Paper Title\* (use style: paper title)

\*Note: Sub-titles are not captured in Xplore and should not be used

line 1: 1st Given Name Surname   
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 4th Given Name Surname  
line 2: *dept. name of organization*  
*(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 2nd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 5th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 3rd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 6th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

*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 των αναλύσεων 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 της υποδομής για την βελτιστοποίηση παραμέτρων της. Χρησιμοποιούνται μοντέλα ενισχυτικής μάθησης που χρησιμοποιούν βαθιά νευρωνικά δίκτυα για να αναγνωρίζουν μοτίβα συσχέτισης καταστάσεων-ενεργειών βάση συνάρτησεων κέρδους. Συνοψίζοντας, σκοπός αυτής της εργασίας είναι η ανάπτυξη ενός γρήγορου και αποδοτικού τρόπου για τη βελτιστοποίηση επιλογών ενεργειών σε κάθε πιθανή κατάσταση του συστήματος.

Λέξεις κλειδιά—cloud-edge, βαθιά ενισχυτική μάθηση, QoS μετρικές

# Εισαγωγη

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 αυξομειούμενες απαιτήσεις των εφαρμογών και υπηρεσιών για υπολογιστική και αποθηκευτική ισχύ.

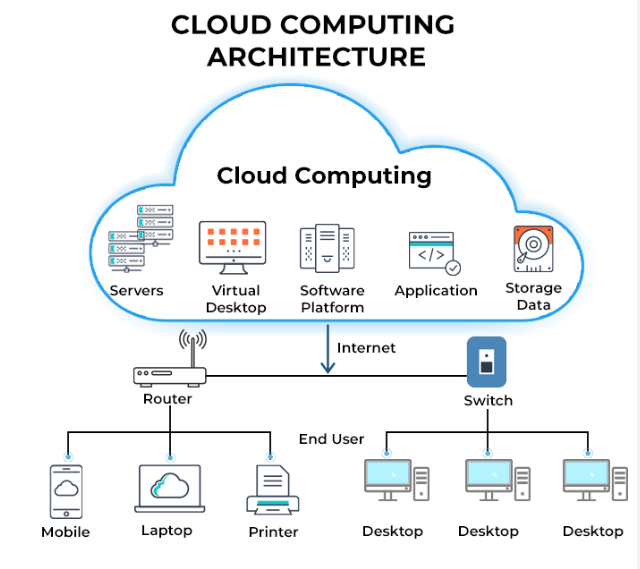


Image 1

Η ανάπτυξη των συσκευών IoT (Internet of Things) και η γενική ανάγκη για επεξεργασία δεδομένων έχει δημιουργήσει μια σύγκρουση με τον τρόπο λειτουργίας του Cloud, διογκώνοντας την ανάγκη για μια κατανεμημένη υποδομή. Η μετάδοση ακατέργαστων δεδομένων σε κεντρικά συστήματα διαχείρισης δεδομένων 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 (Image 2) δημιουργεί ένα νέο επίπεδο επεξεργασίας πιο κοντά στο τελικό σημείο (συσκευή) και χρησιμοποιείται ως σημείο φιλτραρίσματος, 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).

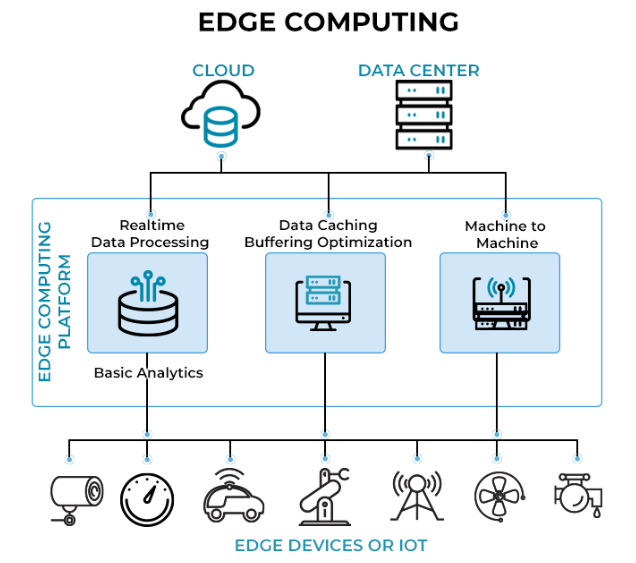


Image 2

Με την ανάπτυξη των κατανεμημένων συστημάτων και την αυτοματοποίησή τους, τα όρια μεταξύ των εννοιών cloud και edge υποχωρούν , δημιουργώντας το cloud-edge continuum. Αυτό το συνεχές αντιπροσωπεύει ένα φάσμα υπολογιστικών πόρων, που κυμαίνονται από τα συγκεντρωτικά κέντρα δεδομένων έως τις κατανεμημένες συσκευές edge, οι οποίες συνεργάζονται απρόσκοπτα για την υποστήριξη ποικιλίας εφαρμογών και υπηρεσιών. Ένα σημαντικό πλεονέκτημα του cloud-edge continuum είναι η δυνατότητα επεξεργασίας και ανάλυσης δεδομένων σε πραγματικό χρόνο, η οποία είναι κρίσιμη για εφαρμογές όπως τα αυτόνομα οχήματα και ο βιομηχανικός αυτοματισμός. Επιπλέον, επιτρέπει την επεξεργασία των δεδομένων πιο κοντά στην πηγή παραγωγής τους, γεγονός που μειώνει εκθετικά τον όγκο των δεδομένων που πρέπει να μεταδίδονται σε κεντρικά συστήματα διαχείρισης δεδομένων [4].

Σύμφωνα με πρόσφατη έρευνα, η παγκόσμια αγορά edge υποδομών αναμένεται να αυξηθεί από 60 δισ. δολάρια το 2024 σε 110 δισ. δολάρια έως το 2029, αντιπροσωπεύοντας CAGR 13% [1]. Αυτή η ταχεία ανάπτυξη οφείλεται στην αυξανόμενη ζήτηση για επεξεργασία με χαμηλή καθυστέρηση, 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, διαπιστώθηκε ότι η υιοθέτηση του edge computing είναι ζωτικής σημασίας για τους οργανισμούς που επιθυμούν να αξιοποιήσουν αναδυόμενες τεχνολογίες 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 δημιουργούνται και θα υποβάλλονται σε επεξεργασία σε edge υποδομές [3].

Ωστόσο, η ενσωμάτωση του edge computing στην υπάρχουσα υποδομή cloud φέρει δυσκολίες σε σχέση με τη διαχείριση μιας κατανεμημένης υποδομής. Σε αυτό το σημείο αξίζει να αναλυθεί πως έγινε η μετάβαση από τα συγκεντρωτικά συστήματα στα κατανεμημένα και πως η ανάπτυξη των τελευταίων φέρει σημαντικά πλεονεκτήματα για το cloud-edge continuum.

Η εξέλιξη του cloud computing έχει οδηγήσει σε μετατόπιση από τα παραδοσιακά συγκεντρωτικά συστήματα σε κατανεμημένα 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 για ταχύτερες και πιο ευέλικτες εφαρμογές. Τα κατανεμημένα 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.

Στο πλαίσιο του cloud-edge continuum, τα κατανεμημένα συστήματα επιτρέπουν την απρόσκοπτη ενσωμάτωση ετερογενών πηγών δεδομένων τόσο από το cloud όσο και από την υποδομή edge. Το cloud παρέχει ένα συγκεντρωτικό περιβάλλον για την αποθήκευση, την επεξεργασία και την ανάλυση δεδομένων, ενώ η edge υποδομή επιτρέπει την επεξεργασία και την ανάλυση δεδομένων πιο κοντά στην πηγή παραγωγής των δεδομένων, μειώνοντας την καθυστέρηση και βελτιώνοντας τις επιδόσεις. Ο συνδυασμός υποδομών cloud και edge δημιουργεί ένα δίκτυο σύνδεσης υπολογιστικών πόρων που συνεργάζονται για την υποστήριξη ποικίλων εφαρμογών και υπηρεσιών.

Αρκετές μελέτες έχουν τονίσει τη σημασία των κατανεμημένων συστημάτων συγκριτικά με τα συγκεντρωτικά συστήματα στον τομέα αυτόν (Image 3). Η δυνατότητα για παράλληλη επεξεργασία δεδομένων κοντά στην πηγή παραγωγής τους αυξάνει σημαντικά την απόδοση των συστημάτων και μειώνει την καθυστέρηση καθώς πραγματοποιείται φιλτράρισμα και μεταφορά μόνο ορισμένων δεδομένων στα συγκεντρωτικά συστήματα [5][6].

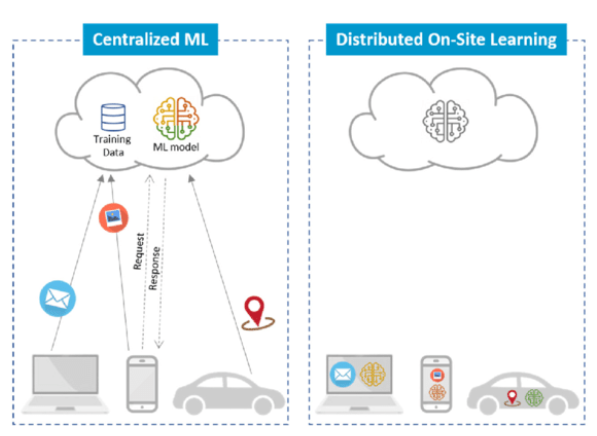


Image 3

Ωστόσο, παρά τα οφέλη των κατανεμημένων 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 κατανεμημένων 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 [7]-[9].

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. Στην εργασία μας, μοντελοποιούμε μια υπολογιστική και αποθηκευτική υποδομή που συνδυάζει τις θεωρητικές δομές Cloud, Edge υπό πραγματικές συνθήκες. 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 εκτέλεση, 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. Ηλύση μας προσφέρει πολλές δυνατότητες για ορισμό παραμέτρων και αποτελεσματικής βελτίωσής τους μέσω της προσομοίωσης υποδομών.

Η συνέχεια της διπλωματικής είναι οργανωμένη όπως ακολουθεί. 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.

# Θεωρητικο Υποβαθρο

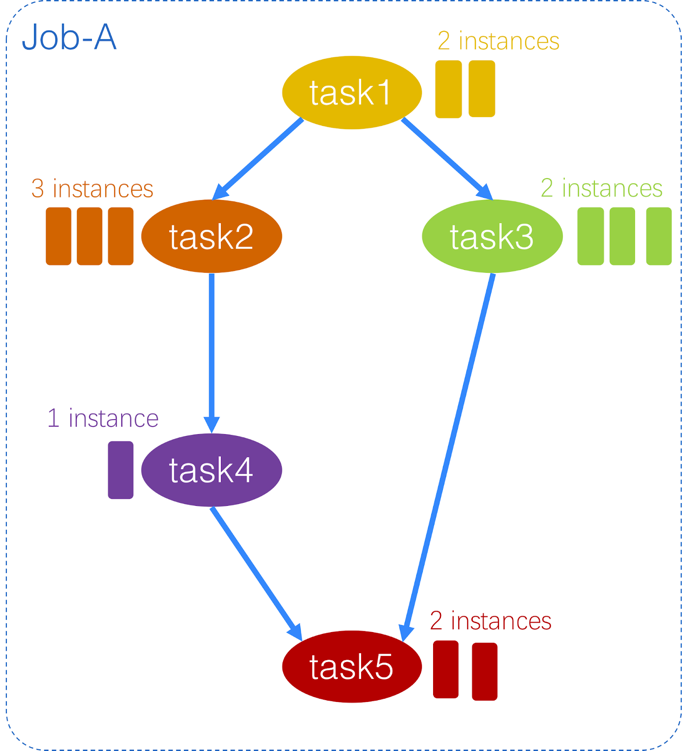
## Δομή Εργασιών

1. *Εισαγωγή*

Στις υπηρεσίες που προσφέρουν οι Cloud-Edge 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.

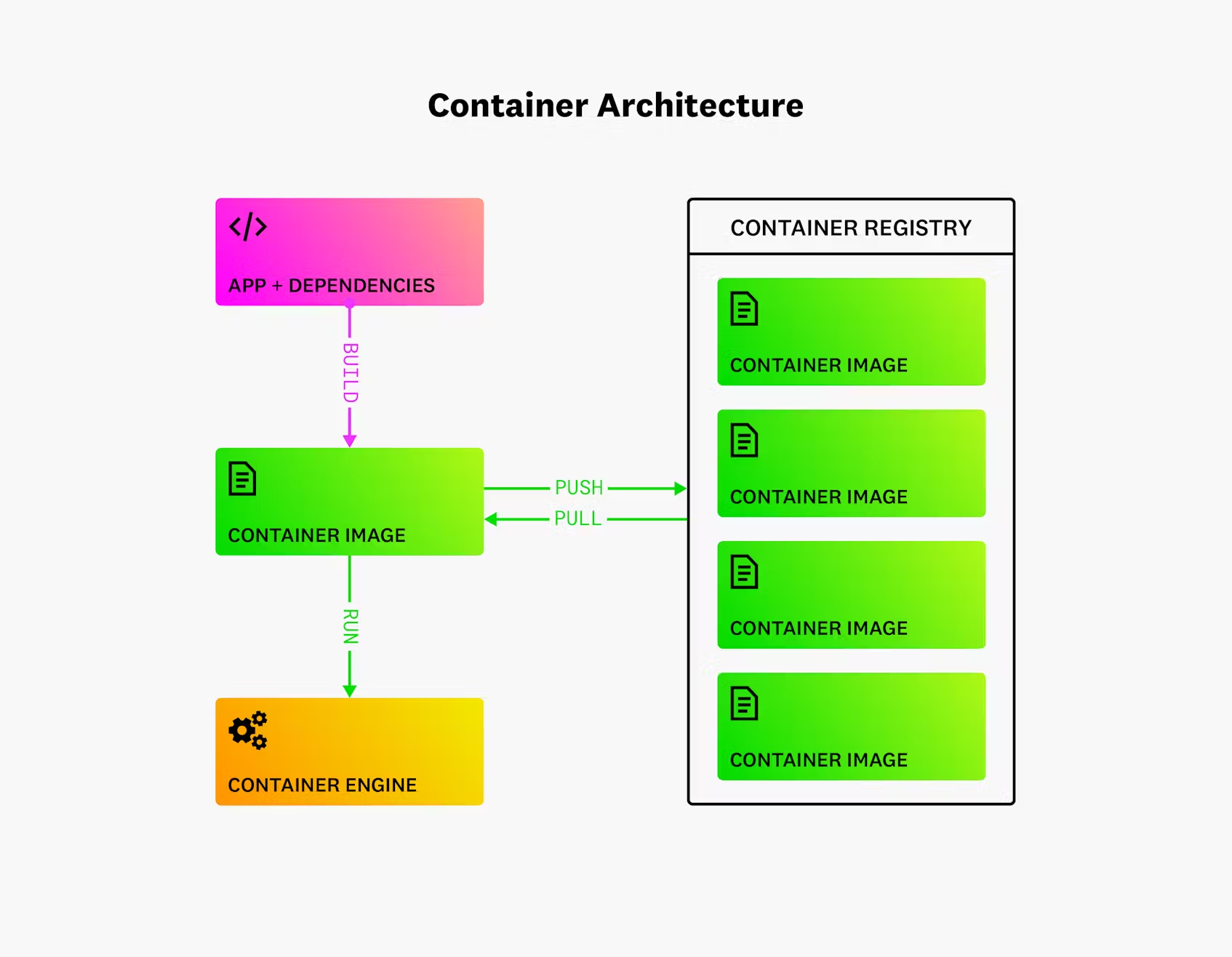
1. *Δομή Εργασιών*

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 αλληλεξαρτήσεις 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.

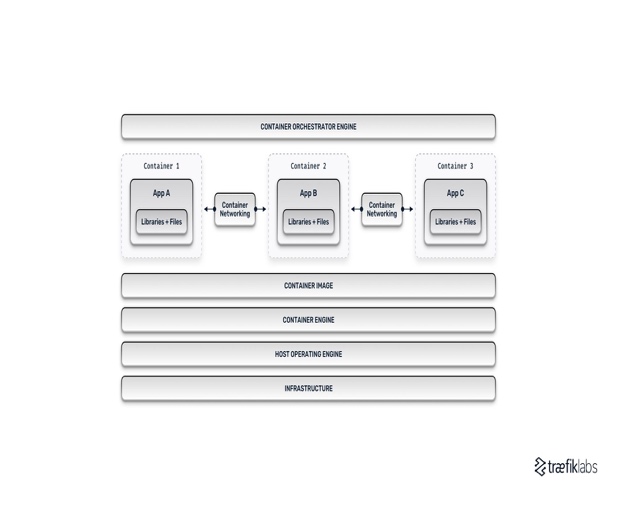


Ο χρόνος ολοκλήρωσης ενός job καθορίζεται από τα tasks. Κάθε task περιέχει τουλάχιστον ένα task instance, το μικρότερο προς κατανομή κομμάτι εργασίας στην υλοποίησή μας. Κάθε task instance του ίδιου task έχει τον ίδιο δυαδικό κώδικα και τις ίδιες ανάγκες σε πόρους. Ομοίως, η διάρκεια ενός task εξαρτάται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

****

**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.**



1. ***Διαχείριση εργασιών σε κατανεμημένα συστήματα***

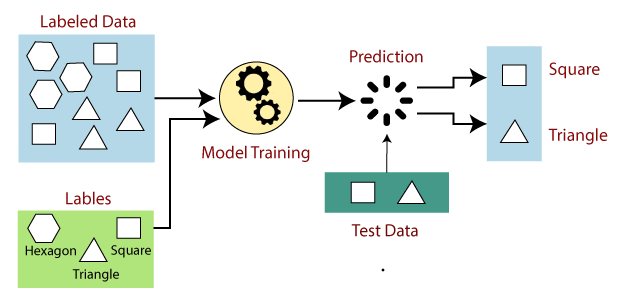
**Σε κατανεμημένα συστήματα, λόγω της φύσης της αρχιτεκτονικής λύσης, πραγματοποιείται συνεχής μεταφορά δεδομένων και εργασιών μεταξύ κόμβων του δικτύου. Η μεταφορά εργασιών, εφόσον η εκτέλεσή της έχει ξεκινήσει, μπορεί να πραγματοποιείται είτε από το σημείο που είχε μείνει, είτε από την αρχή. Στην περίπτωση 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 checkpoints 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 κατανεμημένες υποδομές, προσφέροντας γνώσεις σχετικά με τη περίπλοκη δυναμική και τις προκλήσεις που ενυπάρχουν στην ενορχήστρωση υπολογιστικών εργασιών σε όλο το συνεχές νέφους-ακμής. 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.

1. *Μηχανική Μάθηση*

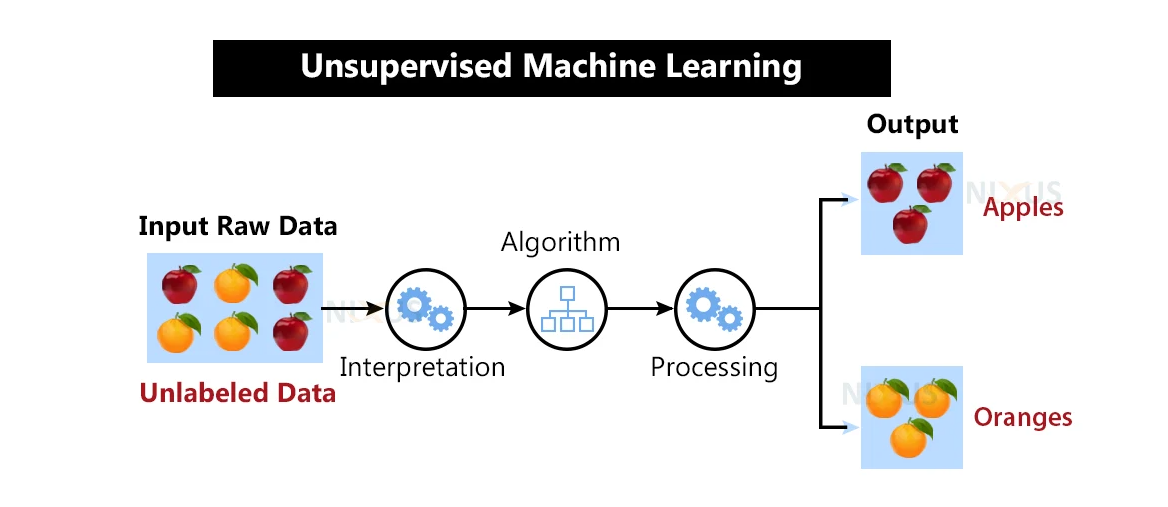
Οι αλγόριθμοι μηχανικής μάθησης μπορούν να αναγνωρίσουν συσχετίσεις και αποκλίνουσες συμπεριφορές δεδομένων μέσω περίπλοκων υπολογισμών όπως προσαρμογή της διάστασης του συστήματος και συνδυασμού διαφόρων παραμέτρων του συστήματος σε μαθηματικά και στατιστικά μοντέλα. Οι αλγόριθμοι μηχανικής μάθησης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

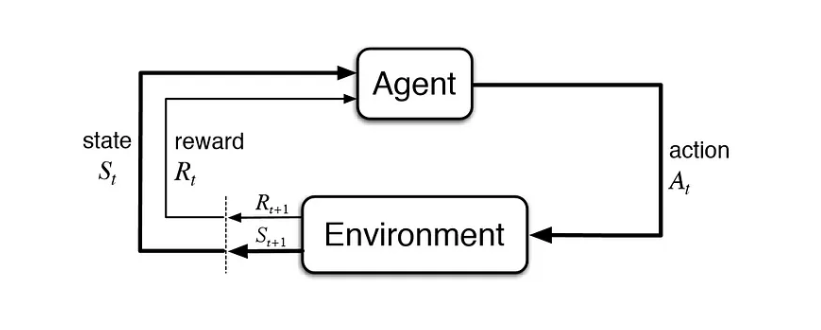


Οι αλγόριθμοι *μη επιβλεπόμενης μάθησης* μπορούν να εφαρμοστούν για την ανίχνευση ανωμαλιών χωρίς την ανάγκη δεδομένων εκπαίδευσης με ετικέτες. Ο στόχος αυτών των αλγορίθμων είναι να ανακαλύψουν κρυμμένα μοτίβα και σχέσεις στα δεδομένα, παρόμοια με την εκτίμηση της πυκνότητας στη στατιστική. Οι αλγόριθμοι 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. 

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.



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 ανάλυσή 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.

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:

όπου, V\* αποτελεί τη βέλτιστη συνάρτηση value.

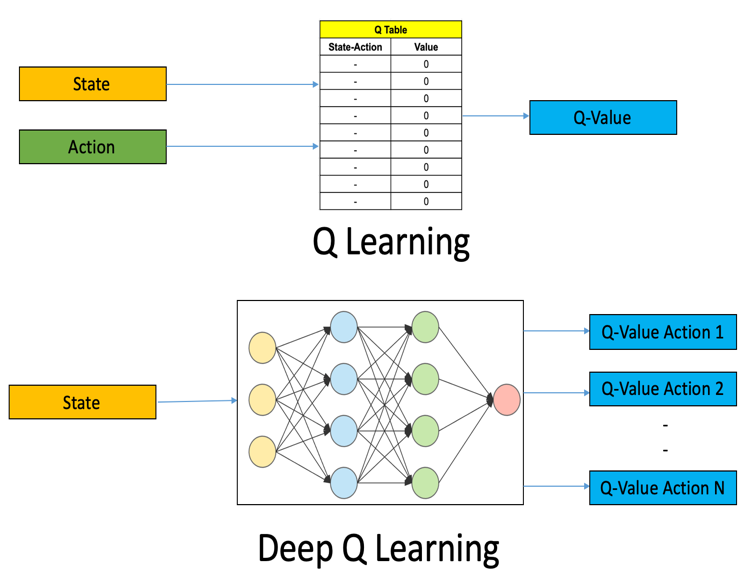
Για τον υπολογισμό της πολιτικής με τη βοηθητική συνάρτηση value, υπολογίζουμε τη V μέσω της Q, που θα οδηγήσει και στον υπολογισμό της συνάρτησης πολιτικής του μοντέλου.

και

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. H 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 actor επιλέγει ενέργειες και παρατηρεί την τιμή της V για το εκάστοτε state-action ζεύγος και ενημερώνει το Actor. H 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:

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 a ( learning rate) μειώνεται με την πάροδο του χρόνου καταλλήλως, οι τιμές Q θα συγκλίνουν στις βέλτιστες τιμές Q\* με πιθανότητα 1(Watkins, 1989; Tsitsiklis, 1994; Jaakkola, Jordan, & Singh, 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 ≠ 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:

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.   


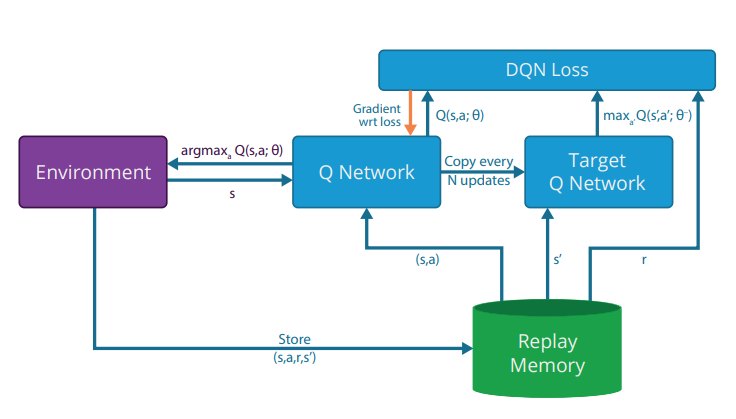
Πιο αναλυτικά, η μέθοδος *Deep Q-Learning* εισάγει ένα νευρωνικό δίκτυο, Deep Q-Network (DQN), και το εκπαιδεύει να λαμβάνει προσεγγιστικά καλές τιμές Q για καταστάσεις που δεν έχει ξαναβρεθεί. Η εκπαίδευση του DQN γίνεται μέσω της τροφοδότησης <s,a,r,s’> διανυσμάτων, που περιέχουν δηλαδή την τρέχουσα 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 <s,a> ζεύγους με βάση τον τύπο Bellman:

,   
όπου r η επιβράβευση του συστήματος για το ζεύγος <s,a>.

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:

,

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 ανανέωση τους, με σκοπό την σταθεροποίηση των αποτελεσμάτων στις συγκρίσεις που ανανεώνουν τα βάρη του νευρωνικού δικτύου για να μην παρατηρηθούν ανεπιθύμητες αυξομειώσεις.



γνώσεις σχετικά με τη περίπλοκη δυναμική και τις προκλήσεις που ενυπάρχουν στην ενορχήστρωση υπολογιστικών εργασιών σε όλο το συνεχές νέφους-ακμής. 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.

# Σχετικεσ Εργασιεσ

Η βέλτιστη σύνδεση και λειτουργία των Cloud-Edge-IoT 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 καθυστερήσεων και κόστους [10]. Ωστόσο, η χρήση edge υποδομών φέρει κάποιες δυσκολίες συγκριτικά με το Cloud που συνοψίζονται σε περιορισμένους πόρους για επεξεργασία, ετερογενή συστήματα επεξεργασίας με διαφορετικές αρχιτεκτονικές και φόρτο εργασίας που αλλάζει δυναμικά με το χρόνο [11]. 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[12][13]. 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, o Khazaei, et al 2012, χρησιμοποιεί το μοντέλο M/G/m/m+r για την ουρά του συστήματος[14], ενώ γίνεται χρήση και της αλυσίδας Markov για την αποτύπωση της πιθανοτικής συνάρτησης των χρόνων καθυστέρησης δρομολόγησης εργασιών και του ρυθμού άφιξής τους[17][18]. Ακόμη, χρησιμοποιούνται ευριστικοί αλγόριθμοι και αναπτύσσονται συναρτήσεις που προσπαθούν να βελτιστοποιήσουν μετρικές του συστήματος, όπως είναι ο particle swarm optimization[19] και ο genetic optimization αλγόριθμος [15][16], σεβόμενοι τις πολύπλοκες αλληλεξαρτήσεις τους.  
Σε αυτό το πρόβλημα, η έρευνα και ανάπτυξη AI μοντέλων που βασίζονται σε ML αλγορίθμους 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. 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. Μια τεχνική xρήσης 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.

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. O 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.

Όσον αφορά συσκευές σε edge, o Shang et al. [2023], χρησιμοποιεί μια μέθοδος για τη βελτιστοποίηση της απόφασης αποφόρτωσης του συστήματος από την υπολογιστική χρησιμοποίηση. 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.  
 Υπάρχουν πολλές προσπάθειες ανάπτυξης μοντέλων βαθιάς ενισχυτικής μάθησης που έχουν ρόλο δρομολογητή εργασιών σε ένα Cloud περιβάλλον. Πιο αναλυτικά, Ο Wei et al. [2018] καθώς και ο Dong et al. [2020], 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. 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 υπερπαραμέτρων. 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.

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. Επιπλέον, η εργασία της Javanetti et al. [2023], εισάγει το μοντέλο Actor-Critic DQN, χρησιμοποιώντας πολλαπλούς μεμονωμένους Actors για τη βελτιστοποίηση επιλογών δρομολόγησης eργασιών σε Cloud/Edge.

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. Οι 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. Ο Chen et al. [2019] προτείνει μια μέθοδο κατανομής πόρων βασισμένη σε RL με γνώμονα κέρδους τη μείωση της καθυστέρησης στην κατανομή και εκτέλεση εργασιών.  
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. 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.

# Υποδομη και Εργασιεσ

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.

## 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 κατανεμημένο, δηλαδή όλοι οι κόμβοι μπορούν να επικοινωνήσουν με όλους, ανεξαρτήτου cluster και θέσης μέσα στο δίκτυο.

## 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

## Δρομολόγηση Εργασιών

Αναπτύξαμε ένα σύστημα δρομολόγησης εργασιών, κατά το οποίο όταν φτάνει στο κεντρικό σύστημα μια εργασία, αυτό την κατανείμει 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.

# Μοντελο Ενισχυτικησ Μαθησησ

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 <S,a,r,S’>, 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 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:

1. 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

1. 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:

1. *Κενή ενέργεια*, η οποία δηλώνει την επιλογή ότι η κατάσταση δε χρειάζεται κάποια αλλαγή
2. *Near-Edge domain Scale Up* - Αύξηση των δεσμευμένων κόμβων στο Near-Edge domain κατά 10 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον δεν έχει φτάσει στο όριο των δυνατοτήτων του.
3. *Near-Edge domain Scale Down* - Μείωση των δεσμευμένων κόμβων του Near-Edge domain κατά 10 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον υπάρχει.
4. *Far-Edge domain Scale Up* - Αύξηση των δεσμευμένων κόμβων στο Near-Edge domain κατά 10 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον δεν έχει φτάσει στο όριο των δυνατοτήτων του.
5. *Far-Edge domain Scale Down* - Μείωση των δεσμευμένων κόμβων του Near-Edge domain κατά 10 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον υπάρχει.
6. *Cloud domain Scale Up* - Αύξηση των δεσμευμένων κόμβων στο Near-Edge domain κατά 10 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον δεν έχει φτάσει στο όριο των δυνατοτήτων του.
7. *Cloud domain Scale Down* - Μείωση των δεσμευμένων κόμβων του Near-Edge domain κατά 10 κόμβους 3 μηχανημάτων με τα προβλεπόμενα χαρακτηριστικά εφόσον υπάρχει.
8. *Μεταφορά 10 workloads* από το Near-Edge στο Far-Edge domain
9. *Μεταφορά 10 workloads* από το Near-Edge στο Cloud domain
10. *Μεταφορά 10 workloads* από το Far-Edge στο Near-Edge domain
11. *Μεταφορά 10 workloads* από το Far-Edge στο Cloud domain
12. *Μεταφορά 10 workloads* ο Cloud domain στο Near-Edge domain
13. *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.

1. 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:

1. 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.

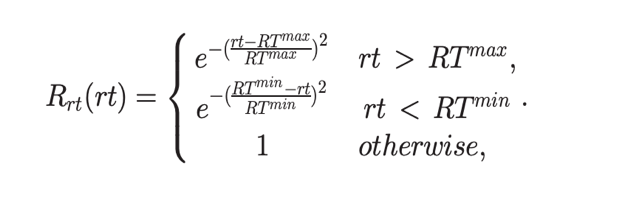
Ενώ από τη μία, η ύπαρξη ενός ελλιπώς εκμεταλλευόμενου περιβάλλοντος μπορεί να προσφέρει στο σύστημα υψηλές QoS μετρικές (χαμηλό response time), η σπατάλη πόρων δεν είναι αποδεκτή από τους πελάτες σε cloud-edge υποδομές. Η υπο-εκμετάλλευση διαθέσιμων πόρων αυξάνει το κόστος από την άποψη της χρηματικής 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 αποτελεί την μετρική που εκφράζει το ποσοστό χρησιμοποίησης των πόρων του κόμβου και το ορίζει τη μέγιστη αποδεκτή χρησιμοποίηση για τον αντίστοιχο πόρο j (και οι δύο μετρικές παίρνουν τιμές στην κλίμακα 0 με 1). Έστω, επίσης, ότι είναι Ν 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 Rut γίνεται αρνητική στους αντίστοιχους κόμβους και έτσι επηρεάζει αρνητικά την τελική ανταμοιβή.

(1)

1. 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 (2) όπου

και 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.

 (2)

1. 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 όποιο είναι δεσμευμένα μηχανήματα από την υποδομή μας, ορίζουμε ένα κόστος χρήσης μηχανημάτων σε κάθε φυσικό μέρος του δικτύου μας με βάση ρεαλιστικές 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, έχουμε:

* = 0.056 για κάθε μηχάνημα
* = 0.054 για κάθε μηχάνημα
* = 0.052 για κάθε μηχάνημα

# υλοποιηση

Στην υλοποίησή μας, θεωρήσαμε μία υποδομή τριών δικτυακών παράλληλων επιπέδων, 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. 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. Βάση αρχών των συγκεκριμένων εργαλείων, κάθε 50 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.

# Αποτελεσματα

##### Βιβλιογραφικεσ Αναφορεσ

1. Edge Computing Market by Component (Hardware, Software, Services), Application (Real-time Data Processing & Analysis, Predictive Maintenance, & Optimization), Organization Size (Large Enterprises, SMEs), Vertical and Region - Global Forecast to 2029
2. Deloitte. Unlocking the Business Value of Edge Computing. 2019.
3. Gartner. Top Strategic IoT Trends and Technologies Through 2023. March 2019.
4. L. Ai, B. Tan, J. Zhang, R. Wang and J. Wu, "Dynamic Offloading Strategy for Delay-Sensitive Task in Mobile-Edge Computing Networks," in IEEE Internet of Things Journal, vol. 10, no. 1, pp. 526-538, 1 Jan.1, 2023, doi: 10.1109/JIOT.2022.3202797. keywords: {Task analysis;Servers;Mobile handsets;Delays;Optimization;Internet of Things;Energy consumption;Mobile-edge computing (MEC);reinforcement learning (RL);task offloading}
5. Sahni, Yuvraj & Cao, Jiannong & Yang, Lei & Wang, Shengwei. (2022). Distributed resource scheduling in edge computing: Problems, solutions, and opportunities. Computer Networks. 219. 109430. 10.1016/j.comnet.2022.109430.
6. Mohammed Laroui, Boubakr Nour, Hassine Moungla, Moussa Ali Cherif, Hossam Afifi, et al.. Edge and fog computing for IoT: a survey on current research activities & future directions. Computer Communications, 2021, 180, pp.210-231. ff10.1016/j.comcom.2021.09.003ff. ffhal-03414397f
7. Young DJ. Wireless powering and data telemetry for biomedical implants. Annu Int Conf IEEE Eng Med Biol Soc. 2009;2009:3221-4. doi: 10.1109/IEMBS.2009.5333163. PMID: 19964060.
8. Liu, Shuo, Shi Qiu, Huayi Li, and Ming Liu. 2023. "Real-Time Telemetry-Based Recognition and Prediction of Satellite State Using TS-GCN Network" Electronics 12, no. 23: 4824. https://doi.org/10.3390/electronics12234824
9. Zhang, Tianyu, Chuanyu Xue, Jiachen Wang, Zelin Yun, Natong Lin and Song Han. “A Survey on Industrial Internet of Things (IIoT) Testbeds for Connectivity Research.” (2024).
10. Toor, A., Islam, S., Ahmed, G. et al. Energy efficient edge-of-things. J Wireless Com Network 2019, 82 (2019). https://doi.org/10.1186/s13638-019-1394-4
11. Hong, Cheol-Ho & Varghese, Blesson. (2019). Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms. ACM Computing Surveys. 52. 1-37. 10.1145/3326066
12. S. Hu, W. Shi and G. Li, "CEC: A Containerized Edge Computing Framework for Dynamic Resource Provisioning," in IEEE Transactions on Mobile Computing, vol. 22, no. 7, pp. 3840-3854, 1 July 2023, doi: 10.1109/TMC.2022.3147800.
13. Y. -W. Chan, H. Fathoni, H. -Y. Yen and C. -T. Yang, "Implementation of a Cluster-Based Heterogeneous Edge Computing System for Resource Monitoring and Performance Evaluation," in IEEE Access, vol. 10, pp. 38458-38471, 2022, doi: 10.1109/ACCESS.2022.3166154.
14. Khazaei, Hamzeh & Mišić, Jelena & Misic, Vojislav. (2012). Performance Analysis of Cloud Computing Centers Using M/G/m/m+r Queuing Systems. Parallel and Distributed Systems, IEEE Transactions on. 23. 1 - 1. 10.1109/TPDS.2011.199.
15. J. Li, X. Zhang, J. Wei, Z. Ji, and Z. Wei, GARLSched: Generative adversarial deep reinforcement learning task scheduling optimization for large-scale high performance computing systems, Future Gener. Comput. Syst., vol. 135, pp. 259–269, 2022.
16. P. Gazori, D. Rahbari, and M. Nickray, Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach, Future Gener. Comput. Syst., vol. 110, pp. 1098–1115, 2020
17. Y. Wei, L. Pan, S. Liu, L. Wu, and X. Meng, DRLscheduling: An intelligent QoS-aware job scheduling framework for applications in clouds, IEEE Access, vol. 6, pp. 55112–55125, 2018
18. T. Dong, F. Xue, C. Xiao, and J. Li, Task scheduling based on deep reinforcement learning in a cloud manufacturing environment, Concurr. Comput. Pract. Exp., vol. 32, no. 11, p. e5654, 2020.
19. Pirozmand, P., Jalalinejad, H., Hosseinabadi, A.A.R. *et al.* An improved particle swarm optimization algorithm for task scheduling in cloud computing. *J Ambient Intell Human Comput* **14**, 4313–4327 (2023). https://doi.org/10.1007/s12652-023-04541-9
20. Moreno-Vozmediano, R., Montero, R.S., Huedo, E., Llorente, I.M.: Efficient resource provisioning for elastic cloud services based on machine learning techniques. Journal of Cloud Computing 8(1), 5 (2019). DOI 10.1186/s13677-019-0128-9
21. Yadav, M.P., Rohit, Yadav, D.K.: Resource provisioning through machine learning in cloud services. Arabian Journal for Science and Engineering (2021). DOI 10.1007/s13369-021-05864-5
22. Witanto, J.N., Lim, H., Atiquzzaman, M.: Adaptive selection of dynamic vm consolidation algorithm using neural network for cloud resource management. Future Generation Computer Systems 87, 35–42 (2018). DOI doi.org/10.1016/j.future.2018.04.075
23. Jin, Y., Bouzid, M., Kostadinov, D., Aghasaryan, A.: Resource management of cloud-enabled systems using model-free reinforcement learning. Annals of Telecommunications 74(9), 625–636 (2019). DOI 10.1007/s12243-019-00720-y
24. Siddesha, K., Jayaramaiah, G.V. & Singh, C. A novel deep reinforcement learning scheme for task scheduling in cloud computing. *Cluster Comput* **25**, 4171–4188 (2022). https://doi.org/10.1007/s10586-022-03630-2
25. S. Swarup, E. M. Shakshuki, and A. Yasar, Energy efficient task scheduling in fog environment using deep reinforcement learning approach, Procedia Comput. Sci., vol. 191, pp. 65–75, 2021.
26. Wang, H., Muñoz-González, L., Eklund, D., & Raza, S. (2021). 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. doi:10.1145/3448300.3467827
27. C. Shang, Y. Sun, H. Luo and M. Guizani, "Computation Offloading and Resource Allocation in NOMA–MEC: A Deep Reinforcement Learning Approach," in IEEE Internet of Things Journal, vol. 10, no. 17, pp. 15464-15476, 1 Sept.1, 2023, doi: 10.1109/JIOT.2023.3264206.
28. J. Chen et al., "Deep Reinforcement Learning Based Resource Allocation in Multi-UAV-Aided MEC Networks," in IEEE Transactions on Communications, vol. 71, no. 1, pp. 296-309, Jan. 2023, doi: 10.1109/TCOMM.2022.3226193.
29. Amanda Jayanetti, Saman Halgamuge, Rajkumar Buyya, Deep reinforcement learning for energy and time optimized scheduling of precedence-constrained tasks in edge–cloud computing environments, Future Generation Computer Systems, Volume 137, 2022, Pages 14-30, ISSN 0167-739X, https://doi.org/10.1016/j.future.2022.06.012.
30. Guangyao Zhou, Ruiming Wen, Wenhong Tian, Rajkumar Buyya, Deep reinforcement learning-based algorithms selectors for the resource scheduling in hierarchical Cloud computing, Journal of Network and Computer Applications, Volume 208, 2022, 103520, ISSN 1084-8045, https://doi.org/10.1016/j.jnca.2022.103520.
31. A. Alsarhan, A. Itradat, A. Y. Al-Dubai, A. Y. Zomaya, and G. Min, “Adaptive resource allocation and provisioning in multi-service cloud environments,” IEEE Transactions on Parallel and Distributed Systems (TPDS), vol. 29, no. 1, pp. 31–42, 2018.
32. Mobile-Edge Computing via Deep Reinforcement Learning,” in 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), Aug. 2018, pp. 1–6, iSSN: 1090-3038.
33. Kuang, Cheng & Duan, Meng & Lv, Tao & Wu, Yingjun & Ren, Xiangyu & Wang, Luping. (2023). ODRL: Application of Reinforcement Learning in Priority Scheduling for Running Cost Optimization. 10.21203/rs.3.rs-3323844/v1.

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)