



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Μηχανισμοί Διαχείρισης Edge Cloud Υποδομών Μέσω Βαθιάς Ενισχυτικής Μάθησης

Μελέτη και υλοποίηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΚΑΡΑΤΖΙΚΟΥ ΑΡΗ

Επιβλέπων: Εμμανουήλ Βαρβαρίγος
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2024



Μηχανισμοί Διαχείρισης Edge Cloud Υποδομών Μέσω Βαθιάς Ενισχυτικής Μάθησης

Μελέτη και υλοποίηση

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΚΑΡΑΤΖΙΚΟΥ ΑΡΗ

Επιβλέπων: Εμμανουήλ Βαρβαρίγος
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 1η Ιουλίου 2024.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Εμμανουήλ Βαρβαρίγος
Καθηγητής Ε.Μ.Π.

.....
Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

.....
Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2024



Copyright © – All rights reserved. Με την επιφύλαξη παντός δικαιώματος.
Άρης Καρατζίκος, 2024.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

.....
Άρης Καρατζίκος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών

1 Ιουλίου 2024

Περίληψη

Στη σύγχρονη τεχνολογική πραγματικότητα της απαίτησης δεδομένων σε πραγματικό χρόνο, δημιουργείται η ανάγκη για βελτιστοποίηση της διαχείρισης των υποδομών που επεξεργάζονται αυτά τα δεδομένα. Ωστόσο, η διόγκωση των μεγεθών των υποδομών αυξάνει σημαντικά και την πολυπλοκότητα της ανάλυσής τους. Οι ανάγκες αυτές οδήγησαν στη χρήση τεχνητής νοημοσύνης και πιο συγκεκριμένα μεθόδων μηχανικής μάθησης, για την κατανόηση και διαχείριση των πολύπλοκων συστημάτων και υποδομών σε πραγματικό χρόνο αυτοματοποιημένα.

Το αντικείμενο αυτής της διπλωματικής εργασίας αποτελεί η ανάπτυξη μεθόδων διαχείρισης και αντίδρασης σε πιθανές καταστάσεις μιας Cloud-Edge υποδομής για την βελτιστοποίηση παραμέτρων του συστήματος καθώς και Quality of Service μετρικών. Χρησιμοποιούνται μοντέλα ενισχυτικής μάθησης που χρησιμοποιούν βαθιά νευρωνικά δίκτυα για να αναγνωρίζουν μοτίβα συσχέτισης καταστάσεων-ενεργειών βάση συνάρτησεων κέρδους.

Λέξεις Κλειδιά

Cloud-Edge, καταμεμημένο σύστημα, μηχανική μάθηση, ενισχυτική μάθηση, βαθιά ενισχυτική μάθηση, QoS μετρικές, target Q-Network, δρομολογητής εργασιών, χρονικές καθυστερήσεις εργασιών, RTT

Abstract

In today's technology world of real-time data demand, there is a need to optimise the management of the infrastructures that process data. However, the growth of infrastructures' size significantly increases the complexity of their analysis accordingly. These needs have led to the use of artificial intelligence, and more specifically machine learning methods, to understand and manage complex systems and infrastructures in real-time in an automated manner.

The subject of this thesis is the development of methods to manage and react to possible states of a Cloud-Edge infrastructure in order to optimize infrastructure and Quality of Service parameters. Reinforcement learning models and deep neural networks are used to identify state-action correlation patterns based on reward feedback from the model.

Keywords

Cloud-Edge, distributed system, machine learning, reinforcement learning, deep reinforcement learning, QoS metrics, target Q-Network, job scheduler, job time delay, RTT

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. Εμμανουήλ Βαρβαρίγο για την επίβλεψη αυτής της διπλωματικής εργασίας.

Επίσης, ευχαριστώ ιδιαίτερα τον Διδάκτορα Πολυζώη Σούμπλη για τη συνεχή καθοδήγησή του και την εξαιρετική συνεργασία που είχαμε.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου, την αδερφή μου, τη γιαγιά μου, την Τζένη μου και τους φίλους μου για την υπομονή και τη στήριξη τους καθόλη τη διάρκεια της εκπόνησης της εργασίας καθώς και όλων των χρόνων που φοίτησα στο Ε.Μ.Π..

Αθήνα, Ιούλιος 2024

Άρης Καρατζίκος

Περιεχόμενα

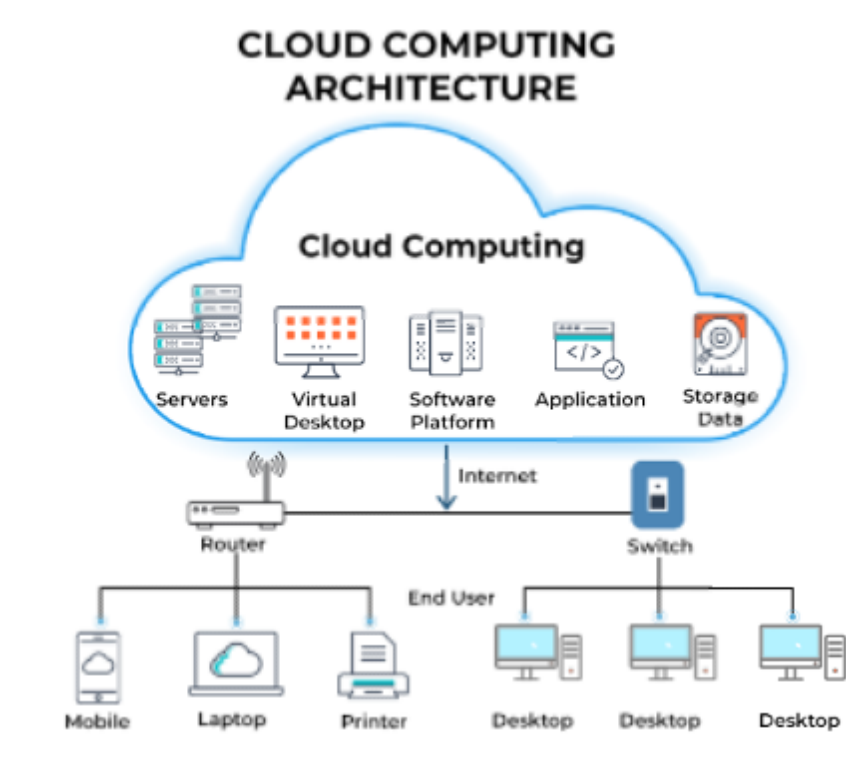
Περίληψη	5
Abstract	7
Ευχαριστίες	9
1 Εισαγωγή	13
1.1 Αντικείμενο της διπλωματικής	16
1.2 Οργάνωση του τόμου	17
I Θεωρητικό Μέρος	19
2 Θεωρητικό υπόβαθρο	21
2.1 Δομή Εργασιών	21
2.1.1 Εισαγωγή	21
2.1.2 Δομή Εργασιών	21
2.1.3 Διαχείριση εργασιών σε κατανεμημένα συστήματα	24
2.2 Μηχανική Μάθηση	24
2.3 Ενισχυτική Μάθηση	26
3 Σχετικές εργασίες	33
II Πρακτικό Μέρος	37
4 Υποδομή και Εργασίες	39
4.1 Υποδομή	40
4.2 Εργασίες	41
4.3 Δομολόγηση Εργασιών	41
5 Μοντέλο Ενισχυτικής Μάθησης	43
6 Συγκεντρωτική περιγραφή του προβλήματος	47
7 Υλοποίηση	49

8 Αποτελέσματα	51
8.1 Παράμετροι Προσομοίωσης	51
8.2 Ανάλυση Αποτελεσμάτων	52
8.2.1 DRL μοντέλα με χρονικές καθυστερήσεις των clusters	53
8.3 Σύγκριση Επιδόσεων με τις βασικές περιπτώσεις δρομολογητών	59
9 Σύνοψη - Συμπεράσματα	63
9.1 Σύνοψη	63
9.2 Συμπεράσματα	63
9.3 Μελλοντικές Επεκτάσεις	64
Βιβλιογραφία	68

Κεφάλαιο 1

Εισαγωγή

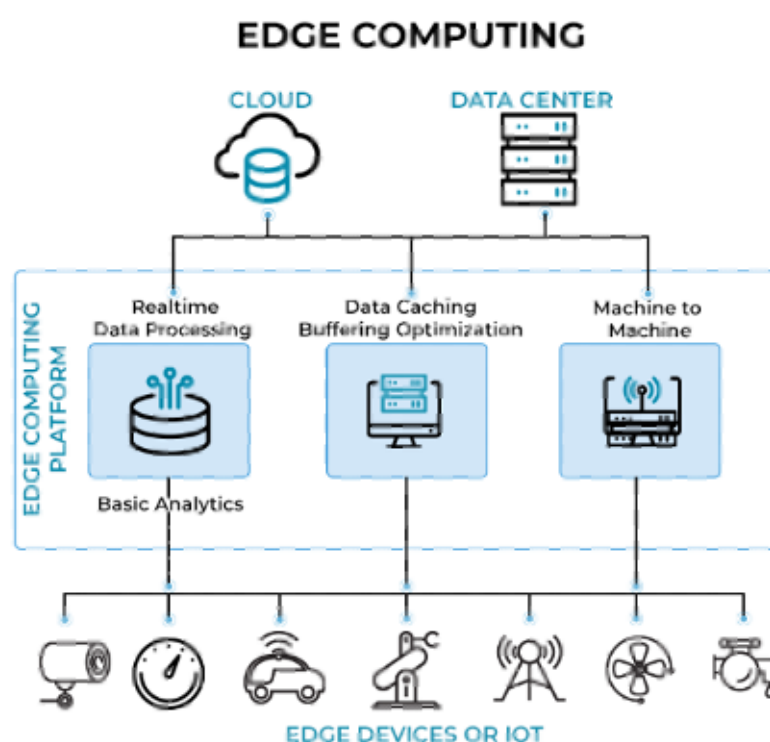
Παραδοσιακά, οι απαιτήσεις επεξεργασίας και αποθήκευσης των εφαρμογών και υπηρεσιών καλύπτονταν από hardware μηχανήματα της εταιρείας που δημιουργούσε τα προϊόντα αυτά. Ωστόσο, στη σύγχρονη εποχή, η κεντρική διαχείριση cloud computing και η πληθώρα ευέλικτων επιλογών χρήσης του έχει δημιουργήσει μια νέα δυναμική στον κόσμο της τεχνολογίας. Το cloud computing (Image 1) αποτελεί ένα πολύ συμβατό σύστημα για τις έντονα αυξανόμενες απαιτήσεις των εφαρμογών και υπηρεσιών για υπολογιστική και αποθηκευτική ισχύ.



Εικόνα 1.1: Αρχιτεκτονική του Cloud computing

Η ανάπτυξη των συσκευών IoT (Internet of Things) και η γενική ανάγκη για γρήγορη επεξεργασία δεδομένων έχει οδηγήσει στην ανάπτυξη και χρήση μαζικών υποδομών πιο κοντά στο χρήστη, καθώς η μετάδοση ακατέργαστων δεδομένων σε κεντρικά συστήματα διαχείρισης δεδομένων, όπως είναι το Cloud, δημιουργεί προβλήματα όσον αφορά τους περιορισμούς του bandwidth του δικτύου και του latency. Ταυτόχρονα, τα αυτόνομα συστήματα, τα δίκτυα 5G

και η μαζική παραγωγή και ανάγκη επεξεργασίας δεδομένων των τεχνολογικών συστημάτων έχουν αυξήσει τις απαιτήσεις για υπολογιστική απόδοση και ελαχιστοποίηση της κατανάλωσης ενέργειας. Η ύπαρξη της edge υποδομής δημιουργεί ένα νέο επίπεδο επεξεργασίας πιο κοντά στο τελικό σημείο (συσκευή) και χρησιμοποιείται επιπρόσθετα και ως σημείο φιλτράρισματος, στο οποίο απορρίπτονται τα περιττά δεδομένα και το φόρτο εργασίας κατανέμεται τόσο στο edge όσο και στο Cloud με βάση την καθυστέρηση και άλλες απαιτήσεις QoS (Quality Of Service). Για τη γρήγορη και αποτελεσματική επικοινωνία και συνεργασία μεταξύ διαφορετικών υποδομών, Cloud και Edge έχουν αναπτυχθεί κατακεκομμένα και ιεραρχικά συστήματα.



Εικόνα 1.2: Αρχιτεκτονική του Edge computing

Με την εξέλιξη της ιεραρχικής αρχιτεκτονικής cloud-edge και την αυτοματοποίησή της, τα όρια μεταξύ των cloud και edge computing εξαλείφονται, δημιουργώντας ένα cloud-edge continuum. Αυτή η δομή αντιπροσωπεύει ένα απρόσκοπτο φάσμα υπολογιστικών πόρων που εκτείνονται από τα κέντρα δεδομένων του Cloud έως κατακεκομμένες edge συσκευές. Αυτή η ιεραρχική δομή επιτρέπει την αποτελεσματική διαχείριση και οργάνωση των πόρων σε πολλαπλά επίπεδα.

Στο υψηλότερο επίπεδο, τα συγκεντρωτικά κέντρα δεδομένων Cloud παρέχουν σημαντική υπολογιστική ισχύ και αποθηκευτική ικανότητα, εξυπηρετώντας την επεξεργασία δεδομένων μεγάλης κλίμακας και μεγάλο φόρτο εργασίας. Στα κατώτερα επίπεδα της ιεραρχίας, περιλαμβάνονται edge συσκευές που τοποθετούνται πιο κοντά στο σημείο παραγωγής ή κατανάλωσης δεδομένων. Αυτές οι συσκευές αποτελούνται από αισθητήρες IoT, έξυπνες συσκευές και αυτόνομα οχήματα κλπ και οι οποίες επιτρέπουν την επεξεργασία δεδομένων σε πραγματικό χρόνο, μειώνοντας την καθυστέρηση και βελτιώνοντας την ανταπόκριση για time critical

εφαρμογές.

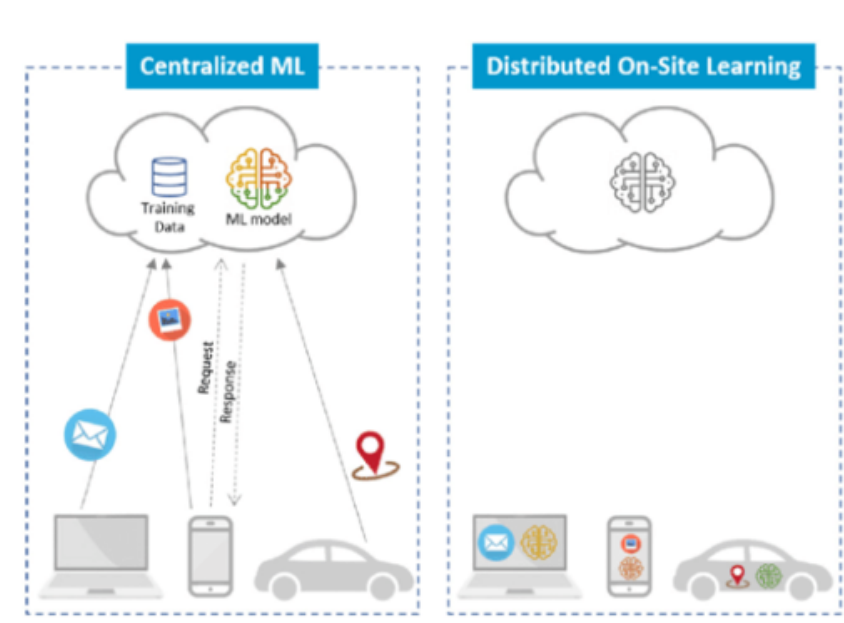
Σύμφωνα με πρόσφατη έρευνα, η παγκόσμια αγορά edge υποδομών αναμένεται να αυξηθεί από 60 δισεκατομμύρια δολάρια το 2024 σε 110 δισεκατομμύρια δολάρια μέχρι το 2029, αντιπροσωπεύοντας CAGR 13% [1].

Αυτή η ταχεία ανάπτυξη οφείλεται στην αυξανόμενη ζήτηση για επεξεργασία χαμηλής καθυστέρησης, στον πολλαπλασιασμό των συσκευών IoT και στην άνοδο των δικτύων 5G. Αρκετές άλλες μελέτες έχουν επίσης επισημάνει τη σημασία του cloud-edge continuum. Για παράδειγμα, έχει παρατηρηθεί ότι η υιοθέτηση του edge computing είναι ζωτικής σημασίας για τους οργανισμούς που στοχεύουν στην αξιοποίηση αναδυόμενων τεχνολογιών όπως η τεχνητή νοημοσύνη, το IoT και το blockchain [2]. Μια άλλη μελέτη προβλέπει ότι έως το 2025, το 75% των δεδομένων που παράγονται από τις επιχειρήσεις θα υποβάλλονται σε επεξεργασία σε edge υποδομές [3].

Η ανάπτυξη των cloud-native εφαρμογών αποτελεί επίσης σημαντική εξέλιξη για τις Cloud-Edge υποδομές, καθώς σηματοδοτεί τη μετάβαση από τις μονολιθικές εφαρμογές μιας ενιαίας κεντρικής διαχείρισης σε μικρά αυτόνομα συνεργαζόμενα τμήματα εφαρμογών. Οι cloud-native εφαρμογές σχεδιάζονται ειδικά για περιβάλλοντα cloud, εκμεταλλευόμενες την ευελιξία και την κλιμάκωση που προσφέρει η αρχιτεκτονική microservices. Αυτή η προσέγγιση επιτρέπει την ανάπτυξη, τη συντήρηση και την αναβάθμιση εφαρμογών με μεγαλύτερη ευελιξία και ταχύτητα, σε αντίθεση με τις παραδοσιακές μονολιθικές εφαρμογές που η πολυπλοκότητά τους δημιουργεί δυσκολίες στη διαχείριση και την αναβάθμιση.

Επιπρόσθετα, με την κυριάρχηση των cloud-native εφαρμογών στη σύγχρονη τεχνολογική πραγματικότητα, η παγκόσμια αγορά μετατοπίζεται από τα παραδοσιακά συγκεντρωτικά συστήματα σε κατακευματισμένα συστήματα που λειτουργούν σε cloud-edge περιβάλλοντα. Η μετατόπιση αυτή, όπως αναφέρθηκε και προηγουμένως, οφείλεται στην αύξηση των συνδεδεμένων συσκευών, στον αυξανόμενο όγκο δεδομένων και στην ανάγκη παράλληλης εκτέλεσης υπό-εργασιών εφαρμογών. Τα κατακευματισμένα συστήματα αποτελούνται από πολλαπλούς κόμβους που συνεργάζονται για την εκτέλεση μιας συγκεκριμένης εργασίας, αντί να βασίζονται σε ένα ενιαίο κεντρικό σύστημα για τη διεκπεραίωση όλων των λειτουργιών.

Αρκετές μελέτες έχουν τονίσει τη σημασία των κατακευματισμένων συστημάτων συγκριτικά με τα συγκεντρωτικά συστήματα στον τομέα αυτόν. Η δυνατότητα για παράλληλη επεξεργασία δεδομένων κοντά στην πηγή παραγωγής τους αυξάνει σημαντικά την απόδοση των συστημάτων και μειώνει τον απαιτούμενο χρόνο αναμονής [4][5].



Εικόνα 1.3: Προσέγγιση των Συγκεντρωτικών και Καταναμημένων υποδομών για την εκτέλεση εργασιών από IoT συσκευές

Ωστόσο, παρά τα οφέλη των ιεραρχικών και καταναμημένων συστημάτων, η διαχείριση και η ενσωμάτωση ετερογενών πηγών δεδομένων στο cloud-edge σύστημα αποτελεί ένα πολύπλοκο έργο. Ως εκ τούτου, είναι απαραίτητη η ανάπτυξη νέων τεχνικών και τεχνολογιών για την αποτελεσματική διαχείριση των καταναμημένων συστημάτων και τον μετριασμό των σχετικών προκλήσεων.

Για την κάλυψη της ανάγκης για αυτοματοποιημένη και πολύπλοκη διαχείριση ετερογενών πηγών δεδομένων, χρησιμοποιείται η τηλεμετρία σε συνδυασμό με μηχανισμούς τεχνητής νοημοσύνης, όπως αλγορίθμους μηχανικής μάθησης και μοντέλα ενισχυτικής μάθησης. Η τηλεμετρία συνιστά μία διαδικασία καταγραφής και μεταφοράς δεδομένων από απομακρυσμένες πηγές στο κεντρικό σύστημα διαχείρισης. Οι δυνατότητές της για γρήγορη και φθηνή μεταφορά, καθώς και το ευρύ φάσμα εφαρμογών της, έχουν οδηγήσει στην επικράτησή της σε θέματα καταγραφής δεδομένων σε πεδία όπως το διάστημα, η βιολογία καθώς και σε ενεργειακές μετρήσεις [6][7][8].

Αντίστοιχα, η χρήση της τεχνητής νοημοσύνης ενσωματώνεται ολοένα και περισσότερο σε διάφορες πτυχές των συστημάτων διαχείρισης δεδομένων, όπως είναι η κατανομή πόρων, η προβλεπτική συντήρηση, η βελτιστοποίηση QoS μετρικών του δικτύου, η ανίχνευση ανωμαλιών καθώς και η βελτιστοποίηση της διαχείρισης της ενέργειας.

1.1 Αντικείμενο της διπλωματικής

Στην εργασία μας, θεωρούμε μια υπολογιστική και αποθηκευτική υποδομή που συνδυάζει τις θεωρητικές δομές Cloud και Edge. Στην υποδομή ανατίθεται φόρτο εργασίας εφαρμογών, το οποίο κατανέμεται και μεταφέρεται μεταξύ των διαθέσιμων επιπέδων και πόρων. Η οργάνωση της υποδομής έχει ένα συγκεντρωτικό σύστημα που διαχειρίζεται τρεις συστάδες (clusters) μηχανημάτων, τα οποία εκτελούν εργασίες. Οι εργασίες προς εκτέλε-

ση, φέρουν μια ιεραρχική δομή τριών επιπέδων οργάνωσης που ορίζει τον τρόπο και την ακολουθία ολοκλήρωσης τους. Η παραπάνω δομή αποτελεί το περιβάλλον ανάπτυξης και εκπαίδευσης ενός συστήματος αναγνώρισης καταστάσεων και των καλύτερων δυνατών ενεργειών σε πραγματικό χρόνο μέσω της χρήσης ενισχυτικής μάθησης. Η λύση μας προσφέρει πολλές δυνατότητες για ορισμό παραμέτρων και αποτελεσματικής βελτίωσής τους μέσω της προσομοίωσης υποδομών.

1.2 Οργάνωση του τόμου

Η συνέχεια της διπλωματικής είναι οργανωμένη με τον ακόλουθο τρόπο. Στο 2ο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο πάνω στο οποίο βασίζεται η διπλωματική. Συγκεκριμένα, αρχικά γίνεται λόγος για την δομή και τον τρόπο εκτέλεσης εργασιών σε μια κατανεμημένη υποδομή. Στη συνέχεια, αναφέρονται οι κατηγορίες στις οποίες εντάσσονται οι αλγόριθμοι μηχανικής και ενισχυτικής μάθησης και ο τρόπος που λειτουργούν. Τέλος, περιγράφονται ορισμένες διαδικασίες που θα. Στο 3ο κεφάλαιο, παρουσιάζεται μια ολοκληρωμένη εικόνα από επιστημονικά άρθρα στον τομέα της διαχείρισης πόρων και φόρτων εργασίας σε cloud-edge υποδομές και σε σχετικά θέματα. Στο 4ο κεφάλαιο, περιγράφεται η υποδομή και οι παραδοχές που έχουμε κάνει. Στο 5ο κεφάλαιο, αναλύουμε τα χαρακτηριστικά του μοντέλου ενισχυτικής μάθησης που αναπτύξαμε. Στο 6ο κεφάλαιο, αναφερόμαστε στις λεπτομέρειες της υλοποίησής μας. Στο 7ο κεφάλαιο παρατίθενται, συγκρίνονται και σχολιάζονται τα αποτελέσματα του μοντέλου ενισχυτικής μάθησης. Στο 8ο κεφάλαιο, πραγματοποιείται μια σύνοψη του έργου και παρατίθενται σημεία για περισσότερη ανάλυση και έρευνα.

Μέρος I

Θεωρητικό Μέρος

Θεωρητικό υπόβαθρο

2.1 Δομή Εργασιών

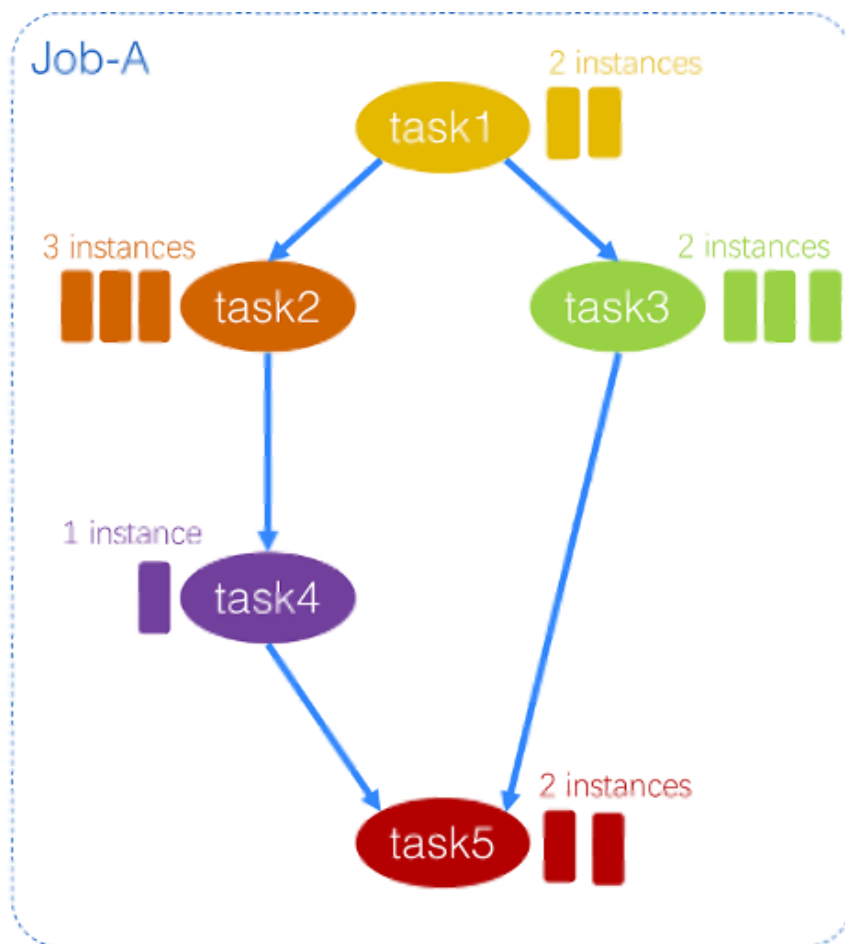
2.1.1 Εισαγωγή

Στις υπηρεσίες που προσφέρουν οι “λουδ-Εδγε κατανεμημένες υποδομές, η διαχείριση των φόρτων εργασίας έχει αναδειχθεί σε πρόκληση. Τα φορτία εργασίας, που περιλαμβάνουν εργασίες batch και online services, αποτελούν την πλειονότητα των υπολογιστικών εργασιών που εκτελούνται σε περιβάλλοντα Cloud-Edge συνεχούς. Λόγω της πολύπλοκης φύσης των υπηρεσιών και εφαρμογών που δημιουργούν αυτές τις υπολογιστικές ανάγκες, οι εργασίες φέρουν συγκεκριμένους κανόνες και εξαρτίσεις που διαφοροποιούν τον τρόπο εκτέλεσης τους. Ακόμη, τα κατανεμημένα συστήματα, διαθέτουν διάφορες πολιτικές στον τρόπο μεταφοράς δεδομένων μεταξύ διαφορετικών σημείων του δικτύου, αναλόγως με τις συνθήκες του συστήματος, που αναλύονται ακολούθως.

2.1.2 Δομή Εργασιών

Οι εργασίες υπηρεσιών και εφαρμογών αποτελούν σύνθετες δομές αλληλεξαρτώμενων υπό-εργασιών που φέρουν συγκεκριμένους κανόνες κατά την εκτέλεσή τους. Πιο αναλυτικά, χρησιμοποιούμε δύο τύπους εργασιών που παρουσιάζουν διαφορετικά δομικά χαρακτηριστικά.

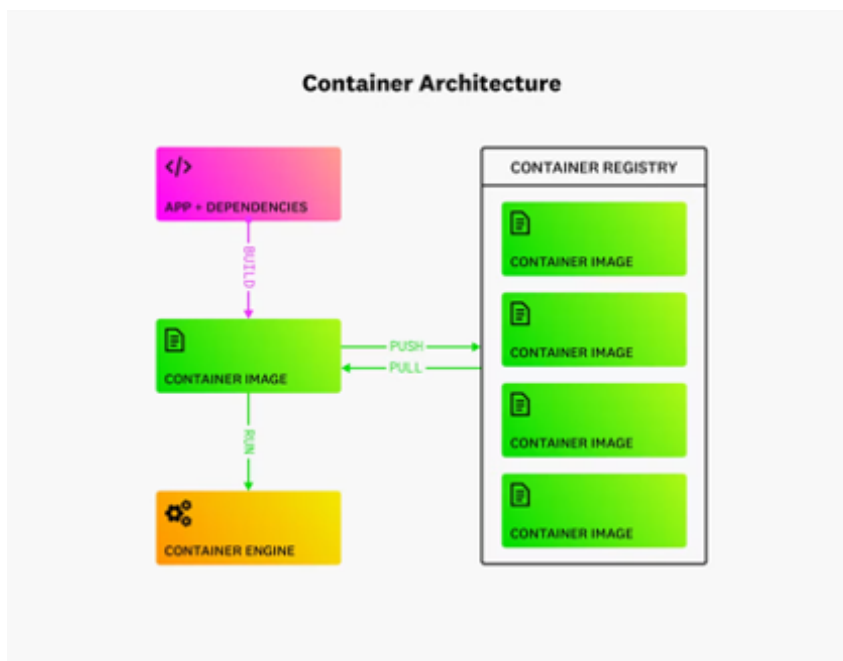
Οι εργασίες batch αφορούν λειτουργικότητες διαχείρισης μνήμης και επεξεργασίας και ανάλυσης δεδομένων όπως το MapReduce και προβλήματα μηχανικής μάθησης, τα οποία υποβάλλονται από εσωτερικούς χρήστες. Αυτές οι εργασίες δεν είναι εργασίες παραγωγής, το οποίο σημαίνει ότι δεν είναι εργασίες με ανάγκη για άμεση εκτέλεση, ενώ εκτελούνται απευθείας στον physical host. Οι εργασίες batch μπορούν να περιγραφούν ως ένα μοντέλο "Job-Task-Instance". Κάθε υποβαλλόμενη εργασία-job χωρίζεται σε πολλαπλές εργασίες-tasks με διαφορετικά στοιχεία υπολογιστικής λογικής, χρονικών και δομικών προδιαγραφές. Τα tasks, μπορεί να φέρουν και αλληλεξαρτίσεις μεταξύ τους, σε σειρά προτεραιότητας εκτέλεσης, σχέσεις που συνήθως περιγράφονται μέσω της χρήσης Ακυκλικών Κατευθυνόμενων Γράφων (DAG) και τις οποίες πρέπει να λαμβάνει υπόψη του το σύστημα κατά την κατανομή φορτίων σε κόμβους για την αποτροπή περιττών καθυστερήσεων.



Εικόνα 2.1: Εσωτερική δομή εργασιών

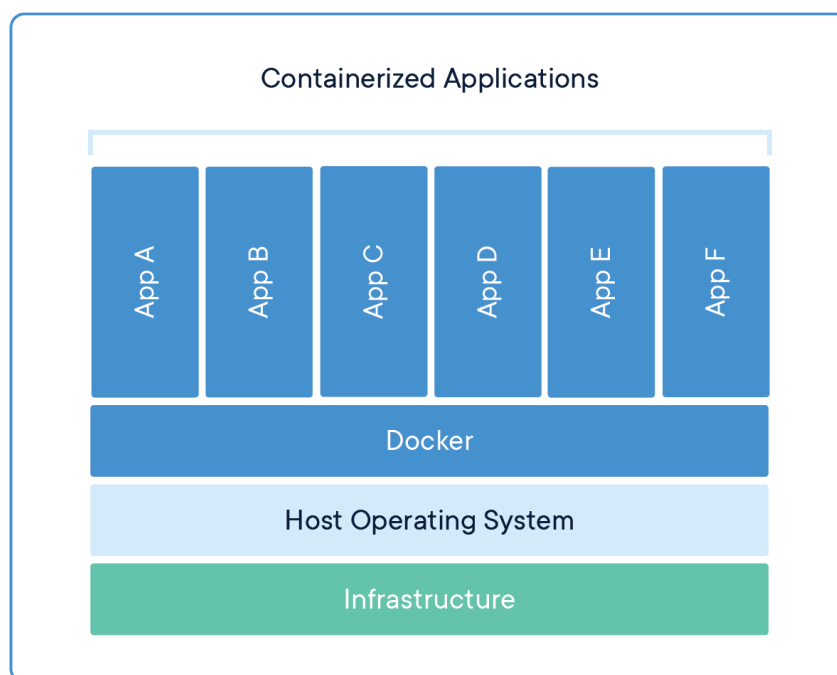
Ο χρόνος ολοκλήρωσης ενός job καθορίζεται από τα tasks. Κάθε task περιέχει τουλάχιστον ένα task instance, το μικρότερο προς κατανομή κομμάτι εργασίας στην υλοποίησή μας. Κάθε task instance του ίδιου task έχει τον ίδιο δυαδικό κώδικα και τις ίδιες ανάγκες σε πόρους. Ομοίως, η διάρκεια ενός task εξαρτάται από όλα τα instances του.

Η δεύτερη κατηγορία αποτελείται από το φόρτο εργασιών online υπηρεσιών. Οι εργασίες που χρησιμοποιούμε παράγονται συνήθως από users των υπηρεσιών, οι οποίες απαιτούν τη λειτουργία διαφόρων υπηρεσιών που σηκώνονται σε containers και τρέχουν για μεγάλα χρονικά διαστήματα. Οι υπηρεσίες αυτές πρέπει να εκτελούνται άμεσα χωρίς καθυστερήσεις και φέρουν προτεραιότητα έναντι στα batch jobs.



Εικόνα 2.2: Αρχιτεκτονική των containers

Οι εργασίες online services φέρουν οργάνωση σε container λογική, που περιέχει την «εικόνα» του service που πρέπει να εκτελεστεί, και συγκεκριμένες πληροφορίες και εξαρτήσεις για τη σωστή λειτουργία της. Τα jobs αυτά διαφέρουν σε οργάνωση από τα batch jobs, καθώς η λογική των containers αφορά την παράλληλη και ανεξάρτητη λειτουργία από jobs. Ωστόσο σε ένα επίπεδο networking, χρησιμοποιώντας εργαλεία όπως το service mesh, που περιγράφει εξαρτήσεις μεταξύ εφαρμογών και υπηρεσιών, αναπτύσσονται αλληλεπιδράσεις, ανταλλαγές δεδομένων και χρονικές εξαρτήσεις μεταξύ διαφορετικών υπηρεσιών και άρα εργασιών.



Εικόνα 2.3: Αρχιτεκτονική εφαρμογών που χρησιμοποιούν containers

2.1.3 Διαχείριση εργασιών σε καταναμημένα συστήματα

Σε καταναμημένα συστήματα, λόγω της φύσης της αρχιτεκτονικής λύσης, πραγματοποιείται συνεχής μεταφορά δεδομένων και εργασιών μεταξύ κόμβων του δικτύου. Η μεταφορά εργασιών, εφόσον η εκτέλεσή της έχει ξεκινήσει, μπορεί να πραγματοποιείται είτε από το σημείο που είχε μείνει, είτε από την αρχή. Στην περίπτωση της μεταφοράς από το σημείο που διακόπηκε η εκτέλεση της εργασίας, απαιτείται μεταφορά μεγάλων όγκων μεταδεδομένων με επιβαρύνσεις στη δικτυακή κίνηση καθώς και στη συχνότητα αποθήκευσης ζητησκεινις των καταστάσεων των εργασιών. Η μεταφορά εργασιών με σκοπό την επανεκκίνηση από την αρχή, φέρει ως λογική σημαντικές χρονικές επιβαρύνσεις, ειδικά όταν μιλάμε για εργασίες που έχουν χρονικές εξαρτίσεις μεταξύ τους. Αναλόγως την υποδομή και τη λύση, επιλέγονται και οι δύο λύσεις καθώς και υβριδικά συστήματα. Διασχίζοντας αυτούς τους αναλυτικούς τομείς, η διατριβή μας στοχεύει να παράσχει μια ολιστική κατανόηση της διαχείρισης του φόρτου εργασίας σε καταναμημένες υποδομές, προσφέροντας γνώσεις σχετικά με τη περίπλοκη δυναμική και τις προκλήσεις που ενυπάρχουν στην ενορχήστρωση υπολογιστικών εργασιών σε όλο το συνεχές νέφους-ακμή. Μέσω αυστηρής ανάλυσης και πειραματισμού, προσπαθούμε να συμβάλουμε στην προώθηση αποτελεσματικών, κλιμακούμενων και ανθεκτικών λύσεων διαχείρισης φόρτου εργασίας, έτοιμων να ανταποκριθούν στις εξελισσόμενες απαιτήσεις των σύγχρονων υπολογιστικών περιβαλλόντων.

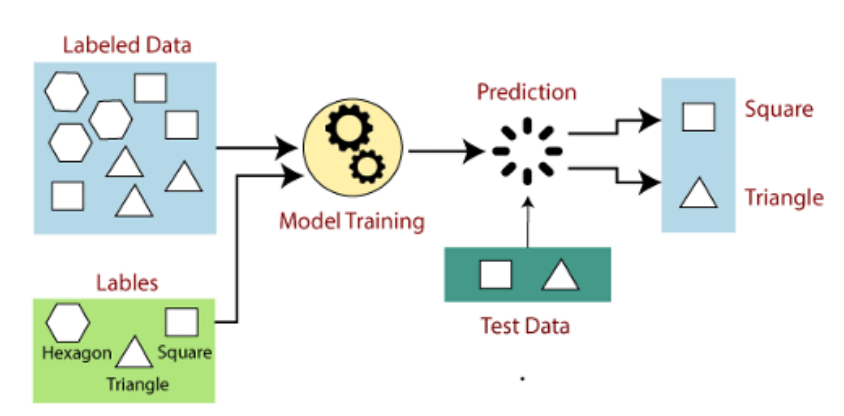
2.2 Μηχανική Μάθηση

Οι αλγόριθμοι μηχανικής μάθησης μπορούν να αναγνωρίσουν συσχετίσεις και αποκλίνουσες συμπεριφορές δεδομένων μέσω περίπλοκων υπολογισμών όπως προσαρμογή της διάστασης του συστήματος και συνδυασμού διαφόρων παραμέτρων του συστήματος σε μαθη-

ματικά και στατιστικά μοντέλα. Οι αλγόριθμοι μηχανικής μάθησης μπορούν να ταξινομηθούν σε δύο μεγάλες κατηγορίες με βάση τη φύση της εισόδου και την αναμενόμενη έξοδό τους.

- Supervised Learning (Επιβλεπόμενη μάθηση)
- Unsupervised Learning (Μη επιβλεπόμενη μάθηση)

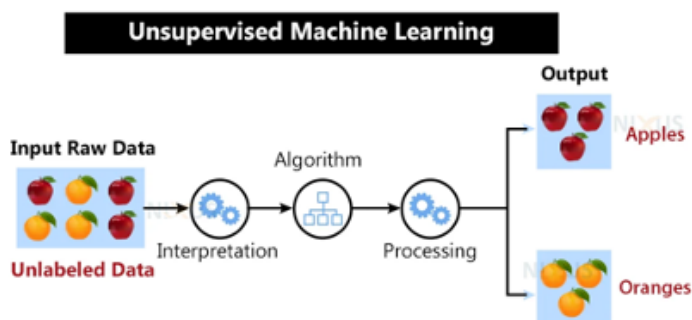
Οι αλγόριθμοι επιβλεπόμενης μάθησης είναι αποτελεσματικοί στον εντοπισμό ανωμαλιών όταν υπάρχει σαφής διάκριση μεταξύ των κλάσεων και τα δεδομένα εκπαίδευσης είναι καλά κατηγοριοποιημένα. Ωστόσο, η εξάρτησή τους από καλά κατηγοριοποιημένα δεδομένα περιορίζει την εφαρμογή τους σε δυναμικά περιβάλλοντα πραγματικού χρόνου, όπου οι κατανομές των δεδομένων μπορούν να αλλάξουν γρήγορα. Επιπλέον, η συχνή ενημέρωση του συνόλου δεδομένων εκπαίδευσης μπορεί να είναι δαπανηρή και χρονοβόρα, περιορίζοντας περαιτέρω τη χρησιμότητά τους στην ανίχνευση ανωμαλιών για υποδομές cloud-edge, όπου το περιβάλλον υπόκειται σε συχνές αλλαγές.



Εικόνα 2.4: Ροή εκπαίδευσης μοντέλου με χρήση Μηχανικής Μάθησης με Επίβλεψη

Οι αλγόριθμοι μη επιβλεπόμενης μάθησης μπορούν να εφαρμοστούν για την ανίχνευση ανωμαλιών χωρίς την ανάγκη δεδομένων εκπαίδευσης με ετικέτες. Ο στόχος αυτών των αλγορίθμων είναι να ανακαλύψουν κρυμμένα μοτίβα και σχέσεις στα δεδομένα, παρόμοια με την εκτίμηση της πυκνότητας στη στατιστική. Οι αλγόριθμοι ομαδοποιούν τα δεδομένα εισόδου σε κλάσεις με βάση τις στατιστικές τους ιδιότητες, χωρίς να κάνουν υποθέσεις σχετικά με την υποκείμενη κατανομή των δεδομένων. Προκειμένου να βελτιωθεί η ακρίβεια, γενικά υποτίθεται ότι οι κανονικές περιπτώσεις δεδομένων είναι συχνότερες στο σύνολο δεδομένων από τις ανώμαλες περιπτώσεις.

Οι τεχνικές μη επιβλεπόμενης μάθησης είναι κατάλληλες για την ανίχνευση άγνωστων ανωμαλιών σε δυναμικά περιβάλλοντα πραγματικού χρόνου, όπως η υποδομή cloud-edge, όπου μπορεί να μην υπάρχει ακριβής ορισμός των χαρακτηριστικών των ανωμαλιών.

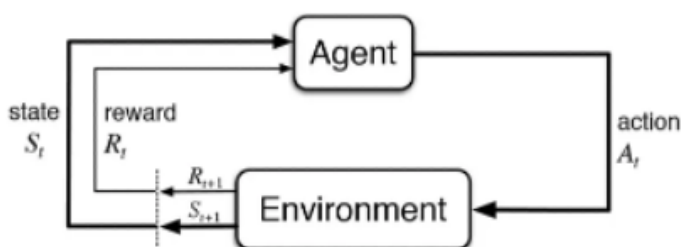


Εικόνα 2.5: Ροή εκπαίδευσης μοντέλου με χρήση Μηχανικής Μάθησης χωρίς Επίβλεψη

2.3 Ενισχυτική Μάθηση

Ωστόσο, χάρη στην ανάπτυξη των βαθιών νευρωνικών δικτύων την τελευταία δεκαετία, μία ακόμη μέθοδος μηχανικής μάθησης, η ενισχυτική μάθηση, έχει εισχωρήσει στη διαχείριση περιπτώσεων ανώμαλης συμπεριφοράς συστημάτων.

Η ενισχυτική μάθηση είναι ένας τύπος μηχανικής μάθησης όπου ένας πράκτορας μαθαίνει να παίρνει αποφάσεις σε ένα περιβάλλον λαμβάνοντας ανατροφοδότηση με τη μορφή ανταμοιβών ή ποινών. Ο πράκτορας αλληλοεπιδρά με το περιβάλλον και αναλαμβάνει δράσεις για να μεγιστοποιήσει την ανταμοιβή του με την πάροδο του χρόνου.

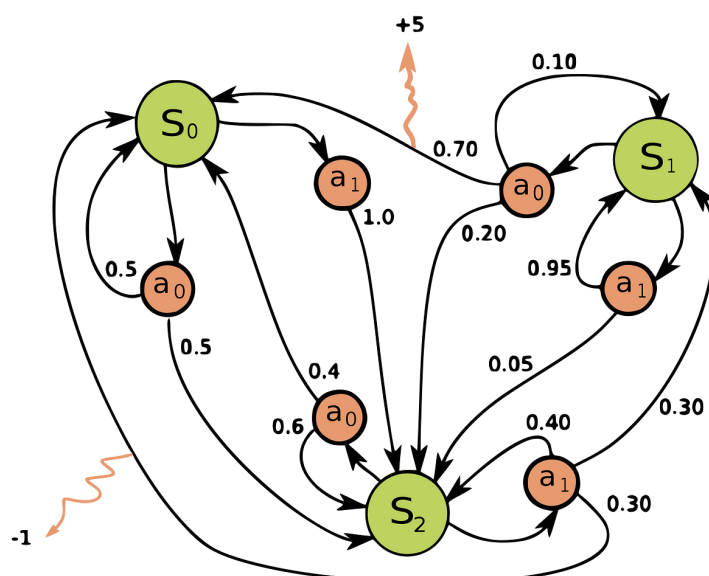


Εικόνα 2.6: Βάση της ροής εκπαίδευσης μοντέλου με χρήση Ενισχυτικής Μάθησης

Η διαφορά του RL(Reinforcement Learning) σε σχέση με supervised μεθόδους έγκειται στο ότι το μοντέλο δε χρειάζεται δεδομένα με ετικέτες. Ωστόσο, μία από τις απαιτήσεις αυτού του είδους μάθησης, αποτελεί η ανάγκη επαρκούς περιγραφής στοιχείων του περιβάλλοντος. Λόγω της δυναμικής και πολύπλοκης φύσης του περιβάλλοντος, δε μπορούν να χρησιμοποιηθούν στις περισσότερες περιπτώσεις αλγόριθμοι δυναμικού προγραμματισμού για την ντετερμινιστική περιγραφή του. Χρησιμοποιούνται, όμως, μαθηματικοί μέθοδοι όπως είναι η διαδικασία αποφάσεων Markov (MDP), η οποία είναι μια στοχαστική διαδικασία ελέγχου διακριτού χρόνου. Παρέχει ένα μαθηματικό πλαίσιο για τη μοντελοποίηση της λήψης αποφάσεων σε καταστάσεις όπου τα αποτελέσματα είναι εν μέρει τυχαία και εν μέρει υπό τον έλεγχο ενός λήπτη αποφάσεων. Οι MDP είναι χρήσιμες για τη μελέτη προβλημάτων βελτιστοποίησης καθώς ορίζουν πλήρως το μοντέλο και επιτρέπουν την ανάλυση

του σε βάθος χρόνου. Πιο συγκεκριμένα, σε κάθε χρονικό βήμα, η διαδικασία βρίσκεται σε κάποια κατάσταση s , και ο λήπτης αποφάσεων μπορεί να επιλέξει οποιαδήποτε ενέργεια a που είναι διαθέσιμη στην κατάσταση s . Το μοντέλο ανταποκρίνεται μετακινούμενη σε μια νέα κατάσταση, s' , και δίνει στο μοντέλο μια αντίστοιχη ανταμοιβή $R(s,s')$. Η πιθανότητα να μεταβεί η διαδικασία στη νέα της κατάσταση s' επηρεάζεται από την επιλεγμένη ενέργεια και την παρούσα κατάσταση s . Για να ικανοποιείται η ιδιότητα Markov στις μεταβάσεις μεταξύ καταστάσεων του περιβάλλοντος, οφείλει το μοντέλο να είναι ανεξάρτητο από τις προηγούμενες καταστάσεις κάθε δεδομένη στιγμή. Η MDP ορίζει πλήρως το περιβάλλον κάθε προβλήματος RL(Reinforcement Learning) με τους εξής όρους:

- Το πεδίο καταστάσεων (state space), που περιγράφει την κατάσταση του συστήματος σε μία χρονική στιγμή. Σε αυτό το διάνυσμα περιέχονται στοιχεία που κρίνονται σημαντικά για πιθανές επιλογές ενεργειών του μοντέλου.
- Το πεδίο ενεργειών (action space), που αποτελεί το σύνολο των πιθανών ενεργειών σε κάθε δεδομένη κατάσταση. Θεωρητικά μπορεί κάθε κατάσταση να έχει διαφορετικές επιλογές σε ενέργειες που περιγράφονται από αυτό το σύνολο.
- Τη συνάρτηση επιβράβευσης, που αποτελεί τον τρόπο που παραμετροποιείται το κάθε σύστημα ώστε να αξιολογηθεί η μετάβαση από μία κατάσταση σε μια άλλη με βάση την επιλεγμένη ενέργεια.
- Τη συνάρτηση πιθανότητας μετάβασης, που αποτελεί την πιθανοτική αναπαράσταση της μετάβασης του συστήματος από μία κατάσταση σε μία άλλη με βάση μια επιλεγμένη ενέργεια.



Εικόνα 2.7: Αποτύπωση των χαρακτηριστικών του συστήματος με βάση τη μέθοδο MDP

Τα προβλήματα RL κατηγοριοποιούνται σε προβλήματα με γνώση του μοντέλου ή χωρίς.

Τα προβλήματα που βασίζονται στη γνώση του υπάρχοντος μοντέλου, έχουν γνώση και όλων των χαρακτηριστικών που αναφέρονται παραπάνω. Σκοπός τους είναι ο υπολογισμός της πολιτικής π του μοντέλου, μίας δηλαδή συνάρτησης που οδηγεί στη βελτιστοποίηση των επιλεγόμενων ενεργειών σε κάθε κατάσταση. Η συνάρτηση αυτή μπορεί να υπολογιστεί είτε μέσω αλλαγών στη συνάρτηση που περιγράφει την ίδια την πολιτική είτε μέσω αλλαγών σε μια βοηθητική συνάρτηση V . Ο τύπος της βέλτιστης συνάρτησης της πολιτικής είναι ο εξής:

$$\pi^*(s) = \arg \max_a \left(R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^*(s') \right)$$

όπου, V^* αποτελεί τη βέλτιστη συνάρτηση value.

Για τον υπολογισμό της πολιτικής με τη βοηθητική συνάρτηση value, υπολογίζουμε τη V μέσω της Q , που θα οδηγήσει και στον υπολογισμό της συνάρτησης πολιτικής του μοντέλου.

$$Q(s, a) := R(s, a) + \gamma \sum_{s' \in S} T(s, a, s') V^*(s')$$

και

$$\pi(s) := \max_a Q(s, a)$$

Στην περίπτωση που δε γνωρίζουμε το μοντέλο, δηλαδή τις συναρτήσεις κέρδους R και πιθανότητας μετάβασης T , υπάρχουν λύσεις που δίνουν προτεραιότητα στον υπολογισμό αυτών των συναρτήσεων για την εύρεση των βέλτιστων τιμών του συστήματος (model-based) και λύσεις που υπολογίζουν τις τιμές αυτές με απλοποιήσεις στους τύπους που μόλις αναλύθηκαν.

Εμείς εξετάζουμε την περίπτωση model-free, καθώς η αναπαράσταση της πιθανότητας μετάβασης στο δυναμικό περιβάλλον των εργασιών και μηχανημάτων είναι πολύ δύσκολο να προσεγγιστεί επαρκώς. Στη model-free λογική επικρατούν δύο κυρίαρχες μέθοδοι, η ευριστική μέθοδος Actor - Critic και η μέθοδος Q-learning. Η μέθοδος Actor - Critic βασίζεται στην ιδέα της βελτιστοποίησης της συνάρτησης πολιτικής μέσω της συνάρτησης value V . Σε ένα τέτοιο μοντέλο υπάρχουν δύο κομμάτια, το Actor που εισαγάγει τη λογική της ενισχυτικής μάθησης και προσπαθεί, αντλώντας πληροφορίες από το critic component για τις τιμές της συνάρτησης V , να βελτιστοποιήσει την πολιτική του μοντέλου και το Critic, που βάσει της πολιτικής του αστορ επιλέγει ενέργειες και παρατηρεί την τιμή της V για το εκάστοτε state-action ζεύγος και ενημερώνει το Actor. Η μέθοδος Q-learning χρησιμοποιεί μόνο μια συνιστώσα, η οποία βασίζεται στον προσδιορισμό των τιμών Q για κάθε state-action. Από τους τύπους που αναφέρθηκαν παραπάνω, οδηγούμαστε στον τύπο:

$$Q(s, a) := Q(s, a) + a \left(r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right)$$

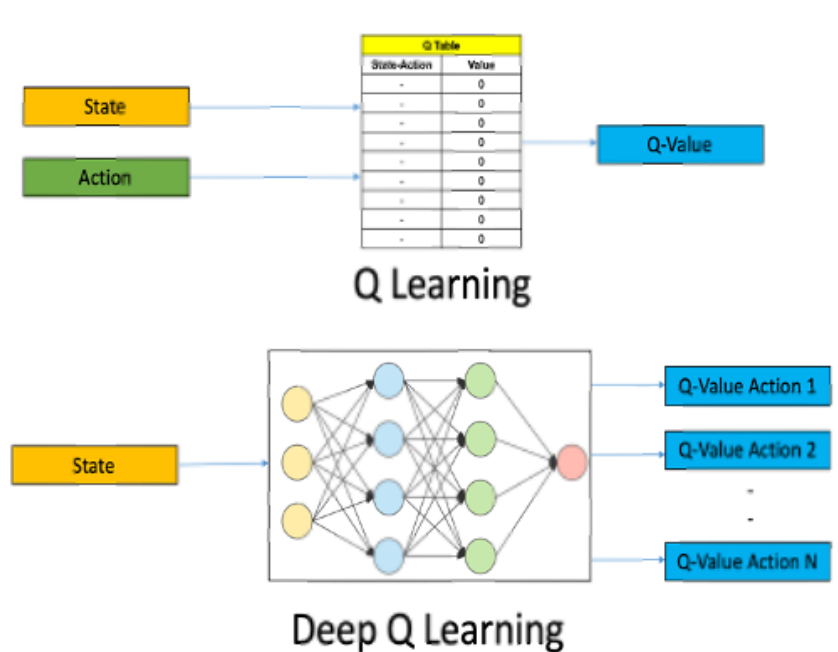
Παρατηρούμε ότι η τιμή της Q δεν εξαρτάται ούτε από τη γνώση της συνάρτησης κέρδους ούτε της συνάρτησης πιθανότητας μετάβασης, παρά μόνο από τις παρατηρήσεις επί του συστήματος. Έχει υπολογιστεί πως με την επιλογή κάθε ενέργειας άπειρες φορές σε μια

εκτέλεση άπειρων βημάτων, και δεδομένου ότι η τιμή μάθησης α (learning rate) μειώνεται με την πάροδο του χρόνου καταλλήλως, οι τιμές Q θα συγκλίνουν στις βέλτιστες τιμές Q^* με πιθανότητα 1 (Watkins et. al 1989, Tsitsiklis et. al 1994, Jaakkola et. al 1994).

Μια σημαντική μέθοδος που χρησιμοποιείται στις παραπάνω διαδικασίες είναι η TD(λ). Η συγκεκριμένη μέθοδος αφορά τον τρόπο με τον οποίο πραγματοποιείται η ανανέωση των τιμών του πίνακα Q , που περιέχει τις τιμές της συνάρτησης για όλους τους δυνατούς συνδυασμούς κατάστασης-ενέργειας. Πιο συγκεκριμένα στους παραπάνω τύπους ορίζεται ο τρόπος που γίνεται ο υπολογισμός των τιμών Q για μια κατάσταση με βάση τον οποίο γίνεται και η ανανέωση των τιμών τους σε πιθανές αλλαγές των τιμών Q των επόμενων καταστάσεων τους. Αυτός ο τύπος λαμβάνει υπόψη του την ανανέωση μόνο της αμέσως προηγούμενης κατάστασης και ορίζεται ως TD(0). Για λ διάφορο του 0 έχουμε διάφορες τεχνικές υπολογισμού μιας συνάρτησης ϵ , που ορίζει οι τιμές Q ποιων καταστάσεων θα ανανεωθούν για κάθε πιθανή αλλαγή τιμής. Ο τύπος που προκύπτει είναι ο εξής:

$$Q(s, a) := Q(s, a) + \alpha \left(r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right) \cdot \epsilon(u)$$

Η χρήση Q-Learning, όπως είδαμε, με τον απαραίτητο αριθμό επαναλήψεων και εκπαίδευσης, μπορεί να οδηγήσει σε βελτιστοποίηση της απόδοσης. Ωστόσο, σε προβλήματα που το πεδίο καταστάσεων αυξάνεται ανεξέλεγκτα, γεγονός που παρατηρείται σε δυναμικά πραγματικά περιβάλλοντα, η μνήμη και η υπολογιστική δύναμη που απαιτούνται καθώς και οι χρονικές καθυστερήσεις στους υπολογισμούς, κάνουν τη μέθοδο Q-Learning ακατάλληλη. Για την προσομοίωση του πίνακα των τιμών Q , χρησιμοποιείται μια μέθοδος που εισάγει την ιδέα των νευρωνικών δικτύων στη συγκεκριμένη διαδικασία μάθησης.



Εικόνα 2.8: Αναπαράσταση του τρόπου εξαγωγής Q Values με τις μεθόδους Q-Learning και Deep Q-Learning αντίστοιχα

Πιο αναλυτικά, η μέθοδος Deep Q-Learning εισάγει ένα νευρωνικό δίκτυο, Deep Q-

Network (DQN), και το εκπαιδεύει να λαμβάνει προσεγγιστικά καλές τιμές X για καταστάσεις που δεν έχει ξαναβρεθεί. Η εκπαίδευση του DQN γίνεται μέσω της τροφοδότησης $\langle s, a, r, s' \rangle$ διανυσμάτων, που περιέχουν δηλαδή την τρέχουσα κατάσταση, την επιλεγμένη ενέργεια, την επιβράβευση του συστήματος και την επόμενη κατάσταση, όπως αυτά έχουν παρατηρηθεί κατά την εκτέλεση του μοντέλου, και διόρθωσης του σφάλματος της πρόβλεψης του DQN για την τιμή Q του συγκεκριμένου $\langle s, a \rangle$ ζεύγους με βάση τον τύπο Bellman:

$$Q(s, a) := r + \gamma \max_{a'} Q(s', a')$$

όπου r η επιβράβευση του συστήματος για το ζεύγος $\langle s, a \rangle$.

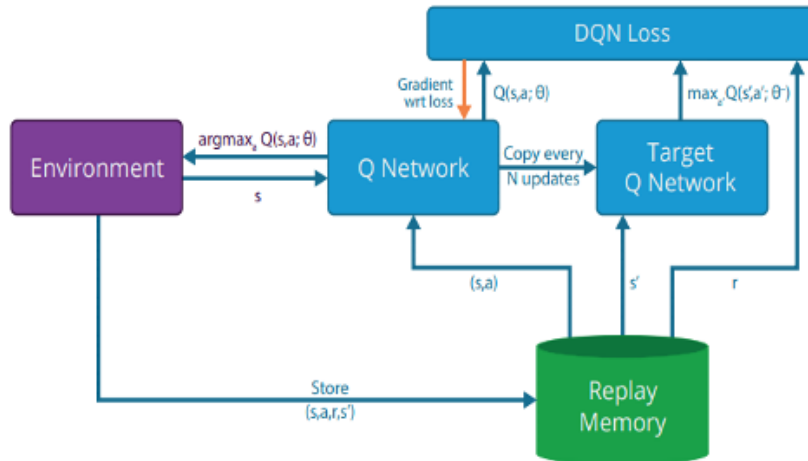
Με τον ορισμό του σφάλματος μέσω κατάλληλων συναρτήσεων και τη διόρθωση του με χρήση μεθόδων όπως είναι η οπισθοδιάδοση (Back Propagation), προσαρμόζονται τα βάρη των νευρώνων του δικτύου με βάση το σφάλμα και απεικονίζονται σωστά οι τιμές Q .

Για τον προσδιορισμό του σφάλματος χρησιμοποιείται η συνάρτηση Mean Squared Error (MSE) ή Συνάρτηση Μέσου Τετραγωνικού Σφάλματος που εκφράζεται ως:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

και αποτελεί δηλαδή το μέσο όρο της τετραγωνικής διαφοράς μεταξύ της Q τιμής που προκύπτει από την εξίσωση Bellman και από την προϋπάρχουσα Q τιμή.

Για τη σταθεροποίηση του συστήματος στις συνεχείς ανανεώσεις των τιμών Q για τις διάφορες καταστάσεις, χρησιμοποιούνται δύο νευρωνικά μοντέλα. Το Q -Network, στο οποίο πραγματοποιούνται οι αλλαγές μετά από κάθε πιθανή σύγκριση Q τιμών, μέσω του αποτελέσματος της εξίσωσης Bellman, και της προϋπάρχουσας τιμής Q για κάποια κατάσταση s . Το δεύτερο μοντέλο είναι το Target Q -Network, που αποτελεί ένα μοντέλο βάση του οποίου υπολογίζονται οι Q τιμές της εξίσωσης Bellman, και το οποίο αντιγράφει τις τιμές και τα βάρη του Q -Network μοντέλου με μικρότερη συχνότητα από την ανανέωσή τους, με σκοπό την σταθεροποίηση των αποτελεσμάτων στις συγκρίσεις που ανανεώνουν τα βάρη του νευρωνικού δικτύου για να μην παρατηρηθούν ανεπιθύμητες αυξομειώσεις.



Εικόνα 2.9: Αναπαράσταση του τρόπου εκπαίδευσης του μοντέλου με χρήση δύο νευρωνικών δικτύων Q-Network και Target Q-Network και replay buffer

Batch Size στη Μηχανική-Ενισχυτική Μάθηση

Στην ενισχυτική μάθηση, το batch size αναφέρεται στον αριθμό των εμπειριών (π.χ., καταστάσεις, δράσεις, ανταμοιβές, και επόμενες καταστάσεις) που χρησιμοποιούνται σε μία ενημέρωση των παραμέτρων του αλγορίθμου. Η επιλογή του μεγέθους παρτίδας μπορεί να επηρεάσει σημαντικά την απόδοση και την αποδοτικότητα της εκπαιδευτικής διαδικασίας.

Ορίζουμε το σύνολο εμπειριών ως

$$\mathcal{D} = \{(s_1, a_1, r_1, s'_1), (s_2, a_2, r_2, s'_2), \dots, (s_N, a_N, r_N, s'_N)\}$$

όπου s_i είναι οι καταστάσεις, a_i οι δράσεις, r_i οι ανταμοιβές, και s'_i οι επόμενες καταστάσεις. Κατά τη διάρκεια της εκπαίδευσης, οι εμπειρίες χωρίζονται σε παρτίδες μεγέθους B . Κάθε παρτίδα αποτελείται από B εμπειρίες:

$$\mathcal{B} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^B$$

Τα μικρά μεγέθη παρτίδας (B) οδηγούν σε περισσότερες ενημερώσεις ανά εποχή και μπορούν να προσφέρουν μια πιο ακριβή εκτίμηση της κλίσης, πιθανώς με αποτέλεσμα μια πιο ακριβή σύγκλιση του μοντέλου. Ωστόσο, αυτό συνοδεύεται από αυξημένο υπολογιστικό φόρτο και μεγαλύτερους χρόνους εκπαίδευσης. Αντιθέτως, τα μεγάλα μεγέθη παρτίδας επωφελούνται από την παράλληλη επεξεργασία και τον μειωμένο χρόνο εκπαίδευσης ανά εποχή, αλλά μπορεί να οδηγήσουν σε λιγότερο ακριβείς εκτιμήσεις της κλίσης και σε υποβέλτιστη σύγκλιση.

Σχετικές εργασίες

Η βέλτιστη σύνδεση και λειτουργία των Cloud-Edge υποδομών αποτελεί ένα καθοριστικό θέμα που απασχολεί τους ερευνητές καθώς η παρουσία των συστημάτων αυτών έχει αφομοιωθεί πλήρως από την σύγχρονη τεχνολογική κατάσταση και μπορεί να επιφέρει σημαντικά αποτελέσματα σε θέματα καθυστέρησης και κόστους [9]. Ωστόσο, η χρήση εδγε υποδομών φέρει κάποιες δυσκολίες συγκριτικά με το “λουδ που συνοψίζονται σε περιορισμένους πόρους για επεξεργασία, ετερογενή συστήματα επεξεργασίας με διαφορετικές αρχιτεκτονικές και φόρτο εργασίας που αλλάζει δυναμικά με το χρόνο [10]. Λόγω αυτών, έχει αυξηθεί σημαντικά η πολυπλοκότητα των συστημάτων και δικτύων και έχει δημιουργηθεί η ανάγκη για καλύτερη διαχείρισή τους.

Η απαίτηση για δυναμική διαχείριση ετερογενών πόρων και δομών δεν ικανοποιείται, όμως, από τα υπάρχοντα δημοφιλή συστήματα ενορχήστρωσής τους, όπως τα Kubernetes και Docker [11][12]. Πιο συγκεκριμένα, προκύπτουν θέματα συμβατότητας καθώς τα Kubernetes και Docker βασίζονται σε ένα σταθερό περιβάλλον για την εκτέλεση των containers και την ενορχήστρωση των deployments τους. Όταν οι edge κόμβοι έχουν διαφορετικό hardware ή λειτουργικά συστήματα, μπορεί να είναι δύσκολο να διασφαλιστεί η συμβατότητα και να αποφευχθούν προβλήματα που μπορεί να επηρεάσουν την απόδοση των εφαρμογών. Στον τομέα της ανάδειξης καταλλήλων αλγορίθμων και μεθόδων διαχείρισης εργασιών προς κατανομή σε μια υποδομή έχουν χρησιμοποιηθεί διάφορες μέθοδοι. Έχουν αναπτυχθεί θεωρητικές προσεγγίσεις που χρησιμοποιούν θεωρίες σχετικά με τον τρόπο αναπαράστασης της ουράς ενός δρομολογητή εργασιών και προβλέψεων της συμπεριφοράς του εισερχόμενου φόρτου. Χαρακτηριστικά, ο Khazaei, et al 2012, χρησιμοποιεί το μοντέλο M/G/m/m+ r για την ουρά του συστήματος [13], ενώ γίνεται χρήση και της αλυσίδας Markov για την αποτύπωση της πιθανοτικής συνάρτησης των χρόνων καθυστέρησης δρομολόγησης εργασιών και του ρυθμού άφιξής τους [14] [15]. Ακόμη, χρησιμοποιούνται ευριστικοί αλγόριθμοι και αναπτύσσονται συναρτήσεις που προσπαθούν να βελτιστοποιήσουν μετρικές του συστήματος, όπως είναι ο particle swarm optimization [16] και ο genetic optimization αλγόριθμος [17][18], σεβόμενοι τις πολύπλοκες αλληλεξαρτήσεις τους.

Σε αυτό το πρόβλημα, η έρευνα και ανάπτυξη AI μοντέλων που βασίζονται σε ML αλγόριθμους φαίνεται να συνεισφέρουν σημαντικά. Οι δυνατότητες που προσφέρουν να νευρωνικά δίκτυα για κατανόηση μη γραμμικών σχέσεων μεταξύ μετρικών του συστήματος, οδηγούν σε σημαντική βελτίωση της απόδοσης. Πιο συγκεκριμένα, τα μοντέλα εκπαιδεύονται με στόχο την βελτίωση των αποτελεσμάτων των παραμέτρων του συστήματος μέσω της βελτιστοποίησης

της κατανομής των πόρων, των δικτυακών στοιχείων και του φόρτου εργασίας στους κόμβους του δικτύου, της πρόβλεψης φόρτου εργασίας, της ανάδειξης σχέσεων μεταξύ του συστήματος και των QoS (Quality of Service) μετρικών του σε real-time δεδομένα, της μείωσης της ενεργειακής κατανάλωσης καθώς και της αναγνώρισης ανωμαλιών του συστήματος.

Η Moreno et al. [2019], παρουσίασε μια μέθοδο αυτόματης κλιμάκωσης πόρων για την προσαρμοστική παροχή ελαστικών υπηρεσιών Edge, βασισμένη στην πρόβλεψη χρονοσειρών και τη θεωρία ουρών αναμονής, με στόχο τη βελτιστοποίηση του χρόνου απόκρισης [19]. Χρησιμοποιούνται Support Vector Regression (SVM), για την πρόβλεψη φόρτου και προσαρμογή των πόρων που ανατίθενται σε κάθε κόμβο. Ομοίως, μια άλλη προσέγγιση πρόβλεψης παρουσιάστηκε από τον Yadav et al. [2021], με χρήση Long Short-Term Memory (LSTM) δικτύου [20]. Μια τεχνική χρήσης Νευρωνικού Δικτύου παρουσιάστηκε από τον Witanto et al. [2018], όπου προτείνει τη δυνατότητα επιλογής διαχείρισης πόρων του συστήματος βάση του τρέχοντος περιβάλλοντος και την προτεραιότητα που δίνεται σε ενεργειακές προϋποθέσεις και SLA παραβιάσεις [21].

Η εργασία μας εστιάζει στη χρήση ενισχυτικής μάθησης ως βάση του μοντέλου ανάπτυξης, που λόγω της δυνατότητας προσαρμογής του σε δυναμικά και πολύπλοκα συστήματα, όπως είναι οι Cloud-Edge υποδομές, βελτιώνει την απόδοση έναντι συμβατικών μοντέλων μηχανικής μάθησης. Οι εργασίες που στοχεύουν σε βελτιστοποίηση ποιοτικών μετρικών Cloud-Edge συστημάτων με χρήση ενισχυτικής μάθησης χρησιμοποιούν σε πολλές περιπτώσεις ως βάση το εισερχόμενο φόρτο εργασίας και προσπαθούν να βελτιστοποιήσουν την κατανομή και δέσμευση πόρων για την εξυπηρέτηση των εργασιών αυτών. Χαρακτηριστικά, ο Jin et al. 2019, πρότεινε έναν αλγόριθμο μάθησης model-free, βασισμένο στην μέθοδο Q-learning, μέσω του οποίου το μοντέλο προσαρμόζεται σε ιδιαιτερότητες του συστήματος, όπως η κίνηση των εργασιών, για να παράγει ενέργειες (από)κλιμάκωσης των πόρων του [22]. Ο Siddesha et al. [2022], χρησιμοποιεί ένα μοντέλο βαθιάς ενισχυτικής μάθησης για την ανάθεση εργασιών σε εικονικά μηχανήματα με στόχο τη βελτιστοποίηση μετρικών του συστήματος όπως ο χρόνος εκτέλεσης των εργασιών, η χρησιμοποίηση των πόρων και η ενεργειακή κατανάλωση [23]. Όσον αφορά συσκευές σε edge, ο Shang et al. [2023], χρησιμοποιεί μια μέθοδος για τη βελτιστοποίηση της απόφασης αποφόρτωσης του συστήματος από την υπολογιστική χρησιμοποίηση [24]. Ο Chen et al. [2022] προτείνει έναν αλγόριθμο στρατηγικής εκφόρτωσης υπολογισμών με βάση το DQN για την ελαχιστοποίηση του μακροπρόθεσμου σταθμισμένου αθροίσματος της καθυστέρησης εκτέλεσης εργασιών και των ενεργειακών απαιτήσεων της υποδομής [25].

Υπάρχουν πολλές προσπάθειες ανάπτυξης μοντέλων βαθιάς ενισχυτικής μάθησης που έχουν ρόλο δρομολογητή εργασιών σε ένα Cloud περιβάλλον. Πιο αναλυτικά, Ο Wei et al. [2018] [14] καθώς και ο Dong et al. [2020] [15], πρότειναν μια προσέγγιση χρονοπρογραμματισμού εργασιών με βάση QoS μετρικές. Ο Li et al. [2022], πρότεινε έναν γενετικό αλγόριθμο προγραμματισμού εργασιών ενισχυτικής μάθησης, ο οποίος μπορεί να χρησιμοποιήσει τη βέλτιστη πολιτική που του δίνεται από μια πηγή αλγορίθμων και μεθόδων, για να καθοδηγήσει τον πράκτορα να μάθει δυναμικό προγραμματισμό εργασιών [17]. Ο Gazori et al. 2020, χρησιμοποίησε μοντέλο διπλής βαθιάς Q-μάθησης, που βοηθά στη σταθεροποίηση των τιμών Q, για τον προγραμματισμό εργασιών που υποβάλλονται από χρήστες και την ελαχιστοποίηση της μέσης καθυστέρησης εξυπηρέτησης μέσω της ανάλυσης υπερπαρα-

μέτρων [18]. Ο Swarup et al. [2021], πρότεινε έναν αλγόριθμο δρομολόγησης εργασιών που βασίζεται σε βαθιά ενισχυτική μάθηση, ο οποίος στοχεύει στη μείωση του κόστους και της καθυστέρησης εξυπηρέτησης εργασιών IoT σε περιβάλλοντα fog [26].

Ο Wang et al. [2021], χρησιμοποίησε ένα πλαίσιο βασισμένο σε DRL για την επίτευξη αποδοτικής κατανομής πόρων στο edge, η οποία επιτεύξε σημαντική εξοικονόμηση ενέργειας μαζί με την ικανοποίηση των απαιτήσεων των χρηστών [27]. Επιπλέον, η εργασία της Javanetti et al. [2023], εισάγει το μοντέλο Actor-Critic DQN, χρησιμοποιώντας πολλαπλούς μεμονωμένους Actors για τη βελτιστοποίηση επιλογών δρομολόγησης εργασιών σε Cloud/Edge [28].

Χρησιμοποιώντας τον αλγόριθμο Deep Q-network (DQN), ο Zhu et al. [2022], ανέπτυξε ένα ιεραρχικό μοντέλο για την δυναμικά προσαρμοστική επιλογή αλγορίθμου κατανομής πόρων σε Cloud υποδομές [29]. Ο Alsarhan et al. [2018] σχεδίασε ένα πλαίσιο SLA βασισμένο σε RL για την εξαγωγή μιας πολιτικής χρήσης VM, η οποία μπορεί να προσαρμοστεί στις δυναμικές αλλαγές του συστήματος και να ικανοποιήσει τις απαιτήσεις QoS διαφορετικών πελατών στο cloud περιβάλλον [30]. Ο Xiong et al. [2020] προτείνει μια μέθοδο κατανομής πόρων βασισμένη σε RL με γνώμονα κέρδους τη μείωση της καθυστέρησης στην κατανομή και εκτέλεση εργασιών [31].

Ο Kuang et al. [2023] χρησιμοποιεί DRL σε συνδυασμό με ένα μοντέλο κατηγοριοποίησης εργασιών με βάση τη συμπεριφορά τους, για τη βελτιστοποίηση ενός συνδυασμού ποιοτικών μετρικών του συστήματος [32]. Παρατηρούμε ότι έχουν γίνει πολλές απόπειρες για καταγραφή των δυναμικών τόσο στο cloud όσο και στο edge, καθώς και στη σχέση μεταξύ τους, μέσω συναρτήσεων κέρδους σε rl μοντέλα, και έχουν εξεταστεί πολλοί παράμετροι προς βελτιστοποίηση, από ενεργειακούς ως χρόνους καθυστέρησης των εργασιών. Ωστόσο, η πλειονότητα των περιπτώσεων εξετάζει το σύστημα που πραγματοποιεί την αρχική κατανομή πόρων/εργασιών και συνήθως εξετάζει τη βελτιστοποίηση μεμονωμένων παραμέτρων του συστήματος.

Εμείς αναπτύξαμε ένα μοντέλο βαθιάς ενισχυτικής μάθησης, που σκοπός του είναι η βελτιστοποίηση των αντιδράσεων του σε επίπεδο διαχείρισης workloads μιας cloud-edge υποδομής με βάση αυξομειώσεις ορισμένων ποιοτικών μετρικών. Σε αντίθεση, με τα περισσότερα εφαρμοσμένα μοντέλα, το δικό μας συνυπολογίζει πληθώρα μετρικών από κόμβους, μηχανήματα και workloads και καλείται να πάρει συγκεντρωτικές ενέργειες για την βελτιστοποίηση του ενός σταθμισμένου συνόλου τους. Ακόμη, δεν εστιάζουμε στην ανάπτυξη μοντέλου που να εκτελεί το ρόλο του δρομολογητή των εργασιών, αλλά θεωρώντας δεδομένη την περίπτωση άφιξης σε μία μη βέλτιστη κατάσταση, τόσο σε επίπεδο υπερεκμετάλλευσης όσο και υποεκμετάλλευσης πόρων του συστήματος και παραβίαση SLA κανόνων, καθώς τα μοντέλα βασίζονται σε συστήματα πρόβλεψης που μπορεί να αποκλίνουν από την πραγματικότητα, δημιουργήσαμε ένα μοντέλο ενισχυτικής μάθησης που να αποκρίνεται σε αυτές τις ανωμαλίες και να εκτελεί τη βέλτιστη ενέργεια με βάση διάφορους παράγοντες που αναλύονται εκτενώς παρακάτω. Τέλος, χρησιμοποιούμε δεδομένα από πραγματική υποδομή ώστε να μπορέσουμε να αποδώσουμε καλύτερη τις σχέσεις και τη συμπεριφορά των εργασιών και των μηχανημάτων. Τέλος, σε αντίθεση με τις περισσότερες έρευνες, ποσοτικοποιούμε τις συσχετίσεις μεταξύ cloud και edge σε ποιοτικές παραμέτρους και εκπαιδεύουμε το μοντέλο να επιλέγει τις καλύτερες δυνατές κινήσεις μεταξύ των διαφορετικών cluster δεδομένων αυτών.

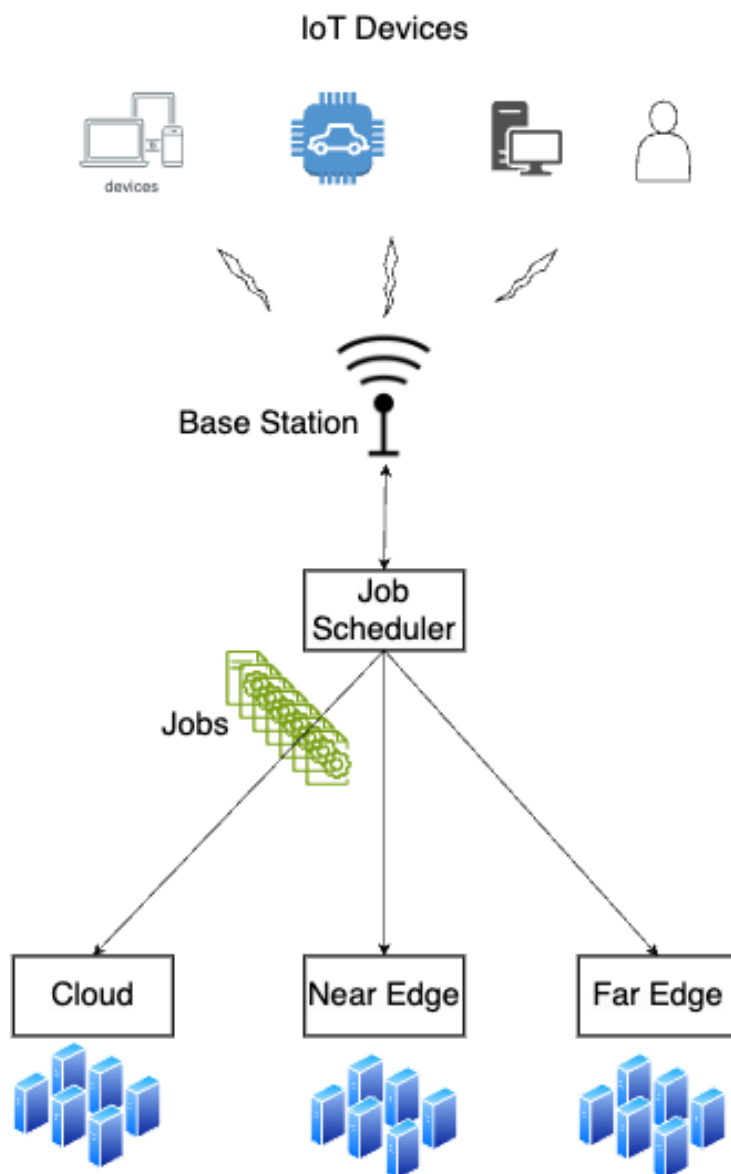
Μέρος 

Πρακτικό Μέρος

Κεφάλαιο **4**

Υποδομή και Εργασίες

Σε αυτό το κεφάλαιο, περιγράφονται αναλυτικά όλες οι εσωτερικές λειτουργίες και παραδοχές του συστήματος που προσομοιώσαμε. Πιο συγκεκριμένα, αναλύονται η υποδομή, τα χαρακτηριστικά των εργασιών και ο τρόπος δρομολόγησής τους.



Εικόνα 4.1: Αναπαράσταση της Cloud-Edge υποδομής.

4.1 Υποδομή

Προσομοιώσαμε μία Cloud-Edge υποδομή που αποτελείται από τρία cluster, το Near Edge, το Far Edge και το Cloud. Κάθε cluster αποτελείται από κόμβους. Κάθε κόμβος αποτελείται από τρία μηχανήματα. Τα μηχανήματα φέρουν συγκεκριμένα χαρακτηριστικά σε πόρους και αποτελούν το μικρότερο στοιχείο υποδομής:

- CPU πυρήνες
- GB RAM

Επίσης, κάθε cluster έχει διαφορετική μέγιστη χωρητικότητα σε κόμβους με βάση τη θέση του στο σύστημα. Το Near Edge έχει τη μικρότερη ενώ το Cloud έχει τη μεγαλύτερη.

Το σύστημα θεωρούμε πως είναι πλήρως καταναμημένο, δηλαδή όλοι οι κόμβοι μπορούν να επικοινωνήσουν με όλους, ανεξαρτήτου cluster και θέσης μέσα στο δίκτυο.

4.2 Εργασίες

Οι εργασίες που εκτελούνται στο σύστημά μας, όπως περιγράφηκε και στη θεωρία, έχουν τη δομή Job-Task-Task Instance. Κάθε Job αποτελείται από Tasks, τα οποία έχουν διαφορετικό ρόλο-λειτουργία και απαιτήσεις μεταξύ τους σε CPU και RAM, και τα οποία όταν ολοκληρώνονται, ολοκληρώνεται το Job. Τα task instances, απεναντίας, που συνιστούν το κάθε Task, έχουν τις ίδιες απαιτήσεις μεταξύ τους. Κάθε Task Instance φέρει ορισμένες προδιαγραφές/απαιτήσεις σε:

- CPU πυρήνες
- GB RAM

4.3 Δομολόγηση Εργασιών

Αναπτύξαμε ένα σύστημα δρομολόγησης εργασιών, κατά το οποίο όταν φτάνει στο κεντρικό σύστημα μια εργασία, αυτό την καταναίμει τυχαία σε ένα εκ των τριών clusters. Στη συνέχεια, χρησιμοποιείται ένας αλγόριθμος αντιστοίχισης των τασκ ινστανσες σε κάποιο μηχάνημα ενός κόμβου, στον οποίο να μπορεί να εξυπηρετηθεί χωρίς να αναμένει. Οι αλγόριθμοι που χρησιμοποιούνται είναι ο First και ο Best Fit. Στην περίπτωση που δεν υπάρχει κάποιο διαθέσιμο μηχάνημα προς εξυπηρέτηση της εργασίας, αυτό διατηρείται στην ουρά του κεντρικού δρομολογητή, και αυξάνεται ο χρόνος καθυστέρησης εκτέλεσης, μέχρι να βρεθεί κάποιος διαθέσιμος πόρος.

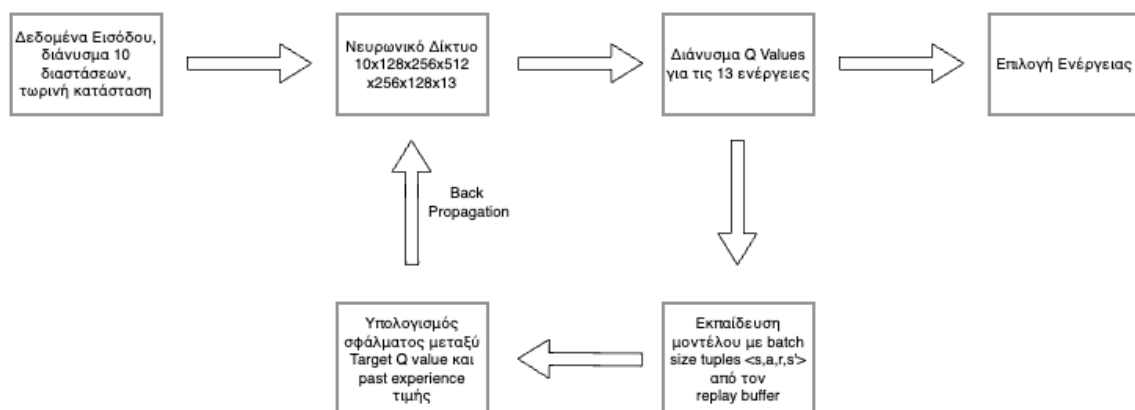
Κεφάλαιο 5

Μοντέλο Ενισχυτικής Μάθησης

Σε αυτό το στάδιο εξηγούμε τη διαδικασία εκπαίδευσης και τις ακριβείς συνθήκες του μοντέλου ενισχυτικής μάθησης, με βάση την κατάσταση του συστήματος, να παίρνει ενέργειες που να κάνουν το σύστημα πιο αποδοτικό σε επίπεδο χρησιμοποίησης πόρων και κατανομής φόρτου εργασίας.

Εκπαιδεύουμε ένα Deep Q Learning model να παίρνει συγκεντρωτικά ενέργειες για το σύστημα και να λαμβάνει μια συγκεντρωτική επιβράβευση που θα είναι το άθροισμα των συναρτήσεων κέρδους των επιμέρους κόμβων. Το μοντέλο θα εκπαιδεύεται να είναι πιο ακριβές στις προβλέψεις του με updates βάση υπολογισμού της διαφοράς μεταξύ των προβλέψεων του μοντέλου στις Q τιμές και υπολογισμού των πραγματικών τιμών τους, μέσω των κατάλληλων εξισώσεων που αναλύονται στο κεφ. 2, σε στιγμιότυπα παρελθοντικών στιγμών που έχουν αποθηκευτεί σε replay buffer με την εξής tuple μορφή $\langle S, a, r, S' \rangle$, όπου :

- S : η τρέχουσα κατάσταση του συστήματος
- a : η επιλεγμένη ενέργεια
- r : η αθροιστική τιμή των συναρτήσεων κέρδους του συστήματος
- S' : η επόμενη κατάσταση του συστήματος



Εικόνα 5.1: Αναπαράσταση της λειτουργίας του DRL μοντέλου

Σε αυτή τη διαδικασία χρησιμοποιείται και ένα target Q network, που είναι αντίγραφο του Q network και ανανεώνει τις τιμές των παραμέτρων του δικτύου με πιο αργό ρυθμό από

το δεύτερο, προκειμένου να αποφύγει παροδικές συσχετίσεις των δεδομένων. Θεωρούμε ότι το μοντέλο μας υπακούει στους νόμους της Markov Decision Process. Πιο συγκεκριμένα, σε κάθε χρονικό βήμα, η διαδικασία βρίσκεται σε κάποια κατάσταση s , και ο λήπτης αποφάσεων μπορεί να επιλέξει οποιαδήποτε ενέργεια a που είναι διαθέσιμη στην κατάσταση s . Το μοντέλο ανταποκρίνεται μετακινούμενη σε μια νέα κατάσταση, s' , και δίνει στο μοντέλο μια αντίστοιχη ανταμοιβή $R(s,s')$. Η πιθανότητα να μεταβεί η διαδικασία στη νέα της κατάσταση s' επηρεάζεται από την επιλεγμένη ενέργεια και την παρούσα κατάσταση s . Συμπεραίνουμε, ότι το μοντέλο είναι ανεξάρτητο από όλες τις προηγούμενες καταστάσεις και ενέργειες, και έτσι οι μεταβάσεις καταστάσεων ενός MDP ικανοποιούν την ιδιότητα Markov. Για να οριστεί το RL μοντέλο καλούμαστε, λοιπόν, να ορίσουμε τα εξής πεδία :

A. Πεδίο καταστάσεων Το πεδίο καταστάσεων αποτελεί τον τρόπο που ορίζουμε ολοκληρωτικά την κατάσταση του συστήματος μας την τρέχουσα χρονική στιγμή. Το μοντέλο αποδομεί τα στοιχεία που χρειάζεται να γνωρίζει από τα επιμέρους clusters καθώς τα στοιχεία που τον ενδιαφέρουν για τις αποφάσεις που καλείται να πάρει είναι :

- Ποσοστά χρησιμοποίησης πόρων των 3 clusters (CPU cores, GB of RAM)
- Μέσος όρος Response time (RT) των 3 mid-level επιπέδων
- Πλήθος από workloads που εκτελούνται στα 3 mid-level επίπεδα
- Πλήθος από workloads που βρίσκονται σε ουρά δρομολογητών

B. Πεδίο ενεργειών Το πεδίο ενεργειών του συστήματος αποτελεί τις επιλογές που έχει το σύστημα για να αντιμετωπίσει ανωμαλίες που διακρίνει στο πεδίο καταστάσεων. Έτσι, συγκεντρωτικά για το μοντέλο μας έχουμε :

- Κενή ενέργεια, η οποία δηλώνει την επιλογή ότι η κατάσταση δε χρειάζεται κάποια αλλαγή
- Near-Edge domain Scale Up - Αύξηση των δεσμευμένων κόμβων στο Near-Edge domain κατά 5 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον δεν έχει φτάσει στο όριο των δυνατοτήτων του.
- Near-Edge domain Scale Down - Μείωση των δεσμευμένων κόμβων του Near-Edge domain κατά 5 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον υπάρχει.
- Far-Edge domain Scale Up - Αύξηση των δεσμευμένων κόμβων στο Far-Edge domain κατά 5 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον δεν έχει φτάσει στο όριο των δυνατοτήτων του.
- Far-Edge domain Scale Down - Μείωση των δεσμευμένων κόμβων του Far-Edge domain κατά 5 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον υπάρχει.

- Cloud domain Scale Up - Αύξηση των δεσμευμένων κόμβων στο Cloud domain κατά 5 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον δεν έχει φτάσει στο όριο των δυνατοτήτων του.
- Cloud domain Scale Down - Μείωση των δεσμευμένων κόμβων του Cloud domain κατά 5 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον υπάρχει.
- Μεταφορά 10 workloads από το Near-Edge στο Far-Edge domain
- Μεταφορά 10 workloads από το Near-Edge στο Cloud domain
- Μεταφορά 10 workloads από το Far-Edge στο Near-Edge domain
- Μεταφορά 10 workloads από το Far-Edge στο Cloud domain
- Μεταφορά 10 workloads από το Cloud domain στο Near-Edge domain
- Μεταφορά 10 workloads από το Cloud domain στο Far-Edge domain

Για τη διαχείριση των παραπάνω ενεργειών σε επίπεδο κόμβων και μηχανημάτων θεωρούμε πως η επιλογή κόμβων προς καταστροφή είναι τυχαία, ενώ η επιλογή κόμβων για αποστολή και λήψη εργασιών προς εκτέλεση επιλέγεται με βάση την μέγιστη ή ελάχιστη χρήση των πόρων του κόμβου αντίστοιχα. Ακόμη, σημειώνουμε πως κατά την καταστροφή κάποιου κόμβου που φέρει ενεργές εργασίες, αυτές αρχικοποιούνται και μεταφέρονται στον δρομολογητή του συγκεκριμένου cluster.

Γ. Συνάρτηση επιβράβευσης Η συνάρτηση επιβράβευσης αποτελεί την ανταπόκριση του συστήματος με βάση τις παρατηρούμενες αλλαγές και συνθήκες του περιβάλλοντος στην επόμενη κατάσταση και αποτελεί τον τρόπο που εκπαιδεύεται το μοντέλο μας να παίρνει αποδοτικές ενέργειες.

Η συνάρτηση έχει τρία μέρη:

1ο Μέρος Το πρώτο μέρος αφορά την αποδοτική εκμετάλλευση των ενεργών πόρων, δηλαδή την επιβράβευση του συστήματος ώστε να διατηρεί ενεργά τα ελάχιστα δυνατά μέσα κατά την λειτουργία του.

Ενώ από τη μία, η ύπαρξη ενός ελλιπώς εκμεταλλεμένου περιβάλλοντος μπορεί να προσφέρει στο σύστημα υψηλές QoS μετρικές (χαμηλό response time), η σπατάλη πόρων δεν είναι αποδεκτή από τους πελάτες σε cloud-edge υποδομές. Η υπο-εκμετάλλευση διαθέσιμων πόρων αυξάνει το κόστος από την άποψη της χρηματικής αξίας καθώς και της κατανάλωσης ενέργειας. Ως εκ τούτου, πρέπει να λάβουμε υπόψη μας τους χρησιμοποιήσιμους αναλογικά με τους διαθέσιμους πόρους κάθε κόμβου προκειμένου να επιβραβεύουμε περισσότερο την υψηλή εκμετάλλευσή τους. Η τιμή u_j αποτελεί την μετρική που εκφράζει το ποσοστό χρησιμοποίησης των πόρων του κόμβου και το U_{jmax} ορίζει τη μέγιστη αποδεκτή χρησιμοποίηση για τον αντίστοιχο πόρο θ (και οι δύο μετρικές παίρνουν τιμές στην κλίμακα 0 με 1). Έστω, επίσης, ότι είναι N οι κόμβοι που έχει ως είσοδο το μοντέλο μας. Υψηλότερα ποσοστά, στο πλαίσιο των φυσιολογικών τιμών, χρησιμοποίησης των πόρων έχουν θετικές επιπτώσεις στην τελική τιμή ανταμοιβής. Ωστόσο, εάν η χρησιμοποίηση παραβιάζει το μέγιστο όριο, η τιμή

του R_{rt} γίνεται αρνητική στους αντίστοιχους κόμβους και έτσι επηρεάζει αρνητικά την τελική ανταμοιβή.

$$R = \frac{\sum_{j=1}^N (U_j^{\max} - u_j)}{N} \quad (5.1)$$

2ο Μερσ Το δεύτερο μέρος αφορά τον χρόνο αναμονής κάθε αιτήματος προς εξυπηρέτηση από πόρους της υποδομής μας.

Έστω rt ο μέσος χρόνος απόκρισης των αιτημάτων κατά τη διάρκεια του χρονικού διαστήματος t έως $t+1$. Τότε, η ανταμοιβή του rt υπολογίζεται με βάση την εξίσωση 5.2 όπου:

RT_{\max} και RT_{\min} είναι οι μέγιστες και ελάχιστες αποδεκτές τιμές για φυσιολογική λειτουργία του συστήματος. Η συνθήκη ελάχιστης τιμής είναι για να ληφθούν υπόψη περιπτώσεις που μια εφαρμογή φτάνει σε μη ανταποκρίσιμη κατάσταση, στην οποία δε μπορεί να εξυπηρετηθεί κανένα αίτημα, με αποτέλεσμα το rt να πέφτει σε σχεδόν μηδενική τιμή. Όταν το rt έχει μια τιμή μεταξύ των \min και \max κατωφλιών και επομένως ικανοποιεί τις φυσιολογικές συνθήκες, η ανταμοιβή θα είναι 1. Όταν το rt παραβιάζει το άνω όριο, η συνάρτηση θα οδηγείται σε μηδενικές τιμές.

$$R_{rt}(rt) = \begin{cases} e^{-\left(\frac{rt-RT_{\max}}{RT_{\max}}\right)^2}, & \text{if } rt > RT_{\max} \\ e^{-\left(\frac{rt-RT_{\min}}{RT_{\min}}\right)^2}, & \text{if } rt < RT_{\min} \\ 1, & \text{otherwise} \end{cases}$$

(5.2)

3ο Μερσ Το τρίτο μέρος αφορά το κόστος δέσμευσης κάθε μηχανήματος/κόμβου της υποδομής μας για κάθε δεδομένη χρονική στιγμή.

Για κάθε χρονική στιγμή (time interval) που παίρνουμε ανταπόκριση από το σύστημά μας και θεωρούμε ως το ελάχιστο χρονικό διάστημα κατά το οποίο είναι δεσμευμένα μηχανήματα από την υποδομή μας, ανάγουμε το κόστος δέσμευσης μηχανημάτων σε κάθε φυσικό μέρος του δικτύου μας με βάση αληθινές τιμές.

Οι τιμές για τα σημεία δέσμευσης πόρων διαφέρουν καθώς η διαθεσιμότητα πόρων στο Edge είναι πολύ πιο περιορισμένη συγκριτικά με τις μαζικές υποδομές που διατίθενται στο Cloud. Πιο συγκεκριμένα, στο σύστημα μας χρησιμοποιούμε τιμές δέσμευσης μηχανημάτων σε Near Edge, Far Edge και Cloud υποδομές σε ευρώ, τις οποίες έχουμε λάβει από το δημόσιο τιμοκατάλογο υπηρεσιών της Amazon με βάση τα χαρακτηριστικά των μηχανημάτων της υποδομής μας και ως βάση μας την Ελλάδα. Οι τιμές που χρησιμοποιήσαμε είναι οι εξής:

- Κόστος χρησιμοποίησης Near Edge μηχανήματος ανά ώρα: 0.305 €
- Κόστος χρησιμοποίησης Far Edge μηχανήματος ανά ώρα: 0.24 €
- Κόστος χρησιμοποίησης Cloud μηχανήματος ανά ώρα: 0.1645 €

Κεφάλαιο 6

Συγκεντρωτική περιγραφή του προβλήματος

Το πρόβλημα που λύνουμε είναι η ανάπτυξη ενός μοντέλου βαθιάς ενισχυτικής μάθησης που αντιδράει σε πιθανές καταστάσεις μιας Cloud-Edge υποδομής, με διαθέσιμες επιλογές ενεργειών ανακατανομής εργασιών και πόρων μεταξύ των clusters, με στόχο τη βελτιστοποίηση ενός συνδυασμού μετρικών QoS, χρήσης πόρων και κόστους του συστήματος. Πιο αναλυτικά, αναπτύξαμε ένα σύστημα δρομολόγησης εργασιών, κατά το οποίο όταν φτάνει στο κεντρικό σύστημα μια εργασία, αυτό την καταναίμει τυχαία σε ένα εκ των τριών clusters. Στη συνέχεια, χρησιμοποιείται ένας αλγόριθμος αντιστοίχισης των task instances σε κάποιο μηχάνημα ενός κόμβου, στον οποίο να μπορεί να εξυπηρετηθεί χωρίς να αναμένει. Οι αλγόριθμοι που χρησιμοποιούνται είναι ο First και ο Best Fit. Στην περίπτωση που δεν υπάρχει κάποιο διαθέσιμο μηχάνημα προς εξυπηρέτηση της εργασίας, αυτό διατηρείται στην ουρά του κεντρικού δρομολογητή, και αυξάνεται ο χρόνος καθυστέρησης εκτέλεσης, μέχρι να βρεθεί κάποιος διαθέσιμος πόρος.

Κατά την εκκίνηση του δρομολογητή μας, κάθε 5 χρονικές στιγμές σταματάμε τη λειτουργία του και την εκτέλεση των εργασιών του συστήματος με σκοπό την χρησιμοποίηση του μοντέλου ενισχυτικής μάθησης για την πραγματοποίηση ενεργειών.

Το μοντέλο ενισχυτικής μάθησης είναι ένα Deep Q Learning model που παίρνει συγκεντρωτικά ενέργειες για το σύστημα και λαμβάνει μια συγκεντρωτική επιβράβευση που είναι το άθροισμα των συναρτήσεων κέρδους των επιμέρους κόμβων. Το μοντέλο θα εκπαιδεύεται να είναι πιο ακριβές στις προβλέψεις του με updates βάση υπολογισμού της διαφοράς μεταξύ των προβλέψεων του μοντέλου στις Q τιμές και υπολογισμού των πραγματικών τιμών τους, μέσω των κατάλληλων εξισώσεων, σε στιγμιότυπα παρελθοντικών στιγμών που έχουν αποθηκευτεί σε replay buffer με την εξής tuple μορφή $\langle S, a, r, S' \rangle$, όπου:

- S: η τρέχουσα κατάσταση του συστήματος
- a: η επιλεγμένη ενέργεια
- r: η αθροιστική τιμή των συναρτήσεων κέρδους του συστήματος
- S': η επόμενη κατάσταση του συστήματος

Σε αυτή τη διαδικασία χρησιμοποιείται και ένα target Q network, που είναι αντίγραφο του Q network και ανανεώνει τις τιμές των παραμέτρων του δικτύου με πιο αργό ρυθμό από το δεύτερο, προκειμένου να αποφεύγει παροδικές συσχετίσεις των δεδομένων.

Θεωρούμε ότι το μοντέλο μας υπακούει στους νόμους της Markov Decision Process. Πιο συγκεκριμένα, σε κάθε χρονικό βήμα, η διαδικασία βρίσκεται σε κάποια κατάσταση s , και ο λήπτης αποφάσεων μπορεί να επιλέξει οποιαδήποτε ενέργεια a που είναι διαθέσιμη στην κατάσταση s . Το μοντέλο ανταποκρίνεται μετακινούμενη σε μια νέα κατάσταση, s' , και δίνει στο μοντέλο μια αντίστοιχη ανταμοιβή $R(s,s')$. Η πιθανότητα να μεταβεί η διαδικασία στη νέα της κατάσταση s' επηρεάζεται από την επιλεγμένη ενέργεια και την παρούσα κατάσταση s . Συμπεραίνουμε, ότι το μοντέλο είναι ανεξάρτητο από όλες τις προηγούμενες καταστάσεις και ενέργειες, και έτσι οι μεταβάσεις καταστάσεων ενός MDP ικανοποιούν την ιδιότητα Markov. Έτσι, σε κάθε χρονική παύση το μοντέλο μας επιλέγει κάποια κίνηση, την οποία εκτελεί το σύστημα μας. Ταυτόχρονα, καλείται η συνάρτηση εκπαίδευσης του μοντέλου μας, κατά τη χρήση της οποίας το μοντέλο παίρνει batch size αριθμό δειγμάτων $\langle S,a,r,S' \rangle$ από το replay buffer που έχουμε δημιουργήσει, και ακολούθως συγκρίνει τις τιμές από το target Q network για ένα συνδυασμό κατάστασης-ενέργειας με το αποτέλεσμα από την εξίσωση:

$$Q(s, a) := r + \gamma \max_{a'} Q(s', a')$$

η οποία χρησιμοποιεί ως $Q(s,a)$, την τιμή του Q network για το 'state-action' ζευγάρι, την τιμή του γ , όπως ορίζεται στην εκάστοτε προσομοίωση στο εύρος $[0,1]$, και η οποία αποτελεί τη μετρική συνεισφοράς μελλοντικών επιβραβεύσεων του συστήματος στην τωρινή κατάσταση. Τέλος, το $Q(s',a')$ αποτελεί τη μέγιστη Q value της επόμενης κατάστασης s' του συστήματος. Αφού λάβουμε τα δύο αποτελέσματα, χρησιμοποιούμε τη Mean Squared Error (MSE) ή Συνάρτηση Μέσου Τετραγωνικού Σφάλματος που εκφράζεται ως:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

και αποτελεί δηλαδή το μέσο όρο της τετραγωνικής διαφοράς μεταξύ της Q τιμής που προκύπτει από την εξίσωση Bellman και από την προϋπάρχουσα Q τιμή. Με βάση το αποτέλεσμα αυτής της συνάρτησης εφαρμόζουμε Back Propagation στο νευρωνικό μας δίκτυο για την εκπαίδευση των βαρών των νευρώνων για τις αλλαγές που πρέπει να πραγματοποιηθούν.

Αξίζει να σημειωθεί πως συγκρίνουμε την τιμή του target Q network με το αποτέλεσμα της εξίσωσης της DQL, λόγω της μεγαλύτερης σταθερότητας στις τιμές που λαμβάνει, καθώς δεν ανανεώνονται οι τιμές του σε κάθε χρονική παύση, όπως συμβαίνει στο Q network, αλλά κάθε 10 χρονικές στιγμές.

Κεφάλαιο **7**

Υλοποίηση

Στην υλοποίησή μας, θεωρήσαμε μία υποδομή τριών clusters, Near Edge, Far Edge και Cloud. Κάθε cluster διαθέτει κόμβους. Θεωρήσαμε ως μέγιστη χωρητικότητα μηχανημάτων σε κάθε cluster, τις εξής τιμές:

- Near Edge: 141 μηχανήματα
- Far Edge: 192 μηχανήματα
- Cloud: 282 μηχανήματα

Θεωρούμε, ακόμη, ως είσοδο στο σύστημά μας τα δεδομένα από μια πραγματική Cloud υποδομή της Alibaba [33]. Οι εργασίες του συστήματος έχουν τα εξής χαρακτηριστικά:

- CPU: 0.1 - 0.5 cores
- RAM: 0.1 - 0.5 GB
- Execution time: 0.1 - 1 seconds

Όλος ο όγκος εργασιών καταφθάνει ταυτόχρονα στον κεντρικό δρομολογητή του συστήματος. Για την ανάπτυξη του προσομοιωμένου περιβάλλοντος του δρομολογητή χρησιμοποιήσαμε την python 3.11 και τη βιβλιοθήκη simpy. Η αρχική ανάπτυξη του δρομολογητή, πάνω στην οποία βασιστήκαμε και εξελίξαμε πολύ για να ανταπεξέρχεται στις απαιτήσεις του προβλήματός μας ανήκει στον FC-Li [34]. Βάση αρχών των συγκεκριμένων εργαλείων, κάθε 5 χρονικές στιγμές σταματάμε τη λειτουργία του δρομολογητή και την εκτέλεση των εργασιών του συστήματος με σκοπό την χρησιμοποίηση του μοντέλου ενισχυτικής μάθησης για την πραγματοποίηση ενεργειών. Για την ανάπτυξη του μοντέλου μας, χρησιμοποιήσαμε την python 3.11 και τη βιβλιοθήκη torch, σε συνδυασμό με το βοηθητικό εργαλείο numpy.

Η αρχιτεκτονική των δύο νευρωνικών δικτύων που χρησιμοποιούνται από το σύστημα μας, είναι κοινή και παρουσιάζει την εξής εσωτερική δομή:

- 6 επίπεδα νευρώνων με αριθμό νευρώνων εξόδου του κάθε εσωτερικού επιπέδου: 128x256x512x256x128

Λόγω του μεγάλου αριθμού καταστάσεων του μοντέλου μας και του όγκου δεδομένων προς εκπαίδευσή του, θεωρήσαμε απαραίτητη τη χρήση ενός μοντέλου με πολλαπλά επίπεδα και νευρώνες για την αναγνώριση σχέσεων και σημαντικών πληροφοριών. Ο αριθμός των επιπέδων και το πλήθος των νευρώνων επιλέχθηκε ύστερα από έρευνα με βάση των αριθμό των καταστάσεων-ενεργειών του μοντέλου σε προσεγγίσεις άλλων εργασιών, καθώς και από την τελική επιλογή του μεταξύ τριών αρχιτεκτονικών λύσεων στις οποίες εκπαιδεύτηκε το μοντέλο μας για μικρό αριθμό επεισοδίων. Οι δομές που εξετάστηκαν ήταν οι εξής:

- 128x256x512x256x128
- 256x512x1024x512x256
- 64x128x128x64

Επιπλέον, στη λύση μας θεωρούμε την εξής καθυστέρηση εύρεσης διακομιστή για εκτέλεση μιας εργασίας και επικοινωνίας από άκρη σε άκρη (RTT) για τα τρία διαφορετικά clusters με βάση αποτελέσματα από την έρευνα του Charyyev et al. [2020] [35]:

- Near Edge: 5 ms
- Far Edge: 20 ms
- Cloud: 60 ms

Όσον αφορά το μοντέλο ενισχυτικής μάθησης, κάναμε τις εξής παραδοχές:

- Το πεδίο καταστάσεων αποτελείται από 10 τιμές που δίνονται ως είσοδος στο νευρωνικό μας δίκτυο. Για απλοποίηση του μεγέθους του συνόλου τιμών του πεδίου, αρχικά κανονικοποιήσαμε τις συνεχείς τιμές στο πλαίσιο [0,1] και ακολούθως ορίσαμε κάποια διακριτά όρια-thresholds για την κατηγοριοποίηση τους. Με βάση αυτά τα όρια ο αριθμός των διαθέσιμων τιμών για κάθε μεταβλητή του πεδίου κατάστασης είναι $6 \times 6 \times 6 \times 2 \times 6 \times 6 \times 6 \times 2 \times 2 \times 2$. Προσεγγιστικά, το πεδίο τιμών αποτελείται από 746.500 πιθανές καταστάσεις.
- Όταν μεταφέρονται Jobs μεταξύ διαφορετικών clusters, τα Task Instances που έχουν ολοκληρωθεί, καθώς και αυτά που διακόπηκαν κατά την εκτέλεσή τους, αρχικοποιούνται και εκτελούνται από την αρχή, ενώ ο χρόνος αναμονής τους διατηρείται.
- Οι τιμές της συνάρτησης κέρδους, που αποτελεί την επιβράβευση του συστήματος, έχουν ως πεδίο τιμών το διακριτό διάστημα [-3, -2, -1, -0.5, 0, 0.5, 1, 2, 3, 5] με βάση τη σύγκριση της τιμής του αθροίσματος των επιμέρους συναρτήσεων κέρδους μιας κατάστασης συγκριτικά με την τιμή της προηγούμενης.

Ο κώδικας όλου του συστήματος που αναπτύξαμε και χρησιμοποιήθηκε για την εξαγωγή των αποτελεσμάτων βρίσκεται στο εξής Github repository [36].

Αποτελέσματα

8.1 Παράμετροι Προσομοίωσης

Προχωρώντας στο πειραματικό μέρος της εργασίας, θα αξιολογήσουμε την επίδοση του συστήματος προσομοιώνοντας ένα περιβάλλον Cloud-Edge υποδομής με τα χαρακτηριστικά που ορίστηκαν στο προηγούμενο κεφάλαιο. Τα δύο νευρωνικά δίκτυα που χρησιμοποιήθηκαν κατά την υλοποίηση του μοντέλου ενισχυτικής μάθησης, Q-network και Target Q-network έχουν ως είσοδο τη κατάσταση si μεγέθους $|si| = 10$ και ως έξοδο το πλήθος των δυνατών ενεργειών σε κάθε κατάσταση, όπου είναι $|a| = 13$, όπως προκύπτει από την ανάλυση του μοντέλου στο 5ο κεφάλαιο. Ακόμη, τα επίπεδα των νευρωνικών δικτύων που χρησιμοποιήθηκαν είναι fully connected (FC).

Η σύγκριση των αποτελεσμάτων του μοντέλου μας γίνεται μεταξύ των ακόλουθων:

- Δρομολογητής εργασιών που χρησιμοποιεί τον αλγόριθμο First Fit
- Δρομολογητής εργασιών που χρησιμοποιεί τον αλγόριθμο Best Fit
- Δρομολογητής εργασιών που χρησιμοποιεί τον αλγόριθμο First Fit και ένα μοντέλο ενισχυτικής μάθησης που λαμβάνει υπόψη του επιπρόσθετες καθυστερήσεις εκτέλεσης εργασιών στα clusters της υποδομής

Επιπλέον, αναλύουμε τα αποτελέσματα του μοντέλου μας με σταθερή και μεταβλητή τιμή της σταθεράς εξερεύνησης ϵ , του πεδίου ενεργειών μας (Q Values), κατά την εκπαίδευσή του καθώς και σε σχέση με τον αριθμό των επεισοδίων στα οποία έχει εκπαιδευτεί.

Ακόμη, για την εκπαίδευση των μοντέλων μας εκτελέσαμε 1200 επεισόδια στο καθένα, όπου το κάθε επεισόδιο αποτελείται από τη δρομολόγηση και εκτέλεση 100 εργασιών στο σύστημα. Το επεισόδιο ολοκληρώνεται όταν εκτελεστούν όλες οι εργασίες.

Τέλος, ορίζουμε στην κατάσταση του συστήματος μια boolean μεταβλητή που περιγράφει αν ο μέσος όρος καθυστέρησης των εργασιών του συστήματος έχουν ξεπεράσει κάποιο όριο που ορίζουμε εμείς. Με βάση την τιμή αυτής της μεταβλητής, το μοντέλο μας αντιδράει μειώνοντας ή αυξάνοντας τους πόρους του συστήματος. Εξετάζουμε τα αποτελέσματα του μοντέλου για τρία διαφορετικά χρονικά όρια, 100, 200 και 400 ms.

Παράμετρος	Τιμή
CPU Εργασίας	0.1 - 0.5 cores
RAM Εργασίας	0.1 - 0.5 GB
Χρόνος Εκτέλεσης εργασίας	0.01 - 0.1 sec
Προϋπάρχων Χρόνος Καθυστερήσης Εκτέλεσης εργασίας	0.01 - 0.1 sec
Μέγιστη χωρητικότητα μηχανημάτων σε CPU	3 cores
Μέγιστη χωρητικότητα μηχανημάτων σε RAM	3 GB
Μέγιστο πλήθος μηχανημάτων στο Near Edge	141
Μέγιστο πλήθος μηχανημάτων στο Far Edge	192
Μέγιστο πλήθος μηχανημάτων στο Cloud	282
Επεισόδια, E	1200
χρονικό όριο	[1000, 2000, 4000]
Learning rate, lr	0.0001
Learning rate decay	0.9
Exploration rate, e	0.2 - 0.7
Exploration rate, e decay	0.995 - 1.0
Q-Network update frequency σε βήματα	2
Target Q-Network update frequency σε βήματα	10
Buffer Size	2000
Batch Size	16
Gamma, γ	0.9

8.2 Ανάλυση Αποτελεσμάτων

Για την παρακολούθηση της διαδικασίας κρατάμε ένα δείγμα σε κάθε επεισόδιο με μετρικές απόδοσης του συστήματος για την εκτέλεση συγκεκριμένου αριθμού εργασιών. Πραγματοποιήσαμε βασικές περιπτώσεις δρομολόγησης εργασιών που είχαν ως αλγορίθμους κατανομής εργασιών σε πόρους του συστήματος, τους First Fit και Best Fit. Ακόμη, χρησιμοποιήσαμε το μοντέλο ενισχυτικής μάθησης με τα δύο νευρωνικά δίκτυα και τα χαρακτηριστικά τους, όπως αναλύθηκαν παραπάνω. Αναπτύξαμε ένα μοντέλο που λαμβάνει υπόψη του τις επιπρόσθετες καθυστερήσεις εύρεσης διακομιστή και τον RTT χρόνο, ανάλογα με το εκάστοτε επίπεδο cluster.

Οι μετρικές απόδοσης του συστήματος, με βάση τις οποίες αξιολογούμε τα αποτελέσματα του συστήματος και των μοντέλων μας είναι οι εξής:

- Συνολικός χρόνος εκτέλεσης των εργασιών
- Μέσο κόστος δέσμευσης μηχανημάτων σε ευρώ
- Μέσος χρόνος αναμονής των εργασιών στο σύστημα
- Μέση χρησιμοποίηση των πόρων κάθε cluster της υποδομής (συγκριτικά με τη μέγιστη χωρητικότητα της υποδομής)

Παρακάτω, πρώτα αναλύουμε αποτελέσματα από την εκπαίδευση των μοντέλων σε σχέση με τον αριθμό των επεισοδίων στα οποία έχουν εκπαιδευτεί. Ακολούθως, συγκρίνουμε τα τελικά μοντέλα με τα baseline μοντέλα καθώς και μεταξύ τους.

8.2.1 DRL μοντέλα με χρονικές καθυστερήσεις των clusters

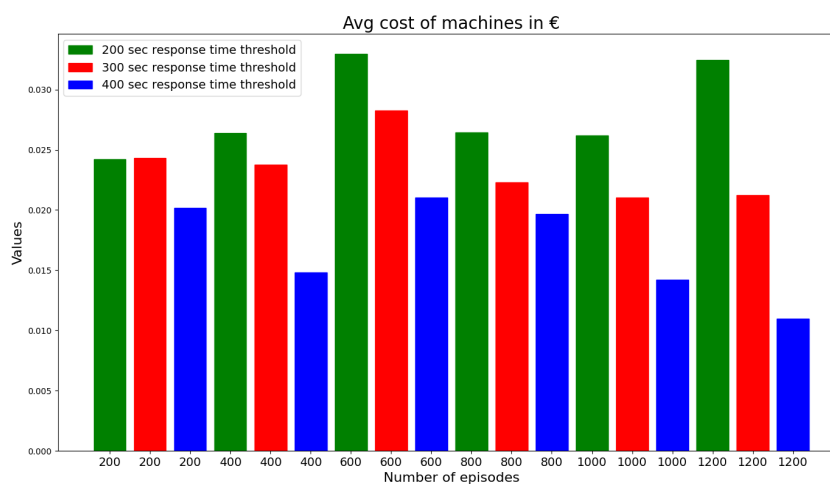
Για το μοντέλο χρησιμοποιήσαμε δύο προσεγγίσεις για την εκπαίδευση του, μία με σταθερό exploration rate και μια με μεταβητό ανά τα επεισόδια εκπαίδευσής του.

Σταθερό ϵ

Στην πρώτη περίπτωση, η τιμή του βαθμού εξερεύνησης ορίστηκε στο 0.4.

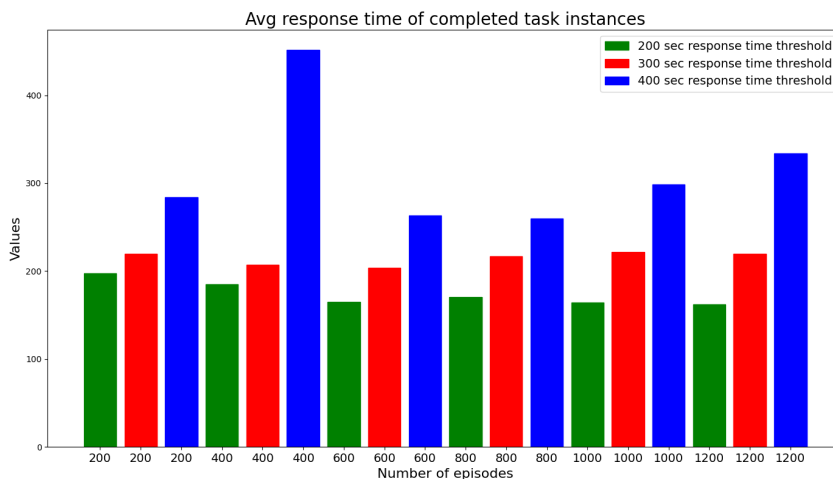
Παρακάτω παρουσιάζουμε γραφήματα με τα αποτελέσματα των μετρικών του συστήματος που αναφέρθηκαν παραπάνω αναλογικά με τον αριθμό των επεισοδίων εκπαίδευσης του μοντέλου, που αναγράφεται στον οριζόντιο άξονα.

Με πράσινο αναπαριστούμε τα επεισόδια με χρονικό όριο καθυστέρησης εκτέλεσης εργασίας $r=200$ ms, με κόκκινο τα επεισόδια με $r=300$ ms και με μπλε εκείνα με $r=400$ ms.



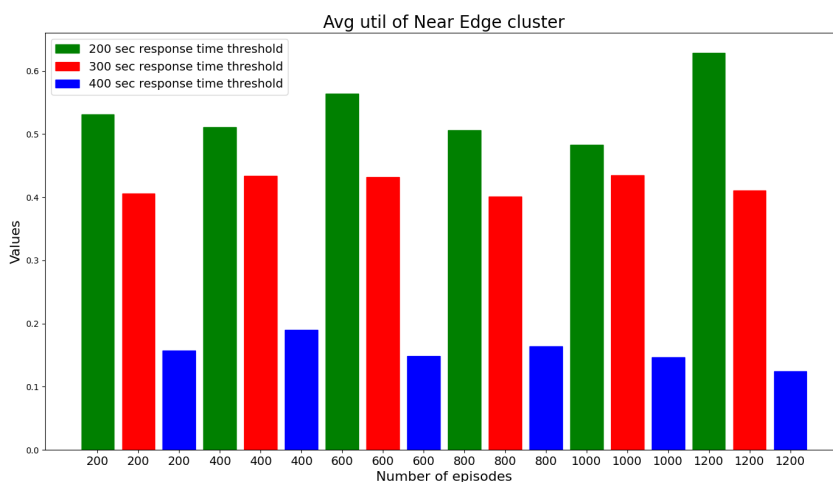
Εικόνα 8.1: Μέσο κόστος δέσμευσης πόρων σε ευρώ

Το μοντέλο στα πρώτα 200 επεισόδια παρουσιάζει μικρές διαφορές στο μέσο κόστος δέσμευσης πόρων για τα τρία χρονικά όρια. Ακολούθως, παρατηρούμε πως η μέση κατανάλωση αυξάνεται για το μικρότερο χρονικό όριο και καταλήγει να έχει την τιμή 0.032, για το $r=300$ ms δεν παρουσιάζει μεγάλη διακύμανση αλλά διατηρεί μια τιμή μεταξύ των δύο ακραίων περιπτώσεων χρονικών ορίων. Τέλος, για $r=400$ ms, το μοντέλο σταδιακά μειώνει το μέσο κόστος δέσμευσης μηχανημάτων και καταλήγει στην τιμή 0.012.



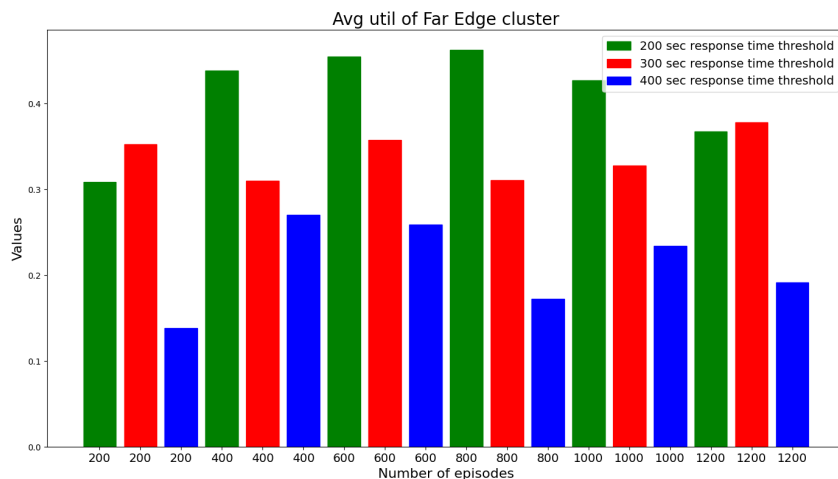
Εικόνα 8.2: Μέσος χρόνος αναμονής εκτέλεσης εργασίας του συστήματος σε ms

Το μοντέλο για $r=200$ ms μειώνει σταδιακά το μέσο χρόνο αναμονής των εργασιών του συστήματος με την αύξηση του αριθμού επεισοδίων στα οποία εκπαιδεύεται. Παρουσιάζει μικρή αύξηση για $r=300$ ms, ενώ για $r=400$ ms ο χρόνος αναμονής των εργασιών διογκώνεται σταδιακά με την πάροδο των επεισοδίων εκπαίδευσης του μοντέλου.



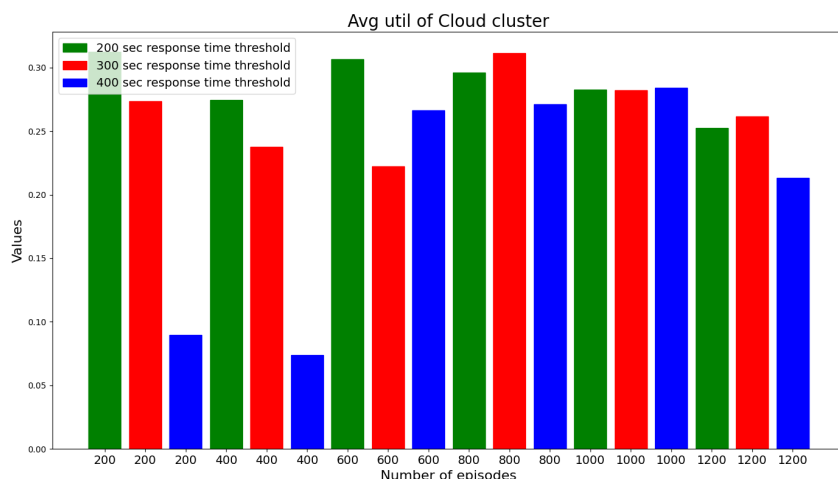
Εικόνα 8.3: Μέση χρήση των πόρων του Near Edge cluster αναλογικά με τη μέγιστη χωρητικότητά του

Σε σχέση με τη μέση χρήση των πόρων του Near Edge cluster, για $r=200$ ms παρατηρείται σταδιακή άνοδος στην τιμή αυτή, η οποία καταλήγει να φτάνει κοντά στο 60%, τιμή σαφώς μεγαλύτερη από αυτή των δύο άλλων περιπτώσεων. Για $r=300$ και $r=400$ ms παρατηρείται η διατήρηση της τιμής τους εντός εύρους μικρών αυξομειώσεων, με την τιμή της πρώτης να είναι κοντά στο 45% ενώ της δεύτερης στο 15%.



Εικόνα 8.4: Μέση χρήση των πόρων του Far Edge cluster αναλογικά με τη μέγιστη χωρητικότητά του

Για $r=200$ ms παρατηρείται η σταδιακή αύξηση και ύστερα μείωση της τιμής της μέσης χρήσης του Far Edge cluster με την αύξηση του αριθμού των επεισοδίων, με την τιμή να καταλήγει περίπου στο 38%. Για $r=300$ ms παρατηρείται αύξηση της τιμής κατά τις τελευταίες συστάδες επεισοδίων, γεγονός που προκύπτει από την αντίστοιχη μείωση χρήσης του Cloud. Για $r=400$ ms παρατηρείται η μείωση της τιμής χρήσης του συγκεκριμένου επιπέδου της υποδομής.



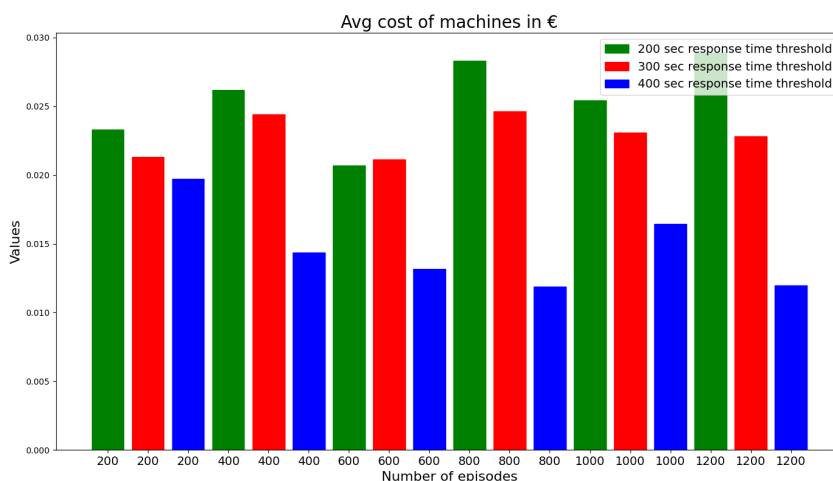
Εικόνα 8.5: Μέση χρήση των πόρων του Cloud cluster αναλογικά με τη μέγιστη χωρητικότητά του

Για $r=200$ ms παρατηρείται η σταδιακή μείωση της τιμής της μέσης χρήσης του Cloud cluster με την αύξηση του αριθμού των επεισοδίων, με την τιμή να καταλήγει περίπου στο 22%. Για $r=300$ ms παρατηρείται αυξομείωση της τιμής, ενώ για $r=400$ ms παρατηρείται μεγάλη αύξηση του ποσοστού χρήσης της Cloud υποδομής μετά τα πρώτα 400 επεισόδια εκπαίδευσης και ύστερα μικρή μείωση.

Μεταβαλλόμενο ϵ

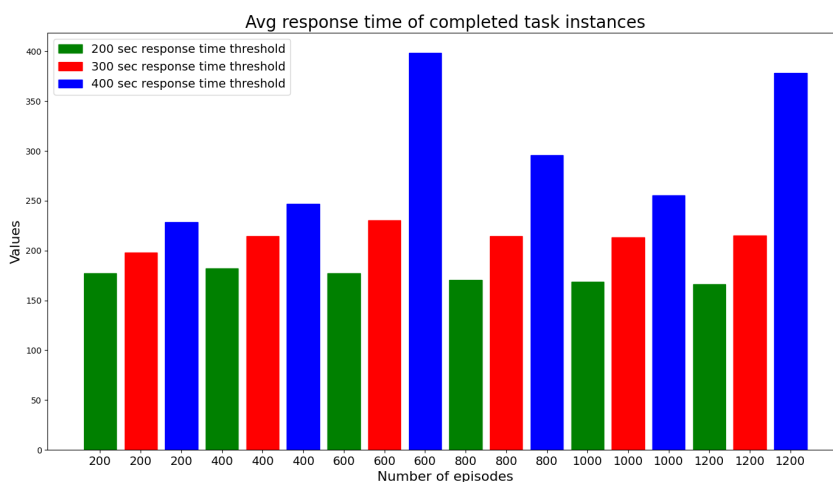
Στη δεύτερη περίπτωση, η τιμή του βαθμού εξερεύνησης ξεκινάει από το 0.7 και μειώνεται κατά 0.1 μέχρι το 0.2 ανά 200 επεισόδια.

Παρακάτω παρατηρούμε γραφήματα με τα αποτελέσματα των μετρικών του συστήματος που αναφέρθηκαν παραπάνω αναλογικά με τον αριθμό των επεισοδίων εκπαίδευσης του μοντέλου, που αναγράφεται στον οριζόντιο άξονα.



Εικόνα 8.6: Μέσο κόστος δέσμευσης πόρων σε ευρώ

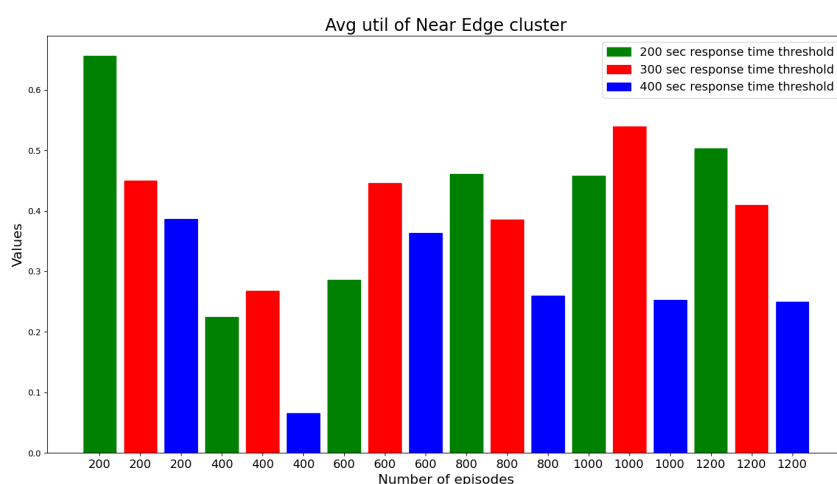
Το μοντέλο στα πρώτα 200 επεισόδια παρουσιάζει μικρές διαφορές στο μέσο κόστος δέσμευσης πόρων για τα τρία χρονικά όρια. Ακολούθως, παρατηρούμε πως η μέση κατανάλωση αυξάνεται για το μικρότερο χρονικό όριο και καταλήγει να έχει την τιμή 0.03, για το $r=300$ ms δεν παρουσιάζει μεγάλη διακύμανση αλλά διατηρεί μια τιμή μεταξύ των δύο ακραίων περιπτώσεων χρονικών ορίων. Τέλος, για $r=400$ ms, το μοντέλο σταδιακά μειώνει το μέσο κόστος δέσμευσης μηχανημάτων και καταλήγει στην τιμή 0.012.



Εικόνα 8.7: Μέσος χρόνος αναμονής εκτέλεσης εργασίας του συστήματος σε ms

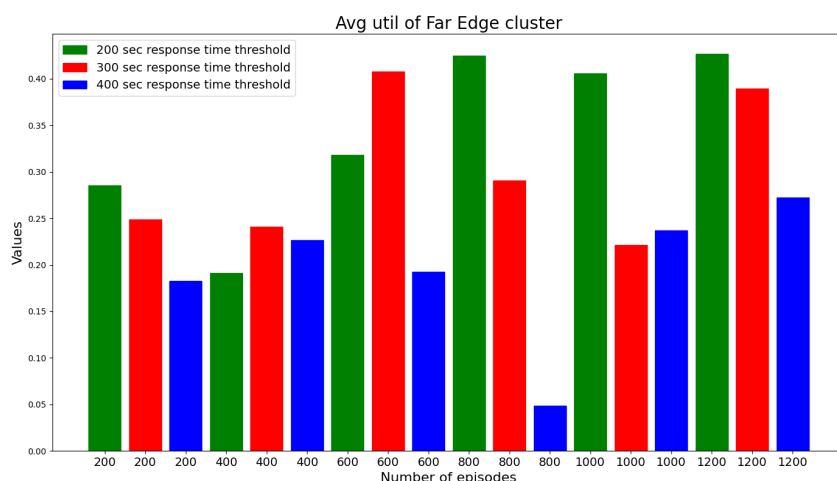
Το μοντέλο για $r=200$ ms μειώνει σταδιακά σε πολύ μικρό βαθμό το μέσο χρόνο αναμονής των εργασιών του συστήματος με την αύξηση του αριθμού επεισοδίων στα οποία εκπαιδεύε-

ται. Παρουσιάζει μικρή αύξηση για $r=300$ ms, ενώ για $r=400$ ms ο χρόνος αναμονής των εργασιών διογκώνεται με την πάροδο των επεισοδίων εκπαίδευσης του μοντέλου.



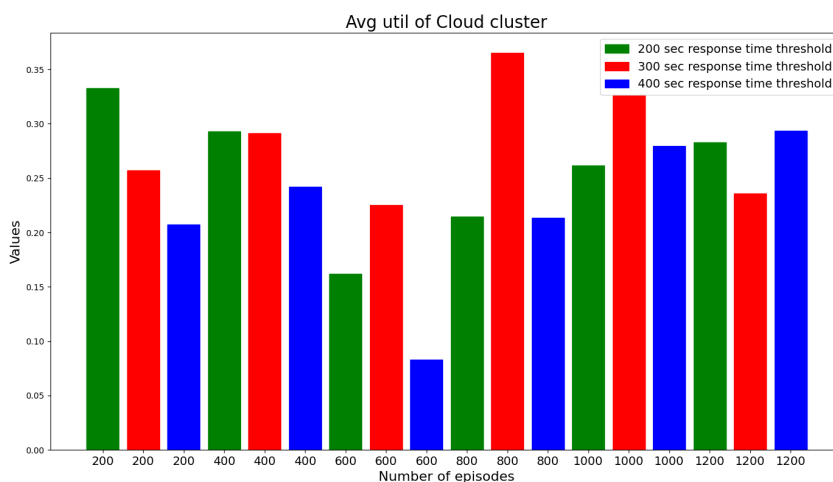
Εικόνα 8.8: Μέση χρήση των πόρων του Near Edge cluster αναλογικά με τη μέγιστη χωρητικότητα του

Σε σχέση με τη μέση χρήση των πόρων του Near Edge cluster, για $r=200$ ms παρατηρείται μεγαλύτερη του 60% μετά τα πρώτα 200 επεισόδια, η οποία ακολούθως διορθώνεται και σταδιακά αυξάνεται και καταλήγει κοντά στο 50%. Για $r=300$ ms υπάρχει αυξομείωση της τιμής μεταξύ 40-50%. Τέλος, για $r=400$ ms παρατηρείται σταθερή τιμή για τις τελευταίες συστάδες επεισοδίων στο 25-30%.



Εικόνα 8.9: Μέση χρήση των πόρων του Far Edge cluster αναλογικά με τη μέγιστη χωρητικότητα του

Για $r=200$ ms παρατηρείται η σταδιακή αύξηση της τιμής της μέσης χρήσης του Far Edge cluster με την αύξηση του αριθμού των επεισοδίων, με την τιμή να καταλήγει περίπου στο 40%. Για $r=300$ ms παρατηρείται ασταθής συμπεριφορά με τη μέση τιμή να είναι κάτω από το 30%. Για $r=400$ ms παρατηρείται μικρή αύξηση της μέσης χρήσης του συγκεκριμένου cluster.



Εικόνα 8.10: Μέση χρήση των πόρων του Cloud cluster αναλογικά με τη μέγιστη χωρητικότητά του

Για $r=200$ ms παρατηρείται η σταδιακή μείωση της τιμής της μέσης χρήσης του Cloud cluster με την αύξηση του αριθμού των επεισοδίων, με την τιμή να καταλήγει περίπου στο 20%. Για $r=300$ ms παρατηρείται αυξομείωση της τιμής, ενώ για $r=400$ ms παρατηρείται αύξηση του ποσοστού χρήσης της Cloud υποδομής μετά τα πρώτα 600 επεισόδια εκπαίδευσης.

Σχόλια Επιδόσεων μεταξύ των δύο μοντέλων αναλογικά με τα επεισόδια εκπαίδευσης

Παρατηρούμε πως τα αποτελέσματα του μοντέλου με το σταθερό βαθμό εξερεύνησης ϵ , συγκλίνουν πιο γρήγορα σε καλύτερα αποτελέσματα στις πρώτες ομάδες 200 επεισοδίων εκπαίδευσης για μικρές χρονικές καθυστερήσεις. Αυτό συμβαίνει, καθώς το μοντέλο βρίσκει ένα τοπικό μέγιστο στις τιμές Q και τις επιλέγει με μεγαλύτερη συχνότητα. Κατά την εξέλιξη της εκπαίδευσης του μοντέλου παρατηρούμε πως τα δύο μοντέλα παρουσιάζουν παρόμοια αποτελέσματα, γεγονός που δικαιολογείται καθώς και τα δύο συγκλίνουν σε βέλτιστες Q τιμές.

Επιπλέον, παρατηρούμε πως κατά την εκπαίδευση του μοντέλου, στα αποτελέσματα που λαμβάνουμε μετά από 200 και 400 επεισόδια αντίστοιχα, ο χρόνος αναμονής στο Near Edge είναι μεγαλύτερος από ότι στα άλλα cluster, γεγονός που εξηγείται μόνο με τη συσσώρευση φορτίου που δε διαμειράζεται ανάλογα και στις υπόλοιπες συστάδες. Αυτό το φαινόμενο, εξαλείφεται όσο εκπαιδευτεί το μοντέλο, δεδομένου ότι εστιάζει στον περιορισμό της αύξησης του χρόνου αναμονής του συστήματος.

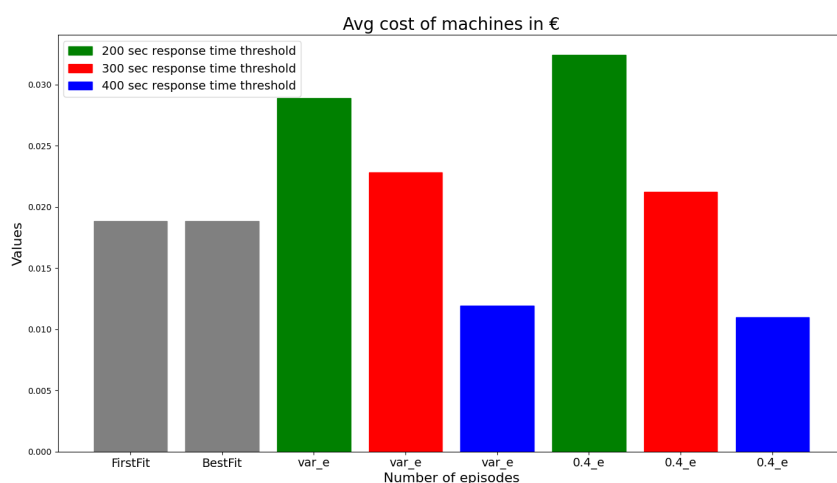
Συγκριτικά, με τα αποτελέσματα ανάλογα με το εκάστοτε χρονικό όριο, από τα διαγράμματα είναι φανερό πως σε όλα τα στάδια της εκπαίδευσης του μοντέλου, όσο μικρότερο είναι το χρονικό όριο τόσο μεγαλύτερη είναι η χρησιμοποίηση των κατά σειρά Near και Far Edge clusters έναντι του Cloud, καθώς το μοντέλο μας αναγνωρίζει πως πρέπει να βελτιστοποιήσει τη διαχείριση των εργασιών ως προς το χρόνο αναμονής τους, που αυξάνεται σημαντικά στις απομακρυσμένες υποδομές. Πιο αναλυτικά, δίνει προτεραιότητα στην μετρίαση της αύξησης του μέσου χρόνου αναμονής των εργασιών συγκριτικά με την αύξηση του μέσου κόστους δέσμευσης πόρων, που παρουσιάζει μεγάλη άνοδο συγκριτικά με τις περιπτώσεις

που το χρονικό όριο αναμονής είναι στα 400 ms.

8.3 Σύγκριση Επιδόσεων με τις βασικές περιπτώσεις δρομολογητών

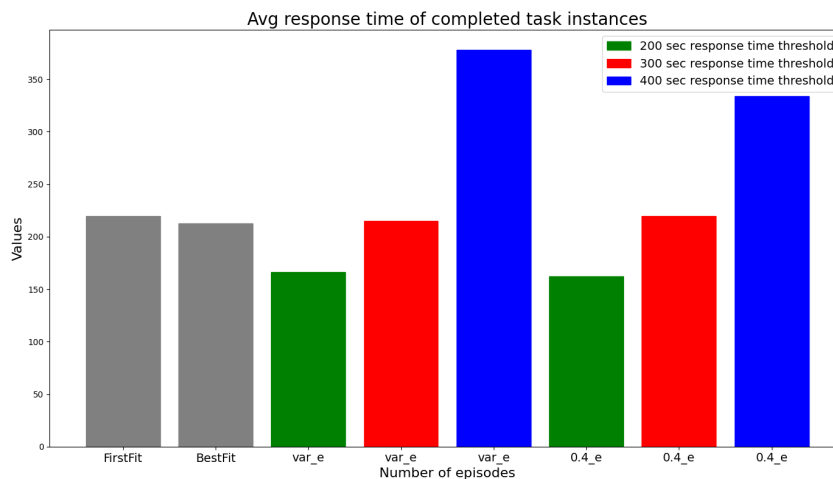
Παρακάτω, παρουσιάζουμε τα αποτελέσματα από τα δύο τελικά μοντέλα που αναλύσαμε παραπάνω συγκριτικά με τις ποιοτικές μετρήσεις που λάβαμε για την εκτέλεση των 100 εργασιών με δύο βασικούς δρομολογητές που έχουν ως βάση κατανομής των εργασιών στους κόμβους, τους αλγορίθμους First και Best Fit.

Με το 0.4_e αναπαριστούμε το μοντέλο με το μεταβαλλόμενο, κατά την εκπαίδευσή του βαθμό εξερεύνησης, και με το var_e ορίζουμε το μοντέλο με το σταθερό e .



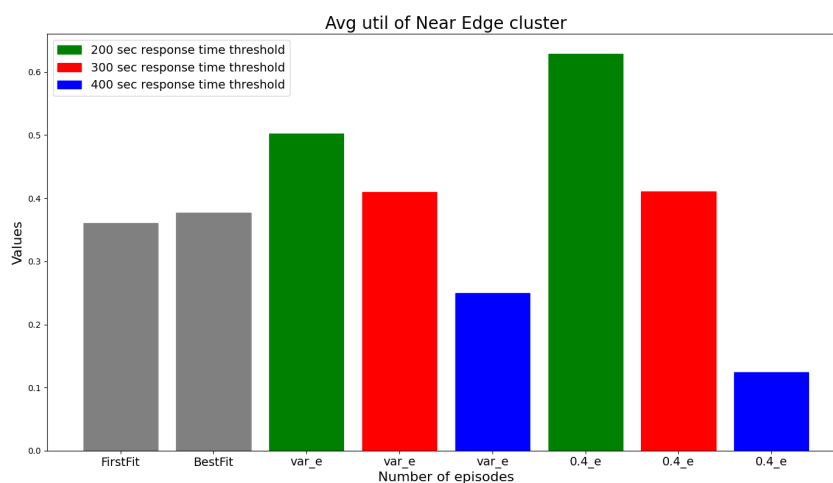
Εικόνα 8.11: Μέσο κόστος δέσμευσης πόρων σε ευρώ

Αρχικά, στο 1ο γράφημα όπου προβάλλεται η μέση ενεργειακή κατανάλωση των πόρων του συστήματος κατά την εκτέλεση ενός επεισοδίου 100 εργασιών, παρατηρούμε πως τα δύο μοντέλα έχουν μεγαλύτερη μέση κατανάλωση για τις περιπτώσεις όπου το χρονικό όριο έχει τεθεί στα 100 και 200 ms, ενώ στην τρίτη περίπτωση η κατανάλωση είναι αισθητά μικρότερη από αυτή των δύο βασικών περιπτώσεων. Πιο συγκεκριμένα, για $r=200$ ms υπολογίζεται 53% και 72% αύξηση συγκριτικά με τις βασικές περιπτώσεις. Αντιθέτως, για $r=400$ ms, παρατηρούμε 37% και 42% μείωση. Έτσι, βγάζουμε το συμπέρασμα πως τα μοντέλα μας επιτυχημένα αυξάνουν ή μειώνουν τη μέση κατανάλωση τους έναντι των βασικών λύσεων με βάση τις απαιτήσεις του συστήματος.



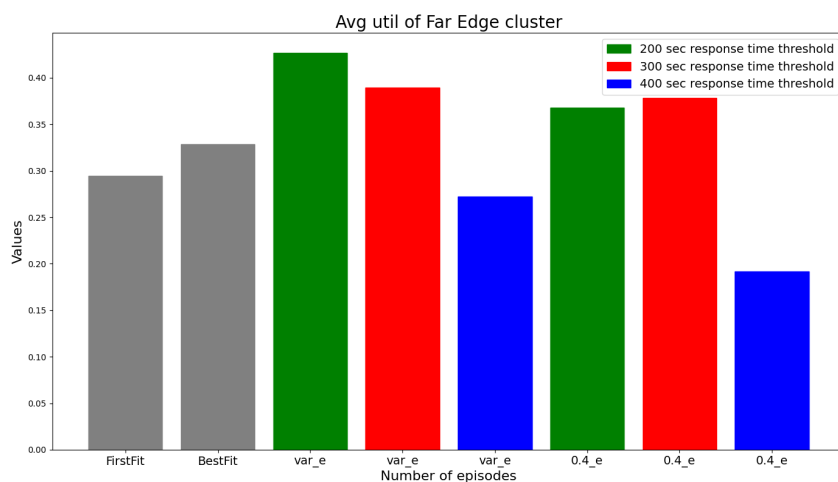
Εικόνα 8.12: Μέσος χρόνος αναμονής εκτέλεσης εργασίας του συστήματος σε ms

Στο 2ο γράφημα, αρχικά παρατηρούμε πως ο Best Fit έχει λίγο καλύτερα αποτελέσματα σε σχέση με τον First Fit, όπως αναμενόταν. Συγκριτικά με τα DRL μοντέλα, παρατηρούμε βελτίωση κατά 22% και 24% του μέσου χρόνου αναμονής των εργασιών του επεισοδίου για την περίπτωση όπου $r=200$ ms σε σχέση με τον Best Fit. Ακόμη, παρατηρούμε τις πιο αυξημένες τιμές για τις άλλες δύο περιπτώσεις χρονικών ορίων, γεγονός που συνάδει με τους στόχους τους.



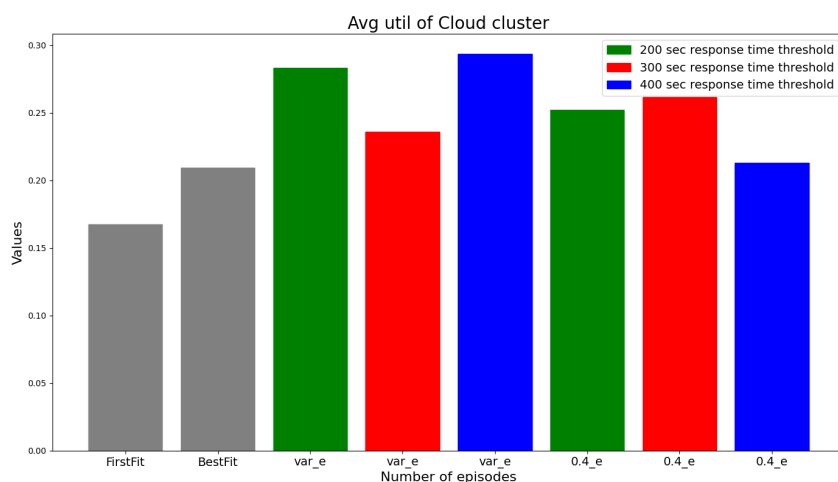
Εικόνα 8.13: Μέση χρήση των πόρων του Near Edge cluster αναλογικά με τη μέγιστη χωρητικότητά του

Στο 3ο γράφημα, διαπιστώνουμε πως όσο πιο αυστηρό είναι το χρονικό όριο που έχει τεθεί από το σύστημα τόσο μεγαλύτερη είναι η χρησιμοποίηση του Near Edge. Έτσι, παρατηρούμε πως για $r=200$ ms υπολογίζεται αύξηση χρήσης του Near Edge κατά 33% και 67% συγκριτικά με τον Best Fit αλγόριθμο, για $r=300$ ms παρατηρείται μικρή αύξηση της τάξης του 5%, ενώ για $r=400$ ms παρατηρείται μείωση κατά 34% και 67%.



Εικόνα 8.14: Μέση χρήση των πόρων του Far Edge cluster αναλογικά με τη μέγιστη χωρητικότητά του

Στο 4ο γράφημα, διαπιστώνουμε πως για τα δύο μικρότερα χρονικά όρια η χρήση του Far Edge είναι αντίστοιχη, ενώ μειώνεται σημαντικά για το πιο μεγάλο χρονικό όριο. Έτσι, παρατηρούμε πως για $r=200$ ms υπολογίζεται αύξηση χρήσης του Far Edge κατά 30% και 12% συγκριτικά με τον Best Fit αλγόριθμο, για $r=300$ ms παρατηρείται αύξηση 18% και 15%, ενώ για $r=400$ ms παρατηρείται μείωση κατά 17% και 42%.



Εικόνα 8.15: Μέση χρήση των πόρων του Cloud cluster αναλογικά με τη μέγιστη χωρητικότητά του

Στο 5ο γράφημα, διαπιστώνουμε αυξημένη χρήση του Cloud και στις τρεις περιπτώσεις εφαρμογής του μοντέλου μας συγκριτικά με τα αποτελέσματα των δύο βασικών αλγορίθμων. Έτσι, παρατηρούμε πως για $r=200$ ms υπολογίζεται αύξηση χρήσης του Near Edge κατά 35% και 21% συγκριτικά με τον Best Fit αλγόριθμο, για $r=300$ ms παρατηρείται αύξηση 13% και 25%, ενώ για $r=400$ ms παρατηρείται αύξηση κατά 40% και 0.1%.

Σύνοψη - Συμπεράσματα

9.1 Σύνοψη

Σε αυτή την εργασία αναπτύξαμε ένα σύστημα δρομολογητή εργασιών σε μία cloud-edge υποδομή και τη χρήση μοντέλων βαθιάς ενισχυτικής μάθησης για την βελτίωση ποιοτικών μετρικών του. Διατυπώσαμε τις συναρτήσεις ορισμού του DRL μοντέλου με τρόπο που να λαμβάνει υπόψη του κατά την απόφαση τόσο το φόρτο όσο και SLA χρονικές προσδιαγραφές των εργασιών του συστήματος, καθώς και ποσοστά εκμετάλλευσης/διαθεσιμότητας πόρων του συστήματος. Ακόμη, συνδυάσαμε εξαρτήσεις καθυστέρησης εκτέλεσης εργασιών και ποσοστών και κόστους χρησιμοποίησης πόρων του συστήματος για την βελτίωση της απόδοσης του τελευταίου. Χρησιμοποιήσαμε αληθινά δεδομένα και μεγάλο μέγεθος πιθανών καταστάσεων για την ανάπτυξη ενός ρεαλιστικού μοντέλου.

9.2 Συμπεράσματα

Τα ποιοτικά αποτελέσματα των μετρικών του συστήματος, έδειξαν σαφή βελτίωση στη χρησιμοποίηση των πόρων και την κατανομή των εργασιών ανάλογα με την εκάστοτε συνθήκη του συστήματος.

Πιο αναλυτικά, παρατηρούμε ότι και τα δύο μοντέλα βαθιάς ενισχυτικής μάθησης βελτίωσαν τα αποτελέσματα συγκριτικά με τις βασικές περιπτώσεις των δρομολογητών. Όπως αναλύθηκε και στο προηγούμενο κεφάλαιο, παρατηρήσαμε σαφή προσαρμογή των επιλογών ενεργειών του μοντέλου με βάση τις εκάστοτε συνθήκες του συστήματος. Στις περιπτώσεις των μικρών χρονικών ορίων, αυξήθηκε σημαντικά η χρήση όλων των clusters, με προτεραιότητα σε εκείνα που παρουσιάζουν τις μικρότερες χρονικές καθυστερήσεις, Near και Far Edge, έναντι του Cloud. Ακόμη, από τα αποτελέσματα βγαίνει το πόρισμα πως αυξάνοντας τον επιτρεπτό μέσο χρόνο αναμονής, μειώνεται το μέσο κόστος δέσμευσης πόρων του συστήματος. Συνοψίζοντας, η χρήση μηχανισμών αντίδρασης σε δυναμικές καταστάσεις του συστήματος, αποτελεί μια σημαντική στήριξη στην εξάλειψη σφαλμάτων που μπορεί να προκύψουν από τα ήδη ανεπτυγμένα συστήματα που χρησιμοποιούνται και έχουν ως στόχο κυρίως τη βελτίωση της αρχικής κατανομής πόρων και εργασιών. Καταληκτικά, το DRL μοντέλο επιφέρει μεγάλη βελτίωση σε ποιοτικές μετρικές του συστήματος, μεταβλητού ενδιαφέροντος στην εκάστοτε περίπτωση, συγκριτικά με τις βασικές περιπτώσεις First και Best Fit, και έτσι κρίνεται κατάλληλο για την περαιτέρω μελέτη όσον αφορά τη διαχείριση της σύνθετης δομής

των cloud-edge υποδομών και την προσαρμογή του σε μεταβλητού ενδιαφέροντος ποιοτικά χαρακτηριστικά του εκάστοτε συστήματος/υποδομής.

9.3 Μελλοντικές Επεκτάσεις

Το σύστημα που αναπτύχθηκε στα πλαίσια αυτής της διπλωματικής εργασίας θα μπορούσε να βελτιωθεί και να επεκταθεί περαιτέρω, τουλάχιστον ως προς πέντε κατευθύνσεις. Συγκεκριμένα, αναφέρονται τα ακόλουθα :

- Αύξηση των διαθέσιμων ενεργειών του συστήματος(ανάπτυξη μοντέλου με συνδυασμό υπάρχοντων ενεργειών) για τη δυνατότητα καλύτερης προσαρμογής του μοντέλου στις εκάστοτε συνθήκες του συστήματος.
- Ανάπτυξη ιεραρχικού DRL μοντέλου 3 επιπέδων, που το συγκεντρωτικό επίπεδο να αποτελείται από το δικό μας μοντέλο, και ακολούθως, ανάλογα με την επιλεγμένη ενέργεια του, να υπάρχει νέος κύκλος επιλογών του εκάστοτε επιλεγμένου cluster για επιλογή του node που θα εκτελέσει την επιλεγμένη ενέργεια. Συνηρημικά, μπορεί και το node να έχει ένα DRL μοντέλο ως κέντρο ελέγχου επιλογής μηχανήματος για την εκτέλεση της επιλεγμένης εργασίας. Αυτό το ιεραρχικό μοντέλο, μπορεί να οδηγήσει σε πολύ μεγάλη προσαρμοστική ακρίβεια στις επιλογές του με βάση την κατάσταση του συστήματος.
- Χρήση μηχανικής μάθησης για την ανάπτυξη μοντέλου πρόβλεψης μελλοντικού φόρτου εργασιών για τη βελτίωση της επιλογής ενεργειών του συστήματος με βάση όχι μόνο την τωρινή κατάσταση, αλλά και τη μελλοντική (ενδεικτικά μπορούν να χρησιμοποιηθούν LSTM και encoders για το μοντέλο πρόβλεψης).
- Χρήση LSTM νευρωνικών δικτύων που να διατηρούν ως τρέχουσα κατάσταση, μετρικές και στοιχεία του συστήματος από αρκετές σειριακές χρονικές στιγμές, με σκοπό την ακριβέστερη ανάλυση των χαρακτηριστικών και της συμπεριφοράς του συστήματος.
- Τέλος, μπορεί να χρησιμοποιηθεί η μέθοδος Few Shot Learning(FSL), και να συγκροτηθεί ένα συγκεντρωτικό μοντέλο που να εκπαιδεύεται με βάση τη συμπεριφορά και τα δεδομένα από πολλές διαφορετικές cloud-edge υποδομές. Αυτό το μοντέλο ακολούθως θα μπορεί να χρησιμοποιηθεί από ανεξερεύνητες υποδομές, και έχοντας τη δυνατότητα αντιστοίχισης των στοιχείων της με υποδομές στις οποίες έχει εκπαιδευτεί, να μπορεί να διαχειριστεί αυτές τις υποδομές με μεγάλη ακρίβεια σε πραγματικό χρόνο.

Βιβλιογραφία

- [1] *edge-computing*. <https://www.marketsandmarkets.com/PressReleases/edge-computing.asp>. Ημερομηνία πρόσβασης: 13-11-2023.
- [2] *Unlocking Business Value from Your Organization's Transition to Cloud*. <https://www2.deloitte.com/xe/en/pages/human-capital/articles/unlocking-business-value-from-your-organizations-transition-to-cloud.html>. Ημερομηνία πρόσβασης: 13-11-2023.
- [3] *Cloud Statistics*. <https://www.gartner.com/en/documents/3890506>. Ημερομηνία πρόσβασης: 13-11-2023.
- [4] Yuvraj Sahni, Jiannong Cao, Lei Yang και Shengwei Wang. *Distributed resource scheduling in edge computing: Problems, solutions, and opportunities*. *Computer Networks*, 219:109430, 2022.
- [5] Mohammed Laroui, Boubakr Nour, Hassine Moun gla, Moussa Ali Cherif, Hossam Afifi και Mohsen Guizani. *Edge and fog computing for IoT: a survey on current research activities & future directions*. *Computer Communications*, 180:210–231, 2021.
- [6] Darrin J. Young. *Wireless powering and data telemetry for biomedical implants*. *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, σελίδες 3221–3224, 2009.
- [7] Shuo Liu, Shi Qiu, Huayi Li και Ming Liu. *Real-Time Telemetry-Based Recognition and Prediction of Satellite State Using TS-GCN Network*. *Electronics*, 12(23), 2023.
- [8] Tianyu Zhang, Chuanyu Xue, Jiachen Wang, Zelin Yun, Natong Lin και Song Han. *A Survey on Industrial Internet of Things (IIoT) Testbeds for Connectivity Research*, 2024.
- [9] Asfa Toor, Saif ul Islam, Ghufuran Ahmed, Sohail Jabbar, Shehzad Khalid και Abdullahi Mohamud Sharif. *Energy efficient edge-of-things*. *EURASIP Journal on Wireless Communications and Networking*, 2019(82), 2019.
- [10] Cheol Ho Hong και Blesson Varghese. *Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms*. *ACM Comput. Surv.*, 52(5), 2019.
- [11] Shihong Hu, Weisong Shi και Guanghui Li. *CEC: A Containerized Edge Computing Framework for Dynamic Resource Provisioning*. *IEEE Transactions on Mobile Computing*, 22(7):3840–3854, 2023.

- [12] Yu Wei Chan, Halim Fathoni, Hao Yi Yen και Chao Tung Yang. *Implementation of a Cluster-Based Heterogeneous Edge Computing System for Resource Monitoring and Performance Evaluation*. *IEEE Access*, 10:38458–38471, 2022.
- [13] Hamzeh Khazaei, Jelena Mistic και Vojislav B. Mistic. *Performance Analysis of Cloud Computing Centers Using M/G/m/m+r Queuing Systems*. *IEEE Transactions on Parallel and Distributed Systems*, 23(5):936–943, 2012.
- [14] Yi Wei, Li Pan, Shijun Liu, Lei Wu και Xiangxu Meng. *DRL-Scheduling: An Intelligent QoS-Aware Job Scheduling Framework for Applications in Clouds*. *IEEE Access*, 6:55112–55125, 2018.
- [15] Tingting Dong, Fei Xue, Chuangbai Xiao και Juntao Li. *Task scheduling based on deep reinforcement learning in a cloud manufacturing environment*. *Concurrency and Computation: Practice and Experience*, 32, 2020.
- [16] Poria Pirozmand, Hoda Jalalinejad, Ali Asghar Rahmani Hosseinabadi, Seyedsaedi Mirkamali και Yingqiu Li. *An improved particle swarm optimization algorithm for task scheduling in cloud computing*. *Journal of Ambient Intelligence and Humanized Computing*, 14:1–15, 2023.
- [17] Jingbo Li, Xingjun Zhang, Jia Wei, Zeyu Ji και Zheng Wei. *GARLSched: Generative adversarial deep reinforcement learning task scheduling optimization for large-scale high performance computing systems*. *Future Generation Computer Systems*, 135:259–269, 2022.
- [18] Pegah Gazori, Dadmehr Rahbari και Mohsen Nickray. *Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach*. *Future Generation Computer Systems*, 110:1098–1115, 2020.
- [19] Rafael Moreno-Vozmediano, Rubén Montero, Eduardo Huedo και Ignacio Llorente. *Efficient resource provisioning for elastic Cloud services based on machine learning techniques*. *Journal of Cloud Computing*, 8, 2019.
- [20] Rohit Yadav, M.P. και D.K. Yadav. *Resource Provisioning Through Machine Learning in Cloud Services*. *Arabian Journal for Science and Engineering*, 47:1483–1505, 2022.
- [21] Joseph Nathanael Witanto, Hyotaek Lim και Mohammed Atiquzzaman. *Adaptive selection of dynamic VM consolidation algorithm using neural network for cloud resource management*. *Future Gener. Comput. Syst.*, 87(“):35–42, 2018.
- [22] Yue Jin, Makram Bouzid, Dimitre Kostadinov και Armen Aghasaryan. *Resource management of cloud-enabled systems using model-free reinforcement learning*. *Annals of Telecommunications*, 74, 2019.
- [23] K. Siddesha, G. V. Jayaramaiah και Chandrapal Singh. *A novel deep reinforcement learning scheme for task scheduling in cloud computing*. *Cluster Computing*, 25(6):4171–4188, 2022.

- [24] Ce Shang, Yan Sun, Hong Luo και Mohsen Guizani. *Computation Offloading and Resource Allocation in NOMA-MEC: A Deep Reinforcement Learning Approach*. *IEEE Internet of Things Journal*, 10(17):15464–15476, 2023.
- [25] Jingxuan Chen, Xianbin Cao, Peng Yang, Meng Xiao, Siqiao Ren, Zhongliang Zhao και Dapeng Oliver Wu. *Deep Reinforcement Learning Based Resource Allocation in Multi-UAV-Aided MEC Networks*. *IEEE Transactions on Communications*, 71(1):296–309, 2023.
- [26] Shashank Swarup, Elhadi M. Shakshuki και Ansar Yasar. *Energy Efficient Task Scheduling in Fog Environment using Deep Reinforcement Learning Approach*. *Procedia Computer Science*, 191:65–75, 2021. Τη 18τη Ιντερνατιοναλ ὄνφερενζε ον Μοβιλε Σψστεμς ανδ Περασιε ὀμπυτινγ (ΜοβιΣΙΤ*), Τη 16τη Ιντερνατιοναλ ὄνφερενζε ον Φυτυρε Νετωρκς ανδ ὀμμυνιζατιονς (ΦΝ*), Τη 11τη Ιντερνατιοναλ ὄνφερενζε ον Συσταιναβλε Ενεργψ Ινφορματιον Τεσνηολογψ.
- [27] Han Wang, LuisMu noz González, David Eklund και Shahid Raza. *Non-IID data re-balancing at IoT edge with peer-to-peer federated learning for anomaly detection*. *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '21*, σελίδα 153–163, New York, NY, USA, 2021. Association for Computing Machinery.
- [28] Amanda Jayanetti, Saman Halgamuge και Rajkumar Buyya. *Deep reinforcement learning for energy and time optimized scheduling of precedence-constrained tasks in edge-cloud computing environments*. *Future Gener. Comput. Syst.*, 137(°):14–30, 2022.
- [29] Guangyao Zhou, Ruiming Wen, Wenhong Tian και Rajkumar Buyya. *Deep reinforcement learning-based algorithms selectors for the resource scheduling in hierarchical Cloud computing*. *J. Netw. Comput. Appl.*, 208(°), 2022.
- [30] Ayoub Alsarhan, Awni Itradat, Ahmed Y. Al-Dubai, Albert Y. Zomaya και Geyong Min. *Adaptive Resource Allocation and Provisioning in Multi-Service Cloud Environments*. *IEEE Transactions on Parallel and Distributed Systems*, 29(1):31–42, 2018.
- [31] Xiong Xiong, Kan Zheng, Lei Lei και Lu Hou. *Resource Allocation Based on Deep Reinforcement Learning in IoT Edge Computing*. *IEEE Journal on Selected Areas in Communications*, 38(6):1133–1146, 2020.
- [32] Cheng Kuang, Meng Duan, Tao Lv, Yingjun Wu, Li Li και Luping Wang. *ODRL: Reinforcement Learning in Priority Scheduling for Running Cost Optimization*. *2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS)*, σελίδες 2410–2419, 2023.
- [33] *Alibaba-Dataset*. <https://github.com/alibaba/clusterdata/blob/master/cluster-trace-v2018/>. Ημερομηνία πρόσβασης: 13-11-2023.

- [34] *Job Scheduler repo*. <https://github.com/FC-Li/CloudSimPy/blob/master/README.en.md>.
Ημερομηνία πρόσβασης: 13-11-2023.
- [35] Batyr Charyyev, Engin Arslan και Mehmet Hadi Gunes. *Latency Comparison of Cloud Datacenters and Edge Servers*. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, σελίδες 1-6, 2020.
- [36] *GitHub repo*. https://github.com/ariskaratzikos/CloudSimPy/blob/more_states_torch_nn. Ημερομηνία πρόσβασης: 13-11-2023.