Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης & Υπολογιστων

Πρόβλεψη Πολυχαρακτηριστικών Χρονοσειρών Υπολογιστικού Φορτίου & Διαχείριση Πόρων με Τεχνικές Μηχανικής Μάθησης

Μελέτη και υλοποίηση

Διπλωματικη Εργασια

του

ΑΛΕΞΑΝΔΡΟΥ Ν. ΤΣΑΦΟΥ

Επιβλέπων: Δημήτριος Τσουμάκος
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2023

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης & Υπολογιστων

Πρόβλεψη Πολυχαρακτηριστικών Χρονοσειρών Υπολογιστικού Φορτίου & Διαχείριση Πόρων με Τεχνικές Μηχανικής Μάθησης

Μελέτη και υλοποίηση

Διπλωματικη Εργασια

του

ΑΛΕΞΑΝΔΡΟΥ Ν. ΤΣΑΦΟΥ

Επιβλέπων: Δημήτριος Τσουμάκος
Αναπληρωτής Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18η Οκτωβρίου 2023.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Δημήτριος Τσουμάκος	Γεώργιος Γκούμας	Ιωάννης Κωνσταντίνου
Αναπληρωτής Καθηγητής Ε.Μ.Π.	Αναπληρωτής Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2023

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης & Υπολογιστων

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.

(Υπογραφή)

Αλέξανδρος Τσάφος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

30 Σεπτεμβρίου 2023

Περίληψη

Τα τελευταία χρόνια το υπολογιστικό νέφος είναι ένας από τους πιο επιδραστικούς κλάδους της επιστήμης των υπολογιστών. Οι cloud υπηρεσίες γίνονται ολοένα και πιο δημοφιλείς και ο φόρτος εργασίας των παρόχων συνεχώς και αυξάνεται. Σύμφωνα με τον Διεθνή Οργανισμό Ενέργειας η κατανάλωση των Data Centres αποτελεί το 1-1.5% της παγκόσμιας κατανάλωσης ενέργειας, με το νούμερο αυτό να αναμένεται ακόμα και να εξαπλασιαστεί μέχρι το 2031. Η ανάγκη για μείωση χρησιμοποιούμενων πόρων είναι πιο επιτακτική από ποτέ. Αυτό προϋποθέτει μια ακριβή πρόβλεψη της κατάστασης του φορτίου εργασίας του συστήματος και χρήση αυτής για την διαχείριση των πόρων. Στην παρούσα διπλωματική, ασχοληθήκαμε με την πρόβλεψη πολυμεταβλητών χρονοσειρών φορτίου εργασίας και εξετάσαμε αν είναι σκόπιμη η υλοποίηση ενός συστήματος αυτόματης διαχείρισης. Για τις προβλέψεις εξετάστηκαν τέσσερα διαφορετικά μοντέλα γραμμικής παλινδρόμησης (Γραμμική Παλινδρόμηση, Παλινδρόμηση Lasso, Παλινδρόμηση Ridge, Παλινδρόμηση ElasticNet) καθώς και νευρωνικά δίκτυα (DNN, CNN, LSTM), τα οποία συγκρίθηκαν για κάθε σύνολο δεδομένων με τρεις μετρικές (MAE, ME, PRMSE). Ο μικρός χρόνος προσαρμογής τους, επέτρεψε την επιλογή του βέλτιστου μοντέλου πρόβλεψης για κάθε υπολογιστικό σύστημα μετά από σύγκριση. Με βάση τις προβλέψεις δείξαμε πως ένα σύστημα αυτόματης διαχείρισης θα μπορούσε να μειώσει την χρήση πόρων έως και 60% σε σχέση με συστήματα που δεν χρησιμοποιούν κάποια έξυπνη πολιτική. Τα δεδομένα προέκυψαν από ανάλυση και επεξεργασία πραγματικών log data κατανεμημένων συστημάτων αλλά και από δημιουργία συνθετικών δεδομένων χρονοσειρών. Τέλος, εξετάστηκαν και αναφέρονται μελλοντικές επεκτάσεις της εργασίας, ώστε να υλοποιηθεί το αυτόματο σύστημα διαχείρισης των πόρων.

Λέξεις Κλειδιά

Υπολογιστικό Νέφος, Ανάλυση Δεδομένων, Πρόβλεψη χρονοσειρών, Μηχανική Μάθηση, Γραμμική Παλινδρόμηση, Βαθιά Νευρωνικά Δίκτυα, Αναδρομικά Νευρωνικά Δίκτυα, Συνελικτικά Νευρωνικά Δίκτυα, LSTM

Abstract

In recent years, cloud computing has become one of the most influential fields in computer science. Cloud services are increasingly popular, and the workload of providers continues to grow steadily. According to the International Energy Agency, data center energy consumption accounts for 1-1.5% of global energy consumption, which is expected to, in some cases, rise by six times by 2031. The need for optimal resource management is more pressing than ever, requiring accurate workload forecasting. In this dissertation, we focused on predicting workload multivariate time series data and explored the feasibility of implementing an automated resource management system. Four different linear regression models (Linear Regression, Lasso Regression, Ridge Regression, ElasticNet Regression), as well as neural networks (DNN, CNN, LSTM) were examined, whose performance was compared for each dataset using three metrics (MAE, ME, PRMSE). Their fast fit time allowed the selection of the optimal prediction model for each computing system after comparison. Using the predictions, we showed that an automated management system could decrease the used resources by up to 60% compared to systems that do not use a smart management policy. The data were generated by analyzing and processing real-world distributed systems log data as well as the creation of synthetic time series data. Finally, we examined and mentioned future extensions of the project for the implementation automatic resource provisioning system.

Keywords

Cloud Computing, Data Analysis, Time Series Forecasting, Machine Learning, Linear Regression, Deep Neural Networks, Recurrent Neural Networks, Convolutional Neural Networks, LSTM

στους γονείς μου & στην αδερφή μου

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. Τσουμάκο Δημήτριο για την ευκαιρία και για την επίβλεψη αυτής της διπλωματικής εργασίας. Επιπλέον ευχαριστώ ιδιαίτερα τους υποψήφιους διδάκτορες Χαλβαντζή Νικόλαο και Μπιτσάκο Κωνσταντίνο για την βοήθεια και την καθοδήγησή τους στο πλαίσιο της εργασίας, καθώς και για την εξαιρετική συνεργασία που είχαμε. Τέλος, ευχαριστώ βαθύτατα τους κοντινούς μου ανθρώπους, τους φίλους και τους συμφοιτητές μου για όλες τις όμορφες αναμνήσεις των τελευταίων πέντε χρόνων, για την αμοιβαία υποστήριξη τόσο ακαδημαϊκά, όσο και έξω από το πανεπιστήμιο, για τα όνειρα και τις φιλοδοξίες τις οποίες μοιραζόμαστε.

Αθήνα, Σεπτέμβριος 2023

Αλέξανδρος Τσάφος

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Τα υπολογιστικά νέφη καθίστανται όλο και περισσότερο δημοφιλή την τελευταία δεκαετία. Υπηρεσίες Cloud, όπως το Google Docs για εφαρμογές γραφείου, το Google Drive για αποθήκευση αρχείων και το Amazon Web Services (AWS) για φιλοξενία εφαρμογών, παρέχονται από εταιρείες - κολοσσούς στον τομέα, όπως η Google, η Amazon, η Microsoft. Αυτές οι υπηρεσίες δεν απαιτούν κάποια σημαντική υπολογιστική ισχύ από τον χρήστη τους, αφού αξιοποιούν απομακρυσμένους πόρους, σε συστάδες υπολογιστών των προαναφερθέντων εταιρειών. Με αυτόν τον τρόπο, ο χρήστης δεν χρειάζεται να προβεί σε ακριβές αναβαθμίσεις εξοπλισμού για την εκτέλεση εργασιών μεγάλης κλίμακας, ούτε και στην συντήρηση αυτού. Γι' αυτούς τους λόγους οι υπηρεσίες Cloud αξιοποιούνται από συνεχώς αυξανόμενο αριθμό χρηστών, συμπεριλαμβανομένων και εταιρειών.

Ωστόσο, προκύπτει ανησυχία για την βιωσιμότητα των συστάδων υπολογιστών και την επίδρασή τους στο περιβάλλον. Σύμφωνα με τον Παγκόσμιο Οργανισμό Ενέργειας (International Energy Agency - IEA), ποσοστό 1-1.5% της παγκόσμιας κατανάλωσης ενέργειας προέρχεται από την λειτουργία μόνο των κέντρων διαχείρισης μνήμης (Data Centres), με συγκεκριμένες χώρες να παρουσιάζουν πολύ μεγαλύτερα ποσοστά. Για παράδειγμα, η κατανάλωση ενέργειας από Data Centres έχει τριπλασιαστεί από το 2015 στην Ιρλανδία, φτάνοντας το 18% της συνολικής κατανάλωσης το 2022. Στην Δανία το αντίστοιχο ποσοστό αναμένεται να εξαπλασιαστεί μέχρι το 2030, με αποτέλεσμα να φτάσει το 15% της κατανάλωσης της χώρας [5].

Παράλληλα, υπάρχει και η οπτική των επιχειρήσεων που παρέχουν αυτούς τους πόρους. Μια αποτελεσματική διαχείριση των υπολογιστικών πόρων μπορεί να πολλαπλασιάσει τα έσοδά τους, κρατώντας τους χρήστες τους το ίδιο ευχαριστημένους. Η πιο απλή και συνηθισμένη πολιτική είναι η στατική κατανομή πόρων, με επεξεργαστική δύναμη και μνήμη αρκετές ώστε να καλύπτουν κάθε στιγμή τις ανάγκες φορτίου. Όμως, οι ανάγκες αυτές μεταβάλλονται συνεχώς, με αποτέλεσμα οι πόροι που κατανέμονται να μένουν αχρησιμοποίητοι για μεγάλα χρονικά διαστήματα και οι εταιρείες να αφήνουν πιθανά κέρδη να χαθούν.

1.1 Αντικείμενο της διπλωματικής

Για τους λόγους που αναφέρθηκαν, η ανάγκη για ένα αυτόματο σύστημα διαχείρισης πόρων, που θα μειώνει την ανεκμετάλλευτη κατανάλωση, είναι επιτακτική. Αντικείμενο της διπλωματικής αποτελεί η αυτοματοποιημένη διαχείριση πόρων σε υπολογιστικά συστήματα νέφους. Το πρόβλημα της διαχείρισης πόρων σε περιβάλλοντα υπολογιστικών νεφών απασχολεί την ερευνητική κοινότητα εδώ και αρκετά χρόνια λόγω της δυσκολίας της αυτοματοποίησης της διαδικασίας κλιμάκωσης και της κατανομής ακριβούς ποσότητας πόρων για να διασφαλιστεί η ποιότητα υπηρεσιών (QoS).

Υπάρχουν δύο είδη αυτόματης κλιμάκωσης: η αντιδραστική (reactive) κατανομή πόρων, η οποία ανταποκρίνεται σε αλλαγές του συστήματος και ανακατανέμει κατάλληλα τους πόρους, και η προληπτική (proactive) κατανομή πόρων, κατά η οποία οι πόροι ανακατανέμονται πριν προκύψει κάποια αλλαγή. Η αντιδραστική προσέγγιση αντιμετωπίζει προκλήσεις στη διαχείριση μεγάλων αλλαγών στο φορτίο εργασίας, αφού οι μεγάλες ανακατανομές χρειάζονται χρόνο και δημιουργούν μεγάλες αναμονές για τους χρήστες. Αντίθετα, οι προληπτικές προσεγγίσεις μπορούν να προετοιμάσουν την ανακατανομή των πόρων πριν από την ανάγκη, εξασφαλίζοντας την άμεση διαθεσιμότητα κατά την προβλεπόμενη αύξηση του φορτίου εργασίας. Βέβαια, εγκυμονεί ο κίνδυνος λανθασμένης πρόβλεψης, στην οποία περίπτωση οι χρήστες δεν εξυπηρετούνται. Στο πλαίσιο της διπλωματικής θα εξετασθεί το δεύτερο είδος κλιμάκωσης, σε συνθετικά και πραγματικά δεδομένα υπολογιστικών νεφών.

1.2 Σχετικό Έργο

Στο κεφάλαιο αυτό γίνεται μια περιγραφή των σχετικών εργασιών με βάση την ελαστικότητα και την διαχείριση πόρων στα υπολογιστικά νέφη. Η δυναμική χρονοδρομολόγηση πόρων έχει ερευνηθεί από την σκοπιά της πρόβλεψης του φορτίου εργασίας σε πραγματικό χρόνο, όμως δεν έχουν συνδυαστεί με κάποια state-of-the-art τεχνική με την χρήση της ελαστικότητας και της ενισχυτικής μάθησης.

Ο Tiramola [6] είναι μία υπηρεσία που χρησιμοποιείται για να αυξομειώνει αυτόματα το μέγεθος των υπολογιστικών συστάδων σε περιβάλλοντα υπολογιστικών νεφών. Σχεδιάστηκε για να διαχειρίζεται αυτόματα και σε πραγματικό χρόνο την αύξηση ή την μείωση των υπολογιστικών πόρων, σε NoSQL clusters, ανάλογα με τις προδιαγραφές και τις απαιτήσεις του εκάστοτε χρήστη. Η λήψη αποφάσεων σχετικά με τις προτεινόμενες ενέργειες γίνεται μοντελοποιώντας τη συστάδα ως μια μαρκοβιανή διαδικασία αποφάσεων που προσδιορίζει συνεχώς την πιο επωφελής δράση ανάλογα με την συνάρτηση επιβράβευσης που ορίζει ο χρήστης. Το πρόβλημα με αυτήν την υλοποίηση ήταν ότι στις μαρκοβιανές διαδικασίες ο χώρος καταστάσεων είναι διακριτές τιμές και οι παράμετροι εισόδου είναι συνεχείς μεταβλητές. Αυτό οδηγεί σε έναν πολύ μεγάλο χώρο καταστάσεων που είναι δύσκολο να διαχειριστεί η υπηρεσία.

Με βάση την αρχιτεκτονική του Tiramola έγιναν περαιτέρω έρευνες ώστε να βρεθεί μια πιο αποδοτική τεχνική λήψης αποφάσεων. Το 2017 προτάθηκε ένα άλλο σύστημα αποφάσεων που βασίζεται σε μια προσαρμοσμένη μορφή ενισχυτικής μάθησης [7]. Η προσέγγιση αυτή προτείνει έναν αλγόριθμο που μοντελοποιεί το περιβάλλον ως μια μαρκοβιανή διαδικασία και

χρησιμοποιεί δέντρα αποφάσεων χωρίζοντας δυναμικά τον χώρο κατάστασης όταν χρειάζεται, σύμφωνα με τις οδηγίες της συμπεριφοράς του συστήματος. Με αυτόν τον τρόπο μπορεί να γενικεύσει τις καταστάσεις του περιβάλλοντος ώστε να λειτουργεί για μεγαλύτερο χώρο καταστάσεων σε σχέση με την αρχική υλοποίηση.

Εν συνεχεία των παραπάνω, το 2018 προτάθηκε το DERP[8], το οποίο είναι μια πρωτοποριακή υπηρεσία δυναμικής διαχείρισης υπολογιστικών πόρων με χρήση βαθιάς ενισχυτικής μάθησης. Λαμβάνοντας υπόψιν τον αλγόριθμο βαθιάς ενισχυτικής μάθησης που παρουσίασε η DeepMind δημιουργήθηκαν τρεις διαφορετικοί πράκτορες (Single Deep Q-Learning, Full Deep Q-Learning, Double Deep Q-Learning) που αναλάμβαναν την λήψη αποφάσεων στο περιβάλλον των υπολογιστικών νεφών. Αποδείχτηκε ότι αυτή η τεχνική ήταν περισσότερο αποδοτική από τις προηγούμενες. Το σύστημα του DERP, συγκρίθηκε με ένα άλλο προτεινόμενο σύστημα στην πτυχιακή εργασία της Ε. Κουλέτου [9].

Στην παρούσα διπλωματική, το κύριο αντικείμενο δεν είναι να βελτιωθούν άμεσα τα παραπάνω συστήματα. Αντ' αυτού, θα εξετασθεί η αξία του συνδυασμού των παραπάνω μηχανισμών με την χρήση μηχανικής μάθησης για την ακριβή πρόβλεψη υπολογιστικού φορτίου. Στην πτυχιακή εργασία της Ε. Κουλέτου, έγινε μια αρχή πάνω στο θέμα, εξετάζοντας φορτίο που ορίστηκε με μία διάσταση (αντί να διαχωριστεί σε επεξεργαστικό φορτίο και φορτίο μνήμης) και προέκυψε μόνο από παραγωγή συνθετικών δεδομένων. Στην περίπτωσή μας, θα αναλυθούν και θα αξιολογηθούν συστήματα που χρησιμοποιούν πραγματικά δεδομένα, κάνοντας ένα βήμα παραπάνω προς την τελική δημιουργία ενός βέλτιστου proactive συστήματος διαχείρισης υπολογιστικών πόρων.

1.3 Συνεισφορά

Σε αυτή την ενότητα, περιγράφονται οι βασικές συνεισφορές της παρούσας πτυχιακής εργασίας. Η έρευνα επικεντρώθηκε στον χειρισμό δεδομένων και την ακριβή πρόβλεψη υπολογιστικού φορτίου, με τελικό σκοπό την τροφοδότησή τους ως είσοδο σε κάποιο autoscaling σύστημα. Συνοψίζονται τα βασικά αποτελέσματα της μελέτης, τονίζοντας τους συγκεκριμένους τρόπους με τους οποίους η εργασία έχει προσθέσει στο υπάρχον σώμα γνώσεων στον τομέα της αυτόματης κλιμάκωσης Cloud συστημάτων.

Αναλυτικά, στην παρούσα εργασία:

- Αναζητήθηκαν, βρέθηκαν και αναλύθηκαν δεδομένα από πραγματικά log data υπολογιστικών νεφών. Τα δεδομένα αυτά καθαρίστηκαν από κενές και "βρόμικες" τιμές, και μετατράπηκαν σε δεδομένα χρονοσειρών, τα οποία διατίθενται σε δημόσιο αποθετήριο κώδικα [10].

- Παρήχθησαν συνθετικά δεδομένα παρόμοιας δομής με τα πραγματικά, αξιοποιώντας βασικές αρχές χαρακτηριστικών των δεδομένων χρονοσειρών.

- Μελετήθηκαν, δοκιμάστηκαν και προτάθηκαν μοντέλα μηχανικής μάθησης που κρίθηκαν κατάλληλα για την πρόβλεψη χρονοσειρών. Οι προτάσεις που έγιναν, κατάφεραν στην περίπτωσή μας να λειτουργήσουν αποδοτικά για περίπλοκες χρονοσειρές φορτίου εργασίας, πραγματοποιώντας προβλέψεις με PRMSE 1.5-8%.

- Εφαρμόστηκε προσαρμοσμένη μέθοδος πρόβλεψης για κάθε χαρακτηριστικό και κάθε σύνολο δεδομένων. Λόγω των μικρών χρόνων προσαρμογής, η τακτική αυτή προτείνεται ως βέλτιστη σε κάθε πιθανό εξεταζόμενο σύστημα, ώστε να γίνει προσαρμογή στα δικά του μοναδικά μοτίβα, επιτρέποντας ταυτόχρονα την επανεξέταση της επιλογής βέλτιστου predictor ανά τακτά χρονικά διαστήματα.

- Έγιναν πειράματα για την εξέταση ενός συστήματος αυτόματης διαχείρισης υπολογιστικών πόρων. Συγκεκριμένα, μοντελοποιώντας μία απλή πολιτική που θα ακολουθούσε τις προβλέψεις λαμβάνοντας υπόψιν ένα διάστημα εμπιστοσύνης 10-20% οι χρησιμοποιούμενοι πόροι παρουσίασαν έως και 64% σε σχέση με μια βέλτιστη πολιτική κατανομής σταθερών πόρων.

1.4 Οργάνωση του τόμου

Η εργασία αυτή είναι οργανωμένη σε έξι κεφάλαια:

- Στο **Κεφάλαιο 2** δίνεται το θεωρητικό υπόβαθρο των βασικών τεχνολογιών που σχετίζονται με την παρούσα διπλωματική. Αρχικά περιγράφονται τα υπολογιστικά νέφη, στη συνέχεια τα δεδομένα χρονοσειρών και τέλος οι τεχνολογίες μηχανικής μάθησης καθώς και οι μετρικές αξιολόγησής τους.

- Στο **Κεφάλαιο 3** αναλύονται τα δεδομένα που χρησιμοποιήθηκαν ως προς τη δομή και τα χαρακτηριστικά τους. Επίσης, περιγράφονται οι διαδικασίες που ακολουθήθηκαν για να δημιουργηθούν τα συνθετικά και τα πραγματικά δεδομένα χρονοσειρών.

- Στο **Κεφάλαιο 4** περιγράφεται η υλοποίηση και η αξιολόγηση των προβλεπτικών μοντέλων μηχανικής μάθησης

- Στο **Κεφάλαιο 5** μελετάται ένα παράδειγμα πολιτικής κλιμάκωσης των πόρων, με σκοπό να αποδειχθεί η σκοπιμότητα υλοποίησης ενός πραγματικού αντίστοιχου συστήματος

- Στο **Κεφάλαιο 6** παρουσιάζονται τα συμπεράσματα αυτής της διπλωματικής εργασίας, καθώς και μελλοντικές επεκτάσεις.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Στο κεφάλαιο αυτό παρουσιάζονται αναλυτικά οι βασικές τεχνολογίες που έχουν σχέση με την παρούσα διπλωματική. Πιο συγκεκριμένα, αναλύονται οι βασικές έννοιες για τα περιβάλλοντα υπολογιστικών νεφών, τα δεδομένα χρονοσειρών και τις μετρικές αξιολόγησης των τεχνικών πρόβλεψής τους. Επίσης, γίνεται εισαγωγή στις βασικές έννοιες της μηχανικής μάθησης και εμβάθυνση στις τεχνικές που χρησιμοποιήθηκαν.

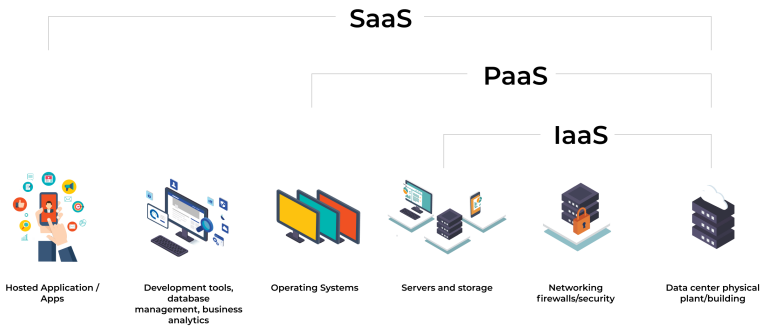
2.1 Υπολογιστικό Νέφος (Cloud Computing)

Υπολογιστικό Νέφος ονομάζεται η κατ' αίτηση διαδικτυακή κεντρική διάθεση υπολογιστικών πόρων (όπως δίκτυο, εξυπηρετητές, εφαρμογές και υπηρεσίες) με υψηλή ευελιξία, ελάχιστη προσπάθεια από τον χρήστη και υψηλή αυτοματοποίηση [11]. Το οικονομικό μοντέλο διάθεσης των πόρων όπως εμφανίζεται σήμερα λέγεται As A Service (XaaS) και χωρίζεται στις εξής τρεις μεγάλες κατηγορίες:

- **Software-as-a-Service (SaaS)**: είναι μια υπηρεσία υπολογιστικού νέφους που παρέχει στους χρήστες πρόσβαση σε λογισμικό που φιλοξενείται κεντρικά σε cloud ενός προμηθευτή. Αντί να προμηθεύονται και να εγκαθιστούν το λογισμικό στο δικό τους μηχάνημα, οι χρήστες έχουν απομακρυσμένη πρόσβαση σε αυτό μέσω του φυλλομετρητή τους. Παράδειγμα SaaS αποτελεί το Google Docs.

- **Platform-as-a-Service (PaaS)**: είναι μια υπηρεσία υπολογιστικού νέφους που παρέχει στους χρήστες ένα περιβάλλον cloud στο οποίο μπορούν να αναπτύξουν και να διαχειριστούν υπολογιστικές πλατφόρμες. Εκτός από την αποθήκευση και άλλους πόρους υπολογιστών, οι χρήστες μπορούν να χρησιμοποιήσουν μια σειρά από προεγκατεστημένα εργαλεία για να αναπτύξουν, να προσαρμόσουν και να δοκιμάσουν τις δικές τους εφαρμογές. Παράδειγμα PaaS αποτελεί το Google Colab.

- **Infrastructure-as-a-Service (IaaS)**: είναι μια υπηρεσία υπολογιστικού νέφους στην οποία ένας προμηθευτής παρέχει στους χρήστες πρόσβαση σε υπολογιστικούς πόρους όπως διακομιστές, αποθήκευση και δικτύωση. Τα μηχανήματα τα οποία παρέχει ο προμηθευτής συνήθως παρέχονται μέσω εικονοποίησης, δηλαδή δημιουργείται ένα εικονικό λογισμικό σύστημα - διεπαφή, ώστε το ¨κομμάτι¨ πόρων που χρησιμοποιεί ο χρήστης να φαίνεται αυτόνομο. Παράδειγμα IaaS αποτελεί το AWS.



Εικόνα 2.1: *Είδη Υπηρεσιών Cloud [1]*

Cloud Elasticity

Ελαστικότητα του νέφους (Cloud Elasticity) αποκαλείται η ιδιότητα ενός νέφους να μεγεθύνει ή να συρρικνώνει την χωρητικότητα επεξεργαστικής ισχύος, της μνήμης και του δίσκου, ώστε να προσαρμόζεται στις αλλαγές των απαιτήσεων. Μπορεί να είναι αυτόματη, χωρίς να χρειάζεται ένα πλάνο της χωρητικότητας εκ των προτέρων, ή μπορεί να είναι χειροκίνητη διαδικασία όταν το σύστημα ειδοποιείται πως στερεύει από ελεύθερους πόρους και αποφασίζεται εκ των υστέρων αν θα αυξηθεί ή θα μειωθεί η χωρητικότητα όταν χρειάζεται[12].

Ένα από τα βασικά οφέλη του Cloud Elasticity είναι η δυνατότητα προσαρμογής των ψηφιακών υπηρεσιών σε αλλαγές στις απαιτήσεις τους. Οι επιχειρήσεις μπορούν να αντιδρούν αμέσως σε διακυμάνσεις στη ζήτηση ή στο φόρτο εργασίας, αυξάνοντας ή μειώνοντας τους ψηφιακούς τους πόρους όπως CPU, μνήμη, αποθήκευση και δίκτυο. Αυτό οδηγεί στην αποτελεσματική χρήση των πόρων, αποφεύγοντας την υπερκατανάλωση και τη σπατάλη πόρων, με αποτέλεσμα τη μείωση των λειτουργικών εξόδων.

Επίσης, το Cloud Elasticity επιτρέπει στις επιχειρήσεις να αντιμετωπίσουν απρόβλεπτες καταστάσεις και αιχμές κίνησης. Κατά τη διάρκεια περιόδων αιχμής, οι επιχειρήσεις μπορούν να αυξήσουν δυναμικά τους πόρους τους για να αντιμετωπίσουν τον υψηλό φόρτο εργασίας, ενώ κατά τις περιόδους χαμηλής ζήτησης μπορούν να μειώσουν τους πόρους για εξοικονόμηση κόστους. Αυτό επιτρέπει την αποτελεσματική χρήση των υποδομών χωρίς την ανάγκη για υποδομές που θα παραμένουν ανενεργές κατά τη διάρκεια μεγάλων χρονικών περιόδων.

Επιπλέον, η χρήση του Cloud Elasticity δίνει τη δυνατότητα για τη δημιουργία ψηφιακών υπηρεσιών υψηλής διαθεσιμότητας και αξιοπιστίας. Οι πόροι μπορούν να διανέμονται σε διάφορες γεωγραφικές τοποθεσίες, και αν εντοπιστεί ένα πρόβλημα σε μια τοποθεσία, οι υπηρεσίες μπορούν να ανακατευθυνθούν αυτόματα σε άλλες λειτουργικές περιοχές χωρίς διακοπή της υπηρεσίας.

Η ελαστικότητα στοχεύει στο να ταιριάξει το ποσό των πόρων που διατίθενται σε μια υπηρεσία με το ποσό των πόρων που πραγματικά χρειάζεται, αποφεύγοντας την υπερβολική παροχή ή την υποεκτίμηση. Η υπερβολική παροχή (over-provisioning), δηλαδή η κατανομή περισσότερων πόρων από ό,τι απαιτείται, θα πρέπει να αποφεύγεται, καθώς ο πάροχος υπη-

ρεσιών πρέπει συχνά να πληρώνει για τους πόρους που διατίθενται στην υπηρεσία. Έτσι, τα έξοδα του παρόχου υπηρεσιών είναι υψηλότερα από το βέλτιστο και το κέρδος τους μειώνεται. Η ανεπαρκής παροχή (under-provisioning), δηλαδή η κατανομή λιγότερων πόρων από ό,τι απαιτείται, πρέπει να αποφευχθεί, διαφορετικά η υπηρεσία δεν μπορεί να εξυπηρετήσει τους χρήστες της με μια καλή υπηρεσία. Οι χρήστες του Διαδικτύου σταματούν τελικά να έχουν πρόσβαση στην υπηρεσία, κι έτσι ο πάροχος χάνει πελάτες. Μακροπρόθεσμα, το εισόδημα του παρόχου θα μειωθεί.

Αυτόματη Κλιμάκωση (Auto-Scaling)

Η αυτόματη κλιμάκωση [13] αποτελεί κρίσιμο στοιχείο στη διαχείριση των υπολογιστικών υποδομών, ειδικά στον κόσμο του cloud computing. Οι δύο βασικές κατηγορίες αυτόματης κλιμάκωσης είναι οι προληπτικοί (proactive) και οι αντιδραστικοί (reactive) μηχανισμοί, και κάθε ένας από αυτούς έχει τα δικά του χαρακτηριστικά και οφέλη [14].

- Οι **proactive** μηχανισμοί λειτουργούν προληπτικά, προβλέποντας τις μελλοντικές ανάγκες για πόρους και προσαρμόζοντας τη χωρητικότητα πριν από την εμφάνιση των αιτημάτων. Αυτό έχει ως αποτέλεσμα τη βελτιστοποίηση της απόδοσης και την αποτροπή των διακοπών λειτουργίας λόγω υπερφόρτωσης. Παράλληλα, επιτυγχάνουν εξοικονόμηση κόστους με τη βελτιστοποίηση της χρήσης των πόρων. Ωστόσο, η υλοποίησή τους μπορεί να είναι πολύπλοκη, και απαιτεί ανάλυση και πρόβλεψη των αναγκών πόρων, με τη χρήση σύνθετων αλγορίθμων και προγνωστικών μοντέλων. Υπάρχει, παράλληλα, το ρίσκο της λανθασμένης πρόβλεψης. Σε αυτή την περίπτωση, το σύστημα θα οδηγηθεί σε υπερκλιμάκωση είτε σε ανεπαρκή παροχή υπολογιστικών πόρων.

- Οι **reactive** μηχανισμοί αντιδρούν αμέσως μετά τις αυξημένες ανάγκες για πόρους, βασιζόμενοι σε συγκεκριμένα κριτήρια. Αυτοί οι μηχανισμοί είναι απλοί στην υλοποίηση και μπορούν να αντιμετωπίσουν απρόβλεπτες αυξήσεις της κίνησης. Ωστόσο, υπάρχει μια καθυστέρηση μεταξύ της ανάγκης για πόρους και της παροχής τους. Το γεγονός αυτό εμπεριέχει μεγάλο κόστος, αφού, την στιγμή που θα επιβεβαιωθεί η ανάγκη δέσμευσης παραπάνω πόρων, το σύστημα ήδη τελεί υπό πίεση.

Εξισορρόπηση Φορτίου (Load Balancing)

Η εξισορρόπηση φορτίου [14] αποτελεί κρίσιμο στοιχείο στη διαχείριση των υπολογιστικών υποδομών, ειδικά στον κόσμο του cloud computing. Ο βασικός στόχος της εξισορρόπησης φορτίου είναι να διασφαλίσει ότι οι πόροι του συστήματος χρησιμοποιούνται αποτελεσματικά και ισότιμα, εξαλείφοντας τυχόν ανισορροπίες στον φόρτο εργασίας μεταξύ των διακομιστών ή των πόρων.

Η ανάγκη για εξισορρόπηση φορτίου προκύπτει λόγω πολλών παραγόντων. Καταρχάς, ο φόρτος εργασίας δεν είναι πάντα ομοιογενής, και ορισμένοι διακομιστές μπορεί να υπερφορτώνονται ενώ άλλοι παραμένουν ανεκμετάλλευτοι. Επιπλέον, κατά τη διάρκεια κατακόρυφων καμπυλών φόρτου, είναι αναγκαίο να προσαρμοστούν οι πόροι για να ανταποκριθούν στις αυξημένες απαιτήσεις. Τέλος, διάφορες εφαρμογές μπορεί να απαιτούν διαφορετικούς

πόρους ανάλογα με τον τύπο της εργασίας. Για παράδειγμα, μια εφαρμογή βάσης δεδομένων μπορεί να χρειάζεται περισσότερη μνήμη, ενώ μια εφαρμογή ανάλυσης δεδομένων μπορεί να χρειάζεται περισσότερο επεξεργαστικό χρόνο.

Για την επίτευξη αυτών των στόχων, υπάρχουν διάφοροι αλγόριθμοι εξισορρόπησης φορτίου. Ορισμένοι από αυτούς περιλαμβάνουν την μέθοδο Round Robin, όπου ο διακομιστής παρέχει τις υπηρεσίες του σε σειρά στους πελάτες, ανεξαρτήτως του φόρτου, τον αλγόριθμο Least Connections που επιλέγει τον επόμενο πελάτη με βάση τον αριθμό των τρεχουσών συνδέσεων στον κάθε διακομιστή και τον αλγόριθμο Weighted Round Robin, όπου οι διακομιστές αξιολογούνται με βάση την απόδοσή τους και ο διακομιστής με τη υψηλότερη αξιολόγηση λαμβάνει περισσότερο φόρτο.

Η εξισορρόπηση φορτίου προσφέρει ουσιαστικά οφέλη, όπως τη βελτιωμένη απόδοση του συστήματος, τη βελτίωση της ανθεκτικότητας και τη μείωση της απόσυρσης υπολογιστικών πόρων. Ωστόσο, απαιτεί προσεκτική διαμόρφωση και εφαρμογή, καθώς και παρακολούθηση της απόδοσης για να προσαρμόζεται σε μεταβαλλόμενες απαιτήσεις.

2.2 Χρονοσειρές

2.2.1 Εισαγωγή στις Χρονοσειρές

Οι χρονοσειρές αποτελούν σημαντικό πεδίο στα στατιστικά και υπολογιστικά επιστημονικά πεδία και έχουν εφαρμογές σε πολλούς τομείς, από την οικονομία μέχρι την πρόβλεψη του καιρού και την αναγνώριση προτύπων. Μια χρονοσειρά είναι ένα σύνολο δεδομένων που συλλέγονται ή καταγράφονται σε χρονική σειρά, όπου κάθε παρατήρηση σχετίζεται με ένα συγκεκριμένο χρονικό σημείο.

Τα δεδομένα αυτά μπορεί να είναι ακολουθίες μετρήσεων, όπως η ημερήσια θερμοκρασία, η τιμή των μετοχών στο χρηματιστήριο, η παραγωγή ενέργειας από ανανεώσιμες πηγές κλπ. Κάθε παρατήρηση αποτελεί ένα σημείο δεδομένων που συνήθως συνοδεύεται από ένα χρονικό σήμα, όπως η ημερομηνία και η ώρα, που καταγράφει πότε πραγματοποιήθηκε η παρατήρηση.

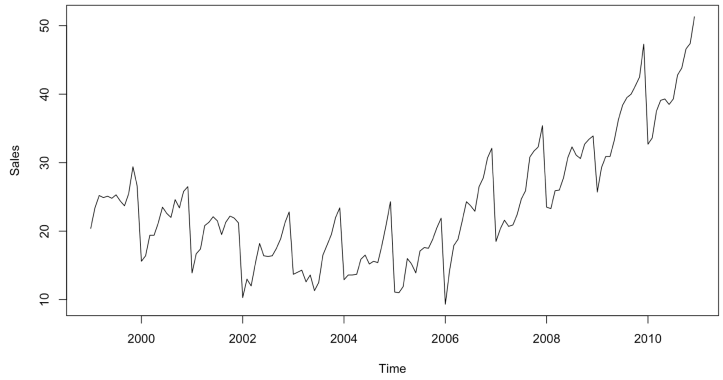
Στην ανάλυση χρονοσειρών, υπάρχει η έννοια της αποσύνθεσης. Η αποσύνθεση αναφέρεται σε μια αναλυτική διαδικασία όπου μια πολύπλοκη χρονοσειρά διαχωρίζεται σε πιο απλές συνιστώσες, γνωστές και ως υποσειρές, με σκοπό την καλύτερη κατανόηση και ανάλυση της συμπεριφοράς της. Κατά την αποσύνθεση, προσπαθούμε να αναλύσουμε τη χρονοσειρά σε τέσσερα βασικά χαρακτηριστικά. Κάθε χαρακτηριστικό εξυπηρετεί έναν συγκεκριμένο σκοπό στην ανάλυση και τον προσδιορισμό της συμπεριφοράς των χρονοσειρών. Η κατανόηση αυτών των χαρακτηριστικών είναι ουσιώδης για την εξαγωγή εργαλείων και πληροφοριών από τις χρονοσειρές [15].

- **Τάση - Trend (T)**: Το χαρακτηριστικό αυτό αναφέρεται στην μακροπρόθεσμη τάση αύξησης ή μείωσης των παρατηρήσεων στη χρονοσειρά. Η ανίχνευση της τάσης επιτρέπει τον προσδιορισμό της γενικής κατεύθυνσης των δεδομένων στο χρόνο.

- **Κυκλικότητα - Cyclicity (C)**: Το χαρακτηριστικό αυτό αναφέρεται στην ύπαρξη κυκλικών μοτίβων ή περιοδικών διακυμάνσεων στη χρονοσειρά. Επιτρέπει τον εντοπισμό

περιοδικών τάσεων που επαναλαμβάνονται σε σταθερά χρονικά διαστήματα.

- **Εποχικότητα - Seasonality (S)**: Το χαρακτηριστικό αυτό αναφέρεται στις περιοδικές αλλαγές που επαναλαμβάνονται σε συγκεκριμένα χρονικά διαστήματα, όπως η εποχικότητα. Επιτρέπει τον χειρισμό εποχικών μοτίβων και αναλύσεων.
- **Τυχαιότητα - Randomality (R)**: Το χαρακτηριστικό αυτό αναφέρεται στον τυχαίο και ασταθή χαρακτήρα των παρατηρήσεων στη χρονοσειρά. Χρησιμοποιείται για την περιγραφή των τυχαίων διακυμάνσεων και την εκτίμηση της αβεβαιότητας στα δεδομένα.



Εικόνα 2.2: *Παράδειγμα Χρονοσειράς*

Στασιμότητα

Η στασιμότητα των χρονοσειρών είναι ένα κρίσιμο στατιστικό χαρακτηριστικό που επηρεάζει σημαντικά την ανάλυση και τη μοντελοποίησή τους. Μια στάσιμη χρονοσειρά έχει σταθερά χαρακτηριστικά, όπως ο μέσος όρος, η διακύμανση και η αυτοσυσχέτιση, και αυτό επιτρέπει την εφαρμογή πιο απλών στατιστικών μεθόδων για την ανάλυση και την πρόβλεψη. Είναι προφανές πως μια χρονοσειρά που χαρακτηρίζεται από τάση, εποχικότητα ή/και κυκλικότητα δεν είναι στάσιμη. Αντίθετα, μια χρονοσειρά που περιέχει μόνο θόρυβο θεωρείται στάσιμη, αφού δεν σχετίζεται με τον χρόνο.

2.2.2 Τεχνικές Προβλέψεων Χρονοσειρών

Η μεγαλύτερη πρόκληση στην ανάλυση χρονοσειρών είναι η πρόβλεψη, δηλαδή πώς η ακολουθία των παρατηρήσεων θα συνεχιστεί στο μέλλον. Το ζητούμενο είναι να ακολουθεί μια διαδικασία που θα εξασφαλίσει ότι θα παραχθούν όσο τον δυνατόν πιο ακριβείς προβλέψεις, αξιοποιώντας στο έπακρο όλη την διαθέσιμη ιστορική πληροφορία. Οι τεχνικές πρόβλεψης χρονοσειρών περιλαμβάνουν διάφορες μεθόδους και μοντέλα που χρησιμοποιούνται για την πρόβλεψη μελλοντικών τιμών σε μια χρονοσειρά. Οι μέθοδοι αυτές χωρίζονται σε στατιστικές και κριτικές, ενώ τα τελευταία χρόνια χρησιμοποιούνται και μέθοδοι μηχανικής μάθησης[16].

Στατιστικές Μέθοδοι

Εκθετική Εξομάλυνση: Μέθοδος πρόβλεψης η οποία εξομαλύνει τα ιστορικά δεδομένα. Υπολογίζεται ο μέσος όρος των δεδομένων, με την χρήση συντελεστών βαρύτητας με τα πιο πρόσφατα δεδομένα να έχουν μεγαλύτερη βαρύτητα. Οι συντελεστές βαρύτητας μειώνονται με εκθετικό τρόπο, όσο παλαιότερα είναι τα δεδομένα. Στόχο της μεθόδου αποτελεί η απομόνωση του προτύπου των δεδομένων από τις τυχαίες διακυμάνσεις. Χρησιμοποιείται ευρέως για βραχυπρόθεσμο σχεδιασμό, καθώς είναι σχετικά εύκολη στην χρήση και απαιτεί ελάχιστα ιστορικά δεδομένα και χρόνο υπολογισμού. Χωρίζεται σε τέσσερα βασικά μοντέλα:

- Σταθερού Επιπέδου, για πρόβλεψη ενός βήματος. Χρησιμοποιείται σε χρονοσειρές που περιέχουν υψηλό θόρυβο ή τυχαιότητα.
- Γραμμικής τάσης: Για σταθερή αύξηση στο μέλλον.
- Εκθετικής τάσης: Για εκθετική αύξηση στο μέλλον (π.χ. στις αρχές του κύκλου ζωής ενός προϊόντος). Συνήθως χρησιμοποιούνται για βραχυπρόθεσμες προβλέψεις, αφού είναι υπεραισιόδοξες για μακροπρόθεσμες.
- Φθίνουσας τάσης: Για μεσοπρόθεσμες προβλέψεις.

Μέθοδος Θ: Η μέθοδος Θ είναι μια μονοδιάστατη μέθοδος πρόβλεψης. Βασίζεται στην μεταβολή των τοπικών καμπυλοτήτων μιας χρονοσειράς μέσα από την παράμετρο θ (Theta), η οποία εφαρμόζεται απευθείας (πολλαπλασιαστικά) στις διαφορές δεύτερης τάξης των δεδομένων. Η καινούργια χρονοσειρά που δημιουργείται διατηρεί την μέση τιμή και κλίση της αρχικής χρονοσειράς αλλά όχι και τις τοπικές καμπυλότητες. Οι χρονοσειρές που παράγονται με αυτή την διαδικασία ονομάζονται γραμμές Θ (Theta Lines). Βασικό ποιοτικό χαρακτηριστικό αυτών των γραμμών είναι η καλύτερη προσέγγιση της μακροπρόθεσμης συμπεριφοράς–τάσης των δεδομένων ή ανάδειξη-τονισμός των βραχυπρόθεσμων χαρακτηριστικών, ανάλογα με την τιμή της παραμέτρου θ (<,>1).

Box-Jenkins Μοντέλα - ARIMA: Ολοκληρωμένα αυτοπαλινδρομικά μοντέλα κινητών μέσων όρων (Auto Regressive Integrated Moving Average). Ανήκουν στα στοχαστικά μοντέλα πρόβλεψης και μελετήθηκαν από τους Box & Jenkins (1971) και συχνά συναντώνται στη βιβλιογραφία με την αντίστοιχη ονομασία. Προσεγγίζουν τη λογική των κλασικών μοντέλων παλινδρόμησης και εκθετικής εξομάλυνσης με την έννοια ότι συσχετίζουν τις μελλοντικές τιμές τις χρονοσειράς με παρελθοντικές της ή/και σφάλματα που εντοπίστηκαν. Η ιδιομορφία τους έγκειται στο ότι η γραμμική συσχέτιση γίνεται χωρίς την άμεση χρήση εξομάλυνσης ή την αξιοποίηση ερμηνευτικών μεταβλητών. Πιο συγκεκριμένα, κάθε μοντέλο ARIMA μπορεί να εκφραστεί ως γραμμικός συνδυασμός των παραπάνω παραγόντων και στόχος είναι να ανακαλυφθεί εκείνος που παράγει τις καλύτερες προβλέψεις. Έτσι, αν το μοντέλο περιλαμβάνει αποκλειστικά παράγοντες αυτοπαλινδρόμησης αναφέρεται ως AR(p), αν περιλαμβάνει αποκλειστικά παράγοντες κινητών μέσω όρων ως MA(q), και αν περιλαμβάνει και τους δύο ως ARMA(p,q), όπου τα p και q δηλώνουν την τάξη του μοντέλου ανά παράγοντα. Ο παράγοντας I(d) αναφέρεται στη διαφόριση της χρονοσειράς πριν την εφαρμογή ενός μοντέλου ARMA(p,q) και έχει ως στόχο την αφαίρεση της τάσης από τα δεδομένα.

2.2.3 Μετρικές Αξιολόγησης Προβλέψεων

Η αξιολόγηση των προβλέψεων αποτελεί κρίσιμο κομμάτι της διαδικασίας πρόβλεψης στον τομέα της ανάλυσης χρονοσειρών και των προβλέψεων. Ενώ η δημιουργία προβλέψεων αποτελεί σημαντικό βήμα, η αξιολόγησή τους αποκτά εξίσου μεγάλη σημασία, καθώς αποτελεί τον τρόπο με τον οποίο μπορούμε να εκτιμήσουμε την ακρίβεια και την απόδοση των μοντέλων πρόβλεψης. Οι μετρικές αξιολόγησης μπορούν να χωριστούν συνολικά σε δύο βασικές κατηγορίες: μετρικές ακρίβειας και μετρικές απόδοσης. Κάθε κατηγορία παρέχει διαφορετική πληροφορία σχετικά με την απόδοση των προβλέψεων.

Οι **μετρικές ακρίβειας** αντιπροσωπεύουν πόσο καλά το μοντέλο πρόβλεψης εκτιμά τις πραγματικές τιμές στο μέλλον. Αυτές οι μετρικές εστιάζουν στην ποσοτική ακρίβεια των προβλέψεων σε σχέση με τις πραγματικές τιμές και μπορούν να παρέχουν σημαντική πληροφορία για την απόδοση του μοντέλου.

Οι **μετρικές ποιοτικής απόδοσης** αποτελούν μια κατηγορία μετρικών που χρησιμοποιούνται για να αξιολογηθεί η ποιότητα των προβλέψεων και η ικανότητα ενός μοντέλου να ανταποκριθεί σε συγκεκριμένες απαιτήσεις. Αυτές οι μετρικές προσπαθούν να αξιολογήσουν το πόσο καλά η πρόβλεψη πληροί συγκεκριμένα κριτήρια ποιότητας και είναι εξαιρετικά χρήσιμες για προβλέψεις σε περιβάλλοντα όπου η ακρίβεια δεν είναι η μοναδική σημαντική πτυχή.

Μέσο Απόλυτο Σφάλμα (Mean Absolute Error - MAE)

Το Μέσο Απόλυτο Σφάλμα (Mean Absolute Error - MAE) είναι μια μετρική ποσοτικής απόδοσης που χρησιμοποιείται για να μετρήσει το μέγεθος του σφάλματος μεταξύ των πραγματικών και των προβλεπόμενων τιμών σε ένα μοντέλο πρόβλεψης. Αυτή η μετρική παρέχει μια εικόνα του μέσου απόλυτου σφάλματος ανάμεσα στις προβλεπόμενες τιμές και τις πραγματικές τιμές, χωρίς να λαμβάνει υπόψη τον προσανατολισμό του σφάλματος. Το MAE είναι ανθεκτικό σε ακραίες τιμές (outliers) και μπορεί να χρησιμοποιηθεί σε περιπτώσεις όπου το σφάλμα πρέπει να μετρηθεί με απόλυτη ακρίβεια.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

όπου n είναι ο συνολικός αριθμός των παρατηρήσεων ή παραδειγμάτων, y_i είναι η πραγματική τιμή για την παρατήρηση i και $\hat{y}_i$ είναι η προβλεπόμενη τιμή για την παρατήρηση i.

Απόλυτο Ποσοστιαίο Σφάλμα (Mean Absolute Percentage Error - MAPE)

Το Απόλυτο Ποσοστιαίο Σφάλμα (Mean Absolute Percentage Error - MAPE) χρησιμοποιείται για να μετρήσει το μέγεθος του σφάλματος σε ποσοστιαία βάση. Το MAPE μετρά το μέσο απόλυτο ποσοστό σφάλματος, δηλαδή το ποσοστό της απόλυτης διαφοράς μεταξύ των πραγματικών και προβλεπόμενων τιμών σε σχέση με τις πραγματικές τιμές. Αυτή η μετρική είναι χρήσιμη για την κατανόηση του μεγέθους των σφαλμάτων σε ποσοστιαία βάση και μπορεί να βοηθήσει στη σύγκριση της απόδοσης του μοντέλου σε διάφορα σενάρια πρόβλεψης.

Το πρόβλημα της μετρικής αυτής, λόγω της διαίρεσης, είναι πως δεν μπορεί να χρησιμοποιηθεί για χρονοσειρές με μηδενικές τιμές, όπως αυτές που θα εξετασθούν στην παρούσα διπλωματική.

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

όπου n είναι ο συνολικός αριθμός των παρατηρήσεων ή παραδειγμάτων, y_i είναι η πραγματική τιμή για την παρατήρηση i και $\hat{y}_i$ είναι η προβλεπόμενη τιμή για την παρατήρηση i.

Τετραγωνικό Ριζικό Σφάλμα (Root Mean Squared Error - RMSE)

Το Τετραγωνικό Ριζικό Σφάλμα (Root Mean Squared Error - RMSE) μετρά το τετραγωνικό μέσο των σφαλμάτων πρόβλεψης και προσφέρει έναν τρόπο να αξιολογηθεί η ακρίβεια του μοντέλου πρόβλεψης. Όσο χαμηλότερο είναι το RMSE, τόσο καλύτερη η ακρίβεια του μοντέλου. Το προβάδισμά του σε σχέση με τα απλά απόλυτα σφάλματα συναντάται στον υπολογισμό ακρίβειας προβλέψεων μεγάλων τιμών, λόγω του τετραγωνισμού του σφάλματος.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

όπου n είναι ο συνολικός αριθμός των παρατηρήσεων ή παραδειγμάτων, y_i είναι η πραγματική τιμή για την παρατήρηση i και $\hat{y}_i$ είναι η προβλεπόμενη τιμή για την παρατήρηση i.

Προσαρμοσμένο Τετραγωνικό Ριζικό Σφάλμα (Percentage RMSE - PRMSE)

Το Προσαρμοσμένο Τετραγωνικό Ριζικό Σφάλμα (PRMSE) είναι μια παραλλαγή του RMSE που εκφράζει το RMSE ως ποσοστό του εύρους, δηλαδή την διαφορά της μέγιστης και της ελάχιστης τιμής των πραγματικών τιμών ($\bar{y}$). Ο προσαρμοσμένος αυτός τύπος επιτρέπει στον αναλυτή να κατανοήσει το μέγεθος του RMSE σε σχέση με τις τιμές που παρατηρούνται στη χρονοσειρά.

$$PRMSE = \frac{RMSE}{range_y} \times 100\%$$

2.3 Μηχανική Μάθηση & Νευρωνικά Δίκτυα

Μηχανική μάθηση είναι κλάδος της επιστήμης των υπολογιστών που αναπτύχθηκε από τη μελέτη της αναγνώρισης προτύπων και της τεχνητής νοημοσύνης. Ο όρος μηχανική μάθηση διατυπώθηκε από τον Arthur Samuel το 1959, επιστήμονα και μηχανικό της IBM, ο οποίος σχεδίασε πρόγραμμα για το γνωστό επιτραπέζιο παιχνίδι checkers με την ικανότητα να διακρίνει τη βέλτιστη στρατηγική. Ο μαθηματικός ορισμός παρουσιάστηκε από τον Tom Mitchell το 1997 ως «Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P, αν η απόδοσή του σε εργασίες από το T, όπως μετριέται από το P, βελτιώνεται μέσω της εμπειρίας E.» [17] [18].

Η μηχανική μάθηση διερευνά τη μελέτη και την κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά. Τέτοιοι αλγόριθμοι λειτουργούν κατασκευάζοντας μοντέλα από πειραματικά δεδομένα, προκειμένου να κάνουν προβλέψεις βασιζόμενες στα δεδομένα ή να εξάγουν αποφάσεις που εκφράζονται ως το αποτέλεσμα. Τα τρία βασικά είδη Μηχανικής Μάθησης είναι:

- **Επιβλεπόμενη Μάθηση (Supervised Learning)**: Η εκπαίδευση γίνεται μέσω ενός συνόλου από παραδείγματα όπου γνωρίζουμε για κάθε είσοδο την επιθυμητή έξοδο (labels) κατασκευάζοντας μία συνάρτηση που τα συσχετίζει. Στόχος είναι η γενίκευση της συνάρτησης αυτής για εισόδους με άγνωστη έξοδο. Τα προβλήματα που μελετούνται με αυτό το είδος είναι προβλήματα ταξινόμησης (Classification) και παλινδρόμησης (Regression).

- **Μη επιβλεπόμενη Μάθηση (Unsupervised Learning)**: Αντίθετα με την επιβλεπόμενη μάθηση, εδώ η εκπαίδευση γίνεται με unlabeled δεδομένα με σκοπό να βρεθούν κοινά χαρακτηριστικά μεταξύ τους. Τα κύρια προβλήματα που μελετώνται με μη επιβλεπόμενη μάθηση είναι η συσταδοποίηση. (Clustering) και η μείωση διαστατικότητας (Dimensionality Reduction).

- **Ενισχυτική Μάθηση (Reinforcement Learning)**: Η λειτουργία της διαφέρει από τις προηγούμενες και θεωρείται ότι είναι αυτή που προσεγγίζει περισσότερο την ανθρώπινη διαδικασία μάθησης. Η μάθηση βασίζεται στην αλληλεπίδραση του υποκειμένου με το περιβάλλον του αναπτύσσοντας εμπειρία των κινήσεων του. Αναπτύσσει στρατηγικές ώστε να καταφέρει να επιτύχει τους στόχους του μέσω εμπειρίας από τα λάθη του έχοντας ως στόχο την μεγιστοποίηση του βραβείου του (reward). Χρησιμοποιείται κυρίως σε προβλήματα σχεδιασμού όπως ο έλεγχος κίνησης ρομπότ καθώς και παιχνίδια στρατηγικής.

2.3.1 Γραμμική Παλινδρόμηση

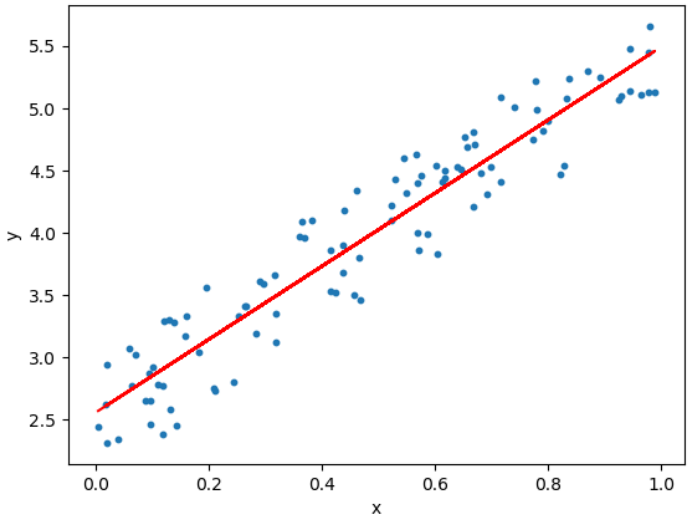
Η γραμμική παλινδρόμηση (linear regression) είναι μία προσέγγιση παλινδρόμησης που μοντελοποιεί τη σχέση μεταξύ μίας εξαρτημένης μεταβλητής y και μίας ή περισσοτέρων ανεξαρτήτων μεταβλητών x. Στο μοντέλο αυτό τα δεδομένα εισόδου, δηλαδή οι ανεξάρτητες μεταβλητές, μοντελοποιούνται χρησιμοποιώντας γραμμικές λειτουργίες, ενώ μέσω αυτών υπολογίζονται οι επιθυμητές τιμές εξόδου, δηλαδή οι άγνωστες παράμετροι. Τέτοιου είδους

μοντέλα καλούνται γραμμικά μοντέλα και συνεπώς η εξαρτημένη μεταβλητή y αποτελεί έναν γραμμικό συνδυασμό των ανεξάρτητων μεταβλητών x [19]. Το μοντέλο της πολλαπλής γραμμικής παλινδρόμησης έχει την παρακάτω μορφή:

$$y = \sum_{i=1}^{n} w_{ij} \cdot x_i + b_j$$

Και βελτιστοποιεί συνάρτηση κόστους ελαχιστοποιώντας το σύνολο των αποστάσεων των σημείων y και $\hat{y}$, βρίσκοντας έτσι την ευθεία που περνά βέλτιστα από τα σημεία:

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



Εικόνα 2.3: *Γραμμική Παλινδρόμηση [2]*

Παλινδρόμηση Lasso

Η παλινδρόμηση Lasso (Least Absolute Shrinkage and Selection Operator Regression) είναι μια μέθοδος για την αντιμετώπιση του προβλήματος της υπερπροσαρμογής (overfitting) και της επιλογής των σημαντικότερων χαρακτηριστικών σε ένα μοντέλο παλινδρόμησης. Στόχος είναι να επιλεγούν μερικές από τις ανεξάρτητες μεταβλητές που συνδέονται με την εξαρτημένη μεταβλητή και να αφαιρεθούν οι υπόλοιπες. Η παλινδρόμηση Lasso επιτυγχάνει αυτόν τον στόχο εισάγοντας έναν όρο κανονικοποίησης L1, που προσθέτει ένα στοιχείο κλίσης (συντελεστή) στην εξίσωση της γραμμικής παλινδρόμησης [20].

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + a \sum_{j=1}^{p} |w_j| \right\}$$

Όπου p είναι ο αριθμός των ανεξάρτητων μεταβλητών και a είναι ο όρος κανονικοποίησης L1 που ελέγχει το βαθμό στον οποίο οι συντελεστές w_j συμπιέζονται προς το μηδέν.

Η παλινδρόμηση Lasso είναι χρήσιμη για την επιλογή και την εκτίμηση σημαντικών μεταβλητών και μπορεί να οδηγήσει σε απλούστερα και ερμηνεύσιμα μοντέλα. Ωστόσο, είναι ευαίσθητη στην επιλογή της παραμέτρου κανονικοποίησης a, και μπορεί να οδηγήσει στον αποκλεισμό σημαντικών μεταβλητών εάν είναι πολύ μεγάλο.

Παλινδρόμηση Ridge

Η παλινδρόμηση Ridge είναι μια μέθοδος για την αντιμετώπιση του προβλήματος της πολυγραμμικότητας (multicollinearity) στη γραμμική παλινδρόμηση, δηλαδή την ύπαρξη συσχέτισης μεταξύ των χαρακτηριστικών x. Παρόμοια με τη Lasso, η Ridge προσθέτει έναν όρο κανονικοποίησης στην εξίσωση της γραμμικής παλινδρόμησης για να αποτρέψει την υπερεκτίμηση των συντελεστών του μοντέλου. Στην περίπτωση της Ridge, ο όρος κανονικοποίησης είναι ο όρος L2, που προσθέτει το άθροισμα των τετραγώνων των συντελεστών στη συνάρτηση κόστους [20].

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + a \sum_{j=1}^{p} w_j^2 \right\}$$

Όπου a είναι ο όρος κανονικοποίησης L2 που ελέγχει το βαθμό που οι συντελεστές β_j συμπιέζονται προς το μηδέν.

Παλινδρόμηση ElasticNet

Η παλινδρόμηση ElasticNet είναι μια επέκταση της γραμμικής παλινδρόμησης που συνδυάζει τα χαρακτηριστικά της Lasso και της Ridge. Αυτό γίνεται με την προσθήκη των όρων κανονικοποίησης L1 και L2 στην εξίσωση της παλινδρόμησης. Αυτό επιτρέπει την επιλογή χαρακτηριστικών (Lasso) και ταυτόχρονα εξασφαλίζει τη σταθερότητα των συντελεστών (Ridge).

$$\min_{\beta} \left\{ \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + a_1 \sum_{j=1}^{p} |w_j| + a_2 \sum_{j=1}^{p} w_j^2 \right\}$$

Όπου a_1 και a_2 είναι οι παράμετροι κανονικοποίησης L1 και L2 που ελέγχουν το βαθμό της κανονικοποίησης Lasso και Ridge αντίστοιχα.

Παλινδρόμηση Power

Η παλινδρόμηση Power είναι μια μέθοδος παλινδρόμησης που χρησιμοποιείται όταν η σχέση μεταξύ των ανεξάρτητων και εξαρτημένων μεταβλητών δεν είναι γραμμική. Σε πολλές περιπτώσεις, οι φυσικές διαδικασίες και οι σχέσεις μεταξύ μεταβλητών δεν ακολουθούν την κλασική γραμμική σχέση και, συνεπώς, απαιτείται μη γραμμική προσέγγιση.

$$y = b + w_1 x_1^{p_1} + w_2 x_2^{p_2} + \ldots + w_n x_n^{p_n} + \varepsilon$$

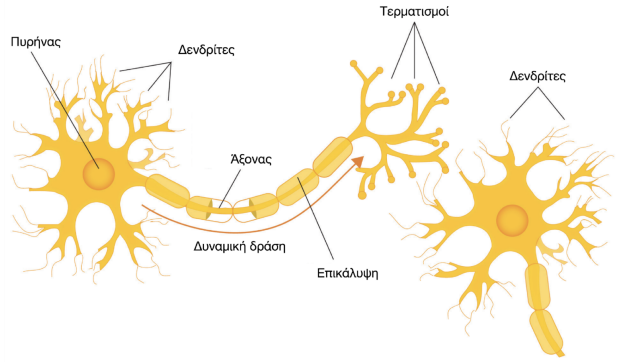
Όπου $p_1, p_2, \ldots, p_n$ είναι οι παράμετροι δύναμης που καθορίζουν τον βαθμό της μη γραμμικότητας για κάθε ανεξάρτητη μεταβλητή και ε είναι το τυχαίο σφάλμα που αντιπροσωπεύει τη διακύμανση που δεν μπορεί να εξηγηθεί από τις ανεξάρτητες μεταβλητές.

Οι παράμετροι $p_1, p_2, \ldots, p_n$ καθορίζουν τον βαθμό της μη γραμμικότητας για κάθε ανεξάρτητη μεταβλητή. Όταν $p_i = 1$ για όλες τις μεταβλητές, η παλινδρόμηση Power μειώνεται στην κλασική γραμμική παλινδρόμηση.

2.3.2 Νευρώνας (Perceptron)

Βιολογικός Νευρώνας

Οι τεχνικές και οι αλγόριθμοι μηχανικής μάθησης βασίζονται σε ανθρώπινες λειτουργίες. Η σημασία του νευρωνικού δικτύου στην βιολογία είναι το ανθρώπινο νευρικό σύστημα, το απλούστερο μέρος του οποίου είναι ο νευρώνας. Ο ρόλος του νευρώνα σε ένα βιολογικό νευρωνικό δίκτυο είναι να λαμβάνει όλα τα σήματα που έρχονται από άλλους νευρώνες, να τα επεξεργάζεται με κατάλληλο τρόπο, και να μεταδίδει περαιτέρω το επεξεργασμένο σήμα σε άλλους νευρώνες, ούτως ώστε ένα σήμα να διαδίδεται μέσω ενός τεραστίου αριθμού νευρώνων. Οι συνδέσεις μεταξύ των νευρώνων, με τους άξονες και τους δενδρίτες, γίνονται στις επαφές που ονομάζονται συνάψεις. Η σύναψη έχει πολύ περίπλοκη δομή και επιτελεί επίσης περίπλοκες διεργασίες κατά την μετάδοση του σήματος. Καθ' όλη την διάρκεια της ζωής ενός οργανισμού οι συνάψεις βρίσκονται σε μία δυναμική ισορροπία, δημιουργούνται καινούργιες και καταστρέφονται παλιές. Η δημιουργία των νέων συνάψεων γίνεται όταν ο εγκέφαλος αποκτά περισσότερες εμπειρίες από το περιβάλλον, μαθαίνει, αναγνωρίζει, κατανοεί, κλπ [21].



Σχήμα 2.1: *Βιολογικός Νευρώνας*

Τεχνητός Νευρώνας

Τα βασικά ευρήματα από τη λειτουργία των βιολογικών νευρώνων επέτρεψαν σε πρώιμους ερευνητές να μοντελοποιήσουν τη λειτουργία απλών τεχνητών νευρώνων. Ένας τεχνητός νευρώνας επεξεργασίας λαμβάνει εισροές ως ερεθίσματα από το περιβάλλον, τους συνδυάζει με έναν ειδικό τρόπο για να σχηματίσει μια «καθαρή» είσοδο, την οποία περνάει από μια γραμμική ή μη γραμμική πύλη συνάρτησης και μεταδίδει το σήμα εξόδουν προς τα εμπρός σε ένα άλλο νευρώνα ή στο περιβάλλον. Ο τεχνητός νευρώνας αποτελεί την βάση για την σχεδίαση μιας μεγάλης οικογένειας νευρωνικών δικτύων και δομείται με τέσσερα βασικά στοιχεία:

- **Συνάψεις**, κάθε μία εκ των οποίων χαρακτηρίζεται από το δικό της βάρος ή δύναμη που συμβολίζεται συνήθως με το γράμμα w. Ανόμοια με το βάρος μίας σύναψης στον ανθρώπινο εγκέφαλο, το βάρος ενός τεχνητού νευρώνα μπορεί να λαμβάνει τόσο αρνητικές όσο και θετικές τιμές.

- Εναν **αθροιστή** για την άθροιση των σημάτων εισόδου, σταθμισμένων από τα αντίστοιχα συναπτικά βάρη του νευρώνα.

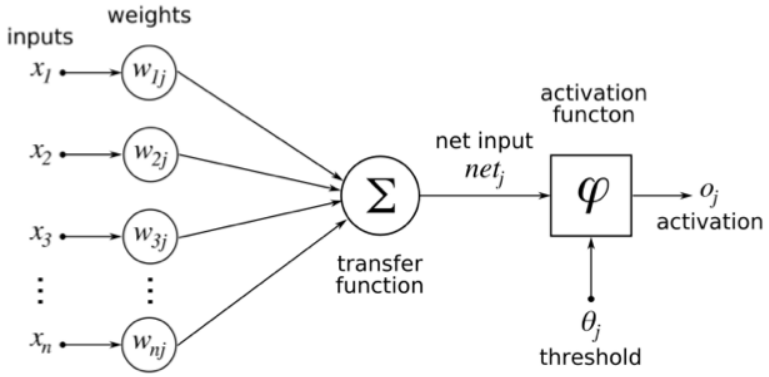
- Μια **συνάρτηση ενεργοποίησης** φ (activation function) για τον περιορισμό του πλάτους του συστήματος εξόδου ενός νευρώνα.

- μια εξωτερικά εφαρμοζόμενη **πόλωση** b_k ή ∂_j, η οποία προκαλεί θετική ή αρνητική προκατάληψη (bias) στο αποτέλεσμα της συνάρτησης ενεργοποίησης.

Η αναλογία μεταξύ του τεχνητού νευρώνα και του βιολογικού νευρώνα είναι ότι οι συνδέσεις μεταξύ των κόμβων αντιπροσωπεύουν τους άξονες και τους δενδρίτες, τα βάρη σύνδεσης αντιπροσωπεύουν τις συνάψεις και το κατώφλι προσεγγίζει τη δραστηριότητα στο σώμα. Η λειτουργία του νευρώνα δίνεται από τον τύπο:

$$y = \varphi(\sum_{i=1}^{n} w_{ij} \cdot x_i + b_j)$$

Εύκολα κανείς παρατηρεί πως η συνάρτηση της γραμμικής παλινδρόμησης είναι ίδια με την ενεργοποίηση ενός μοναδικού νευρώνα, ο οποίος υλοποιεί γραμμική συνάρτηση ενεργοποίησης.



Σχήμα 2.2: *Τεχνητός Νευρώνας*

Συναρτήσεις Ενεργοποίησης

Η πρώτη συνάρτηση ενεργοποίησης που αναπτύχθηκε, ήταν η **Step Function**. Στην ουσία, επιστρέφει την κατάσταση του νευρώνα, δηλαδή αν έχει ενεργοποιηθεί ή όχι.

$$f(x) = \begin{cases} 0, & \text{εάν } x < 0 \\ 1, & \text{εάν } x \geq 0 \end{cases}$$

Η γραμμική συνάρτηση ενεργοποίησης απλά επιστρέφει την είσοδο της χωρίς καμία αλλαγή και δεν εισάγει μη γραμμικότητα στο νευρωνικό δίκτυο. Για αυτόν τον λόγο, συνήθως χρησιμοποιείται στην τελευταία στρώση ενός δικτύου, όπου η έξοδος πρέπει να είναι γραμμική συνάρτηση της εισόδου. Η γραμμική συνάρτηση εκφράζεται με τον ακόλουθο μαθηματικό τύπο:

$$f(x) = x$$

Η συνάρτηση ενεργοποίησης **ReLU** (Rectified Linear Unit) είναι μια δημοφιλής συνάρτηση που χρησιμοποιείται σε νευρωνικά δίκτυα. Ουσιαστικά, η συνάρτηση ReLU ¨ενεργοποιεί¨ τον νευρώνα αν η είσοδος είναι θετική, ενώ εάν είναι αρνητική, διατηρεί τον νευρώνα ανενεργό. Η συνάρτηση ορίζεται ως εξής:

$$f(x) = \max(0, x)$$

Η συνάρτηση ενεργοποίησης **ELU** (Exponential Linear Unit) είναι μια άλλη δημοφιλής συνάρτηση που χρησιμοποιείται σε νευρωνικά δίκτυα. Η συνάρτηση ELU συμπεριφέρεται παρόμοια με τη συνάρτηση ReLU για θετικές τιμές εισόδου, δηλαδή $x > 0$, αλλά διαφέρει όταν η είσοδος είναι αρνητική ($x \leq 0$). Σε αυτήν την περίπτωση, η ELU είναι ομαλή και μη-μηδενική, ενώ η ReLU θα επέστρεφε πάντα μηδέν. Η συνάρτηση ELU ορίζεται ως εξής:

$$f(x) = \begin{cases} x, & \text{αν } x > 0 \\ a \cdot (e^x - 1), & \text{αν } x \leq 0 \end{cases}$$

Όπου a είναι μια θετική παράμετρος που καθορίζει τον ρυθμό σύγκλισης της ELU στο μηδέν όταν η είσοδος είναι αρνητική.

Η συνάρτηση **sigmoid** έχει την ιδιότητα να αντιστρέφεται απότομα γύρω από το κέντρο της γραφικής της παράστασης, και γι' αυτό χρησιμοποιείται κυρίως σε προβλήματα δυαδικής ταξινόμησης (όπου θέλουμε να προβλέψουμε δύο κατηγορίες, όπως 0 ή 1).

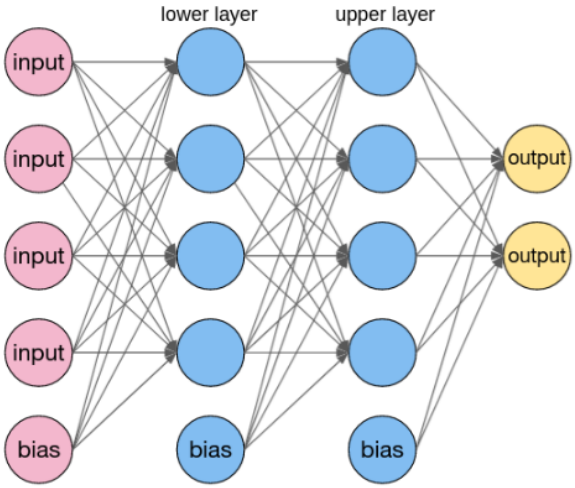
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Η παράγωγός της συνάρτησης ως προς την είσοδο z είναι $\sigma'(z) = \sigma(z)(1 - \sigma(z))$. Αυτή η παράγωγος χρησιμοποιείται συχνά κατά την εκπαίδευση των νευρωνικών δικτύων μέσω του αλγορίθμου προς τα πίσω (backpropagation). Το κύριο πλεονέκτημα της είναι ότι παράγει έξοδο στο εύρος (0, 1), καθιστώντας την κατάλληλη για προβλήματα που αφορούν πιθανότητες. Ωστόσο, έχει και μειονεκτήματα, όπως το πρόβλημα της εξαφάνισης της κλίσης (vanishing gradient) σε βαθιά νευρωνικά δίκτυα. Επίσης, δεν είναι κατάλληλη για προβλήματα ταξινόμησης όπου χρειάζεται να προβλέψουμε περισσότερες από δύο κατηγορίες.

2.3.3 Βαθιά Νευρωνικά Δίκτυα

Perceptron Πολλών Επιπέδων - Multilayer Perceptron (MLP)

Αυτό το είδος δικτύων αποτελείται από πολλαπλά στρώματα υπολογιστικών μονάδων, συνήθως διασυνδεδεμένα με τρόπο πρόσθιας τροφοδότησης (Feedforward Networks. Κάθε νευρώνας σε ένα επίπεδο έχει κατευθύνει τις συνδέσεις με όλους τους νευρώνες του επόμενου στρώματος.



Σχήμα 2.3: *Multilayer Perceptron*

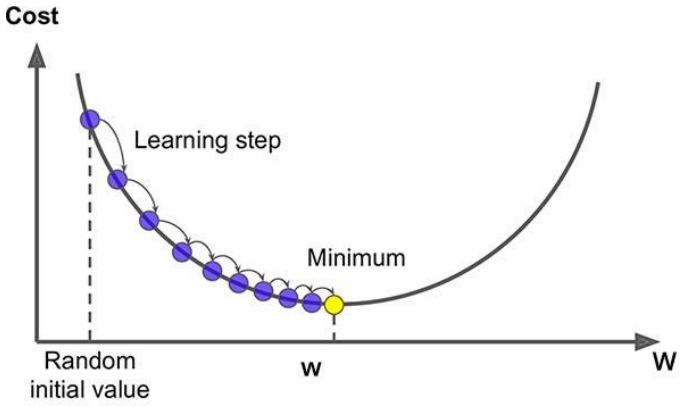
Οπίσθια Διάδοση Σφάλματος - Backpropagation

Τα δίκτυα πολλαπλών επιπέδων χρησιμοποιούν μια ποικιλία τεχνικών μάθησης. Η πιο δημοφιλής είναι η οπίσθια διάδοση. Εδώ, οι τιμές εξόδου συγκρίνονται με τη σωστή απάντηση για να υπολογιστεί η τιμή κάποιας προκαθορισμένης συνάρτησης κόστους (Cost Function). Με διάφορες τεχνικές, το σφάλμα διαδίδεται στην αντίθετη κατεύθυνση, προς την είσοδο. Χρησιμοποιώντας αυτές τις πληροφορίες, ο αλγόριθμος προσαρμόζει τα βάρη κάθε σύνδεσης για να μειώσει την τιμή της συνάρτησης κόστους κατά κάποιο μικρό μέγεθος (ρυθμός μάθησης- learning rate). Μετά την επανάληψη αυτής της διαδικασίας, το δίκτυο συνήθως θα συγκλίνει σε κάποια κατάσταση όπου το σφάλμα των υπολογισμών είναι μικρό. Σε αυτή την περίπτωση, θα λέγαμε ότι το δίκτυο έχει μάθει μια συγκεκριμένη λειτουργία στόχου.

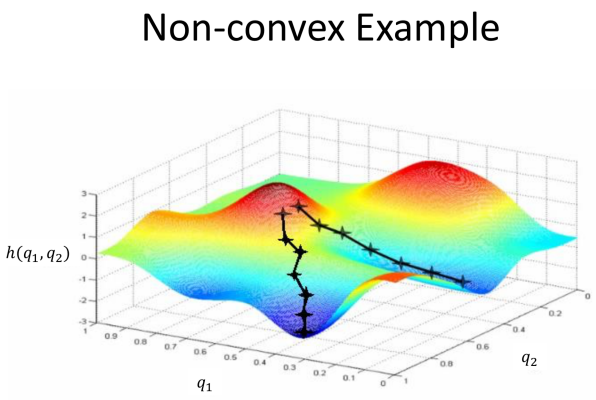
Κατάβαση Κλίσης - Gradient Descent

Για την σωστή ρύθμιση των βαρών, εφαρμόζεται μια γενική μέθοδος για τη μη γραμμική βελτιστοποίηση που ονομάζεται Κατάβαση Κλίσης. Για το σκοπό αυτό, το δίκτυο υπολογίζει το παράγωγο της συνάρτησης σφάλματος σε σχέση με τα βάρη του δικτύου και αλλάζει τα βάρη έτσι ώστε το σφάλμα να μειώνεται (συνεπώς να κατεβαίνει προς τα κάτω στην επιφάνεια της συνάρτησης σφάλματος). Η κατάβαση γίνεται με συγκεκριμένο βήμα. Για το λόγο αυτό, η οπίσθια διάδοση μπορεί να εφαρμοστεί μόνο σε δίκτυα με διαφοροποιήσιμες λειτουργίες

ενεργοποίησης. Το πιο σύνηθες πρόβλημα που εμφανίζεται με την εκμάθηση μέσω του αλγορίθμου Gradient Descent είναι ότι ο αλγόριθμος μπορεί να παγιδευτεί στα τοπικά ελάχιστα στις μη-κυρτές καμπύλες και δεν βρίσκει την βέλτιστη λύση (ολικό ελάχιστο).



Εικόνα 2.4: *Απεικόνιση Gradient Descent ενός χαρακτηριστικού [3]*



Εικόνα 2.5: *Απεικόνιση Gradient Descent δύο χαρακτηριστικών [4]*

Υπάρχουν τρία ήδη αλγορίθμων Gradient Descent τα οποία διαφέρουν ως προς το πόσα δεδομένα χρησιμοποιούνται για τον υπολογισμό της διαβάθμισης της αντικειμενικής συνάρτησης [22].

1. **Batch Gradient Descent**: Ο πιο παραδοσιακός τύπος κατάβασης κλίσης. Υπολογίζει η κλίση της συνάρτησης κόστους ως προς τις παραμέτρους, χρησιμοποιώντας ολόκληρο το σύνολο εκπαίδευσης. Ο αλγόριθμος Batch Gradient Descent είναι αποτελεσματικός αλλά μπορεί να είναι αργός για μεγάλα σύνολα δεδομένων, καθώς απαιτεί τον υπολογισμό της κλίσης σε όλα τα δεδομένα κατά κάθε επανάληψη. Ωστόσο, είναι εγγυημένο ότι θα συγκλίνει σε μια τοπική ελάχιστη της συνάρτησης κόστους.

$$\theta = \theta - \eta \cdot \nabla J(\theta)$$

Όπου θ είναι οι παράμετροι του μοντέλου, η είναι το learning rate που καθορίζει το βήμα ενημέρωσης και $\nabla J(\theta)$ είναι η κλίση της συνάρτησης κόστους.

2. **Stohastic Gradient Descent**: Πρόκειται για παραλλαγή του αλγορίθμου Batch Gradient Descent. Σε αντίθεση με τον Batch Gradient Descent, όπου ο υπολογισμός της κλίσης γίνεται χρησιμοποιώντας ολόκληρο το σύνολο εκπαίδευσης, ο αλγόριθμος SGD υπολογίζει την κλίση χρησιμοποιώντας μόνο ένα τυχαίο δείγμα από το σύνολο εκπαίδευσης σε κάθε επανάληψη. Αυτό το κάνει κατάλληλο για μεγάλα σύνολα δεδομένων λόγω της ταχύτητάς του. Μπορεί τα προσπεράσει τα τοπικά ελάχιστα σε μη-κυρτές επιφάνειες, δεδομένου ότι το learning rate έχει τεθεί αρκετά ψηλά.

$$\theta = \theta - \eta \cdot \nabla J(\theta, x_i, y_i)$$

3. **Mini-Batch Gradient Descent**: Αποτελεί συνδυασμό των δύο προηγούμενων αλγορίθμων. Χρησιμοποιεί ένα μικρό υποσύνολο των δεδομένων εκπαίδευσης (batch) αντί για ένα τυχαίο δείγμα ή ολόκληρο το σύνολο δεδομένων. Αυτό το κάνει κατάλληλο για την εκπαίδευση μοντέλων σε μεγάλα σύνολα δεδομένων. Το μέγεθος του batch καθορίζει πόσα δείγματα θα χρησιμοποιηθούν σε κάθε επανάληψη. Αυτό είναι μια υπερπαράμετρος που πρέπει να οριστεί πριν από την εκπαίδευση. Οι παράμετροι ενημερώνονται με βάση τον υπολογισμό της κλίσης για το batch, αλλά η ενημέρωση γίνεται μετά από κάθε batch, όχι μετά από κάθε δείγμα.

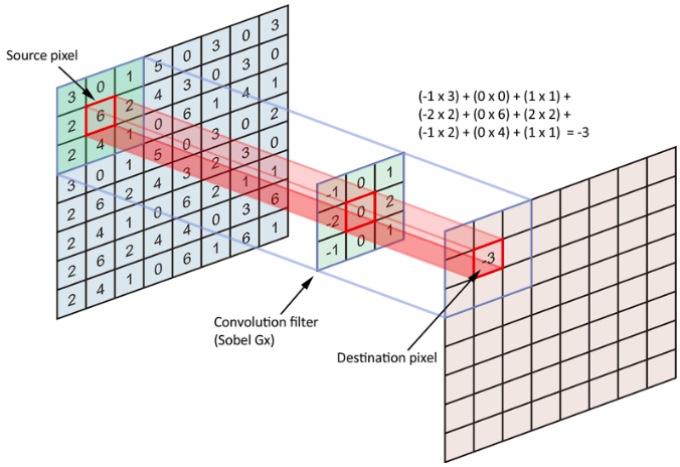
$$\theta = \theta - \eta \cdot \nabla J(\theta; x^{(i:i+b)}, y^{(i:i+b)})$$

Ο Mini-Batch ελαχιστοποιεί τις πιθανότητες να παγιδευτεί το μοντέλο σε τοπικό ελάχιστο, καθιστώντας τον τον πιο δημοφιλή αλγόριθμο σε βιβλιοθήκες Deep Learning.

2.3.4 Συνελικτικά Νευρωνικά Δίκτυα

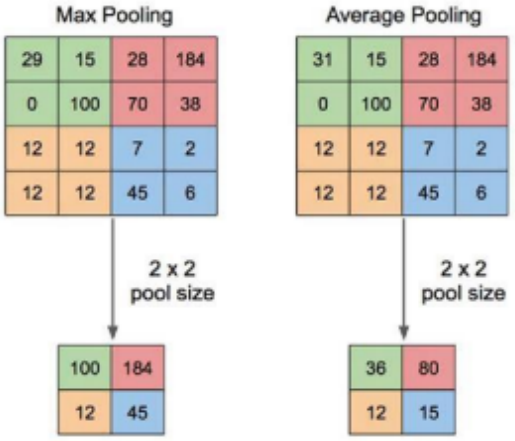
Το τεράστιο ενδιαφέρον για μοντέλα βαθιάς μάθησης και η πρόοδος στις κάρτες γραφικών τις τελευταίες δύο δεκαετίες οδήγησαν στην επικράτηση των συνελικτικών νευρωνικών δικτύων (Convolutional Neural Networks). Πρόκειται για ειδική κατηγορία πολυστρωματικών δικτύων, η αρχιτεκτονική των οποίων αποτελεί τη κυρίαρχη λύση, κυρίως σε προβλήματα όρασης υπολογιστών [23][24]. Βασίζονται σε τρία βασικά είδη επιπέδων:

- **Συνελικτικά (Convolutional)**: Ο βασικός μηχανισμός είναι η συνέλιξη. Εφαρμόζεται στην είσοδο με ένα σύνολο μικρών φίλτρων (ή πυρήνων). Αυτά τα φίλτρα σαρώνουν τον πίνακα εισόδου κατά πλάτος και κατά μήκος, ενισχύοντας συγκεκριμένα χαρακτηριστικά. Οι συνελίξεις μετατρέπουν τον πίνακα σε μια πιο αφαιρετική αναπαράσταση, όπου τα χαρακτηριστικά είναι πιο αναγνωρίσιμα.



Εικόνα 2.6: *Παράδειγμα συνέλιξης ενός Convolutional Layer*

- **Υποδειγματοληψία (Pooling)**: Μετά από κάθε συνέλιξη, συνήθως ακολουθεί ένα επίπεδο υποδειγματοληψίας. Στην υποδειγματοληψία, ο πίνακας χωρίζεται σε περιοχές και εξάγεται ένα αντιπροσωπευτικό στοιχείο από κάθε περιοχή, όπως το μέγιστο (Max Pooling) ή το μέσον (Average Pooling), ξανά με χρήση φίλτρου. Αυτό μειώνει τη διαστατικότητα των δεδομένων, κρατώντας παράλληλα τα σημαντικά χαρακτηριστικά.



Εικόνα 2.7: *Παράδειγμα Pooling Layer*

- **Πλήρως Συνδεδεμένα Επίπεδα (Fully Connected Layers)**: Το τελευταίο τμήμα ενός CNN αποτελείται από πλήρως συνδεδεμένα επίπεδα, όπου γίνεται η ταξινόμηση ή η πρόβλεψη. Αυτά τα επίπεδα λειτουργούν όπως τα κανονικά νευρωνικά δίκτυα και συνδέονται με όλα τα χαρακτηριστικά που έχουν εξαχθεί προηγουμένως.

Φίλτρα

Στην περιγραφή των επιπέδων των συνελικτικών δικτύων αναφέρθηκε ο όρος φίλτρα. Πρόκειται για πίνακες μικρότερων ή ίσων διαστάσεων με αυτές του πίνακα εισόδου, οι οποίοι ολισθαίνουν πάνω σε αυτόν για την εφαρμογή πράξεων. Ένα φίλτρο αποτελείται από:

- Μέγεθος φίλτρου-kernel size: Ορίζει το μέγεθος του φίλτρου, όπως φαίνεται και στις

προηγούμενες εικόνες

- Βήμα-stride: Ρυθμίζει το βήμα μετακίνησης του φίλτρου. Όσο μεγαλύτερο το βήμα, τόσο μικρότερη η διάσταση εξόδου.
- Γέμισμα περιθωρίου-padding: Ορίζει το μέγεθος του γεμίσματος γύρω από τις άκρες της εισόδου.

Το μέγεθος της εξόδου της εφαρμογής ενός φίλτρου σε κάποιο πίνακα εισόδου δίνεται από τον τύπο:

$$\frac{N + 2P - F}{S} + 1$$

όπου N οι διαστάσεις τις εισόδου, P το padding, F οι διαστάσεις του φίλτρου και S το βήμα.

2.3.5 Αναδρομικά Νευρωνικά Δίκτυα

Τα αναδρομικά νευρωνικά δίκτυα (Recurrent Neural Networks - RNNs) είναι ένα είδος τεχνητών νευρωνικών δικτύων που αναπτύχθηκαν για την επεξεργασία δεδομένων σε ακολουθίες, όπως χρονοσειρές, κείμενα και ήχος [25]. Η ιστορία των αναδρομικών νευρωνικών δικτύων περιλαμβάνει την ανάπτυξη διαφόρων αλγορίθμων και αρχιτεκτονικών που επέτρεψαν την αποτελεσματική επεξεργασία ακολουθιών δεδομένων.

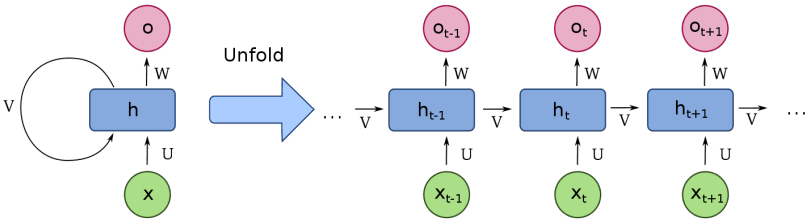
Βασικό δομικό τους στοιχείο αποτελεί ο αναδρομικός νευρώνας. Κάθε αναδρομικό κύτταρο διαθέτει μια εσωτερική κατάσταση (hidden state) που διατηρεί τις πληροφορίες που έχει εξάγει από προηγούμενες εισόδους. Λόγω αυτού, τα RNNs μπορούν να αναπτύσσονται χρονικά. Σε κάθε βήμα της επεξεργασίας, ο αναδρομικός νευρώνας λαμβάνει την τρέχουσα είσοδο και την εσωτερική κατάστασή του από το προηγούμενο βήμα, παράγοντας μια νέα εσωτερική κατάσταση. Αυτό του επιτρέπει να αναγνωρίζει πρότυπα σε ακολουθίες. Προφανώς η εσωτερική κατάσταση είναι ανακυκλούμενη και επαναχρησιμοποιείται σε κάθε βήμα. Αυτό τους επιτρέπει να διατηρούν μνήμη για πολλά βήματα πίσω στο παρελθόν και να επηρεάζουν την πρόβλεψη σε βάθος χρόνου. Οι έξοδοι του κυττάρου δίνονται από τους τύπους:

$$h_t = \phi(W_{hx}x_t + W_{hh}h_{t-1} + b_h)$$

$$y_t = \phi(W_{yh}h_t + b_y)$$

όπου φ η συνάρτηση ενεργοποίησης και Ω τα διανύσματα βαρών που επιδρούν στις εισόδους x_t και h_{t-1}.

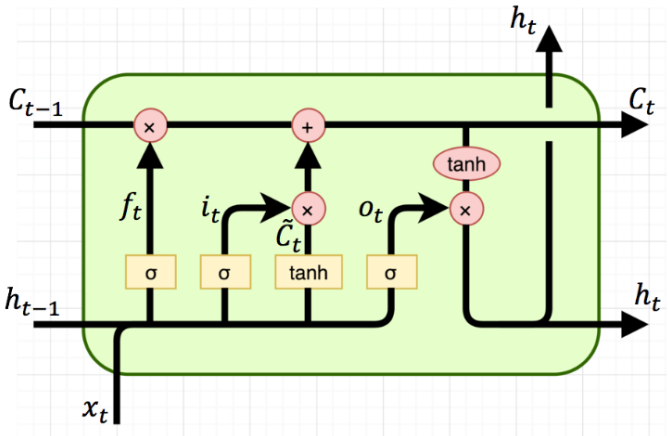
Κατά την εκπαίδευση των RNNs εμφανίζεται το πρόβλημα του vanishing gradient problem. Κατά το backpropagation, οι κλίσεις παρουσιάζουν συνεχώς μικρότερες τιμές. Τέτοιες τιμές κλίσεις εμποδίζουν την αλλαγή της τιμής των βαρών. Ως αποτέλεσμα, το δίκτυο ¨παγώνει¨ κατά την εκπαίδευση και οι παλαιότερες τιμές ξεχνιούνται και τα απλά RNNs δυσκολεύονται να διατηρήσουν μακροπρόθεσμες εξαρτήσεις σε δεδομένα που εκτείνονται πολλές χρονικές στιγμές πίσω. Αυτό, αποκαλείται το πρόβλημα της βραχυπρόθεσμης μνήμης (short-term memory problem) και η λύση του έρχεται με την δημιουργία νέων τύπων κυττάρων.



Σχήμα 2.4: *Αναδρομικό Κύτταρο*

Κύτταρα LSTM

Οι Sepp Hochreiter, Jürgen Schmidhuber πρότειναν το 1997 την αρχιτεκτονική του Long Short-Term Memory Loss (LSTM) [26] ώστε να αντιμετωπίσουν το πρόβλημα του vanishing gradient των RNNs. Η αρχιτεκτονική τους είναι παρόμοια με των απλών αναδρομικών κυττάρων, έχοντας την προσθήκη εσωτερικών μηχανισμών, ονόματι πυλών (gated cells). Οι πύλες αυτές είναι:



Σχήμα 2.5: *Κύτταρο LSTM*

- **Input Gate**: Αυτή η πύλη ελέγχει ποια νέα πληροφορία θα αποθηκευτεί στο κύτταρο. Αυτό γίνεται μέσω των πράξεων συνέλιξης και συνάθροισης, όπως τον υπολογισμό της συνολικής εισόδου. Ο τύπος της πύλης εισόδου είναι:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

όπου σ είναι η συνάρτηση sigmoid (συνάρτηση ενεργοποίησης) και W_i είναι τα βάρη της πύλης εισόδου.

- **Forget Gate**: Αυτή η πύλη ελέγχει ποιες πληροφορίες θα διατηρηθούν στο κελί μνήμης από την προηγούμενη κατάσταση του LSTM και ποιες θα απορριφθούν. Ο τύπος της πύλης λήψης πληροφοριών είναι:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

- **Cell state update**: Μέσω αυτής της πύλης γίνεται η ανανέωση της κατάστασης του κυττάρου:

$$C_t = f_t C_{t-1} + i_t \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c)$$

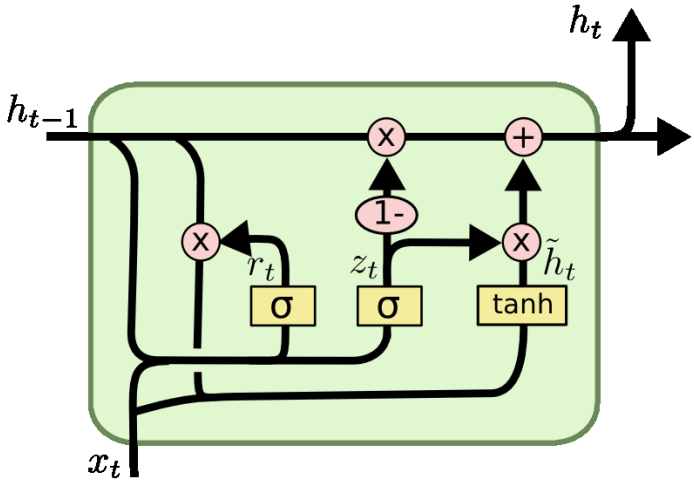
- **Output Gate**: Αυτή η πύλη ελέγχει ποια πληροφορία από το κελί μνήμης θα περάσει στην έξοδο του LSTM. Ο τύπος της πύλης εξόδου είναι:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Λόγω της πύλης forget gate δεν υπάρχει το πρόβλημα με το vanishing gradient που σημαίνει ότι το LSTM μπορεί να μάθει εργασίες που απαιτούν μνήμη από γεγονότα που συνέβησαν χιλιάδες ή ακόμα και εκατομμύρια χρονικά βήματα (time steps) νωρίτερα.

Gated Recurrent Units (GRU)

Τα GRU, παρουσιάστηκαν από τον Cho και τους συνεργάτες του το 2014 [27]. Είναι μια άλλη μορφή RNN που είναι παρόμοια με τα LSTM αλλά χρησιμοποιούν λιγότερες πύλες, κάτι που τα καθιστά υπολογιστικά αποδοτικότερα.



Σχήμα 2.6: *Κύτταρο GRU*

- **Update Gate**: Αυτή η πύλη αποφασίζει πόσο από την προηγούμενη κατάσταση (state) θα διατηρηθεί και πόση νέα πληροφορία θα προστεθεί στην τρέχουσα κατάσταση. Ο τύπος της πύλης ενημέρωσης είναι:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

- **Reset Gate**: Αυτή η πύλη αποφασίζει κατά πόσο η προηγούμενη κατάσταση θα αγνοηθεί και πόσο νέα πληροφορία θα προστεθεί στην τρέχουσα κατάσταση. Ο τύπος της πύλης επιλογής είναι:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

- **Candidate Activation**:

$$\hat{h}_t = \tanh(W_{xn} x_t + r_t \cdot (W_{hn} h_{t-1}) + b_n)$$

- **Output Gate**:

$$h_t = (1 - z_t) \cdot n_t + z_t \cdot h_{t-1}$$

Το GRU τις περισσότερες φορές είναι ισάξιο με το LSTM. Δεδομένου όμως του γεγονότος ότι το GRU έχει λιγότερες παραμέτρους και δεν έχει το cell state, προτιμάται λόγω χαμηλότερης χρήσης της μνήμης και γιατί εκπαιδεύεται πιο γρήγορα.

Κεφάλαιο 3

Δεδομένα

Σε αυτό το κεφάλαιο, θα γίνει ανάλυση της διαδικασίας δημιουργίας των συνθετικών δεδομένων που θα χρησιμοποιηθούν για την εξέταση των προβλεπτικών μοντέλων. Η δημιουργία συνθετικών δεδομένων έχει ως στόχο την διερεύνηση της προσαρμογής των μοντέλων σε δεδομένα με εξασφαλισμένη ύπαρξη συσχετίσεων των χαρακτηριστικών και χρονικών μοτίβων. Στη συνέχεια, θα γίνει μετάβαση στα δεδομένα του πραγματικού κόσμου που προέρχονται από το αποθετήριο GWA του Πολυτεχνείου του Ντελφτ [28]. Παρουσιάζεται μια αρχική ανάλυση των δεδομένων καθώς και τα βήματα καθαρισμού μέσω προεπεξεργασίας και μετασχηματισμού σε ένα ολοκληρωμένο σύνολο δεδομένων χρονοσειρών. Τέλος, παρουσιάζονται εκτενώς η δομή, οι σχέσεις και οι κατανομές των χαρακτηριστικών των συνόλων δεδομένων, με σκοπό μια πρώιμη εκτίμηση της ικανότητας προσαρμογής των προβλεπτικών μεθόδων που θα εξετασθούν στο επόμενο κεφάλαιο.

3.1 Συνθετικά Δεδομένα

Σε αυτήν την ενότητα, παρουσιάζονται οι επιλογές και οι διαδικασίες που ακολουθήθηκαν κατά τη δημιουργία του συνθετικού συνόλου δεδομένων. Αναλύονται οι σχέσεις και οι συναρτήσεις που επιλέχθηκαν για την παραγωγή των στηλών, καθώς και ο τρόπος με τον οποίο επιλέχθηκαν οι παράμετροι για κάθε μία από αυτές. Επίσης γίνεται σύγκρισή του με τα πραγματικά σύνολα δεδομένων. Σκοπός είναι να εξετασθεί ένα σύνολο δεδομένων το οποίο θα περιέχει περίπλοκες σχέσεις μεταξύ των χαρακτηριστικών και του χρόνου. Η επιλογή των ονομάτων των χαρακτηριστικών βασίζεται στα ονόματα των αντίστοιχων στηλών των πραγματικών δεδομένων και αναλύεται στην επόμενη ενότητα.

3.1.1 Δημιουργία Δεδομένων

Ξεκινώντας, δημιουργήθηκε μια χρονοσειρά χρονοσφραγίδων από τον Ιανουάριο του 2018 μέχρι τον Δεκέμβριο του 2021 με ωριαία συχνότητα. Τα χρονοσήματα θα αποτελέσουν τη βάση του τελικού συνθετικού συνόλου δεδομένων.

Για τη δημιουργία του χρόνου επεξεργασίας των επεξεργαστών (CPUTime), επιλέχθηκε ένα σταθερό επίπεδο $1e6$, εποχικότητα, κυκλικότητα, θόρυβος καθώς και τυχαίες αιχμές. Συγκεκριμένα, πρόκειται για τυχαίο θόρυβο που ακολουθεί κανονική κατανομή με $\mu = 1, \sigma = 0.01$ και αιχμές που ακολουθούν τριγωνικό μοτίβο με περίοδο μίας εβδομάδας. Επίσης, εισάγεται και κυκλικότητα υλοποιημένη με ημιτονοειδή συνάρτηση πε-

ριόδου $T = 24h$. Το γινόμενο των παραπάνω και του επιπέδου, μειώνονται κατά 10% για τους μήνες των διακοπών:

$$spikes = 2.5 - 1.5|\frac{(t-12)MOD6}{24*7}| \tag{3.1}$$

$$CPUTime = (1000000) * norm(1, 0.01) * spikes * S * (1 + 0,1 * sin(t\frac{2\pi}{T})), \tag{3.2}$$

όπου t οι ώρες των χρονοσφραγίδων και $S = 0.9$ ή $S = 1$, με βάση τον μήνα των χρονοσφραγίδων.

Η δημιουργία της χρήσης μνήμης (Memory) είναι υψηλά συσχετισμένη με τον χρόνο επεξεργασίας (CPUTime) αλλά με μη γραμμική σχέση. Σε πραγματικές καταστάσεις, συνήθως μια διεργασία με μεγάλες απαιτήσεις σε επεξεργαστική δύναμη, απαιτεί και μεγάλο χώρο στην μνήμη. Οι εξαιρέσεις όμως δεν είναι σπάνιες. Γι' αυτό τον λόγο εισήχθη τυχαίος θόρυβος κανονικής κατανομής με $\mu = 0, \sigma = 20000$ που αντιπροσωπεύει τις διαφορές των ειδών των εγρασιών. Επίσης προστέθηκε μία ημιτονοειδής σχέση με το CPUTime:

$$Memory = CPUTime * 100(1 + 0.5sin(\frac{CPUTime}{800000})) + norm(0, 20000) \tag{3.3}$$

Ο αριθμός των επεξεργαστών (NProcs) δημιουργείται με μη γραμμική σχέση του CPUTime και του Memory. Προκύπτει από άθροισμα του λογαρίθμου του CPUTime και της τετραγωνικής ρίζας του Memory επί της υπερβολικής εφαπτομένης του. Προστέθηκε ξανά τυχαίος θόρυβος κανονικής κατανομής με $\mu = 15, \sigma = 10$, για να διαφοροποιηθεί ξανά το μέγεθος των εργασιών.

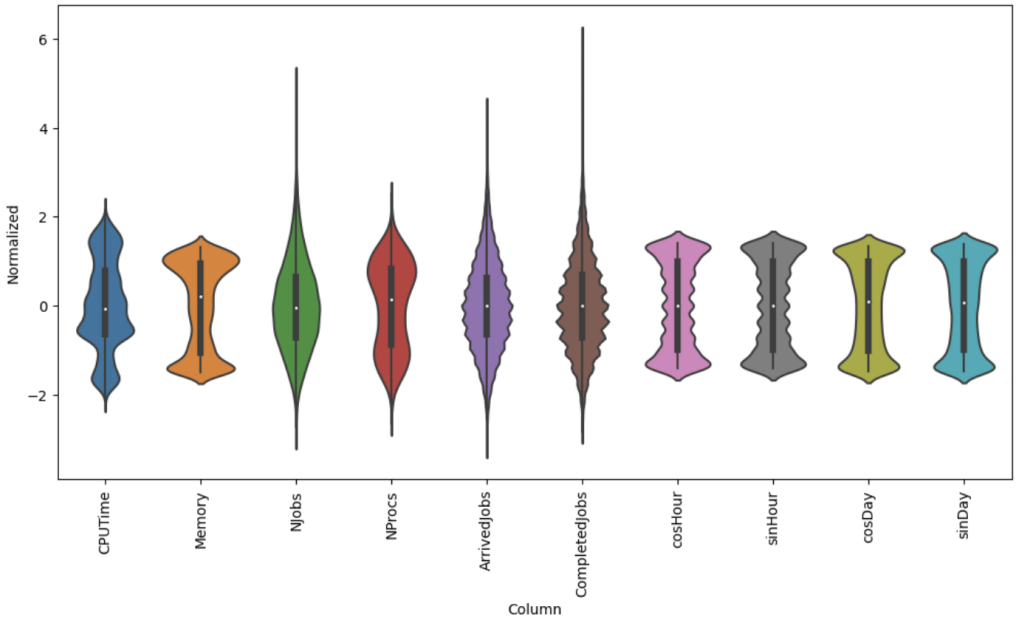
$$NProcs = log(CPUTime) + \frac{\sqrt{Memory} * tanh(\frac{Memory}{1e9})}{10} + norm(15, 10) \tag{3.4}$$

Ο αριθμός των εργασιών (NJobs) συνδέεται με τον αριθμό των επεξεργαστών (NProcs), με τυχαίες διακυμάνσεις. Κάθε ώρα, επιλέγεται ένας τυχαίος αριθμός που ακολουθεί κανονική κατανομή με $\mu = 6, \sigma = 0.5$ και αντιπροσωπεύει τον μέσο όρο των επεξεργαστών που έχει δεσμεύσει η κάθε εργασία στο σύστημα. Φυσικά, αυτός ο αριθμός φράζεται από κάτω στη μονάδα:

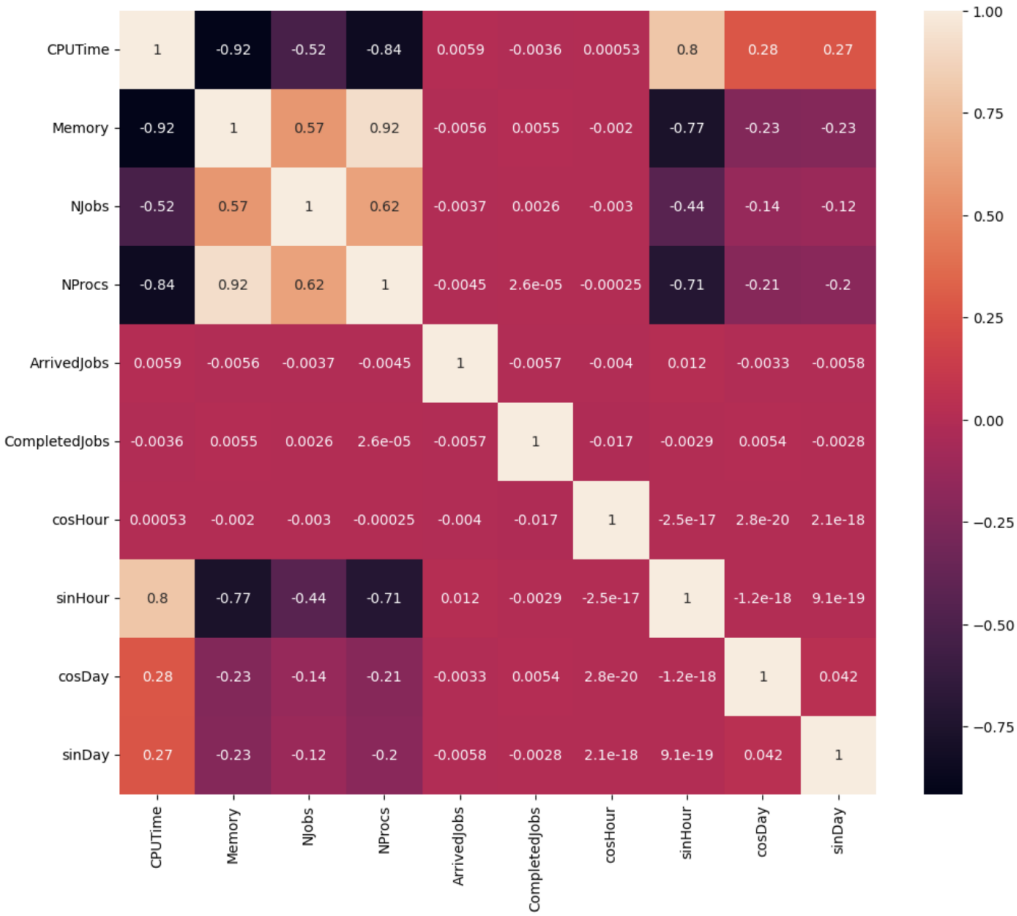
$$NJobs = NProcs/max(1, norm(6, 0.5)) \tag{3.5}$$

Η άφιξη και η ολοκλήρωση των εργασιών ArrivedJobs και CompletedJobs υλοποιήθηκε με τυχαίες μεταβλητές Ποισσον, με $k = 10, \lambda = 1$ και $k = 8, \lambda = 1$. Με αυτόν τον τρόπο, παράγονται ακέραιες τιμές.

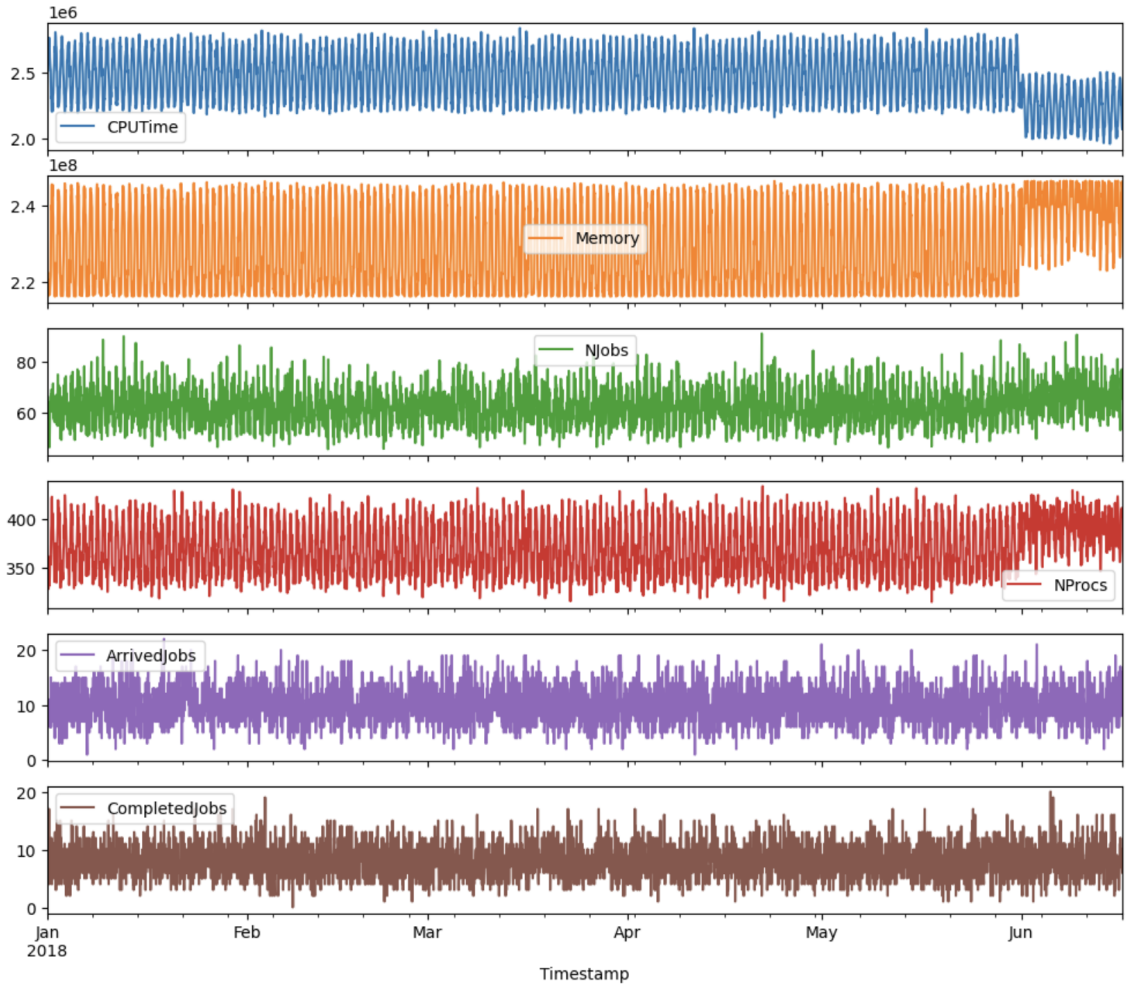
Τέλος, προσθέτουμε επιπλέον τις ημιτονοειδείς στήλες που εξάγονται από τις χρονοσφραγίδες. Συνολικά, αυτές οι επιλογές συμβάλλουν στη δημιουργία ενός ρεαλιστικού συνθετικού συνόλου δεδομένων που μπορεί να χρησιμοποιηθεί, το οποίο όμως παρουσιάζει πιο περίπλοκες σχέσεις μεταξύ των χαρακτηριστικών, βοηθώντας να αναδείξουμε την χρησιμότητα των νευρωνικών δικτύων έναντι της γραμμικής παλινδρόμησης. Η κατάσταση των κανονικοποιημένων συνθετικών δεδομένων φαίνεται στην εικόνα 3.4. Επιπλέον, παρουσιάζονται, ο πρώτος χρόνος των δεδομένων στην εικόνα 3.5 και ο πίνακας συντελεστών συσχέτισης χαρακτηριστικών.



Εικόνα 3.1: *Κατανομή κανονικοποιημένων συνθετικών δεδομένων χρονοσειράς*



Εικόνα 3.2: *Πίνακας συντελεστών συσχέτισης χαρακτηριστικών συνθετικών δεδομένων*



Εικόνα 3.3: *Απεικόνιση συνθετικών δεδομένων χρονοσειράς*

3.2 Πραγματικά Δεδομένα

Σε αυτή την ενότητα, παρουσιάζονται και αναλύονται τα σύνολα δεδομένων της αρχειοθήκης The Grid Workloads Archive - GWA, που παρέχονται δημόσια από το Πολυτεχνείο του Ντελφτ. Η GWA αποτελεί μία σημαντική πηγή δεδομένων για την ερευνητική κοινότητα στον τομέα των κατανεμημένων υπολογιστικών συστημάτων. Πρόκειται για μία συλλογή από log data που προέρχονται από πραγματικά συστήματα κατανεμημένων υπολογιστικών πόρων, τα οποία παρέχουν πληροφορίες για τις εργασίες που ¨τρέχουν¨ στο σύστημα, όπως το υπολογιστικό φορτίο και ο χρόνος εκτέλεσης.

3.2.1 Ανάλυση Δεδομένων

Στο πλαίσιο της παρούσας εργασίας εξετάσθηκαν συνολικά πέντε σύνολα δεδομένων από το GWA. Πρόκειται για δεδομένα εργασιών διαφορετικών κατανεμημένων υπολογιστικών συστημάτων στην Ευρώπη. Ενδεικτικές πληροφορίες για το μέγεθος των δεδομένων φαίνεται στον Πίνακα 3.1.

Όπως ήδη αναφέρθηκε, κάθε γραμμή των δεδομένων περιέχει πληροφορίες σχετικές με την εργασία την οποία προσδιορίζει. Αξίζει να σημειωθεί πως όλα τα σύνολα δεδομένων

Πίνακας 3.1: *Πίνακας διαστάσεων συνόλων δεδομένων GWA πριν του καθαρισμό*

Όνομα	Εργασίες	Χρονικό Πλαίσιο
DAS2	1.12 εκατ.	1.8 έτη
Grid5000	1.02 εκατ.	2.52 έτη
NorduGrid	0.78 εκατ.	3.15 έτη
AuverGrid	0.4 εκατ.	1 έτος
SHARCnet	1.2 εκατ.	1.07 έτη

της συλλογής περιέχουν τα ίδια χαρακτηριστικά (στήλες) για τις εργασίες. Μεταξύ αυτών, το user ID και group ID του χρήστη που αιτήθηκε για την εκτέλεση της εργασίας, καθώς και οι JobID, ExecutableID, QueueID, PartitionID, OrigSiteID, LastRunSiteID, ProjectID οι οποίες περιέχουν διάφορα αναγνωριστικά κατηγοριών των εργασιών, τα οποία είναι ασήμαντα για τον σκοπό της παρούσας εργασίας. Συνολικά από τις τριάντα δύο στήλες, έχουν σημασία για την συνέχεια οι:

- SubmitTime: Η χρονοσφραγίδα υποβολής της εργασίας.
- WaitTime: Ο χρόνος αναμονής της εργασίας αρχίσει να εκτελείται σε δευτερόλεπτα. Σε κάποια από τα σύνολα δεδομένων, η στήλη χρησιμοποιείται για ένδειξη πως κάποια εργασία ακυρώθηκε, δίνοντας την τιμή -1.
- RunTime: Ο χρόνος εκτέλεσης της εργασίας σε δευτερόλεπτα.
- NProc: Ο αριθμός των επεξεργαστών που χρησιμοποιήθηκαν για την εκτέλεση της εργασίας.
- UsedCPUTime: Ο συνολικός χρόνος CPU που χρησιμοποίησε από η εργασία σε δευτερόλεπτα.
- UsedMemory: Η ποσότητα μνήμης που χρησιμοποίησε η εργασία σε MB.

Στη συνέχεια, παρουσιάζεται μια στατιστική ανάλυση (πίνακες 3.2 έως 3.6) των τιμών των σημαντικών στηλών των δεδομένων. Κατά κανόνα, οι τιμές βρίσκονται εντός λογικών ορίων. Οι διαφορετικές τάξεις μεγεθών ανάμεσα στα σύνολα δεδομένων προσφέρουν πληροφορία για τις διαφορές στην κλίμακα και την χρήση των κατανεμημένων συστημάτων.

Παρατηρείται πως, στα Grid5000 και NorduGrid, οι στήλες UsedCPUTime, MemoryUsed και UsedCPUTime αντίστοιχα, εμφανίζουν μόνο την τιμή -1. Το γεγονός αυτό τα καθιστά μη χρησιμοποιήσιμα για το σκοπό της εργασίας, καθώς δεν μπορούν να χρησιμοποιηθούν για την προσαρμογή ή την δοκιμή κάποιου μοντέλου. Όπως θα αναφερθεί αργότερα, ο χρόνος του επεξεργαστή θα είναι η στήλη-στόχος των προβλέψεών μας. Επίσης, βλέπουμε την τιμή -1 να εμφανίζεται ως ελάχιστη τιμή σε αρκετά χαρακτηριστικά, όμως οι φυσιολογικές τιμές μέσου όρου και τυπικής απόκλισης δείχνουν πως πρόκειται για ακραίες τιμές ή για εργασίες οι οποίες ακυρώθηκαν. Οι τιμές αυτές θα πρέπει να αντιμετωπισθούν κατά τον καθαρισμό του συνόλου δεδομένων, γιατί δύνανται να βλάψουν την απόδοση του τελικού μοντέλου.

Πίνακας 3.2: *Πίνακας στατιστικής ανάλυσης DAS2*

	RunTime	NProc	UsedCPUTime	UsedMemory
mean	369.717	4.306	34.052	45895.86
std	3938.101	6.361	308.307	346424.6
min	0	1	0	0
max	5.483e+05	128	51070.15	4.294e+06

Πίνακας 3.3: *Πίνακας στατιστικής ανάλυσης Grid5000*

	RunTime	NProc	UsedCPUTime	UsedMemory
mean	2462.630	5.815	-1	-1
std	24091.420	21.051	0	0
min	-1	1	-1	-1
max	3.015e+06	342	-1	-1

Πίνακας 3.4: *Πίνακας στατιστικής ανάλυσης NorduGrid*

	RunTime	NProc	UsedCPUTime	UsedMemory
mean	89273.91	1.073	-1	1.998e+05
std	2.842e+05	1.273	0	3.065e+05
min	-6471	1	-1	-1
max	1.807e+07	64	-1	2.147e+06

Πίνακας 3.5: *Πίνακας στατιστικής ανάλυσης AuverGrid*

	RunTime	NProc	UsedCPUTime	UsedMemory
mean	21661.28	0.86	18882.036	2.542e+05
std	38815.57	0.346	33011.668	3.342e+05
min	-1	0	-1	-1
max	1.575e+06	1	259316	3.667e+06

Πίνακας 3.6: *Πίνακας στατιστικής ανάλυσης SHARCNet*

	RunTime	NProc	UsedCPUTime	UsedMemory
mean	31654.30	2.993	20757.24	80496.21
std	1.165e+05	24.552	5.154e+06	4.639e+05
min	-1	-1	-2.124e+09	-1
max	1.39e+07	3000	2.087e+09	3.202e+07

3.2.2 Διαμόρφωση Δεδομένων Χρονοσειρών

Για την προσαρμογή ενός προβλεπτικού μοντέλου της ωριαίας κατάστασης του φορτίου εργασίας, τα διαθέσιμα δεδομένα πρέπει αποκτήσουν διαφορετική μορφή. Στόχος είναι η διαμόρφωση συνόλων δεδομένων χρονοσειρών, τα οποία θα είναι έτοιμα για την εκπαίδευση. Στις παρακάτω παραγράφους αναλύονται όλα τα βήματα της επεξεργασίας.

Πρώτο βήμα για την δημιουργία των τελικών συνόλων δεδομένων, είναι ο καθαρισμός από ¨βρόμικες¨ τιμές. Πρόκειται για τις τιμές που περιγράφηκαν κατά την ανάλυση, δηλαδή εργασίες οι οποίες έχουν μηδενικό ή αρνητικό χρόνο εκτέλεσης ή/και χρήση μνήμης. Επίσης, θα πρέπει να διαγραφούν όλες οι στήλες που δεν θα συμπεριληφθούν στις τελικές χρονοσειρές. Συγκεκριμένα, διαγράφηκαν όλες οι στήλες εκτός των SubmitTime, RunTime, NProc, UsedCPUTime και UsedMemory. Στη συνέχεια, αφαιρέθηκαν όλες οι γραμμές με μη θετικές τιμές σε οποιοδήποτε χαρακτηριστικό, ενώ προστέθηκε η στήλη StopTime, η οποία προέκυψε ως άθροισμα των SubmitTime, WaitTime και RunTime. Η κατάσταση των συνόλων δεδομένων μετά από την επεξεργασία φαίνεται στον Πίνακα 3.7. Τα Grid5000 και NorduGrid είναι πλέον κενά, ενώ τα υπόλοιπα τρία σύνολα έχασαν κατά σειρά 3%, 32.5% και 15% των εργασιών τους.

Πίνακας 3.7: *Πίνακας διαστάσεων συνόλων δεδομένων GWA μετά τον καθαρισμό*

Όνομα	Εργασίες	Χρονικό Πλαίσιο
DAS2	1.09 εκατ.	1.8 έτη
Grid5000	∅	∅
NorduGrid	∅	∅
AuverGrid	0.27 εκατ.	1 έτος
SHARCnet	1.02 εκατ.	1.07 έτη

Από τα επεξεργασμένα δεδομένα, είναι εφικτή η δημιουργία ενός ¨καθαρού¨ συνόλου δεδομένων, με την συνολική ωριαία κατάσταση του φορτίου του cluster. Πρόκειται ουσιαστικά για άθροισμα των χαρακτηριστικών των εργασιών που εκτελούνταν κάθε ώρα. Οι στήλες με τις οποίες αρχικοποιείται το σύνολο δεδομένων είναι οι CPUTime, Memory, NJobs, NProcs, ArrivedJobs και CompletedJobs. Η επιλογή των ωρών στις οποίες ήταν ενεργή μια εργασία γίνεται με τη χρήση των SubmitTime και StopTime. Έπειτα, αυξάνονται ανάλογα οι στήλες NJobs, NProcs, ArrivedJobs και CompletedJobs. Για τις CPUTime και Memory, γίνεται η παραδοχή πως το φορτίο ισομοιράζεται στις ώρες που η εργασία ήταν ενεργή. Δηλαδή αν μια εργασία εμφανίζεται σε N διαφορετικές ώρες, κάθε ώρα από αυτές στο τελικό σύνολο δεδομένων θα περιλαμβάνει τα CPUTime και UsedMemory της διαιρεμένα με N. Ο αλγόριθμος 3.1 εξηγεί την διαδικασία αναλυτικότερα. Τελευταίο βήμα, η προσθήκη των στηλών cosHour, sinHour, cosDay, & sinDay. Η στήλη των χρονοσφραγίδων είναι πολύ χρήσιμη για την πρόβλεψη, όμως όχι στην μορφή που βρίσκεται. Προσθέτοντας τις στήλες των ημιτονοειδών συναρτήσεων της ώρας και της ημέρας, δίνεται στα μοντέλα η δυνατότητα ανίχνευσης κάποιου πιθανού μοτίβου στα σήματα με ημερήσια και ετήσια περίοδο.

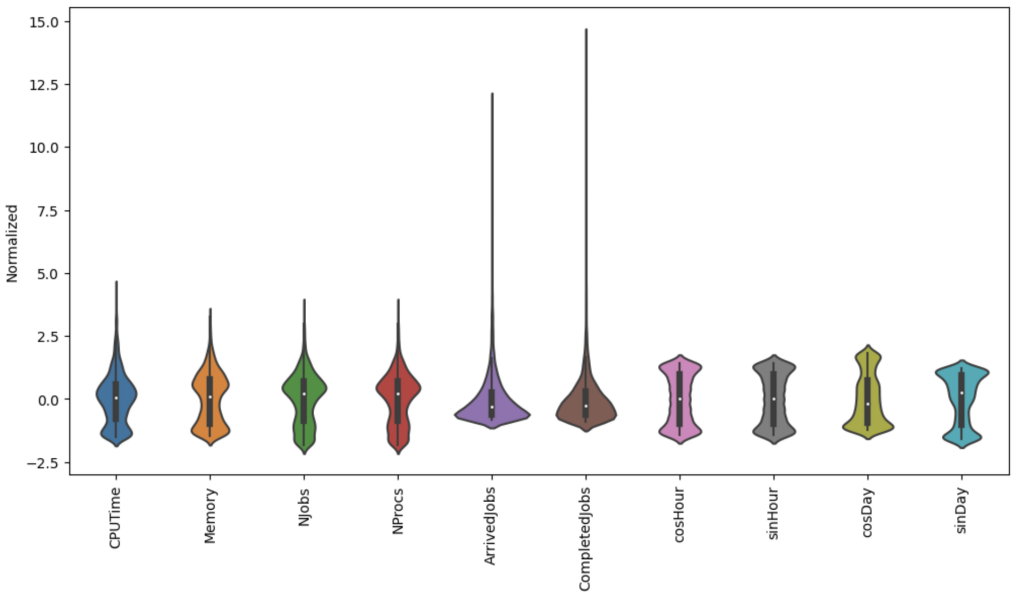
Αλγοριθμος 3.1: *Μετατροπή Log Data σε Δεδομένα Χρονοσειρών*

```
1: DF ← Log Dataset
2: TS ← Time-series Dataset
3: TS[∗] ← 0
4: for DFRow in DF do
5:     Start ← DFRow[SubmitTime]
6:     End ← DFRow[StopTime]
7:     for Hour in Start[Hour] : End[Hour] do
8:         TSRow[Jobs] ← TSRow[Jobs] + 1
9:         TSRow[CPU] ← TSRow[CPUTime] + DSRow[CPU]
10:        TSRow[Memory] ← TSRow[Memory] + DSRow[Memory]
11:        TSRow[Procs] ← TSRow[Procs] + DSRow[Procs]
12:    end for
13:    TSRow[Arrived][Start] ← TSRow[Arrived][Start] + 1
14:    TSRow[Completed][End] ← TSRow[Completed][End] + 1
15: end for
```

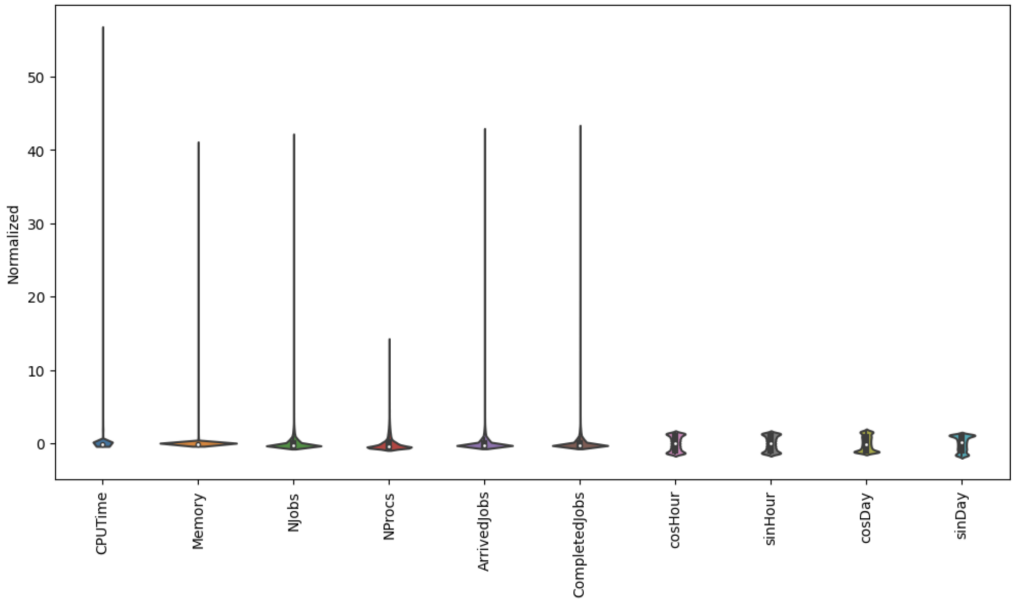
Η τελική κατάσταση των κατανομών των δεδομένων ακριβώς πριν τροφοδοτηθούν στην εκπαίδευση, φαίνεται στις εικόνες 3.4-3.6. Τα δεδομένα, έχουν υποστεί κανονικοποίηση z-score, αφαιρώντας τον μέσο όρο τους και διαιρώντας με την τυπική τους απόκλιση:

$$X_{norm} = \frac{X - E(X)}{\sigma(X)} \tag{3.6}$$

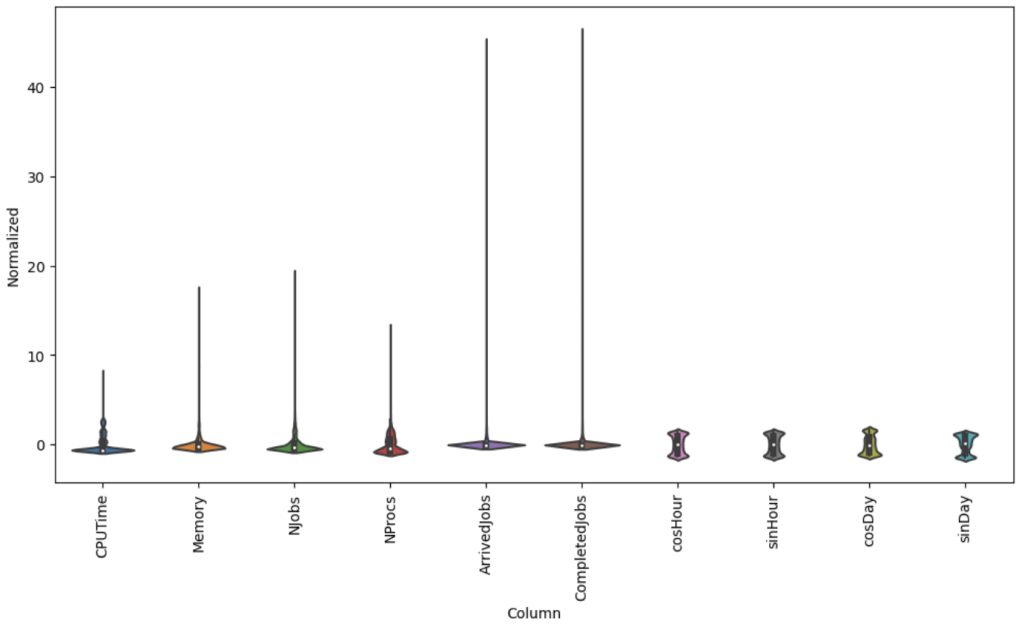
Διαισθητικά, ο παραπάνω τύπος εκφράζει τον αριθμό των τυπικών αποκλίσεων που απέχει μία τιμή από τον μέσο όρο της χρονοσειράς που ανήκει. Τα δεδομένα του AuverGrid φαίνεται να είναι πολύ πιο βοηθητικά για την προσαρμογή ενός μοντέλου. Στα δύο άλλα σύνολα, παρατηρείται συσπείρωση των τιμών στον μέσο όρο, αλλά και ελάχιστες τιμές οι οποίες φτάνουν έως τις 50 τυπικές αποκλίσεις διαφορά από τον μέσο όρο. Είναι πολύ πιθανό, τα σύνολα αυτά να μην μπορούν να προσφέρουν αξία στο πλαίσιο της παρούσας εργασίας.



Εικόνα 3.4: *Κατανομή κανονικοποιημένων δεδομένων χρονοσειράς AuverGrid*



Εικόνα 3.5: *Κατανομή κανονικοποιημένων δεδομένων χρονοσειράς DAS2*



Εικόνα 3.6: *Κατανομή κανονικοποιημένων δεδομένων χρονοσειράς SharcNet*

Επίσης παρουσιάζονται οι πίνακες συσχέτισης των χαρακτηριστικών των δεδομένων. Ξανά, το AuverGrid φαίνεται να είναι το πιο κατάλληλο για την προσαρμογή, παρουσιάζοντας μεγάλους συντελεστές συσχέτισης μεταξύ των CPUTime, Memory, NJobs και NProcs. Συγκεκριμένα για τα δύο τελευταία ο συντελεστής είναι ίσος με την μονάδα, καθώς όλες οι εργασίες στο συγκεκριμένο σύνολο χρησιμοποιούν από έναν πυρήνα, με αποτέλεσμα οι δύο στήλες να ταυτίζονται. Παράλληλα, βλέπουμε πως υπήρχε ποικιλία το προφίλ των εργασιών. Ο συντελεστής μεταξύ της χρήσης επεξεργαστή και μνήμης είναι 0.74, δείχνοντας την διαφορά εργασιών που απαιτούν περισσότερη ή λιγότερη μνήμη σε αναλογία με την επεξεργαστική δύναμη. Επιπλέον η συσχέτιση των αφιχθέντων και των ολοκληρωμένων εργασιών είναι 0.67,

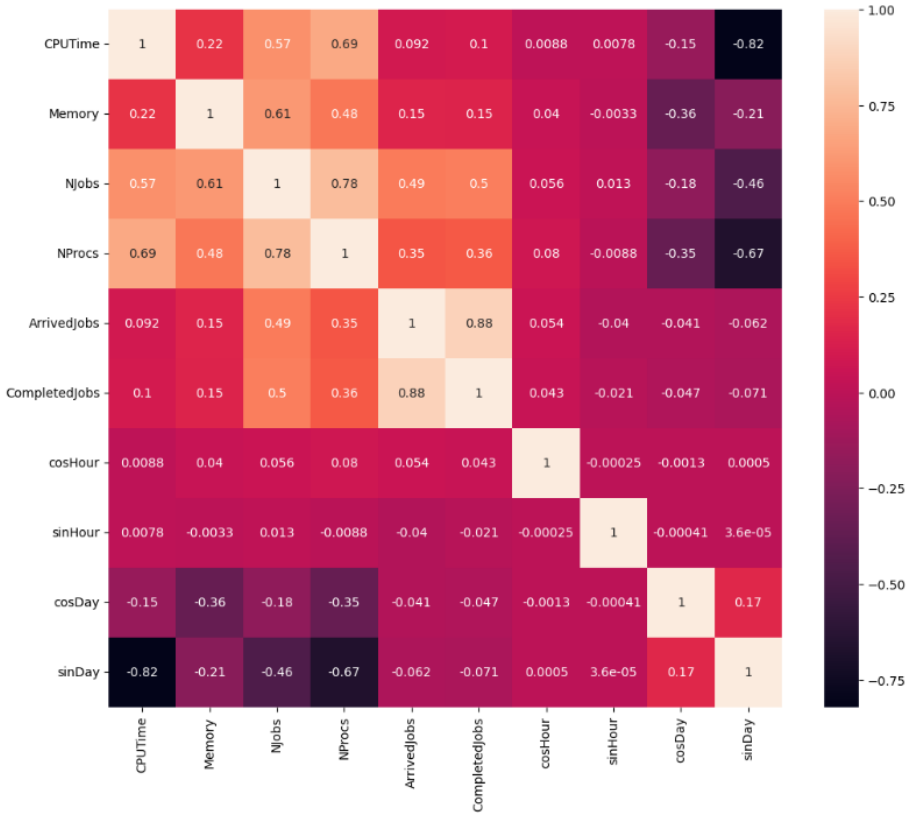
το οποίο εξηγείται από την άφιξη εργασιών οι οποίες ολοκληρώθηκαν την ίδια ώρα, αλλά και εργασιών οι οποίες μπορεί να έμειναν ενεργές για παραπάνω ώρες στο σύστημα.

Στο DAS2, παρατηρούνται συντελεστές άνω του 0.5 μόνο μεταξύ των NJobs, NProcs, ArrivedJobs και CompletedJobs. Σε αντίθεση με το προηγούμενο σύνολο δεδομένων, εδώ φαίνεται να φθάνουν μικρές εργασίες που ολοκληρώνονται κατά κανόνα σε λιγότερο από μια ώρα. Μάλιστα, ο λόγος που ο συντελεστής είναι ίσος με 0.99 και όχι με 1, πιθανώς να είναι πως κάποιες εργασίες έφτασαν στο τέλος μίας ώρας, όμως παρότι διήρκεσαν λίγα λεπτά, ολοκληρώθηκαν την επόμενη. Αξίζει να αναφερθεί πως πρόκειται για το μόνο εκ των τριών, στο οποίο κανένα χαρακτηριστικό δεν παρουσιάζει κάποια συσχέτιση με τα χαρακτηριστικά του χρόνου. Το γεγονός αυτό, επιβεβαιώνει πως είτε επικρατεί μια τυχαιότητα στις εργασίες που καταφθάνουν στο σύστημα, είτε έχει γίνει κακή καταγραφή στα log data.

Στο τελευταίο σύνολο δεδομένων, κυριαρχεί μια ενδιάμεση κατάσταση μεταξύ των υπολοίπων. Κύρια, άξια αναφοράς, σημεία αποτελούν ο μικρός συντελεστής συσχέτισης μεταξύ των CPUTime και Memory, καθώς και ο συντελεστής με τιμή -0.82 μεταξύ CPUTime και sinDay. Υπάρχει, δηλαδή, έντονη εποχιακότητα μέσα στο έτος, με αύξηση περίπου στο δεύτερο εξάμηνο του έτους, αφού το ημίτονο με περίοδο ενός χρόνου εκεί παρουσιάζει φθίνουσα συμπεριφορά.

Εικόνα 3.7: *Πίνακας συντελεστών συσχέτισης χαρακτηριστικών AuverGrid*

Εικόνα 3.8: *Πίνακας συντελεστών συσχέτισης χαρακτηριστικών DAS2*



Εικόνα 3.9: *Πίνακας συντελεστών συσχέτισης χαρακτηριστικών SharcNet*

Λόγω των κακών ενδείξεων που παρατηρήθηκαν, το DAS2 εξερευνήθηκε περαιτέρω. Ο πίνακας 3.8 παρουσιάζει τις τιμές του $25^{ου}$, του $50^{ου}$, του $75^{ου}$ και του $100^{ου}$ (μέγιστη τιμή) εκατοστημορίου των τιμών των τεσσάρων κυρίων χαρακτηριστικών. Επιβεβαιώνεται η υπόθεση πως τα δεδομένα δεν μπορούν να χρησιμοποιηθούν για το σκοπό της εργασίας. Το κατά κανόνα σχεδόν μηδενικό φορτίο παραπέμπει σε πολύ διαφορετικές μεθοδολογίες προβλέψεων χρονοσειρών που δεν απαντούν στο ερώτημα ¨πόσο θα είναι το φορτίο την επόμενη ώρα;¨, αλλά στο ¨πότε θα υπάρξει ξανά μη αμελητέο φορτίο;¨.

Πίνακας 3.8: *Πίνακας τιμών εκατοστημορίων DAS2*

	CPUTime	Memory	NJobs	NProcs
25%	0.19	36614.19	7	37
50%	1.19	1.892e+05	29	137
75%	3.16	6.234e+05	90	371
100%	2.42e+06	1.96e+09	7.009e+03	7.001e+03

Κεφάλαιο 4

Υλοποίηση & Αξιολόγηση Μοντέλων Πρόβλεψης

Στο κεφάλαιο παρουσιάζεται η διαδικασία υλοποίησης των υποψηφίων μοντέλων πρόβλεψης καθώς και τα αποτελέσματα της προσαρμογής αυτών στα τρία τελικά σύνολα δεδομένων. Για την προετοιμασία των δεδομένων και την υλοποίηση της εργασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python 3.11.4 και οι βιβλιοθήκες:

- Pandas, για I/O και χειρισμό των συνόλων δεδομένων,
- NumPy, για χειρισμό Πινάκων,
- Seaborn/MatPlotLib, για δημιουργία γραφημάτων,
- Sci-Kit Learn, για υλοποίηση των Παλινδρομικών μοντέλων,
- TensorFlow, για υλοποίηση Νευρωνικών Δικτύων,

Η εκπαίδευση έγινε χρησιμοποιώντας το πρώτο 65% των συνόλων για εκπαίδευση, το επόμενο 15% για επαλήθευση και το υπόλοιπο 20% για την εξέταση.

4.1 Αρχιτεκτονικές Μοντέλων

Η αρχιτεκτονική και οι υπερπαράμετροι των μοντέλων επιλέχθηκαν πειραματικά και με σύγκριση αποτελεσμάτων. Η διαδικασία επιλογής τους παραλείπεται και αναφέρονται μόνο οι τελικές αρχιτεκτονικές που επιλέχθηκαν. Μεταβλητές στόχους αποτελούν ο χρόνος επεξεργαστή CPUTime, η χρήση μνήμης Memory και ο αριθμός πυρήνων Nprocs.

4.1.1 Μοντέλα Παλινδρόμησης

Το πιο απλό μοντέλο της εργασίας είναι η Γραμμική Παλινδρόμηση και δεν έχει υπερπαραμέτρους προς ανάθεση. Η βιβλιοθήκη Sci-Kit Learn παρέχει υλοποιημένες κλάσεις για όλα τα γνωστά μοντέλα παλινδρόμησης. Οι κλάσεις αυτές χρησιμοποιήθηκαν και για τις πέντε παραλλαγές του κλασσικού μοντέλου γραμμικής παλινδρόμησης που εξετάσθηκαν.

Στην παλινδρόμηση Lasso υλοποιήθηκε αναζήτηση βέλτιστων υπερπαραμέτρων, με πιθανές τιμές για το a τις $[1, 100, 10^4, 10^6]$ και μετρική βελτιστοποίησης το μέσο απόλυτο σφάλμα (MAE).

Στην παλινδρόμηση Ridge υλοποιήθηκε αναζήτηση βέλτιστων υπερπαραμέτρων, με πιθανές τιμές για το a τις $[10^{-10}, 10^{-5}, 10^{-4}, 0.01, 0.1, 1]$ και μετρική βελτιστοποίησης το μέσο απόλυτο σφάλμα (MAE).

Στην παλινδρόμηση ElasticNet υλοποιήθηκε αναζήτηση βέλτιστων υπερπαραμέτρων, με πιθανές τιμές για το a τις $[1, 10, 20, 40, 80, 160, 240]$ και μετρική βελτιστοποίησης το μέσο απόλυτο σφάλμα (MAE).

Στην Πολυωνυμική παλινδρόμηση υλοποιήθηκε αναζήτηση βέλτιστων υπερπαραμέτρων, με πιθανές τιμές για τις δυνάμεις τις $[2, 3]$ και μετρική βελτιστοποίησης το μέσο απόλυτο σφάλμα (MAE).

4.1.2 Βαθύ Νευρωνικό Δίκτυο

Το βαθύ νευρωνικό δίκτυο που απέδωσε καλύτερα και στα τρία σύνολα δεδομένων αποτελούνταν από 3433 παραμέτρους με συνολικά έξι επίπεδα:

- Επίπεδο εισόδου με 8 units
- Κρυφό επίπεδο με 64 units
- Κρυφό επίπεδο με 32 units
- Κρυφό επίπεδο με 16 units
- Κρυφό επίπεδο με 8 units
- Επίπεδο εξόδου με 1 unit και γραμμική συνάρτηση ενεργοποίησης.

Χρησιμοποιήθηκε παράθυρο εισόδου μίας ώρας και ADAM optimizer. Για τα πραγματικά δεδομένα, χρησιμοποιήθηκε γραμμική συνάρτηση ενεργοποίησης, ενώ για τα συνθετικά η ReLU, σε όλα τα επίπεδα εκτός της εξόδου. Η εκπαίδευση έγινε σε batches των 32 βημάτων και χρησιμοποιήθηκε η κλάση EarlyStopping, με υπομονή 10 εποχών, για πρόωρο τερματισμό. Μέγιστος αριθμός εποχών τέθηκαν οι 50, όμως πάντα η εκπαίδευση ολοκληρωνόταν νωρίτερα.

4.1.3 Συνελικτικό Νευρωνικό Δίκτυο

Το συνελικτικό νευρωνικό δίκτυο που απέδωσε καλύτερα και στα τρία σύνολα δεδομένων αποτελούνταν από 10689 παραμέτρους με συνολικά πέντε επίπεδα:

- Συνελικτικό Φίλτρο Conv1D με 64 filters
- Κρυφό επίπεδο με 64 units
- Κρυφό επίπεδο με 32 units
- Κρυφό επίπεδο με 16 units
- Επίπεδο εξόδου με 1 unit και γραμμική συνάρτηση ενεργοποίησης.

Χρησιμοποιήθηκε παράθυρο εισόδου πέντε ωρών και ADAM optimizer. Για τα πραγματικά δεδομένα, χρησιμοποιήθηκε γραμμική συνάρτηση ενεργοποίησης, ενώ για τα συνθετικά η ReLU, σε όλα τα επίπεδα εκτός της εξόδου. Η εκπαίδευση έγινε σε batches των 32 βημάτων και χρησιμοποιήθηκε η κλάση EarlyStopping, με υπομονή 10 εποχών, για πρόωρο τερματισμό. Μέγιστος αριθμός εποχών τέθηκαν οι 50, όμως πάντα η εκπαίδευση ολοκληρωνόταν νωρίτερα.

4.1.4 Αναδρομικό Νευρωνικό Δίκτυο - LSTM

Το LSTM νευρωνικό δίκτυο που απέδωσε καλύτερα και στα τρία σύνολα δεδομένων αποτελούνταν από 6025 παραμέτρους με συνολικά έξι επίπεδα:

- Επίπεδο εισόδου με 8 units
- LSTM επίπεδο με 32 units
- Dropout επίπεδο με ρυθμό 0.2
- Κρυφό επίπεδο με 16 units
- Κρυφό επίπεδο με 8 units
- Επίπεδο εξόδου με 1 unit και γραμμική συνάρτηση ενεργοποίησης.

Χρησιμοποιήθηκε παράθυρο εισόδου πέντε ωρών και ADAM optimizer. Για τα πραγματικά δεδομένα, χρησιμοποιήθηκε γραμμική συνάρτηση ενεργοποίησης, ενώ για τα συνθετικά η ReLU, σε όλα τα επίπεδα εκτός της εξόδου. Η εκπαίδευση έγινε σε batches των 16 και χρησιμοποιήθηκε η κλάση EarlyStopping, με υπομονή 10 εποχών, για πρόωρο τερματισμό. Μέγιστος αριθμός εποχών τέθηκαν οι 50, όμως πάντα η εκπαίδευση ολοκληρωνόταν νωρίτερα.

Για την καλύτερη κατανόηση και σύγκριση, είναι καλό να αποφασιστεί ένα μοντέλο-βάσης (Baseline). Αυτό θα είναι το μοντέλο αφελούς πρόβλεψης (Naive), το οποίο προβλέπει πάντα πως η τιμή της χρονοσειράς την στιγμή t θα είναι ίση με την τιμή της την στιγμή $t-1$. Προφανώς, θεωρούμε το χρόνο εκπαίδευσής του μηδενικό. Οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση του μοντέλου είναι το μέσο σφάλμα (ME) και το μέσο απόλυτο σφάλμα (MAE) κανονικοποιημένα ως ποσοστό επί τοις εκατό του μέσου όρου του test set, για καλύτερη ερμηνευσιμότητα. Ο λόγος που προτιμήθηκε να γίνει η κανονικοποίηση αντί της χρήσης των ποσοστιαίων μέσων σφαλμάτων (MPE, MAPE), είναι η συχνή παρουσία μηδενικών τιμών, μη επιτρέποντας την διαίρεση. Η επιλογή των τριών μετρικών έγινε με σκοπό τον πλήρη προσδιορισμό των σφαλμάτων, ελέγχοντας ταυτόχρονα την προκατάληψη (ME) και την ακρίβεια (MAE) των μοντέλων. Τέλος, χρησιμοποιήθηκε και το percentage root mean squared error (PRMSE), το οποίο είναι αυστηρότερο στα μεγάλα σφάλματα, δίνοντας περισσότερες πληροφορίες για την πρόβλεψη των απότομων αυξήσεων στο φορτίο.

4.2 Αποτελέσματα Προβλέψεων CPUTime

4.2.1 Συνθετικά δεδομένα

Στα συνθετικά δεδομένα, το baseline MAE που έπρεπε να ξεπεραστεί ήταν 2.045% και το baseline PRMSE 6.513%. Όλα τα μοντέλα απέδωσαν καλύτερα σε αυτή την περίπτωση. Τα νευρωνικά δίκτυα έδειξαν να πηγαίνουν λίγο καλύτερα από τις παλινδρομήσεις. Συγκεκριμένα, η γραμμική παλινδρόμηση πέτυχε τις καλύτερες μετρικές στην κατηγορία της, με MAE 1.28% και PRMSE 3.562%. Στα νευρωνικά δίκτυα παρατηρήθηκαν πόλυ καλές επιδόσεις, με καλύτερη απόδοση από την γραμμική παλινδρόμηση. Επιλογή για ένα αληθινό σύστημα με παρόμοια δεδομένα, θα ήταν ένα Βαθύ Νευρωνικό Δίκτυο, με χρόνο προσαρμογής 29 δευτερόλεπτα, ο οποίος επιτρέπει πολλές δοκιμές για την εύρεση των βέλτιστων υπερπαραμέτρων. Αξίζει επίσης να αναφερθεί η τάξεων μεγέθους διαφορά στους χρόνους προσαρμογής των Lasso και ElasticNet.

Πίνακας 4.1: *Μετρικές αξιολόγησης προβλέψεων του CPUTime για συνθετικά δεδομένα*

	MAE (%)	ME (%)	PRMSE (%)	Χρόνος Εκπαίδευσης (s)
Baseline	2.045	0.002	6.513	0
LR	2.02	0	6.456	0.066
Lasso	1.28	0	3.562	7394.46
Ridge	1.28	0	4.291	0.441
ElasticNet	2.28	0	7.213	11461.59
Power	2.02	0	6.456	2.343
DNN	0.84	-0.083	2.868	29.846
CNN	0.855	-0.198	2.881	37.412
LSTM	0.88	0.092	2.979	84.401

4.2.2 AuverGrid

Το AuverGrid αποτέλεσε παράδειγμα πως η πολυπλοκότητα δεν συνεπάγεται καλύτερη επίδοση. Το baseline MAE που έπρεπε να ξεπεραστεί ήταν 3.19% και το baseline PRMSE 1.556%. Εντύπωση δημιουργεί το ήδη χαμηλό σφάλμα, το οποίο μαρτυρά αργές μεταβολές στο φορτίο, και πολύ μεγάλη συσχέτιση της τιμής της προηγούμενης ώρας στην τωρινή. Τα νευρωνικά δίκτυα σε αυτή την περίπτωση δεν είχαν καλύτερη επίδοση από τις παλινδρομήσεις, ούτε από το Baseline μοντέλο. Πρέπει να επισημανθεί, πως δοκιμάστηκαν πολλές διαφορετικές αρχιτεκτονικές για τα νευρωνικά δίκτυα, και αυτές ήταν οι καλύτερες επιδόσεις που σημειώθηκαν. Συγκεκριμένα, η γραμμική παλινδρόμηση και η Ridge πέτυχαν τις καλύτερες μετρικές στην κατηγορία τους, συνυπολογίζοντας και τον χρόνο εκπαίδευσης. Ξανά, οι Lasso και ElasticNet παρουσιάζουν αξιοσημείωτα αποτελέσματα. Τόσο για τον ασύμφορο χρόνο εκτέλεσής τους, όσο και για τα μεγάλα τετραγωνικά σφάλματα. Το γεγονός αυτό, μπορεί εν μέρει να οφείλεται στο κριτήριο αξιολόγησης κατά την επιλογή των βέλτιστων υπερπαραμέτρων στο GridSearch, το οποίο ήταν το MAE.

Πίνακας 4.2: *Μετρικές αξιολόγησης προβλέψεων του CPUTime για AuverGrid*

	MAE (%)	ME (%)	PRMSE (%)	Χρόνος Εκπαίδευσης (s)
Baseline	3.194	0	1.556	0
LR	2.889	0	1.494	0.023
Lasso	2.888	0	31.029	612.67
Ridge	2.956	0	1.492	0.218
ElasticNet	2.905	0	31.027	1991.64
Power	2.889	0	1.494	0.388
DNN	3.41	0.6	1.527	1.72
CNN	3.93	1.52	1.64	3.56
LSTM	4.54	0.68	14.822	15.36

4.2.3 SharcNet

Το τελευταίο διαθέσιμο σύνολο και η απρόσμενα κακή απόδοση των τεχνικών πρόβλεψης των τιμών του, στάθηκε αφορμή για περαιτέρω έρευνα και εμβάθυνση στην λειτουργία των μοντέλων και των μετρικών. Το baseline MAE που έπρεπε να ξεπεραστεί ήταν 1.77%(!) και το baseline PRMSE 2.701%. Όλα τα μοντέλα απέδωσαν χειρότερα στο MAE. Το μόνο μοντέλο που πλησίασε στην ακρίβεια ήταν το DNN. Συγκεκριμένα, είχε MAE 2.32% και PRMSE 1.796%, ξεπερνώντας αρκετά το Baseline στην μείωση των σφαλμάτων σε μεγάλες τιμές. Η επίδοση αυτή ίσως να το καθιστά προτιμότερο μοντέλο σε περιπτώσεις όπου η πρόβλεψη των μεγάλων αυξήσεων φορτίου είναι σημαντικότερη για το σύστημά μας.

Πίνακας 4.3: *Μετρικές αξιολόγησης προβλέψεων του CPUTime για SharcNet*

	MAE (%)	ME (%)	PRMSE (%)	Χρόνος Εκπαίδευσης (s)
Baseline	1.77	0	2.701	0
LR	3.174	0	2.321	0.011
Lasso	3.788	0	33.147	1294.2
Ridge	4.512	0	2.512	0.274
ElasticNet	3.176	0	33.313	2150.73
Power	3.174	0	2.321	0.879
DNN	2.32	0.04	1.796	4.95
CNN	4.21	-0.84	1.64	7.64
LSTM	12.23	-4.82	15.514	12.171

4.3 Αποτελέσματα Προβλέψεων Memory

4.3.1 Συνθετικά Δεδομένα

Το baseline MAE που έπρεπε να ξεπεραστεί ήταν 1.239% και το baseline PRMSE 12.49%. Η παλινδρόμηση Ridge πέτυχε τις καλύτερες μετρικές στην κατηγορία της, με MAE 1.075% και PRMSE 10.371%. Στα νευρωνικά δίκτυα παρατηρήθηκαν πολύ καλές επιδόσεις, με καλύτερη απόδοση από το Ridge. Επιλογή για ένα αληθινό σύστημα με παρόμοια δεδομένα, θα ήταν ένα Βαθύ Νευρωνικό Δίκτυο, με χρόνο προσαρμογής 24 δευτερόλεπτα, MAE 0.542% και PRMSE 5.822%.

Πίνακας 4.4: *Μετρικές αξιολόγησης προβλέψεων του Memory για συνθετικά δεδομένα*

	MAE (%)	ME (%)	PRMSE (%)	Χρόνος Εκπαίδευσης (s)
Baseline	1.239	-0.002	12.49	0
LR	1.233	0	12.276	0.006
Ridge	1.075	0	10.371	0.441
Power	1.218	0	12.235	2.343
DNN	0.542	0.073	5.822	24.197
CNN	0.559	-0.044	5.958	22.492
LSTM	0.549	-0.015	5.844	57.239

4.3.2 AuverGrid

Το baseline MAE που έπρεπε να ξεπεραστεί ήταν 7.931% και το baseline PRMSE 2.77%. Η γραμμική παλινδρόμηση πέτυχε ξανά τις καλύτερες μετρικές στην κατηγορία της, με MAE 1.075% και PRMSE 10.371%. Στα νευρωνικά δίκτυα παρατηρήθηκαν καλές επιδόσεις, όμως με χειρότερη απόδοση από την γραμμική παλινδρόμηση, η οποία είναι και η επιλογή για τα συγκεκριμένα δεδομένα.

Πίνακας 4.5: *Μετρικές αξιολόγησης προβλέψεων του Memory για AuverGrid*

	MAE (%)	ME (%)	PRMSE (%)	Χρόνος Εκπαίδευσης (s)
Baseline	6.643	-0.007	3.748	0
LR	5.544	0	2.794	0.002
Ridge	6.003	0	2.846	0.441
Power	5.544	0	2.794	2.343
DNN	6.675	0.673	3.094	35.688
CNN	6.092	0.75	3.032	47.466
LSTM	6.089	-0.067	3.024	70.059

4.4 Αποτελέσματα Προβλέψεων NProcs

4.4.1 Συνθετικά Δεδομένα

Το baseline MAE που έπρεπε να ξεπεραστεί ήταν 3.322% και το baseline PRMSE 12.577%. Η παλινδρόμηση Ridge πέτυχε τις καλύτερες μετρικές στην κατηγορία της, με MAE 2.507% και PRMSE 9.01%. Στα νευρωνικά δίκτυα παρατηρήθηκαν πολύ καλές επιδόσεις, ξεπερνώντας ξανά τις παλινδρομήσεις. Επιλογή για ένα αληθινό σύστημα με παρόμοια δεδομένα, θα ήταν ένα Βαθύ Νευρωνικό Δίκτυο, με χρόνο προσαρμογής 18 δευτερόλεπτα, MAE 2.224% και PRMSE 5.822%.

Πίνακας 4.6: *Μετρικές αξιολόγησης προβλέψεων του NProcs για συνθετικά δεδομένα*

	MAE (%)	ME (%)	PRMSE (%)	Χρόνος Εκπαίδευσης (s)
Baseline	3.322	-0.001	12.577	0
LR	2.66	0	9.63	0.007
Ridge	2.507	0	9.01	0.441
Power	2.656	0	9.623	2.343
DNN	2.224	0.312	8.451	18.04
CNN	2.237	-0.006	8.51	13.349
LSTM	2.291	0.184	8.697	35.946

4.4.2 AuverGrid

Τέλος, το baseline MAE που έπρεπε να ξεπεραστεί ήταν 6.335% και το baseline PRMSE 3.461%. Η παλινδρόμηση Ridge πέτυχε τις καλύτερες μετρικές στην κατηγορία της, με MAE 5.161% και PRMSE 2.309%. Στα νευρωνικά δίκτυα παρατηρήθηκαν καλές επιδόσεις, με εξαίρεση τα LSTM, που δεν μπόρεσαν να ξεπεράσουν ούτε το baseline. Επιλογή και πάλι για το AuverGrid θα είναι παλινδρόμηση, όμως αυτή την φορά με την βελτιστοποίηση Ridge.

Πίνακας 4.7: *Μετρικές αξιολόγησης προβλέψεων του NProcs για AuverGrid*

	MAE (%)	ME (%)	PRMSE (%)	Χρόνος Εκπαίδευσης (s)
Baseline	6.335	-0.004	3.461	0
LR	5.299	-0.004	2.317	0.002
Ridge	5.161	-0.001	2.309	0.441
Power	5.299	-0.004	2.317	2.343
DNN	7.845	4.566	3.338	7.578
CNN	6.269	0.151	3.066	15.947
LSTM	11.861	6.252	5.064	39.087

4.5 Σχόλια

Σε αυτή την ενότητα παρουσιάζονται καίρια για την εργασία σχόλια, σχετικά με τα τελικά αποτελέσματα και την διαφορά μεταξύ των πραγματικών και των συνθετικών δεδομένων. Τα θέματα προς σχολιασμό συνδέονται μεταξύ τους και έχουν να κάνουν κυρίως με την απλότητα και την γραμμικότητα των πραγματικών δεδομένων. Παρατηρήθηκε πως τα νευρωνικά δίκτυα και ειδικά τα LSTM δεν απέδωσαν ως αναμενόμενα.

1. **Υπεροχή Γραμμικής Παλινδρόμησης στο AuverGrid**: Όπως αναφέρθηκε προηγουμένως, η πολυπλοκότητα των μεθόδων πρόβλεψης δεν είναι πάντοτε ανάλογη των επιδόσεων τους. Στην περίπτωσή μας, παρατηρούνται πολύ απλές και πιθανώς γραμμικές σχέσεις μεταξύ των συγκεκριμένων δεδομένων, σχεδόν καθιστώντας την εφαρμογή νευρωνικών δικτύων υπερβολική. Το γεγονός αυτό αποδεικνύεται από την υπεροχή της Γραμμικής Παλινδρόμησης και των παραλλαγών της. Παρ' όλα αυτά, η επιλογή ενός μόνο perceptron δεν θα έπρεπε να μας παραξενεύει ούτε να αμφισβητείται. Άλλωστε, σύμφωνα με την αρχή του Ξυραφιού του Όκαμ, ¨Η απλούστερη εξήγηση για ένα πρόβλημα είναι συνήθως η καλύτερη¨ [29]. Εφαρμόζοντας την αρχή στο θέμα μας, εάν στα δεδομένα μπορεί να προσαρμοστεί το απλούστερο μοντέλο μηχανικής μάθησης και να τα προβλέψει ικανοποιητικά, ίσως αποτελεί και την καλύτερη επιλογή.

2. **Επίδοση LSTM**: Τα μοντέλα LSTM χρησιμοποιούνται ευρέως στην πρόβλεψη χρονοσειρών λόγω της ικανότητάς τους να αντιλαμβάνονται πολύπλοκες χρονικές εξαρτήσεις. Ωστόσο, στο πλαίσιο της εργασίας, αυτά τα μοντέλα, παρά το φημισμένο τους δυναμικό, επέδειξαν σημαντικά χειρότερη απόδοση σε σύγκριση με πιο παραδοσιακές μεθόδους, όπως η γραμμική παλινδρόμηση. Αυτό το απρόσμενο αποτέλεσμα μπορεί να αποδοθεί σε αρκετούς παράγοντες. Καταρχάς, τα πραγματικά δεδομένα που χρησιμοποιήθηκαν παρουσιάζουν αρκετά απλή δομή, με περιορισμένες χρονικές εξαρτήσεις και μια απλή σχέση μεταξύ των χαρακτηριστικών εισόδου και της μεταβλητής στόχου. Δεύτερον, το χρονικό διάστημα που κάλυπταν τα δεδομένα μας ήταν σχετικά σύντομο, κάτι που μπορεί να μην παρείχε αρκετή χρονική συμπεριφορά για τα μοντέλα LSTM να ξεπεράσουν. Όπως παρατηρήθηκε, στα συνθετικά δεδομένα το φαινόμενο αυτό περιορίστηκε. Τέλος, η μικρή διαστατικότητα των δεδομένων συνέβαλε περαιτέρω στην ανικανότητα του μοντέλου να εξάγει σημαντικά πρότυπα, καθώς η πολυπλοκότητα των αρχιτεκτονικών LSTM δεν είχε επαρκή βάση σε μια τέτοια ρύθμιση. Αυτές οι παρατηρήσεις υποδεικνύουν ότι, σε συγκεκριμένα σενάρια με απλά δεδομένα και μικρές χρονικές περιόδους, η εφαρμογή πιο σύνθετων νευρωνικών δικτύων όπως τα LSTM ενδέχεται να μην προσφέρει την αναμενόμενη βελτίωση έναντι πιο απλών προσεγγίσεων.

3. **Δεδομένα SharcNet**: Το εξαιρετικά χαμηλό σφάλμα του Baseline μοντέλου, καθώς και η αδυναμία των υπόλοιπων μοντέλων να το πλησιάσουν έγιναν αίτια για εμβάθυνση της ανάλυσης του συνόλου δεδομένων. Παρατηρήθηκε πως κατά το πρώτο σχεδόν μισό χρονικό διάστημα (πέντε μήνες), το φορτίο του SharcNet είναι τάξεις μεγέθους μικρότερο από το υπόλοιπο μισό. Κλασσική μέθοδος αντιμετώπισης αποτελεί η διαγραφή των συγκεκριμένων τιμών και η χρήση των υπολοίπων για την εκπαίδευση και

την αξιολόγηση. Η μέθοδος δοκιμάστηκε, όμως τα αποτελέσματά της δεν διέφεραν κατά πολύ από αυτά που παρουσιάστηκαν. Το γεγονός αυτό μπορεί να αποδοθεί στο πλέον υπερβολικά μικρό μέγεθος των δεδομένων, το οποίο δεν επιτρέπει στα μοντέλα να αναγνωρίσουν μοτίβα μέσα στις τιμές. Το SharcNet, λοιπόν, απορρίφθηκε και η μελέτη συνεχσίστηκε για τα υπολειπόμενα δύο μοντέλα.

4. **Απόρριψη Παλινδρομήσεων Lasso και ElasticNet**: Οι δύο παλινδρομήσεις δεν έδειξαν σε καμία περίπτωση την καλύτερη επίδοση, ενώ ταυτόχρονα παρουσίασαν χρόνους προσαρμογής μεγαλύτερους των 30 λεπτών. Τόσο μεγάλοι χρόνοι δυσκολεύουν τον πειραματισμό και την επιλογή διαφορετικών μοντέλων για κάθε εξεταζόμενο σύστημα, δηλαδή έναν από τους στόχους της παρούσας εργασίας. Σε συνδυασμό με την μέτρια επίδοσή τους, οδήγησε στην απόφαση να απορριφθούν και να μην εφαρμοστούν στα επόμενα χαρακτηριστικά του φορτίου.

Κεφάλαιο 5

Απόδειξη Σκοπιμότητας Υλοποίησης Συστήματος Διαχείρισης

Τελευταίο βήμα της εργασίας αποτελεί η μελέτη της αυτοματοποιημένης διαχείρισης πόρων των συστημάτων. Για την υλοποίηση ενός τέτοιου συστήματος υπάρχουν δύο επιλογές.

Η πρώτη επιλογή, είναι η χρήση ενός μοντέλου ενισχυτικής μάθησης, το οποίου θα μπορούσε να ελέγχει την αλλαγή (ή την μη αλλαγή) του αριθμού των επεξεργαστών ή/και των μνημών που διαθέτει ενεργό το σύστημα. Ως είσοδος, θα δινόταν η πρόβλεψη της κατάστασης του συστήματος την επόμενη ώρα, καθώς και η τρέχουσα κατάσταση του φορτίου. Το μοντέλο θα δεχόταν επιβράβευση για την μείωση του χαμένου (εργασίες που δεν εξυπηρετήθηκαν) και για την μείωση των ανεκμετάλλευτων πόρων. Περισσότερα στοιχεία για την υλοποίηση αλλά και την συμπεριφορά της συγκεκριμένης επιλογής παρουσιάζονται στην αναφερθείσα στην εισαγωγή διπλωματική εργασία ¨Αυτοματοποιημένη Διαχείριση Πόρων με Χρήση Τεχνικών Πρόβλεψης Χρονοσειρών και Βαθιά Ενισχυτική Μάθηση¨ [9].

Δεύτερη επιλογή, αποτελεί η εύρεση μίας πολιτικής διαχείρισης των πόρων. Ένα σενάριο πολιτικής, θα ήταν να χρησιμοποιούνται οι προβλέψεις των μοντέλων. Αν προβλεφθεί αύξηση του φορτίου, η αναγκαία ρύθμιση μπορεί να γίνει απευθείας αφού οι εργασίες που είναι ενεργές την τρέχουσα ώρα δε θα επηρεασθούν. Στην πρόβλεψη μείωσης του φορτίου, παρουσιάζεται το πρόβλημα πως, οι πόροι δεν μπορούν να μειωθούν, γιατί οι ενεργές εργασίες θα χάσουν τους πόρους που χρησιμοποιούν. Μία ενδεικτική λύση για την αντιμετώπιση του προβλήματος, θα ήταν η σταδιακή αποδέσμευση πόρων, ακολουθώντας την ολοκλήρωση των ενεργών εργασιών. Επίσης, θα μπορούσε να εισαχθεί ένα περιθώριο ασφαλείας σε μορφή ποσοστού, όπου οι πόροι που θα διατίθεντο από το σύστημα, θα ήταν τα αποτελέσματα των προβλέψεων, αυξημένα κατ΄ αυτό το περιθώριο.

Στο πλαίσιο της παρούσας εργασίας δεν υλοποιήθηκε κάποιο τέτοιο σύστημα, καθώς η προσομοίωση της λειτουργίας του με σκοπό την εξέταση της απόδοσής του απαιτεί την δημιουργία και την αποστολή ροών εργασιών σε ένα πραγματικό cluster για μεγάλο χρονικό διάστημα. Κύριο αντικείμενο προσοχής της παρούσας εργασίας ήταν η εξέταση βραχυπρόθεσμης πρόβλεψης του υπολογιστικού φορτίου σε πραγματικά δεδομένα, ώστε να εξεταστεί αν ένα τέτοιο σύστημα είναι εφικτό και σκόπιμο να υλοποιηθεί (Proof of Concept). Στο κεφάλαιο αυτό, λοιπόν, αναλύεται η διαδικασία που ακολουθήθηκε κατά την εξερεύνηση των οφελών ενός ιδανικού συστήματος αυτόματης διαχείρισης υπολογιστικών πόρων.

5.1 Υλοποίηση Πειράματος

Στην εξέταση του συστήματος χρησιμοποιείται η παραδοχή πως οι αυξομειώσεις των πόρων γίνεται με βέλτιστο τρόπο ακριβώς στην αρχή της ώρας.

Για την προσομοίωση χρησιμοποιήθηκε το test set των δεδομένων που αναλύθηκε στο προηγούμενο κεφάλαιο. Άρα, οι προβλέψεις έχουν την ακρίβεια που έχει αναφερθεί για το εκάστοτε επιλεγμένο μοντέλο. Στο σημείο αυτό, χρειάζεται να ορισθούν - εισαχθούν οι έννοιες των χρησιμοποιούμενων πόρων (UR - Used Resources), ώρες χαμένου φορτίου (LOL - Loss Of Load) και χαμένο φορτίο (QNS - Quantity Not Served). Οι δύο τελευταίες έννοιες χρησιμοποιούνται σε συστήματα ενέργειας, για την αξιολόγηση επάρκειας ισχύος σε ένα ενεργειακό δίκτυο [30]. Με αυτές θα αξιολογηθεί η επάρκεια του συστήματος.

Κατά την προσομοίωση του συστήματος, για να εξασφαλισθεί μεγαλύτερη κάλυψη των αναγκών και να μειωθεί το ρίσκο της ανεπάρκειας, ορίστηκε ένα περιθώριο ασφαλείας (safety factor στην πολιτική δέσμευσης πόρων. Συγκεκριμένα, δεσμεύονται N% επιπλέον πόροι σε σχέση με την πρόβλεψη του εκάστοτε μοντέλου. Η λειτουργία αυτή προστέθηκε ως υλοποίηση ενός πιθανού QoS agreement. Για παράδειγμα, κάποιος πάροχος μπορεί να είναι πιο ελαστικός σε ότι αφορά την αναμονή των εργασιών που φθάνουν στο σύστημα. Σε αυτή την περίπτωση, η ποινή για μια πιθανή ανεπάρκεια θα ήταν μικρότερη και θα βοηθούσε την επιλογή χρήσης μικρότερου περιθωρίου ασφάλειας κλίνοντας υπέρ της μείωσης της κατανάλωσης έναντι της πολύ υψηλής ποιότητας εξυπηρέτησης. Συνολικά θα εξετασθούν δύο περιπτώσεις πολιτικής QoS, μία που θα αφορά έναν πάροχο με αυστηρή πολιτική, και μια που θα αφορά έναν πάροχο με αυστηρή πολιτική.

Όπως και κατά την εξέταση των μοντέλων πρόβλεψης, είναι αναγκαία κάποια πολιτική που θα αποτελέσει την βάση για την σύγκριση των πολιτικών διαχείρισης των πόρων. Ως βάση επιλέχθηκε η πολιτική που χρησιμοποιείται από τα περισσότερα Clusters σήμερα, δηλαδή η διανομή σταθερών πόρων. Συγκεκριμένα για κάθε χαρακτηριστικό του φορτίου κατανεμήθηκαν σταθερά πόροι ίσοι με την μέγιστη τιμή του χαρακτηριστικού. Φυσικά, οι προβλέψεις φράζονται επίσης σε αυτόν τον αριθμό, καθώς δεν θα ήταν δυνατόν το σύστημα να διαθέσει παραπάνω. Στο πείραμα, οι πόροι που κατανεμήθηκαν συγκρίνονται με τις πραγματικές ανάγκες. Αν είναι παραπάνω από όσοι χρειάστηκαν, η περίσσεια φορτίου προστίθεται στο UR. Αν υπήρξε ανεπάρκεια, τότε το φορτίο που δεν εξυπηρετήθηκε προστίθεται στο QNS και το LOL αυξάνεται κατά μία μονάδα. Η αξιολόγηση της επάρκειας θα γίνει με βάση την μείωση κατά ποσοστό του UR σε σχέση με την πολιτική σταθερής κατανομής, το LOL, και το ποσοστό του QNS επί του συνολικού φορτίου.

5.2 Αποτελέσματα - Αυστηρή Πολιτική QoS

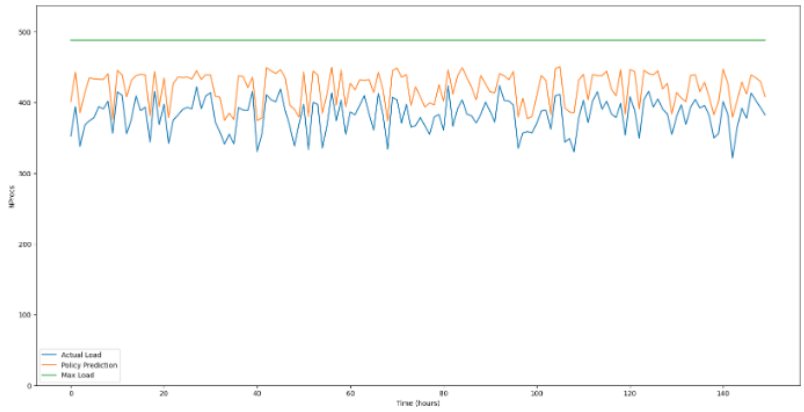
Ως ανώτατο όριο χαμένου φορτίου για την αυστηρή πολιτική ποιότητας εξυπηρέτησης χρησιμοποιήθηκε το 0.5% του συνολικού. Κατά την προσομοίωση τέθηκε safety factor 10% για τα συνθετικά δεδομένα και 20% για τα δεδομένα του Auvergrid. Η επιλογή αυτή έγινε γιατί στα συνθετικά δεδομένα παρατηρήθηκαν πολύ πιο ακριβείς μετρήσεις.

5.2.1 Συνθετικά Δεδομένα

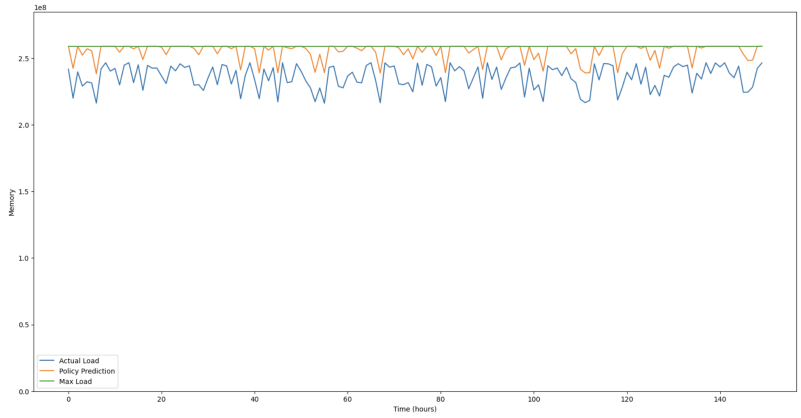
Για τα συνθετικά δεδομένα, η αύξηση δεν παρουσιάζει τα αναμενόμενα αποτελέσματα. Παρατηρείται μείωση κατά μόνο 0.4% στην χρήση πόρων μνήμης, ενώ για 5% μείωση σε αριθμό επεξεργαστών θυσιάζεται 0.019% των απαιτήσεων. Το γεγονός αυτό μπορεί να αποδοθεί στην διαφορετική μορφολογία των δεδομένων. Όπως θα δούμε παρακάτω, στο AuverGrid, παρατηρούνται πολύ μεγαλύτερες διακυμάνσεις στο φορτίο, δίνοντας χώρο για μεγαλύτερη βελτίωση. Για παράδειγμα, η μέγιστη τιμή επεξεργαστών στο AuverGrid είναι 3 φορές μεγαλύτερη από την μέση τιμή της χρονοσειράς. Αν ως μέτρο ανάθεσης των πόρων γινόταν αυτός ο λόγος, στα συνθετικά δεδομένα θα υπήρχε βελτίωση 65-70% στην χρήση πόρων.

Πίνακας 5.1: *Αξιολόγηση διαχείρισης πόρων με αυστηρή πολιτική για συνθετικά δεδομένα*

	UR (%)	LOL (H)	QNS (%)
CPUTime	10.38	0	0.0000
Memory	0.39	0	0.0000
NProcs	5.04	3	0.0019



Εικόνα 5.1: *Πολιτικές διαχείρισης επεξεργαστικών πόρων με αυστηρή πολιτική για συνθετικά δεδομένα*



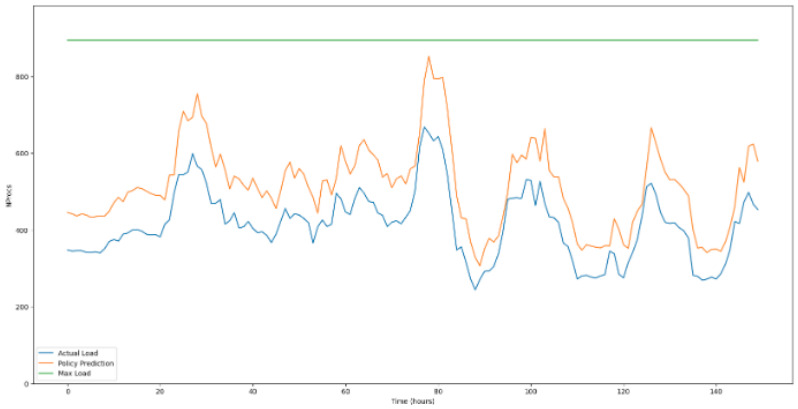
Εικόνα 5.2: *Πολιτικές διαχείρισης πόρων μνήμης με αυστηρή πολιτική για συνθετικά δεδομένα*

5.2.2 AuverGrid

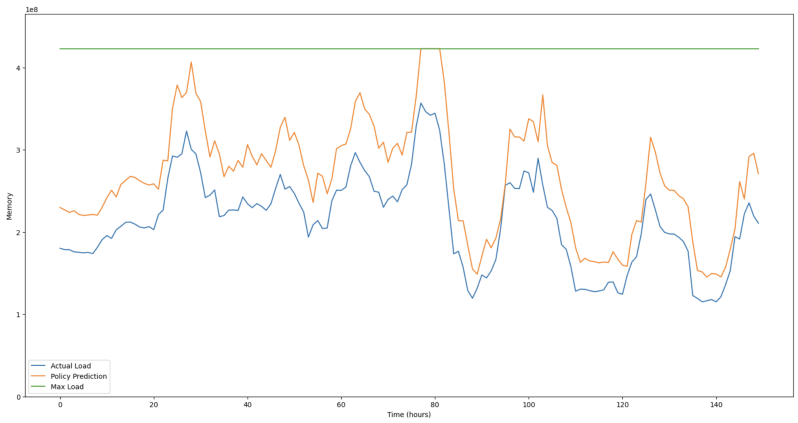
Τα αποτελέσματα για τα πραγματικά δεδομένα είναι πολύ καλύτερα και μπορούν να χαρακτηριστούν ελπιδοφόρα, αφού βλέπουμε μείωση του συνολικού φορτίου κατά 60-64%. Παρατηρούνται μικρές απώλειες φορτίου, μικρότερες του 0.3% σε κάθε περίπτωση. Οι απώλειες αυτές, όμως, δεν αποτελούν λόγο απόρριψης της υλοποίησης ενός συστήματος αυτόματης διαχείρισης, καθώς είναι αρκετά μικρές ώστε να μπορούν να αντιμετωπισθούν και να μηδενιστούν. Οι τρόποι με τους οποίους μπορεί να γίνει αυτό θα αναλυθούν στις μελλοντικές επεκτάσεις.

Πίνακας 5.2: *Αξιολόγηση διαχείρισης πόρων με αυστηρή πολιτική για AuverGrid*

	UR (%)	LOL (H)	QNS (%)
CPUTime	63.74	21	0.1179
Memory	60.61	38	0.2880
NProcs	61.93	32	0.1707



Εικόνα 5.3: *Πολιτική διαχείρισης επεξεργαστικών πόρων με αυστηρή πολιτική για AuverGrid*



Εικόνα 5.4: *Πολιτική διαχείρισης πόρων μνήμης με αυστηρή πολιτική για AuverGrid*

5.3 Αποτελέσματα - Ελαστική Πολιτική QoS

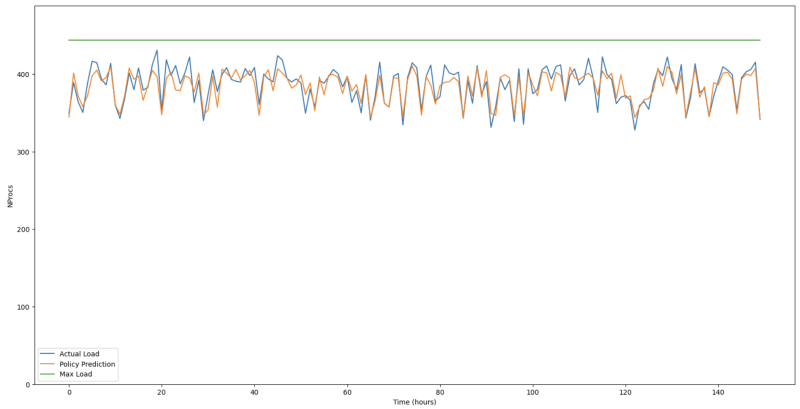
Ως ανώτατο όριο χαμένου φορτίου για την αυστηρή πολιτική ποιότητας εξυπηρέτησης χρησιμοποιήθηκε το 2% του συνολικού. Κατά την προσομοίωση τέθηκε safety factor για τα 5% για τα δεδομένα του Auvergrid, ενώ δεν χρησιμοποιήθηκε κάποιο περιθώριο για τα συνθετικά δεδομένα.

5.3.1 Συνθετικά Δεδομένα

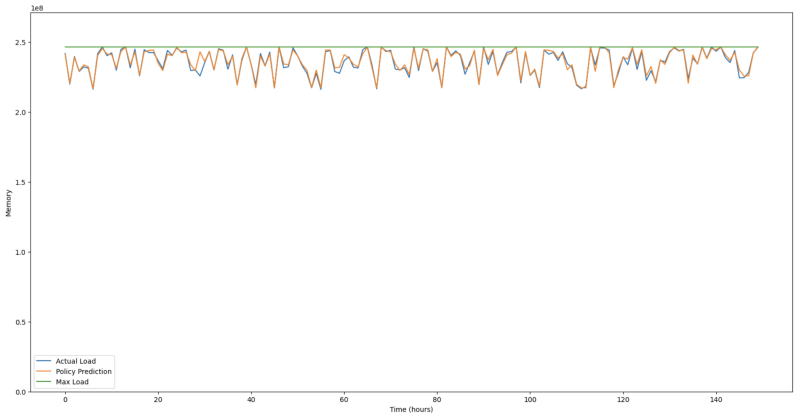
Όπως αναμενόταν, παρουσιάζεται μεγαλύτερη εξοικονόμηση πόρων και μεγαλύτερη ποσότητα μη εξυπηρετημένου φορτίου. Συγκεκριμένα, η μείωση των χρησιμοποιούμενων πόρων μνήμης ανέβηκε στο 4.4%. Παρατηρούνται επίσης εξαιρετικά μεγάλες τιμές LOL, όμως το QNS δεν συμβαδίζει με αυτό. Αυτό συμβαίνει γιατί, ακόμα και τις στιγμές που υπάρχει under-provisioning, τα σφάλματα είναι τόσο μικρά που δεν επιτρέπουν να χαθεί μεγάλη ποσότητα φορτίου. Το γεγονός αυτό μπορεί να διαπιστωθεί και από τις ακόλουθες εικόνες, όπου βρίσκονται σημεία στα οποία η πολιτική έχει αναθέσει λιγότερους πόρους από το πραγματικό φορτίο, όμως η διαφορά είναι πολύ μικρή.

Πίνακας 5.3: *Αξιολόγηση διαχείρισης πόρων με ελαστική πολιτική για συνθετικά δεδομένα*

	UR (%)	LOL (H)	QNS (%)
CPUTime	18.04	3013	0.53
Memory	4.40	2402	0.23
NProcs	13.58	2844	1.27



Εικόνα 5.5: *Πολιτικές διαχείρισης επεξεργαστικών πόρων με ελαστική πολιτική για συνθετικά δεδομένα*



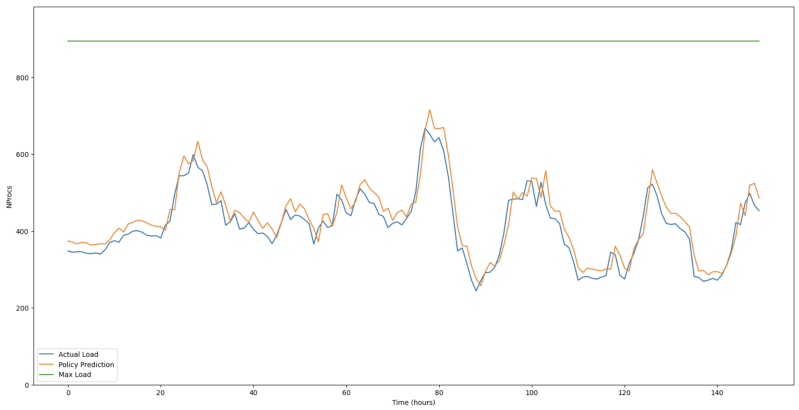
Εικόνα 5.6: *Πολιτικές διαχείρισης πόρων μνήμης με ελαστική πολιτική για συνθετικά δεδομένα*

5.3.2 Auvergrid

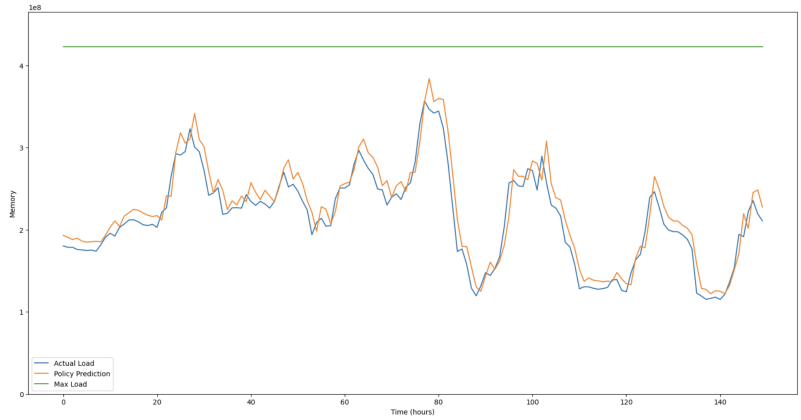
Όπως προηγουμένως, τα αποτελέσματα για τα πραγματικά δεδομένα είναι ικανοποιητικά, αφού παρουσιάζεται μείωση του συνολικού φορτίου κατά 67-69%. Παρατηρούνται μεγαλύτερες απώλειες φορτίου σε σχέση με την αυστηρή πολιτική αλλά και με τα συνθετικά δεδομένα, όμως μικρότερες του 1.5% σε κάθε περίπτωση. Βλέπουμε πως, με την εφαρμογή μιας πιο ελαστικής πολιτικής QoS, μπορεί να υπάρξει μεγαλύτερο ενεργειακό όφελος για τους παρόχους υπηρεσιών Cloud.

Πίνακας 5.4: *Αξιολόγηση διαχείρισης πόρων με ελαστική πολιτική για AuverGrid*

	UR (%)	LOL (H)	QNS (%)
CPUTime	69.12	131	0.47
Memory	66.82	248	1.22
NProcs	67.80	244	1.04



Εικόνα 5.7: *Πολιτική διαχείρισης επεξεργαστικών πόρων με ελαστική πολιτική για AuverGrid*



Εικόνα 5.8: *Πολιτική διαχείρισης πόρων μνήμης με ελαστική πολιτική για AuverGrid*

Κεφάλαιο 6

Επίλογος

6.1 Συμπεράσματα

Σε αυτήν την εργασία προτείνεται μια νέα μορφή συστήματος διαχείρισης υπολογιστικών πόρων που συνδυάζει δύο προϋπάρχουσες τεχνικές. Υπάρχουν πολλές υπηρεσίες που βασίζουν την κλιμάκωση του cloud περιβάλλοντος τους στην πρόβλεψη του φορτίου εργασίας τους με βάση ιστορικά δεδομένα και άλλες οι οποίες χρησιμοποιούν τεχνικές ενισχυτικής μάθησης όπως μαρκοβιανές διαδικασίες, τεχνικές Q-Learning και βαθιάς ενισχυτικής μάθησης με χρήση της ελαστικότητας του περιβάλλοντος. Χρησιμοποιώντας δεδομένα από πραγματικά περιβάλλοντα υπολογιστικών νεφών, αποδείχθηκε πως είναι εφικτό και σκόπιμο να υλοποιηθεί ένα σύστημα που αξιοποιεί μέρη και από τις δύο αυτές προσεγγίσεις για να αποφασίσει την επόμενη κίνηση του. Δόθηκε περισσότερη έμφαση στη διαδικασία παραγωγής χρήσιμων προβλέψεων για ένα μοντέλο που θα μπορούσε να βασίζεται σε έναν πράκτορα βαθιάς ενισχυτικής μάθησης ή κάποια άλλη reactive πολιτική κλιμάκωσης, η οποία εκτός από τα δεδομένα της τρέχουσας κατάστασης του συστήματος λαμβάνει και μία επιπλέον πληροφορία για ένα χρονικό βήμα παραπάνω, ώστε να λειτουργεί εκ των προτέρων. Αναλυτικά, στην παρούσα εργασία:

- Αναζητήθηκαν, βρέθηκαν και αναλύθηκαν δεδομένα από πραγματικά log data υπολογιστικών νεφών. Στη συνέχεια υπέστησαν καθαρισμό και μετατράπηκαν σε δεδομένα χρονοσειρών συνολικής κατάστασης του συστήματος, ώστε να χρησιμοποιηθούν από μοντέλα μηχανικής μάθησης για εκπαίδευση. Τα καθαρά δεδομένα χρονοσειρών προστέθηκαν σε αποθετήριο κώδικα, και δίνονται δημόσια προς λήψη και χρήση από την προγραμματιστική κοινότητα. Επίσης, παρήχθησαν συνθετικά δεδομένα παρόμοιας δομής με τα πραγματικά, βάσει των αρχών της αποσύνθεσης των δεδομένων χρονοσειρών.

- Μελετήθηκαν, δοκιμάστηκαν και προτάθηκαν μοντέλα μηχανικής μάθησης που κρίθηκαν κατάλληλα για την πρόβλεψη χρονοσειρών. Οι προτάσεις που έγιναν, κατάφεραν στην περίπτωσή μας να λειτουργήσουν αποδοτικά για περίπλοκες χρονοσειρές φορτίου εργασίας, που περιλαμβάνουν συχνές αυξομειώσεις και θόρυβο, δείχνοντας ότι μπορούν να δοκιμαστούν σε οποιαδήποτε χρονοσειρά φορτίου παρουσιάζει μοτίβα. Συγκεκριμένα, μπόρεσαν επιτυχώς να προβλέψουν το φορτίο με PRMSE 1.5-8%. Για κάθε χαρακτηριστικό και κάθε σύνολο δεδομένων, εφαρμόστηκε προσαρμοσμένη μέθοδος

πρόβλεψης. Λόγω των μικρών χρόνων προσαρμογής, η τακτική αυτή προτείνεται ως βέλτιστη σε κάθε πιθανό εξεταζόμενο σύστημα, ώστε να γίνει προσαρμογή στα δικά του μοναδικά μοτίβα. Ταυτόχρονα, ευνοείται και συνιστάται η τακτική επανεξέταση του βέλτιστου προβλεπτικού μοντέλου, ανά τακτά χρονικά διαστήματα.

- Έγιναν πειράματα για την εξέταση ενός συστήματος αυτόματης διαχείρισης υπολογιστικών πόρων. Συγκεκριμένα, δείξαμε πως με μία απλή πολιτική που θα ακολουθούσε τις προβλέψεις λαμβάνοντας υπόψιν ένα διάστημα εμπιστοσύνης 10-20% οι χρησιμοποιούμενοι πόροι θα μπορούσαν να μειωθούν έως και 64% σε σχέση με μια βέλτιστη πολιτική κατανομής σταθερών πόρων. Το σύστημα προσαρμόζεται άμεσα στις αλλαγές του υπολογιστικού φορτίου (αφού τις έχει προβλέψει σε προηγούμενο χρόνο). Εν αντιθέσει, υλοποιήσεις παρόμοιες του DERP αργούν κάποια βήματα μέχρι να καταλάβουν και να προσαρμοστούν στην αλλαγή.

Κρίνεται επίσης σκόπιμη μια αναφορά στη διαφορά της επίδοσης μεταξύ των συνόλων δεδομένων που εξετάστηκαν. Αρχικά ερευνήθηκαν πέντε σύνολα δεδομένων από log data πραγματικών συστημάτων Cloud, με τα δύο εξ αυτών να απορρίπτονται πριν ακόμα εφαρμοστεί ο αλγόριθμος μετατροπής τους σε δεδομένα χρονοσειρών. Παρατηρήθηκαν κενές τιμές σε ολόκληρες στήλες, με αποτέλεσμα και την ανικανότητα χειρισμού τους. Εν συνεχεία, ένα ακόμα σύνολο απορρίφθηκε καθώς εμφάνιζε φορτίο σποραδικά, πράγμα που το κατέτασσε σε κατηγορία προβλέψεων διαφορετική από αυτή που προσεγγίστηκε στην εργασία. Τέλος, απορρίφθηκε και τέταρτο σύνολο δεδομένων, αφήνοντας τον τελικό αριθμό σε ένα. Σε αυτή την περίπτωση, το σύνολο δεδομένων παρουσίαζε για σχεδόν το μισό χρονικό εύρος του μηδενικές τιμές φορτίου, μειώνοντας κατά πολύ το μέγεθός του και αποτρέποντας τις τεχνικές πρόβλεψης να ξεπεράσουν ακόμα και ένα μοντέλο Naive. Αξίζει επίσης να τονισθεί και πάλι η διαφορά των σχέσεων μεταξύ των χαρακτηριστικών των πραγματικών και των συνθετικών δεδομένων. Τα συνθετικά δεδομένα δημιουργήθηκαν με σκοπό την ανάδειξη της ευκολίας με την οποία τα νευρωνικά δίκτυα αντιλαμβάνονται και μαθαίνουν τα χρονικά μοτίβα των χρονοσειρών φορτίου. Αντίθετα, στο μοναδικό πραγματικό σύνολο δεδομένων που απέμεινε, οι σχέσεις μεταξύ των χαρακτηριστικών ήταν πολύ πιο απλές, ενώ δεν υπήρχαν τόσο έντονες εξαρτήσεις από τον χρόνο. Καταλήγοντας, αξίζει να τονισθεί η πολυτιμότητα των καθαρών δεδομένων, καθώς και η ανάγκη να συμβαδίζουν με το πρόβλημα προς επίλυση. Για να μπορέσει να προσαρμοστεί ένα μοντέλο μηχανικής μάθησης, είναι απαραίτητη η ύπαρξη μοτίβων τόσο χρονικών, όσο και μεταξύ των χαρακτηριστικών. Τέτοιου τύπου μοτίβα είναι δύσκολο να παρατηρηθούν σε υπολογιστικά συστήματα που χρησιμοποιούνται από πανεπιστήμια, όπως αυτά που εξετάστηκαν. Αντίθετα, το πρόβλημα που πρασπαθήσαμε να λύσουμε απαιτεί συνεχή χρήση του εξεταζόμενου συστήματος, και ύπαρξη μοτίβων συμπεριφοράς στο φορτίο του.

Συνεπώς, μία σωστή καταγραφή στα log data μιας υπηρεσίες Cloud μπορεί να δώσει πολύτιμες πληροφορίες για το φορτίο στο οποίο υποβάλλεται. Επίσης, με τον κατάλληλο χειρισμό των δεδομένων αυτών, αλλά και με την χρήση κατάλληλων τεχνικών πρόβλεψης, μπορούν να εξαχθούν ακριβείς πληροφορίες για την μελλοντική κατάσταση της υπηρεσίας, οι οποίες μπορούν να τροφοδοτηθούν σαν επιπρόσθετη είσοδος σε ένα σύστημα proactive provisiong. Τα αποτελέσματα δείχνουν πως, με την χρήση ενός τέτοιου συστήματος, το

πρόβλημα της κατανάλωσης ενέργειας των συστημάτων Cloud και της μείωσης των χαμένων κερδών των διαχειριστών τους ελαχιστοποιείται.

6.2 Μελλοντικές Επεκτάσεις

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

- Αναζήτηση κι άλλων δεδομένων από πραγματικά περιβάλλοντα Cloud με τελικό σκοπό την εξέταση της συμπεριφοράς των προβλεπτικών μοντέλων σε δεδομένα με μεγαλύτερη διαστατικότητα χαρακτηριστικών ή/και μεγαλύτερου μεγέθους. Θα ήταν επίσης ενδιαφέρον, τα δεδομένα να προέρχονταν από μεγάλης κλίμακας συστήματα με έντονη λειτουργία, όπου θα ήταν πιο πιθανή η εμφάνιση χρονικών μοτίβων στις ανάγκες της υπηρεσίας. Παράλληλα, θα μπορούσε να εξετασθεί η πρόβλεψη ενός παραθύρου μελλοντικών ωρών, με σκοπό την παροχή περισσότερης πληροφορίας στο σύστημα διαχείρισης των πόρων. Τέλος, θα ήταν σκόπιμο να ερευνηθεί η χρήση τεχνικών online learning, στην οποία τα δεδομένα γίνονται διαθέσιμα με διαδοχική σειρά και χρησιμοποιούνται για την ενημέρωση του καλύτερου predictor για μελλοντικά δεδομένα σε κάθε βήμα, σε αντίθεση με τις τεχνικές μάθησης batch που δημιουργούν τον καλύτερο predictor μαθαίνοντας σε ολόκληρο το σύνολο δεδομένων εκπαίδευσης ταυτόχρονα. Το online learning χρησιμοποιείται σε περιπτώσεις όπου είναι απαραίτητο ο αλγόριθμος να προσαρμόζεται δυναμικά σε νέα μοτίβα στα δεδομένα, ή όταν τα ίδια τα δεδομένα παράγονται ως συνάρτηση του χρόνου, όπως στις χρονοσειρές [31] [32].

- Υλοποίηση και εξέταση ενός αυτόματου proactive provisioning συστήματος που θα αξιοποιεί τις προβλέψεις που παράγονται. Υποψήφια συστήματα διαχείρισης προς εξέταση θα μπορούσαν να είναι εμπνευσμένα από το Tiramola, το DERP αλλά και από το προτεινόμενο σύστημα της πτυχιακής εργασίας της Ε. Κουλέτου. Εδώ, περιλαμβάνεται και η ενσωμάτωση του συστήματος στα μηχανήματα, καθώς και η μετατροπή της εξόδου του συστήματος σε πραγματικά actions και κλιμάκωση των πόρων. Η αξιολόγηση των υποψηφίων συστημάτων θα μπορούσε/έπρεπε να γίνει σε αληθινά μηχανήματα που θα προσομοιώνουν την λειτουργία μιας υπηρεσίας Cloud. Προς αυτά τα μηχανήματα θα γινόταν αποστολή ροής εργασιών βασισμένη στα πραγματικά δεδομένα, με σκοπό την μελέτη της συμπεριφοράς των συστημάτων.

- Σχεδίαση και υλοποίηση ενός συστήματος διαχείρισης των περιπτώσεων under-provisioning. Στην παρούσα εργασία έγινε η παραδοχή πως όσο φορτίο ξεπερνά τους πόρους που κατανεμήθηκαν ακυρώνεται και δεν καλύπτεται. Αντ' αυτού θα μπορούσε να υπάρχει ένα σύστημα επαναπρογραμματισμού εργασιών που φθάνουν όταν το σύστημα είναι ¨γεμάτο¨. Παράλληλα, θα μπορούσε επίσης να αξιοποιείται και reactive ανακατανομή των πόρων, κρατώντας τις περισσευούμενες εργασίες σε μία ουρά αναμονής.

Βιβλιογραφία

[1] Arif Wali. *IaaS Vs SaaS Vs PaaS : The Ultimate Guide*. https://tezhost.com/cloud-computing/iaas-vs-saas-vs-paas/. Ημερομηνία πρόσβασης: 17/06/2023.

[2] Animesh Agarwal. *Linear Regression using Python*. https://towardsdatascience.com/linear-regression-using-python-b136c91bf0a2. Ημερομηνία πρόσβασης: 19/06/2023.

[3] Tanay Joshi. *The Gradient Descent Algorithm*. https://tanay-blogs.medium.com/the-gradient-descent-algorithm-f0c14efff5ad. Ημερομηνία πρόσβασης: 19/06/2023.

[4] Shashank Ojha and Kylee Santos. *Implementing parallelism using different architectures*. https://shashank-ojha.github.io/ParallelGradientDescent/. Ημερομηνία πρόσβασης: 19/06/2023.

[5] Vida Rozite, Emi Bertoli και Brendan Reidenbach. *Data Centres and Data Transmission Networks*. https://www.iea.org/energy-system/buildings/data-centres-and-data-transmission-networks. Ημερομηνία πρόσβασης: 02-09-2023.

[6] Ioannis Konstantinou, Evangelos Angelou, Dimitrios Tsoumakos, Christina Boumpouka, Nectarios Koziris και Spyros Sioutas. *Tiramola: elastic nosql provisioning through a cloud management platform*. *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012.

[7] Konstantinos Lolos, Ioannis Konstantinou, Verena Kantere και Nectarios Koziris. *Elastic resource management with adaptive state space partitioning of Markov Decision Processes*. *arXiv preprint arXiv:1702.02978*, 2017.

[8] Constantinos Bitsakos, Ioannis Konstantinou και Nectarios Koziris. *Derp: A deep reinforcement learning cloud system for elastic resource provisioning*. *2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2018.

[9] Ε. Κουλέτου. *Αυτοματοποιημένη Διαχείριση Πόρων με Χρήση Τεχνικών Πρόβλεψης Χρονοσειρών και Βαθιά Ενισχυτική Μάθηση*. Πτυχιακή εργασία, CS Lab, Εθνικό Μετσόβιο Πολυτεχνείο, 2021.

[10] Alexandros Tsafos. *GWA Timeseries Datasets public repository*. https://github.com/alextsaf/GWA-Timeseries-Conversion. Ημερομηνία πρόσβασης: 20/06/2023.

[11] Mell P. και Grance T. *The NIST Definition of Cloud Computing*. *National Institute of Standards and Technology Special Publication*, 2011.

[12] *What is Cloud Elasticity?* `https://www.vmware.com/topics/glossary/content/cloud-elasticity.html`. Ημερομηνία πρόσβασης: 17-09-2023.

[13] Rafael Moreno-Vozmediano, Rubén S Montero, Eduardo Huedo και Ignacio M Llorente. *Efficient resource provisioning for elastic Cloud services based on machine learning techniques*. *Journal of Cloud Computing*, 2019.

[14] Derek DeJonghe. *Load Balancing in the Cloud*. O'Reilly Media, Inc., 2018. ISBN:9781492037996.

[15] Shumway R.H. και Stoffer D.S. *Characteristics of Time Series*. Springer International Publishing, Cham, 2017. ISBN:9783319524528.

[16] Φώτιος Πετρόπουλος και Βασίλειος Ασημακόπουλος. *Επιχειρησιακές Προβλέψεις*. Εκδόσεις Συμμετρία, Αθήνα, 1η έκδοση, 2013.

[17] Mitchell T. *Machine Learning*. McGraw-Hill, 1997.

[18] Σ. Κανδυλάκης και Χ. Ορφανόπουλος. *Ανάλυση Συμμόρφωσης σε Κινητικές Ασκήσεις με Χρήση Τεχνικών Μηχανικής Μάθησης*. Πτυχιακή εργασία, Εθνικό Μετσόβιο Πολυτεχνείο, 2022.

[19] C. L. Chiang. *Statistical methods of analysis*. World Scientific, 2003. ISBN: 9812383107.

[20] Saptashwa Bhattacharyya. *Ridge and Lasso Regression: L1 and L2 Regularization*. `https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b`. Ημερομηνία πρόσβασης: 19/06/2023.

[21] W. Penfield και T. Rusmussen. *The cerebral cortex of man*. McMillan, NY, 1955.

[22] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. *arXivpreprint arXiv:1609.04747*, 2016.

[23] Ashwin B., Maithili B., Pranav G. και Rohan C. *Applications of Convolutional Neural Networks*. *International Journal of Computer Science and Information Technologies*, 7, 2016.

[24] Anastasia Borovykh, Sander Bohte και Cornelis W Oosterlee. *Conditional time series forecasting with convolutional neural networks*. *arXivpreprint arXiv:1703.04691*, 2017.

[25] Danilo P. Mandic και Jonathon A. Chambers. *Recurrent Neural Networks Architectures*. John Wiley and Sons, Ltd, 2001.

[26] Sepp Hochreiter και Jürgen Schmidhuber. *Long short-term memory*. Neural Computation, 1997.

[27] Kyunghyun Cho, Bartvan Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk και Yoshua Bengio. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. CoRR, 2014.

[28] TU Delft. *Πηγή Δεδομένων GWA*. `http://gwa.ewi.tudelft.nl/datasets/`. Ημερομηνία πρόσβασης: 17/06/2023.

[29] Brian. Duignan. *Occam's Razor*. Encyclopedia Britannica, `https://www.britannica.com/topic/Occams-razor`, 2023. Ημερομηνία πρόσβασης: 10-09-2023.

[30] Hannes Weigt, Turhan Demiray, Giovanni Beccuti, Jonas Savelsberg, Moritz Schillinger και Ingmar Schlecht. *System Adequacy Study*. *SFOE Strommarkttreffen Schweiz*, Zürich, Switzerland, 2018.

[31] Oren Anava, Elad Hazan, Shie Mannor και Ohad Shamir. *Online Learning for Time Series Prediction*. *JMLR: Workshop and Conference Proceedings*, 2013.

[32] Vitaly Kuznetsov και Mehryar Mohri. *Time Series Prediction and Online Learning*. *JMLR: Workshop and Conference Proceedings*, 2016.