PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY FERHAT ABBAS SÉTIF 1

Faculty of Sciences
Department of Computer Science

MASTER'S THESIS



Integrating AI into Mobile Cloud Computing Task Offloading

Supervised By:

Dr. Annane Boubakeur

Presented By:

Mr. Rouila Abdennour
Mr. Akarab Mohamed Dhia Eddine

Thesis defended on June 2026, before the jury:

Dr. Annane Boubakeur	(Supervisor)
Dr. Fateh Seghir	(Examiner)
Dr. Abderrahim Lakehal	(Examiner)

2025–2026

Acknowledgements

We begin by expressing our heartfelt gratitude to Allah Almighty, whose endless mercy, guidance, and blessings have supported us throughout this journey. His wisdom has been our source of strength, patience, and perseverance in completing this thesis.

We are profoundly thankful to our supervisor, Dr. Annane Boubakeur, for his exceptional mentorship, continuous encouragement, and invaluable guidance. His expert advice, patience, and constructive remarks greatly contributed to the direction and quality of our research.

We would also like to extend our sincere thanks to the jury members for their time, valuable observations, and thoughtful evaluation of our work. Their participation in this academic process is deeply appreciated.

Our appreciation also goes to our friends and fellow students, whose companionship, motivation, and shared experiences made this academic journey more meaningful and memorable.

Finally, we would like to express our deepest gratitude to our families and close ones for their endless patience, understanding, prayers, and unconditional support. Their belief in us has always been a source of inspiration and strength throughout this journey.

Abstract

This thesis explores the application of Artificial Intelligence (AI) to task offloading in Mobile Cloud Computing (MCC) environments. The study addresses the challenge of efficiently allocating computational tasks among local devices, edge servers, and cloud resources while minimizing execution delay and energy consumption. To achieve this objective, the offloading problem is formulated as a Reinforcement Learning (RL) task, and two Deep Reinforcement Learning (DRL) algorithms, Deep Q-Network (DQN) and Double Deep Q-Network (DDQN), are employed to learn adaptive offloading policies based on real-time system conditions. A closed-loop intelligent framework is proposed, implemented in Python, and evaluated through simulation experiments. The results demonstrate that the proposed DRL-based approach effectively improves task allocation decisions, reduces latency and energy consumption, and achieves stable convergence during training. These findings highlight the potential of AI-driven, context-aware offloading strategies for enhancing the performance and efficiency of next-generation MCC systems.

Keywords: Mobile Cloud Computing (MCC), Task Offloading, Deep Reinforcement Learning (DRL), Deep Q-Network (DQN), Double Deep Q-Network (DDQN), Artificial Intelligence, Energy Efficiency.

Résumé

Ce mémoire explore l'application de l'intelligence artificielle (IA) au déchargement de tâches dans les environnements de Mobile Cloud Computing (MCC). L'étude aborde le défi de répartir efficacement les tâches de calcul entre les appareils locaux, les serveurs de périphérie (edge) et les ressources du cloud, tout en minimisant le délai d'exécution et la consommation d'énergie. Pour atteindre cet objectif, le problème de déchargement est formulé comme une tâche d'apprentissage par renforcement (RL), et deux algorithmes d'apprentissage par renforcement profond (DRL), le Deep Q-Network (DQN) et le Double Deep Q-Network (DDQN), sont employés pour apprendre des politiques de déchargement adaptatives fondées sur les conditions du système en temps réel. Un cadre intelligent en boucle fermée est proposé, implémenté en Python et évalué au moyen d'expériences de simulation. Les résultats démontrent que l'approche proposée fondée sur le DRL améliore efficacement les décisions d'allocation des tâches, réduit la latence et la consommation d'énergie, et atteint une convergence stable durant l'entraînement. Ces résultats mettent en évidence le potentiel des stratégies de déchargement pilotées par l'IA et sensibles au contexte pour améliorer les performances et l'efficacité des systèmes MCC de nouvelle génération.

Mots-clés : Mobile Cloud Computing (MCC), déchargement de tâches, apprentissage par renforcement profond (DRL), Deep Q-Network (DQN), Double Deep Q-Network (DDQN), intelligence artificielle, efficacité énergétique, qualité de service (QoS).

ملخص

يستكشف هذا المذكّرة تطبيق الذكاء الاصطناعي (AI) على تفريغ المهام الحوسبية في بيئات الحوسبة السحابية المتنقّلة .(MCC) تتناول الدراسة تحدّي توزيع المهام الحوسبية بكفاءة بين الأجهزة المحلية وخوادم الحافة وموارد السحابة، مع تقليل زمن التنفيذ واستهلاك الطاقة. ولتحقيق هذا الهدف، تُصاغ مشكلة التفريغ كمسألة تعلّم معزّز ،(RL) وتُستخدَم خوارزميتان من خوارزميات التعلّم المعزّز العميق ،(DRL) وهما شبكة Q العميقة (DQN) وشبكة Q العميقة المزدوجة ،(DDQN) لتعلّم سياسات تفريغ تكيّفية اعتمادًا على حالة النظام في الزمن الحقيقي. يُقتَرَح إطار عمل ذكيّ مغلق الحلقة، نُفّذ بلغة Python وقُيّم من خلال تجارب المحاكاة. تُظهِر النتائج أنّ النهج المقترَح القائم على التعلّم المعزّز العميق يحسّن بفعالية قرارات توزيع المهام، ويقلّل من زمن الاستجابة واستهلاك الطاقة، ويحقّق تقاربًا مستقرًّا أثناء التدريب. تُبرِز هذه النتائج إمكانات استراتيجيات التفريغ المعتمِدة على الذكاء الاصطناعي والمدرِكة للسياق في تعزيز أداء وكفاءة أنظمة الحوسبة السحابية المتنقّلة من الجيل القادم.

الكلمات المفتاحية: الحوسبة السحابية المتنقّلة ،(MCC) تفريغ المهام، التعلّم المعزّز العميق ،(DRL) شبكة Q العميقة ،(DQN) شبكة Q العميقة المزدوجة ،(DDQN) الذكاء الاصطناعي، كفاءة الطاقة، جودة الخدمة .(QoS)

Contents

Acknowledgements i

Abstract ii

Résumé iii

ملخص iv

List of Abbreviations x

General Introduction 1

1 Background and Theoretical Foundations 3

1.1 Introduction . . . 3
1.2 Cloud Computing (CC) . . . 3
1.3 Mobile Cloud Computing (MCC) . . . 4
1.4 Task Offloading in MCC . . . 4
1.5 Traditional Task Offloading Approaches . . . 5
1.6 Mobile Cloud Computing Service Models . . . 5
1.6.1 Infrastructure as a Service (IaaS) . . . 5
1.6.2 Platform as a Service (PaaS) . . . 6
1.6.3 Software as a Service (SaaS) . . . 6
1.7 Challenges of Traditional Offloading Approaches . . . 6
1.8 Artificial Intelligence Overview . . . 6
1.8.1 Artificial Intelligence (AI) . . . 6
1.8.2 Machine Learning (ML) . . . 7
1.8.3 Deep Learning (DL) . . . 8
1.8.4 Autoencoders . . . 12
1.8.5 Transformers . . . 12
1.8.6 Other AI Techniques . . . 12
1.9 AI and Deep Reinforcement Learning in MCC . . . 12
1.10 Classification of Mobile Applications in MCC . . . 13
1.10.1 Lightweight Applications . . . 13

1.10.2 Resource-Intensive Applications . . . 13
1.11 Conclusion . . . 13

2 Literature Review and Related Work 15
2.1 Introduction . . . 15
2.2 Related Works . . . 15
2.3 Comparative Table . . . 16
2.4 Research Gap and Motivation . . . 18
2.5 Conclusion . . . 18

3 Conception and Modelling of the Proposed Framework 19
3.1 Introduction . . . 19
3.2 General Architecture of the Proposed Framework . . . 19
3.3 Task Generation . . . 20
3.4 Context and State Collection . . . 20
3.5 AI Decision Engine . . . 21
3.6 Offloading Decision . . . 21
3.7 Execution Environments . . . 21
3.8 Result Return . . . 22
3.9 Performance Evaluation and Feedback . . . 22
3.10 Learning and Model Update . . . 22
3.11 Modelling the Offloading Problem . . . 22
3.11.1 State Representation . . . 23
3.11.2 Action Space . . . 23
3.11.3 Reward Function . . . 23
3.11.4 Policy Objective . . . 23
3.12 Deep Q-Network (DQN) . . . 23
3.12.1 Description . . . 23
3.12.2 DQN Algorithm . . . 25
3.13 Double Deep Q-Network (DDQN) . . . 25
3.14 Design Objectives . . . 26
3.15 Conclusion . . . 26

4 Implementation and Experimental Evaluation 27
4.1 Introduction . . . 27
4.2 Development Environment and Tools . . . 27
4.2.1 Python Programming Language . . . 27

4.2.2 Visual Studio Code (VS Code) . . . 28
4.2.3 Python Libraries . . . 28
4.2.4 Development Workflow Environment . . . 29
4.3 System Implementation . . . 29
4.3.1 System Architecture . . . 29
4.3.2 Reinforcement Learning Workflow . . . 30
4.4 Experimental Setup . . . 30
4.4.1 Simulation Environment . . . 30
4.4.2 Task Generation and Execution Scenarios . . . 31
4.4.3 Network and Load Conditions . . . 31
4.4.4 Training Hyperparameters . . . 31
4.5 Performance Evaluation Metrics . . . 32
4.5.1 Execution Delay . . . 32
4.5.2 Energy Consumption . . . 32
4.5.3 Reward Evolution . . . 33
4.5.4 Convergence Stability . . . 33
4.6 Results and Discussion . . . 34
4.6.1 Learning Performance Analysis . . . 34
4.6.2 Execution Delay Evaluation . . . 34
4.6.3 Energy Consumption Evaluation . . . 35
4.6.4 Convergence Analysis . . . 35
4.6.5 Discussion and Interpretation of Results . . . 36
4.6.6 Summary of Experimental Findings . . . 36
4.7 Conclusion . . . 37

General Conclusion and Future Perspectives **38**

Bibliography **40**

A Simulation Environment Parameters **41**

List of Figures

1.1 Supervised Learning Framework 7
1.2 Unsupervised Learning Framework 8
1.3 Reinforcement Learning Framework 9
1.4 Multilayer Perceptron (MLP) Architecture 9
1.5 Simplified Convolutional Neural Network (CNN) Architecture 10
1.6 Simplified Bidirectional RNN Architecture 10
1.7 Simplified LSTM Architecture for Sequence Prediction 11
1.8 Simplified GRU Architecture for Sequence Prediction 11

3.1 General Architecture of the Proposed Intelligent Task Offloading Framework 20
3.2 Workflow of the DQN-Based Offloading Decision Process 24

4.1 Python Programming Language 28
4.2 Visual Studio Code IDE 28
4.3 Python Libraries Used in the Project 29
4.4 Overall Development Workflow 30
4.5 Reinforcement Learning Framework for Computational Task Offloading 31
4.6 Reward vs. Episodes — Learning Curve 34
4.7 Comparison of Execution Delay Across Task Scenarios 35
4.8 Energy Consumption Comparison Across Scenarios (Baseline vs. Proposed RL Method) 35
4.9 Convergence Behaviour of the DQN Model 36

List of Tables

2.1 Comparative Overview of Related Works 17

4.1 Training Hyperparameters 32

A.1 Simulation Environment Parameters 41