC server scenario, directly comparing the utility of individual servers is not feasible. However, in an aggregated scenario, we evaluate our proposal against DROO by aligning the maximum capacity of the single MEC server with the aggregated capacity of all MEC servers, i.e., C_M, in our network, as enabled by the SDN controller in our proposed case. The results highlight significant performance improvements in the efficient utilization of limited MEC resources compared to the DROO approach. Whereas the CPBO approach results in an inefficient and imbalanced distribution of computational tasks among the 3 MEC servers, leading to overloading certain servers while under-utilizing others. This is because users may be distributed unevenly, and those close to a low-capacity server might always look to offload tasks to it more frequently, even if it is over-utilized and other neighboring high-capacity MEC servers, e.g., MEC server 1, are less loaded.

Fig. 6 represents the training loss as a measure of the error/difference between the predicted and target values during the training process [56]. Training loss is used to update the parameters of the neural network to improve the agent's performance. As seen in Fig. 6, the training loss gradually decreases during the training process, where the fluctuations seen are mainly because of the randomly sampled training data.

TABLE 2. Comparisons of execution latency.

# of Users	Proposed	DROO
20	39 *ms*	32 *ms*
30	73 *ms*	64 *ms*

C. COMPUTATIONAL COMPLEXITY ANALYSIS

Our analysis reveals that the overall computational complexity of the proposed algorithm is dependent upon several factors, primarily driven by the total number of users in the network (U) and the number of binary offloading combinations predictions (K). Additionally, the system utility and reward calculation complexity ($N \cdot M$) and the training of the neural network ($E \cdot T$) contribute to the computational demands, where E is the size of the training dataset, and T is training iterations. We evaluated the overall computational complexity of the proposed algorithm in terms of its execution latency. This method is pertinent to evaluating computational complexity for NP-hard computational offloading solutions in MEC, as highlighted in prior works [10], [57].

We evaluated and compared the execution latency of the proposed algorithm with the DROO approach. To facilitate fair comparison, we excluded the comparison of execution latency with the CPBO approach, as the conventional approach does not include significant optimization objectives. Instead, the mobile users are offloaded to the nearest MEC servers based on the channel signal strength. The execution latency of the proposed algorithm and its comparison with the DROO approach is provided in Table 2, averaging over 10,000 independent wireless channel realizations, including both offloading action generation and deep neural network training. For example, in the proposed approach, in a network of $U = 20$, it takes 39 *ms* to generate an optimal computational offloading action and train the deep neural network at each time frame. Similar to the DROO approach, the simulations are performed on an Intel Core i5 machine with a 3.2 GHz CPU and 12 GB memory.

The analysis of execution latency, as seen in Table 2, indicates that the proposed approach exhibits execution latencies that are highly comparable to those of the DROO approach, despite the proposed AIDWN approach considering a more comprehensive and complex network scenario and optimization problem involving multiple MEC servers compared to the single MEC server scenario employed in the DROO approach. The AIDWN approach entails exploring numerous (u, j) combinations to optimize user offloading to MEC servers, making the optimization problem significantly harder compared to DROO. Despite this complexity, achieving similar execution latencies to DROO signifies the efficiency of the proposed AIDWN approach. As shown previously in Fig. 4(a), the proposed algorithm mitigates computational complexity through the dynamic reduction of K as the deep neural network learns during the training process in the AI-defined application plane. This reduction in K streamlines the decision-making process, leading to enhanced computational efficiency without compromising the ability of the proposed algorithm to derive optimal offloading strategies.

VII. CONCLUSION, POTENTIAL USE CASES, AND FUTURE WORK

A. CONCLUSION

In this article, we proposed a novel AI-defined wireless networking approach for managing MEC-enabled next-generation wireless networks. The AIDWN implements a new AI-defined application plane to process the network analytics, network topology, and wireless channel information received from the SDN control plane, making intelligent, informed, and optimal control actions and learning optimal application-specific control policies. The AI-defined actions are used to dynamically and efficiently control the network flow information, mobility and handover, resource coordination and allocation, and data plane management in wireless networks. We utilized the proposed approach to dynamically adapt the optimal binary computational offloading and resource allocation decisions in MEC use case scenarios. Finally, the numerical results highlight this approach's benefits in optimizing the use of limited and scarce MEC resources in a typical highly dynamic and complex mobile network scenario.

B. AIDWN APPLICATION DOMAINS

1) FEDERATION OF MEC RESOURCES

The inter-MEC system communication and collaboration between network operators are essential for mutual consumption and sharing of MEC capabilities offered by them [58]. The MEC resources are limited, and the dedicated MEC services are not readily available on each MEC server of a specific network operator, particularly for the highly dynamic and mobile networks scenario, such as the vehicular networks [49], [59]. Therefore, migration of MEC services across different MEC servers and federation of MEC resources between different network operators is essential to ensure service quality in mobile scenarios [49], [60]. Any methods and approaches developed to ensure the continued service quality for mobile network scenarios should be lightweight and meet the stringent performance requirements of their specific use case. One possible solution to ensure continued service quality in highly dynamic and mobile network scenarios is to use the AIDWN approach, where the AI-defined application plane could learn from the past wireless network information and adapt optimal MEC service migration and resource coordination decisions between network operators in real time.

2) NETWORK SLICING ADMISSION CONTROL

The service providers need to instantiate different types of network slices with diverse service requirements to meet the envisioned use cases of 5G and beyond [61], [62]. The dedicated network slices should be implemented at the MEC servers to meet the stringent performance requirements of the 5G and beyond use cases, i.e., ultra-low latency [63]. However, the MEC servers have constrained network resources, and therefore, developing optimal and efficient joint network slice admission control and resource management techniques is essential. The methods should be dynamic and in real-time able to adapt optimal admission control and resource allocation decisions and capable of scaling the network resources to meet the diverse service requirements [51], [63]. The AIDWN approach could be a potential solution to intelligent management of network slice admission control and resource allocation. Using the AI-defined application plane, the SDN controller can make informed decisions on admission control and resource allocation based on past information and network training.

3) SECURITY OF WIRELESS COMMUNICATIONS

AIDWN approach holds the potential to significantly enhance the security of wireless communication networks, with a particular focus on fortifying physical layer security. Physical layer security [64] utilizes the physical attributes of the communication channel, encompassing factors such as signal strength, interference profiles, and propagation characteristics, to support an additional layer of security beyond the conventional cryptography methods. In theory, perfect secrecy is achievable when the Shannon capacity of a potential eavesdropper station remains lower than that of the legitimate receiver. The centralized intelligence provided through the SDWN paradigm offers a comprehensive view of the network's topology [65]. This network information can be harnessed within the AI-defined application plane to effectively allocate legitimate mobile users for computational offloading to APs and their corresponding MEC servers, optimizing the network's secrecy capacity through dynamic handover and mobility management functionalities.

C. FUTURE WORK

For our future work, we will focus on exploring how interference impacts computational offloading scenarios within MEC-enabled wireless networks, particularly dynamic and mobile network scenarios such as V2X and UAV. We plan to investigate how interference from neighboring MEC servers and mobile users can influence the efficacy of offloading computations to MEC servers. We will leverage machine learning and artificial intelligence algorithms to develop adaptive interference management strategies that optimize resource allocation and transmission parameters in real time, mitigating interference effects, specifically in the context of computational offloading in MEC-enabled wireless networks. By incorporating real-time network feedback and environmental data as enabled by the proposed AIDWN approach, our goal is to enhance the reliability and efficiency of offloading processes, thereby maximizing the performance of MEC-enabled wireless networks in dynamic and congested wireless network environments.

REFERENCES

[1] L. Kong et al., "Edge-computing-driven Internet of Things: A survey," *ACM Comput. Surveys*, vol. 55, no. 8, pp. 1–41, Dec. 2022.

[2] Y. Chiang et al., "Management and orchestration of edge computing for IoT: A comprehensive survey," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14307–14331, Aug. 2023.

[3] H. Djigal, J. Xu, L. Liu, and Y. Zhang, "Machine and deep learning for resource allocation in multi-access edge computing: A survey," *IEEE Commun. Sur. Tutor.*, vol. 24, no. 4, pp. 2449–2494, 4th Quart., 2022.

[4] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," *J. Netw. Comput. Appl.*, vol. 169, Nov. 2020, Art. no. 102781.

[5] H. Zhang, L. Feng, X. Liu, K. Long, and G. K. Karagiannidis, "User scheduling and task offloading in multi-tier computing 6G vehicular network," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 446–456, Feb. 2023.

[6] X. Zhang, X. Zhang, and W. Yang, "Joint offloading and resource allocation using deep reinforcement learning in mobile edge computing," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3454–3466, Sep. 2022.

[7] K. Sadatdiynov, L. Cui, L. Zhang, J. Z. Huang, S. Salloum, and M. S. Mahmud, "A review of optimization methods for computation offloading in edge computing networks," *Digit. Commun. Netw.*, vol. 9, pp. 450–461, Mar. 2022.

[8] C. Li, H. Wang, and R. Song, "Intelligent offloading for NOMA-assisted MEC via dual connectivity," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2802–2813, Feb. 2021.

[9] N. Waqar, S. A. Hassan, A. Mahmood, K. Dev, D.-T. Do, and M. Gidlund, "Computation offloading and resource allocation in MEC-enabled integrated aerial-terrestrial vehicular networks: A reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst*, vol. 23, no. 11, pp. 21478–21491, Nov. 2022.

[10] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, Nov. 2020.

[11] X. Li, L. Huang, H. Wang, S. Bi, and Y.-J. A. Zhang, "An integrated optimization-learning framework for online combinatorial computation offloading in MEC networks," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 170–177, Feb. 2022.

[12] H. Guo, J. Liu, J. Ren, and Y. Zhang, "Intelligent task offloading in vehicular edge computing networks," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 126–132, Aug. 2020.

[13] C. Shang, Y. Sun, H. Luo, and M. Guizani, "Computation offloading and resource allocation in NOMA-MEC: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15464–15476, Sep. 2023.

[14] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, Oct. 2018.

[15] H. Zhou, K. Jiang, X. Liu, X. Li, and V. C. Leung, "Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1517–1530, Jan. 2022.

[16] Z. Aghapour, S. Sharifian, and H. Taheri, "Task offloading and resource allocation algorithm based on deep reinforcement learning for distributed AI execution tasks in IoT edge computing environments," *Comput. Netw.*, vol. 223, Mar. 2023, Art. no. 109577.

[17] Y. Ju et al., "Joint secure offloading and resource allocation for vehicular edge computing network: A multi-agent deep reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5555–5569, May 2023.

[18] Z. Xue, C. Liu, C. Liao, G. Han, and Z. Sheng, "Joint service caching and computation offloading scheme based on deep reinforcement learning in vehicular edge computing systems," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6709–6722, May 2023.

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

[20] L. Liu, J. Feng, X. Mu, Q. Pei, D. Lan, and M. Xiao, "Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15513–15526, Dec. 2023.

[21] C. Luo, J. Zhang, X. Cheng, Y. Hong, Z. Chen, and X. Xing, "Computation off-loading in resource-constrained edge computing systems based on deep reinforcement learning," *IEEE Trans. Comput.*, vol. 73, no. 1, pp. 109–122, Jan. 2024.

[22] Y. Wang, H. Wu, R. H. Jhaveri, and Y. Djenouri, "DRL-based URLLC-constraint and energy-efficient task offloading for Internet of Health Things," *IEEE J. Biomed. Health Inform.*, early access, Jul. 24, 2023, doi: 10.1109/JBHI.2023.3297525.

[23] H. Tang, H. Wu, G. Qu, and R. Li, "Double deep Q-network based dynamic framing offloading in vehicular edge computing," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 3, pp. 1297–1310, Jun. 2023.

[24] G. Qu, H. Wu, R. Li, and P. Jiao, "DMRO: A deep meta reinforcement learning-based task offloading framework for edge-cloud computing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3448–3459, Jun. 2021.

[25] W. Qin, H. Chen, L. Wang, Y. Xia, A. Nascita, and A. Pescapè, "MCOTM: Mobility-aware computation offloading and task migration for edge computing in industrial IoT," *Future Gen. Comput. Syst.*, vol. 151, pp. 232–241, Feb. 2024.

[26] J. Bi, Z. Wang, H. Yuan, J. Zhang, and M. Zhou, "Cost-Minimized computation offloading and user association in hybrid cloud and edge computing," *IEEE Internet Things J.*, early access, Jan. 16, 2024, doi: 10.1109/JIOT.2024.3354348.

[27] M. A. Hossain, W. Liu, and N. Ansari, "Computation-efficient offloading and power control for MEC in IoT networks by Meta reinforcement learning," *IEEE Internet Things J.*, early access, Jan. 16, 2024, doi: 10.1109/JIOT.2024.3355023.

[28] N. Lin et al., "Deep reinforcement learning-based computation offloading for servicing dynamic demand in multi-UAV-assisted IoT network," *IEEE Internet Things J.*, early access, Jan. 22, 2024, doi: 10.1109/JIOT.2024.3356725.

[29] S. Bera, S. Misra, and A. V. Vasilakos, "Software-defined networking for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1994–2008, Aug. 2017.

[30] L. I. Barona López, Á. L. Valdivieso Caraguay, L. J. García Villalba, and D. López, "Trends on virtualisation with software defined networking and network function virtualisation," *Inst. Eng. Technol. Netw.*, vol. 4, no. 5, pp. 255–263, Sep. 2015.

[31] D. B. Rawat and S. Reddy, "Recent advances on software defined wireless networking," in *Proc. SoutheastCon*, 2016, pp. 1–8.

[32] Z. Zaidi, V. Friderikos, Z. Yousaf, S. Fletcher, M. Dohler, and H. Aghvami, "Will SDN be part of 5G?" *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3220–3258, 4th Quart., 2018.

[33] S. D. A. Shah, M. A. Gregory, S. Li, R. dos Reis Fontes, and L. Hou, "SDN-based service mobility management in MEC-enabled 5G and beyond vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13425–13442, Aug. 2022.

[34] G. Wu, H. Wang, H. Zhang, Y. Zhao, S. Yu, and S. Shen, "Computation offloading method using stochastic games for software defined network-based multi-agent mobile edge computing," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 17620–17634, Oct. 2023.
[35] K. Yang, X. Wang, Q. He, L. Zhao, Y. Liu, and D. Tarchi, "Knowledge-defined edge computing networks assisted long-term optimization of computation offloading and resource allocation strategy," *IEEE Trans. Wireless Commun.*, early access, Nov. 1, 2023, doi: 10.1109/TWC.2023.3325654.
[36] A. Sarah, G. Nencioni, and M. M. I. Khan, "Resource allocation in multi-access edge computing for 5G-and-beyond networks," *Comput. Netw.*, vol. 227, May 2023, Art. no. 109720.
[37] H. Hu, Q. Wang, R. Q. Hu, and H. Zhu, "Mobility-aware offloading and resource allocation in a MEC-enabled IoT network with energy harvesting," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17541–17556, Dec. 2021.
[38] R. d. R. Fontes and C. E. Rothenberg, "Mininet-Wi-Fi: A platform for hybrid physical-virtual software-defined wireless networking research," in *Proc. ACM SIGCOMM Conf.*, 2016, pp. 607–608.
[39] H. Xiao, J. Huang, Z. Hu, M. Zheng, and K. Li, "Collaborative cloud-edge-end task offloading in MEC-based small cell networks with distributed wireless backhaul," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 4, pp. 4542–4557, Dec. 2023.
[40] Y. Cong, K. Xue, C. Wang, W. Sun, S. Sun, and F. Hu, "Latency-energy joint optimization for task offloading and resource allocation in MEC-assisted vehicular networks," *IEEE Trans. Veh. Tech.*, vol. 72, no. 12, pp. 16369–16381, Dec. 2023.
[41] P. Wang, K. Li, B. Xiao, and K. Li, "Multiobjective optimization for joint task offloading, power assignment, and resource allocation in mobile edge computing," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 11737–11748, Jul. 2021.
[42] H. Ke, J. Wang, L. Deng, Y. Ge, and H. Wang, "Deep reinforcement learning-based adaptive computation offloading for MEC in heterogeneous vehicular networks," *IEEE Trans. Veh. Tech.*, vol. 69, no. 7, pp. 7916–7929, Jul. 2020.
[43] Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu, and F. Tian, "Dynamic task offloading and resource allocation for mobile-edge computing in dense cloud RAN," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3282–3299, Apr. 2020.
[44] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *J. Control Decis.*, vol. 5, no. 1, pp. 42–60, 2018.
[45] H. Badri, T. Bahreini, D. Grosu, and K. Yang, "Energy-aware application placement in mobile edge computing: A stochastic optimization approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 4, pp. 909–922, Nov. 2019.
[46] W. J. Yun et al., "Cooperative multiagent deep reinforcement learning for reliable surveillance via autonomous multi-UAV control," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7086–7096, Oct. 2022.
[47] T. Brys, A. Harutyunyan, P. Vrancx, M. E. Taylor, D. Kudenko, and A. Nowé, "Multi-objectivization of reinforcement learning problems by reward shaping," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2014, pp. 2315–2322.
[48] E. Mustafa et al., "Reinforcement learning for intelligent online computation offloading in wireless powered edge networks," *Cluster Comput.*, vol. 26, no. 2, pp. 1053–1062, 2023.
[49] S. D. A. Shah, M. A. Gregory, S. Li, and R. D. R. Fontes, "SDN enhanced multi-access edge computing (MEC) for E2E mobility and QoS management," *IEEE Access*, vol. 8, pp. 77459–77469, 2020.
[50] H.-C. Chu, C.-E. Wong, W.-M. Cheng, and H.-C. Lai, "User QoS-based optimized handover algorithm for wireless networks," *Sensors*, vol. 23, no. 10, p. 4877, May 2023.
[51] M. A. Hossain and N. Ansari, "Hybrid multiple access for network slicing aware mobile edge computing," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 2910–2921, Sep. 2023.
[52] B. Brik and A. Ksentini, "Toward optimal MEC resource dimensioning for a vehicle collision avoidance system: A deep learning approach," *IEEE Netw.*, vol. 35, no. 3, pp. 74–80, Jun. 2021.
[53] 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 Trans. Commun.*, vol. 67, no. 6, pp. 4132–4150, Jun. 2019.
[54] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, and C. E. Rothenberg, "Mininet-Wi-Fi: Emulating software-defined wireless networks," in *Proc. 11th Int. Conf. Netw. Service Manag. (CNSM)*, 2015, pp. 384–389.
[55] "Multi-access edge computing (MEC); study on MEC support for V2X use cases," Eur. Telecommun. Stand. Inst., Sophia-Antipolis, France, Rep. GR MEC 022, Sep. 2020.
[56] S. E. Li, "Deep reinforcement learning," in *Reinforcement Learning for Sequential Decision and Optimal Control*. Singapore: Springer, 2023, pp. 365–402.
[57] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, Sep. 2018.
[58] "Multi-access edge computing (MEC); federation enablement APIs," Eur. Telecommun. Stand. Inst., Sophia-Antipolis, France, Rep. GS MEC 040, Feb. 2023.
[59] H. M. Do, T. P. Tran, and M. Yoo, "Deep reinforcement learning-based task offloading and resource allocation for industrial IoT in MEC federation system," *IEEE Access*, vol. 11, pp. 83150–83170, 2023.
[60] F. Guo and M. Peng, "Efficient mobility management in mobile edge computing networks: Joint handover and service migration," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 18237–18247, Oct. 2023.
[61] S. D. A. Shah, M. A. Gregory, and S. Li, "Cloud-native network slicing using software defined networking based multi-access edge computing: A survey," *IEEE Access*, vol. 9, pp. 10903–10924, 2021.
[62] H.-W. Kao and E. H.-K. Wu, "QoE sustainability on 5G and beyond 5G networks," *IEEE Wireless Commun.*, vol. 30, no. 1, pp. 118–125, Feb. 2023.
[63] S. D. A. Shah, M. A. Gregory, and S. Li, "Toward network slicing enabled edge computing: A cloud-native approach for slice mobility," *IEEE Internet Things J.*, vol. 11, no. 2, pp. 2684–2700, Jan. 2024.
[64] H. V. Poor and R. F. Schaefer, "Wireless physical layer security," *Proc. Nat. Acad. Sci.*, vol. 114, no. 1, pp. 19–26, 2017.
[65] F. Bouhafs, F. den Hartog, A. Raschella, M. Mackay, Q. Shi, and S. Sinanovic, "Realizing physical layer security in large wireless networks using spectrum programmability," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2020, pp. 1–6.

SYED DANIAL ALI SHAH received the master's degree in engineering from Incheon National University (INU), South Korea, in 2018, and the Ph.D. degree in engineering from the Royal Melbourne Institute of Technology (RMIT) University, Melbourne, Australia, in 2022. He is a Postdoctoral Research Associate with the University of New South Wales, Australia Defence Force Academy. He was a Sessional Academic Staff with RMIT University from 2020 to 2022. He also worked as a Research Assistant with INU from 2016 to 2018. His research interests include 5G and beyond wireless networks, IoT, SDN/NFV, and vehicular networks. He has several years of publication experience and serves as a review and guest editorial panel member for various reputable international journals and conferences.

MARK A. GREGORY (Senior Member, IEEE) received the Ph.D. degree from RMIT University, Melbourne, Australia, in 2008, where he is an Associate Professor with the School of Engineering. His research interests include telecommunications, network design, public policy, and technical risk. In 2009, he received an Australian Learning and Teaching Council Citation for an outstanding contribution to teaching and learning. He is the Managing Editor of *International Journal of Information, Communication Technology and Applications* and the General Co-Chair of ITNAC. He is a Fellow of the Institute of Engineers Australia.

FAYCAL BOUHAFS (Senior Member, IEEE) received the Ph.D. degree in computer science from Liverpool John Moores University, U.K. He is a Senior Lecturer with the School of Systems and Computing, University of New South Wales Canberra, Australia Defence Force Academy. Before, he served as an Assistant Professor of Wireless Communications and Networking. Since 2015, he acted as the Technical Lead for H2020 Project Wi-5. His research interests revolve around spectrum congestion, radio resource management, and beyond 5G communications and architecture.

FRANK DEN HARTOG (Senior Member, IEEE) received the M.Sc. degree from the Eindhoven University of Technology, and the Ph.D. degree in physics from Leiden University. He is a Professor with the Network Engineering and Cyber Security Capability, Faculty of Science and Technology, University of Canberra, where he holds the Cisco Research Chair of Critical Infrastructure. He is also an Adjunct Fellow with the University of New South Wales Canberra, Australia Defence Force Academy. He specializes in complex wireless networked systems research and analysis, with a particular interest in the Internet of Things (IoT) and cyber–physical systems. From 2003 to 2016, he was a Senior Scientist with the Netherlands Organisation for Applied Scientific Research (TNO) and studied Future Internet architectures and heterogeneous consumer networking. Before joining TNO, he worked for the Dutch incumbent telecommunications operator KPN, where he pioneered the home networking and IoT research area.