Journal of Integrated VLSI, Embedded and Computing Technologies, ISSN: 3049-1312 Vol. 3, No. 1, 2026 (pp. 55-63)

RESEARCH ARTICLE

ECEJOURNALS.IN

Embedded and Cloud Computing Integration for Smart Mobile Learning Applications Using Deep Reinforcement Learning

Prerna Dusi*

Assistant Professor, Department of Information Technology, Kalinga University, Raipur, India.

KEYWORDS:

Embedded systems;
Deep reinforcement learning;
Cloud-edge computing;
Smart mobile learning;
Computation offloading;
Energy efficiency

ARTICLE HISTORY:

Submitted : 13.08.2025
Revised : 26.09.2025
Accepted : 08.11.2025

https://doi.org/10.31838/JIVCT/03.01.08

Abstract

Smart mobile learning applications are becoming increasingly dependent on embedded devices which run with harsh limitations with regards to computation power, energy, and network variability. Although cloud computing is capable of providing scalable processing, storage capacities, the integration of embedded platforms to cloud infrastructure is still a big problem especially when it comes to use in resource and intensive and latency-sensitive learning applications. In this paper, the author proposes a smart embedded-cloud computing system of smart mobile learning applications relying on Deep Reinforcement Learning (DRL). This proposed solution defines the computation offloading problem and resource allocation issue as a sequential decision-making problem, according to which the DRL agent is a dynamic, in charge of deciding whether the learning tasks need to be performed locally on the embedded system or offloaded through edge servers or cloud servers. State space is a combination of device status, network status and task properties and the reward mechanism serves to manage the combination of execution latency, energy use and quality of service. Numerous experiments performed in a simulated mobile learning setting prove that the suggested DRL-based mechanism is more favourable, compared to traditional local execution, offloading to the cloud, and heuristic-based approaches, in terms of lower latency and energy consumption. The findings demonstrate the scalability and flexibility of DRA on dynamic mobile systems, so that the suggested model is a prospective indicator to the future-generation smart mobile learning systems combining embedded and cloud computing technologies.

Author's e-mail: ku.PrernaDusi@kalingauniversity.ac.in

How to cite this article: Dusi P. Embedded and Cloud Computing Integration for Smart Mobile Learning Applications Using Deep Reinforcement Learning. Journal of Integrated VLSI, Embedded and Computing Technologies, Vol. 3, No. 1, 2026 (pp. 55-63).

Introduction

The way contemporary education has been changing is due to the high increase of the smart mobile learning systems which have ensured that learners can access instructional materials anytime and anywhere on their mobile and embedded devices. Multimedia classroom tutorials, live evaluation, learning analytics, and adaptive content delivery are applications that are characterized by high computation intensity and low response times, which are necessary to achieve reasonable Quality of Experience (QoE). Nevertheless, these demands are very challenging to perform through mobile devices only, which are limited by restricted processing power, part-memory, battery and storage. Consequently, local execution of computation-intensive learning tasks can be consumptively energy intensive and response times may be very slow, which are extremely detrimental to the efficiency of a system and user experience time.[8, 10]

To overcome such shortcomings, cloud computing and edge computing paradigm have become viable solutions as they provide scaling computing and storage as compared to isolated embedded systems. Mobile learning applications can be implemented to spread computation to the edge servers or remote data centres in a cloud to create a drastic reduction in the workload on the device, as well as enhance responsiveness and performance. According to current work, edge cloud collaboration can successfully deliver latency-sensitive learning services by moving computation to the end users as well as utilising cloud scalability to execute intricate processing operations.[3, 7] However, the nature of network conditions, the movement of users and the heterogeneous task demands often makes the use of static or heuristic-based offloading strategies fail to adapt to the dynamic network conditions resulting in an inefficient resource utilisation.[1, 5]

The offloading of intelligent computation and control of the resources have thus emerged as critical research issues in mobile edge computing settings. The stochastic and time-varying characteristics of mobile networks are not easy to manage with the traditional optimization based techniques. In that regard, Deep Reinforcement Learning (DRL) has been receiving growing interest because of its capability to represent computation offloading as a row-by-row decision-making process and optimise the optimal policies through the environment interaction. The approaches based on DRL have proven to be more robust in dynamically balancing between latency, energy consumption, and the system throughput at different conditions of the networks, and devices.[2, 9, 12, 14, 15] The particular features of DRL ensure that it is especially relevant in smart mobile learning applications where a workload on tasks, network bandwidth, and device conditions are constantly altered.

Based on these observations, a paper has been proposed on the proposal of DRC based embedded-cloud computing integration framework with smart mobile learning applications. The suggested framework allows to make intelligent decisions about the offloading of intelligent tasks by taking into account jointly the state of devices, the state of the network, and the specifics of the task and achieve maximum impacts on execution latency, energy consumption, and Quality of Service (QoS). The key contributions of the work are three-fold, including (i) designing a DRL-based embedded-cloud integration architecture that suits the context of mobile learning systems; (ii) creating an intelligent and dynamic task offloading strategy, and (iii) a more rigorous performance-evaluation that shows that there are significant performance gains over traditional local execution and heuristic offloading techniques.[4, 11, 13]

Related Work

Early studies relating to smart mobile learning systems were mainly on the utilisation of embedded and mobile devices as a means of providing educational content in digital format with main focus being on portability and accessibility. Nonetheless, in reality, due to low-level computational power, energy requirements, and storage, standalone embedded systems can hardly afford high-end learning capabilities, including real-time analytics, multimedia processing, and adaptive customization. These limitations have been observed by some researchers and have shown that a mobile learning architecture that is strictly device centric tends to lead to high latencies and a quick depletion of a battery thus compromising the user experience and reliability of the system.[8, 10]

Mobile learning architecture has extensively manifested the use of cloud computing and lately both edge computing paradigms to address these limitations. Mobile learning systems based on the cloud sale allow computational intensive activities to be off loaded to remote servers which is scalable and centrally administered. Edge computing also enhances this paradigm through having more computing resources closer to the end users, hence, minimising communication latency and enhance responsiveness in learning applications with high delays.[3,7] It has been demonstrated that hybrid edge cloud architecture can be effective in balancing performance and scalability and it can be used in interactive and real time educational services.[11] However, the majority of the current architectures are based on predetermined or fixed resource allocation mechanisms, which restrict their dynamism in the dynamic mobile scenario.

The major approaches to traditional task offloading and task scheduling in mobile edge computing have been either heuristic based, mathematic optimization like policies, or rule based decision processes. These are greedy, threshold-based, and convex optimization methods that are designed to ensure that the latency or energy is minimised. Although those techniques are theoretically optimal, given some fixed assumptions, they are not usually effective when it comes to practical situations, which involve changing network properties, user mobility and unbalanced task demands.[1, 5] In order to circumvent these issues, machine learning-driven solutions like supervised learning models, evolutionary optimization algorithms have been pursued but these models generally demand large labelled data sets, or do not keep up with changes in the environment.[3, 4]

Reinforcement learning and deep reinforcement learning (DRL) has become an influential technique of intelligent offloading of computations in mobile and edge devices in the last few years. The DRL-based techniques view the offloading choices as a sequential decision-making task allowing systems to acquire the best possible policy by continuously interacting with the environment. As a number of studies have proved, DRA makes considerably higher optimisation of latency, energy consumption,

Table 1: Comparison of Related Work on Mobile Learning and Task Offloading

Ref.	Computing Paradigm	Methodology	Application Focus	Key Limitations
[8]	Cloud-based	Edge-assisted architecture	Mobile learning	Limited adaptability
[10]	Edge-Cloud	ML-enabled education systems	Smart education	Static decision models
[1]	Edge computing	DRL-based offloading	MEC systems	Not learning-specific
[5]	Edge computing	Mobility-aware DRL	Mobile networks	High model complexity
[12]	Edge-Cloud	DRL + service caching	MEC	Generic application scope
[14]	Edge computing	DRL for low latency	MEC	Ignores learning QoE
Proposed	Embedded-Edge-Cloud	DRL-based adaptive offloading	Smart mobile learning	Addresses identified gaps

and Quality of Service during the dynamic environment, compared to traditional methods as well as to heuristic and uninformed methods.[2, 9, 12, 14, 15] In spite of these improvements, currently used DRL-based systems are largely adapted to generic mobile edge computing applications and do not give much attention to the particular needs of smart mobile learning applications, including learning QoE, embedded device specifications, and education workload specifications. The breach is the reason a DRL-based embedded-cloud integration model which is specific to smart mobile learning systems is developed.

System Architecture and Problem Formulation

Overall Embedded-Cloud Learning Architecture

The proposed system will entail an integrated embedded, edge and cloud architecture to serve smart learning mobile applications within dynamic operating environments. The architecture, as depicted in (Fig. 1), has three main layers, which include the mobile learner device layer, the edge/cloud computing layer, and the network communication layer. Mobile learner device is an embedded platform meaning a smartphone, tablet, wearable learning device capable of interaction with the user, a small scale computation and contextual information sensing. These devices perform tasks that are related to learning like contents rendering, tracking user behaviour, and initial data pre-processes and allows them to work under stringent energy and computational constraints.

The cloud computing layer and edge offers computational and storage resources that can be scaled to solve learning tasks that consume resources. The edge servers are deployed in geographic proximity to learners to support operations sensitive to latency, like real-time assessment evaluation and interactive content processing, whereas cloud servers are used to perform computation-intensive operations, such as learning analytics, learners in training their models, and massive storage of data. The hierarchical structure allows a workload allocation to be effective and balances between low latency service provision and high computing power. The architecture enables run time linking between embedded devices, edge nodes and cloud servers in accordance with application demands and state of systems as illustrated in (Fig. 1).

The network communication model links the mobile devices to the edge servers and cloud servers via heterogeneous wireless connectivity systems which include Wi-Fi, 4G and 5G networks. Network performance is a time-varying phenomenon caused by the mobility of users, change in channel, and fluctuation of traffic load which has a great influence on the performance of tasks. There are bidirectional flow of Learning content, computation tasks as well as control information through the network. In particular, uncooked/semi cooked learning data is uploaded by the embedded device to edge/cloud, and refined outputs, feedbacks and learning suggestions are sent back to the learner. The management of the flow of data among these layers is needed to ensure reasonable Quality of Service (QoS) in smart mobile learning platforms.

Computation Offloading Problem Definition

Computation offloading is formulated as a decision-making problem in the proposed framework, where every learning task that is created by a mobile device can be either performed locally, or offloaded to an edge server, or also transferred to a cloud. Each job is defined in terms of CPU cycle demand, size of input data, size of output data and deadline. The embedded gadget is limited with the capacity of the battery, processing and memory resources where the edge and cloud servers offer some heterogeneous but limited computational resources that are shared by multiple users. The goal of the offloading strategy is to come up with an best implementation decision that reduces the end to end execution duration as well as the use of energy besides making sure that the learning service delivery is reliable and the QoS of learning is maintained. The current multi-objective optimization problem under the dynamic system conditions is the core of the solution provided by Deep Reinforcement Learning that will be discussed further in the following sections, and the general system interactions and the decision flow can be summarised in (Fig. 1).

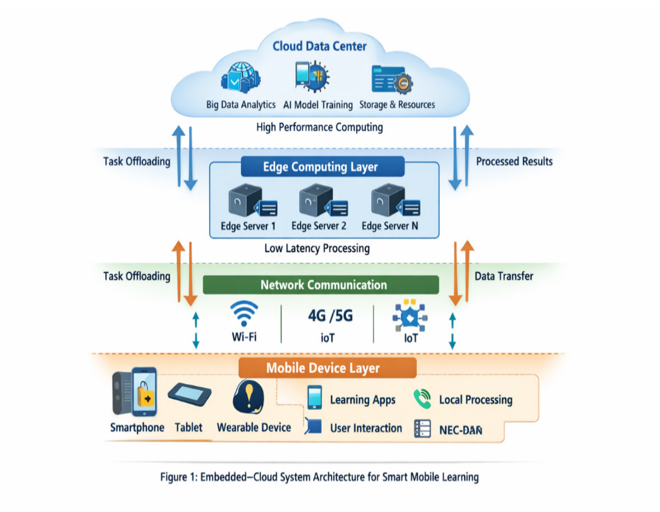

Fig.1: Embedded-Cloud System Architecture Diagram.

Deep Reinforcement Learning-Based Offloading Model

DRL Framework Design

In order to facilitate dynamic and intelligent offloading computation into smart mobile learning settings, the proposed structure presented develops the offloading decisioning process as Deep Reinforcement Learning (DRL) problem. As (Fig. 2) demonstrates, the embedded device inside of the mobile is the agent of the DRL, and it continuously communicates with the environment that is the network conditions and the resources of the edge and cloud computing. The agent takes a note of the system state at every epoch of decision making and chooses a suitable offloading action in order to optimise the performance of learning services taking into consideration the embedded device constraints. Such agent environment interaction enables the system to dynamically respond to the change in user behaviour, task workload and network variability.

The state space is meant to fully represent the environment of operation of the mobile learning system. It contains some of the most important parameters, which include the remaining battery level of the embedded device, the available network bandwidth, the data size of the input of the tasks, the computational demand (CPU cycles), and the device CPU load at the moment. All these state variables are a representation of the internal state of the device of the learner as well as an external computing and communication world. Action space is characterised by three potential execution choices, namely, local execution on the embedded device, offloading to an edge server, or offloading to a remote cloud server. This discrete action design is an actual execution choice in real embedded-cloud learning systems as demonstrated in (Fig. 2).

Reward mechanism is important in allowing the DRA agent to be made to make the best offloading decisions. The proposed model constitutes a weighted sum of the negative execution latency, energy consumption and Quality of Service (QoS) degradation as the reward. The reward functionality will provide the necessary balance between system efficiency and user experience by punishing excessive delay and energy consumption and rewarding constancy in the delivery of learning services. This multi-objective reward objective allows the DRL agent to engage in learning policies that optimise the time and battery consumption of end-to-end task execution under dynamic scenarios and the policy minimising acceptable learning QoS that is in line with the system goals as illustrated in (Fig. 2).

4.2 DRL Algorithm Selection and Training Strategy

Value-based and policy-gradient DRL models that can be considered as evidence DQN (Deep Q-Networks), Proximal Policy Optimization (PPO), and Actor-Critic due to discrete nature of the offloading decision space and dynamics of mobile environments respectively. Actor-critic and PPO algorithms are the most promising in embedded-cloud applications where the stable convergence pattern and sensitivity to continuous state space and reduced variance are constant. To balance between exploration and exploitation in training, a ε-greedy or entropy-regularised exploration strategy is used. Before the DRR model is deployed in the real world, it must undergo offline training in a simulated mobile learning environment to learn faster and minimise overheads in the real world. The trained model is further executed on the embedded device without the need to download additional inference mechanisms, and adapts online to real-time learning conditions at low computation cost because conceptually depicted in (Fig. 2).

Smart Mobile Learning Application Model

The smart mobile learning applications usually contain several functional modules which help to deliver the

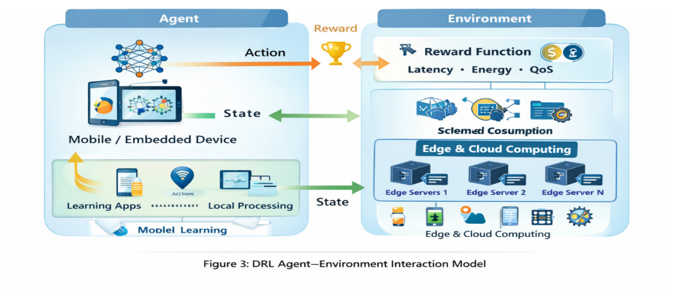

Fig. 2: Deep Reinforcement Learning (DRL) Agent-Environment Interaction Model for Embedded-Cloud Based Smart Mobile Learning Systems.

content, interact with learners, evaluate them, and give them learning analytics. The learning application process within the proposed framework starts with educational content delivery i.e. video lectures, interactive simulation, and digital reading content to an embedded device on the learner. The interactions of learners, such as quiz answers, completing activities, and behavioural feedback, are also constantly gathered and processed to assist in individualised learning experiences. The results of assessment and usage data are sent to the backend systems to be evaluated and generate feedback to permit the real-time and adaptive learning services to be provided as conceptually shown in (Fig. 3).

The computational problems associated with these learning applications have a wide range of resource needs and nature of computational activities. Lightweight jobs e.g. user interface and simple display of content can be effectively addressed by local processing using the embedded device. Conversely, tasks that are computation-intensive, such as video processing and real-time learning analytics in high-resolution, and creation of a personalized suggestions, demand large processing capabilities and memory. Such functions can result in high latencies and high energy usage (when implemented locally) and would therefore be good candidates to be offloaded to edge/cloud servers. The dissimilarity of these tasks makes it necessary to have a smart architecture of the dynamic execution decision-making mechanism in the learning system as illustrated in (Fig. 3).

To overcome this demand, the suggested smart mobile learning framework will pair learning-related computational tasks with Deep Reinforcement Learning (DRL)-based offloading choices. Depending on the state of the system, such as the battery of the device, the bandwidth of the network, task size, and urgency of its execution, the DRL agent decides whether the task could be executed locally, offloaded to an edge server, or processed in cloud. Interactive assessment tasks, which are latency-sensitive, are exposed to the edge with other resource-demanding analytics and model training tasks being offloaded to the cloud. Such adaptive task mapping will allow effective resource use and facilitated learning processes that are smooth and responsive as shown in (Fig. 3).

The proposed application model will have a smooth integration with the existing Learning Management Systems (LMS) (like Moodle or cloud-based educational platforms). The exchange of learning content, learner and performance measure between mobile learning application and LMS backend are conducted by the use of standardised application programming interfaces (APIs). This deployment enables institutions to utilise the suggested DRL based embedded-cloud platform without altering the current learning systems. The proposed model will lead to higher scalability, adaptability, and effectiveness of intelligent mobile learning systems by integrating intelligent offloading and LMS interoperability as summarised in (Fig. 3).

Experimental Setup and Implementation

The experimental analysis of the proposed DRL-based embedded-cloud offloading framework is executed in the form of the representative mobile learning system that consists of the mobile embedded devices, edge servers,

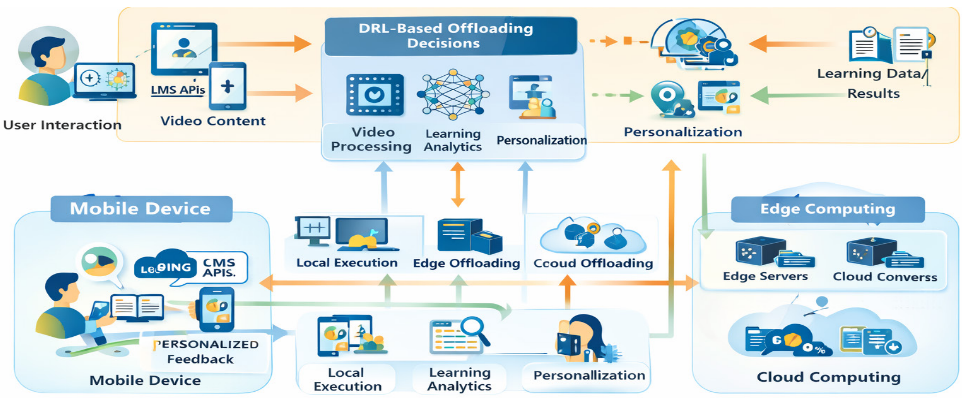

Fig.3: Workflow of Smart Mobile Learning Application.

and cloud infrastructure. The mobile learner device is designed upon the model of embedded platform that has few computational and energy resources, which is based on the nature of modern smartphones and tablets in application in mobile learning. Some of the important parameters are CPU frequency, available memory, battery capacity, and energy consumption rates in which to calculate and communicate. The specifications of these devices are presented in the (Table 2) that gives the base configuration obtained in the experiments. The cloud infrastructure and edge is constructed in the manner so as to represent a realistic hierarchical computing environment. Edge servers are configured to have moderate computing power and are deployed near mobile users to enable learning tasks which have latency constraint whereas cloud servers have high-performance computing and extensive storage so that computation intensive analytics and model training can occur. There are several edge servers, which are assumed to contend resources among users. The network connexions between mobile devices and edge servers and the cloud are modelled using heterogeneous values of bandwidth and latency relating to Wi-Fi and 4G/5G connexions. The specifics of the infrastructure parameters such as server processing capabilities and network properties are presented in (Table 2).

An evaluation methodology using simulation is used to train and test the DRL model in the case of dynamic situations. The simulated environment supports the mobility of the user, the undulating network bandwidth, and the intermittent workloads of the user in smart learning scenarios of the mobile learning process. Learning tasks are created in accordance with realistic application profiles such as video content processing, assessment evaluation and learning analytics. Workload traces and synthetic datasets are used to simulate the various learning behaviours and system loads. The DRL agent is first trained in offline in this simulated world, the agent is then tested in unseen sequences of tasks to determine its generalisation and flexibility. In order to understand the performance of the system fully, there are several assessment metrics that are taken into account. Task latency is calculated as the duration of execution of the tasks between the occurrence of the task and the delivery of the results. The energy consumption is computed depending on the calculation and communication energy expenses spent by the mobile device. System throughput represents the number of learning operations passively handled in one time interval, whereby User Quality of Experience (QoE) depends on a composite measure consisting of a latency, service reliability, and task success. Such performance indicators offer a global assessment of the suggested structure and the foundation of comparative analysis that will be offered in the next Results and Discussion section, where the definitions of metrics and measurement parameters were summed up in (Table 2).

Table 2: Simulation Parameters and System Configuration

Parameter	Configuration
Mobile Device CPU	Quad-core, 2.0 GHz
Mobile Device RAM	4 GB
Battery Capacity	3500 mAh
Edge Server Capacity	8–16 cores, 16 GB RAM
Cloud Server Capacity	32+ cores, 64 GB RAM
Network Technology	Wi-Fi / 4G / 5G
Bandwidth Range	5-100 Mbps
Task Data Size	1-20 MB
Task Deadline	0.5-2.5 s
DRL Algorithm	DQN / PPO / Actor-Critic
Discount Factor (γ)	0.98
Training Episodes	2000

Results and Discussion

The validity of the proposed DRL-based embedded-cloud offloading framework can also be measured through its comparison with three base methods: local execution alone, cloud-only offloading and heuristic-based offloading. The quantitative performance comparison done in (Table 3) is based on task latency, energy consumption, throughput and user Quality of Experience (QoE). The findings indicate that a local execution has a high power consumption and a large latency because embedded devices have inadequate resources and cloud-only offloading has undue dependence on varying network factors. The methods that are based on heuristics provide average enhancements but are not adaptable. Conversely, the suggested DRA-oriented model has significantly lower latency and power usage, and better QoE in dynamic mobile learning systems, which validates the approach of smart decision-making to dynamic mobile learning.

In order to further study the learning effectiveness of the proposed model, convergence behaviour of the DRL agent is studied. As the cumulative reward is increasing until training stabilises, as demonstrated in (Fig. 4), there is effective learning in the policy and convergence occurs at the adequate number of episodes. The first stage of exploration has variations in its performance because the agent interacts between exploration and exploitation. In the long term, the agent is informed about the most effective offloading behaviour that maximises long-term payoffs through simultaneous optimization of latency, energy consumption, and the quality of services. Such convergence behaviour illustrates the stability and strength of the DRM model with respect to applying it to the issue of offloading computation to the embedded-cloud learning systems.

Network variability is also important to control how mobile learning applications perform. The experiments with dissimilar bandwidth and latency states show that the DRL-based framework is more flexible in response to network changes in comparison to the control techniques and strategies. In low-bandwidth or high-latency conditions, the DRL agent simply adapts local execution or edge offloading in order to preserve a reasonable learning performance, whilst cloud-only and heuristics solutions severely degrade in performance. The resilience and stable QoE delivery are clear in the trends of the performance when the DRA policy is varied to different network conditions as the performance stays at a relatively high level across different network conditions (Fig. 4).

The high scale and flexibility a re essential in real-life applications of smart mobile learning systems that demand implementation at the large scale. It can be denoted by the experimental results that the suggested framework is efficient to scale with the rise in the rate of task arrival and the amount of users using the cloud resources that are distributed. In contrast to the offloading solutions addressed by adhering to a fixed policy, the proposed method based on DRL implements a

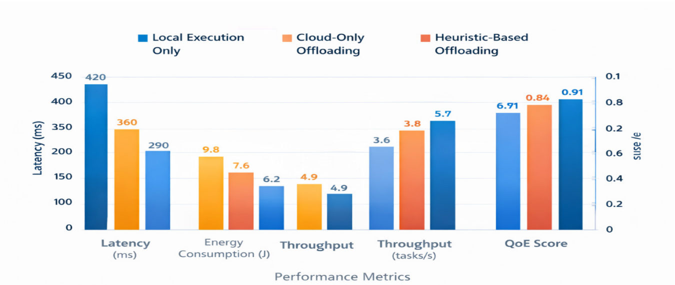

Fig.4: Performance Comparison of Different Computation Offloading Strategies in Smart Mobile Learning Systems.

dynamic reallocation of the tasks depending on the real-time system conditions thus avoiding the performance bottlenecks and overconsumption of resources. As the discussion of the results in (Table 3) and (Fig. 4) confirms, the proposed approach delivers the scaled-up and dynamic solution that can ensure the next-generation smart mobile learning applications in the heterogeneous, dynamic, computing environments.

Practical Implications

The suggested embedded-cloud offloading framework is based on DRL, which exhibits a high potential to be deployed in a real-life smart mobile learning framework. The framework is able to be incorporated into existing learning systems without the need to make specific hardware changes by operating on standard mobile and embedded platforms and using existing edge and cloud infrastructures. The trained DRL model can be deployed on-device through lightweight inference requirements, and this allows the model to make decisions with low overhead costs, and thus such an approach can be used in the real-life applications of resource-constrained mobile learning settings.

In terms of the learner, the quality of the learning experience is greatly improved as proposed system ensures that there is less latency in performing the tasks and is not subjected to any form of dead air during interaction with learning applications. Activities like interactive quizzes, real-time feedback and multimedia content rendering are latency sensitive tasks that can be optimally offloaded through smart decisions when there is a change in network and device characteristics. This has led to increased speed, reliability and continuous provision of learning services to students, which in totality have resulted to increased engagement and satisfaction.

Schools will also have a lot to derive through the implementation of the proposed framework. Distribution of intelligent tasks amongst embedded devices, edge server and cloud resources results in a better use of the computational infrastructure to save on operational cost and unnecessary overprovisioning of cloud facilities.

Table 3: Performance Comparison of Different Offloading Strategies

Offloading Strategy	Latency (ms)	Energy Consumption (J)	Throughput (tasks/s)	QoE Score
Local Execution Only	420	9.8	3.1	0.71
Cloud-Only Offloading	360	7.6	3.8	0.78
Heuristic-Based Offloading	290	6.2	4.5	0.84
Proposed DRL-Based Offloading	**210**	**4.9**	**5.7**	**0.91**

The adaptability of the DRA-guided methodology allows the institutions to dynamically deal with the computational loads in high traffic seasons to sustain the service qualitatively and effectively to efficiently allocate resources.

The proposed framework can provide a versatile and adaptive control mechanism to the management of dynamic mobile learning settings by system designers and software developers. The decision model which was developed using DRL can be scaled or retrained to support new application needs, network technologies or hardware platforms. In addition, the platform is entirely integrated with the current embedded systems, edge computing, and cloud platforms with uniform communication protocols and APIs. The compatibility guarantees a smooth integration with the existing infrastructures as it offered a scalable platform on which intelligent mobile learning applications can be built in the future.

Limitations and Future Work

Although the suggested DRL-based embedded-cloud offloading framework has shown a great performance, diverse limitations should be admitted. The first of these is the training overhead and convergence time of deep reinforcement learning models. Whereas offline training helps to reduce real-time computational costs, stable convergence can be very costly (in terms of training episodes and search together with exploration), especially in a highly dynamic environment. This complexity of training may become a difficulty where the system conditions or application requirements change very often and model retraining or fine tuning needs to be done.

The other critical weakness of the existing framework is scalability. Although experimental measures indicate that the system can perform effectively when used by moderate user numbers, and when used to perform moderate amounts of tasks, there is an increase in computational and communication overheads at the edge and at the cloud computing layer when scaling the system to support large-scale deployments containing thousands of users and tasks simultaneously. Contention of resources between users may affect optimality of decisions and responsiveness in the system. The next-generation work should explore more hierarchical or distributed learning structures that can be more effectively applied to a huge multi-user system and sustain a low latency and energy consumption.

Other vulnerable issues include security and privacy in smart mobile learning systems, and especially when sensitive learner information is offloaded to an edge server or cloud server. Data confidentiality, integrity, and access control it is not explicitly mentioned in the current framework, which assumes a trusted execution environment. The privacy controlling mechanisms, secure mode of communications, and trusted execution technologies need to be included in future studies in order to safeguard the information of learners in the offloading and processing phases. These issues are critical to be addressed in the real world to be adopted in the educational settings where the data protection regulations need to be fulfilled.

A number of potential lines of research may be followed further on the proposed framework. Federated Deep reinforcement Learning can be investigated so that to allow joint learning of policies on different devices without transfer of raw data and to improve privacy. The offloading decisions can be coordinated among the users and edge servers using multi-agent DRA in a distributed fashion. Also, when introducing edge-cloud-IoT co-optimization tactics, the system can be advanced even further through the introduction of the adaptability of contextual information obtained by the diverse devices of the IoT system. These extensions present good prospects to increase scalability, reliability and smarts of the future smart mobile learning systems.

Conclusion

In this article, the authors provided a Deep Reinforcement Learning-based embedded-cloud computing design of smart mobile learning applications to overcome the problem of computational, energy, and latency constraints of mobile and embedded devices working in isolation. The proposed framework provided by carrying out the intelligent planning of the execution of tasks revolving around embedded devices, edge servers, and cloud resources enables adaptive and efficient computation offloading in the dynamic conditions of the network and workload. Experimental evidence showed that there were major better outcomes than local execution, cloud-only offloading, and heuristic-based solutions, resulting in a lower task latency, energy consumption, system throughput, and user Quality of Experience improvement. The results indicate the usefulness of DRL in synthesising the dynamics of complex systems and optimal offloading decisions under real-time conditions. On the whole, this paper highlights the opportunity of DRL-enabled embedded-cloud integration as a major enabler of the next-generation intelligent mobile learning platforms, a scalable and adaptable platform to support the educational systems in the future, which embrace edge intelligence, cloud scalability, and adaptive learning technologies.

References

1. Ale, L., Zhang, N., Fang, X., Chen, X., Wu, S., & Li, L. (2021). Delay-aware and energy-efficient computation offloading in mobile-edge computing using deep reinforcement learning. *IEEE Transactions on Cognitive Communications and Networking*, *7*(3), 881-892.
2. Chen, J., Jin, L., Yao, R., & Zhang, H. (2024). Deep reinforcement learning method for task offloading in mobile edge computing networks based on parallel exploration with asynchronous training. *Mobile Networks and Applications*, 1-19.
3. Grzesik, P., & Mrozek, D. (2024). Combining machine learning and edge computing: Opportunities, challenges, platforms, frameworks, and use cases. *Electronics*, *13*(3), 640.
4. Kumari, S., & Gupta, D. (2025). Task offloading based on deep reinforcement learning in edge computing network. *The Journal of Supercomputing*, *81*(16), 1-25.
5. Li, Y., Liu, Y., Liu, X., Xie, Y., Han, G., Qiu, T., & Wu, P. (2024). Mobility-Aware Partial Task Offloading and Resource Allocation Based on Deep Reinforcement Learning for Mobile Edge Computing. *IEEE Transactions on Vehicular Technology*.
6. Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, *30*.
7. Ma, C., Li, X., Huang, B., Li, G., & Li, F. (2024). Personalized client-edge-cloud hierarchical federated learning in mobile edge computing. *Journal of Cloud Computing*, *13*(1), 161.
8. Mohiuddin, K., Fatima, H., Khan, M. A., Khaleel, M. A., Nasr, O. A., & Shahwar, S. (2022). Mobile learning evolution and emerging computing paradigms: An edge-based cloud architecture for reduced latencies and quick response time. *Array*, *16*, 100259.
9. Nieto, G., de la Iglesia, I., Lopez-Novoa, U., & Perfecto, C. *Deep Reinforcement Learning Techniques for Dynamic Task Offloading in the 5G Edge-Cloud Continuum*, *Journal of Cloud Computing* **13**, 94 (2024).
10. Qiu, S. (2024). Improving performance of smart education systems by integrating machine learning on edge devices and cloud in educational institutions. *Journal of Grid Computing*, *22*(1), 41.
11. Semerikov, S. O., Striuk, A. M., Pinchuk, O. P., Vakaliuk, T. A., Kanevska, O. B., & Ostroushko, O. A. (2025). Learning under pressure: game-based, AI-driven, and crisis-responsive pedagogies in focus of the 12th Workshop on Cloud Technologies in Education. In *CEUR Workshop Proceedings* (pp. 1-14).
12. Xie, M., Ye, J., Zhang, G., & Ni, X. (2024). Deep reinforcement learning-based computation offloading and distributed edge service caching for mobile edge computing. *Computer Networks*, *250*, 110564.
13. Xu, R., & Zhang, T. (2024). Combining Cloud and Mobile Computing for Machine Learning. *arXiv preprint arXiv:2402.04880*.
14. Yang, W., Liu, Z., Liu, X., & Ma, Y. (2024). Deep reinforcement learning-based low-latency task offloading for mobile-edge computing networks. *Applied Soft Computing*, *166*, 112164.
15. Zhou, H., Jiang, K., Liu, X., Li, X., & Leung, V. C. (2021). Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing. *IEEE Internet of Things Journal*, *9*(2), 1517-1530.