ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Βελτιστοποίηση Απόδοσης και Ενορχήστρωση Υπηρεσιών στα Άκρα του Δικτύου μέσω Τεχνολογιών Βαθιάς Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεόδωρος Θεοδωρόπουλος

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2022

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Βελτιστοποίηση Απόδοσης και Ενορχήστρωση Υπηρεσιών στα Άκρα του Δικτύου μέσω Τεχνολογιών Βαθιάς

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Θεόδωρος Θεοδωρόπουλος

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 9 η Φεβρουαρίου 2022.

.....	.....	.....
Θ. Βαρβαρίγου	Σ. Παπαβασιλείου	Ε. Βαρβαρίγος
Καθ. Ε.Μ.Π	Καθ. Ε.Μ.Π.	Καθ. Ε.Μ.Π.

Αθήνα, Φεβρουάριος 2022

.....
Θεόδωρος Θεοδωρόπουλος

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

Περίληψη

Στη παρούσα διπλωματική εργασία παρουσιάζεται πληθώρα λύσεων που συμβάλλουν στη βέλτιστη διαχείριση και ενορχήστρωση πόρων που αποτελούν τμήμα Edge και Cloud πλαισίων. Η διπλωματική εργασία χωρίζεται σε θεωρητικό και πρακτικό μέρος. Στο θεωρητικό μέρος, οι αρχιτεκτονικές ιδιαιτερότητες και σχεδιαστικές απαιτήσεις ενός υπερσύγχρονου πλαισίου για την υποστήριξη εφαρμογών εκτεταμένης πραγματικότητας αναλύονται εκτενώς. Ιδιαίτερη έμφαση δίνεται στις μεθοδολογίες πρόβλεψης φόρτου δικτύου και χρήσης πόρων με μεθοδολογίες βαθιάς μάθησης προκειμένου να επιτευχθεί βέλτιστη ενορχήστρωση δικτύων. Επιπλέον γίνεται ανάλυση διαφόρων τεχνικών βελτιστοποίησης υπερπαραμέτρων νευρωνικών δικτύων, ώστε να εξασφαλιστεί η μέγιστη δυνατή αποδοτικότητα του μηχανισμού πρόβλεψης. Στο πρακτικό μέρος, περιλαμβάνεται η υλοποίηση διαφόρων μηχανισμών πρόβλεψης με χρήση βαθιάς μάθησης σε γλώσσα προγραμματισμού python 3. Μέρος του κώδικα βρίσκεται στο τέλος της διπλωματικής στο παράρτημα.

Η παρούσα διπλωματική πραγματοποιήθηκε στα πλαίσια δραστηριοτήτων των ερευνητικών προγραμμάτων CHARITY και Accordion που έχουν λάβει χρηματοδότηση από το πρόγραμμα έρευνας και καινοτομίας «Ορίζοντας 2020» της Ευρωπαϊκής Ένωσης. Μέρος της διπλωματικής έχει δημοσιευθεί σε τρία επιστημονικά συνέδρια με κριτές. Οι τίτλοι των δημοσιεύσεων είναι (1) Μια προσέγγιση που βασίζεται σε κωδικοποιητές-αποκωδικοποιητές βαθιάς μάθησης για τη πρόβλεψη φόρτου δικτύου σε μορφή πολλαπλών βημάτων (2) Cloud για ολογραφία και επαυξημένη πραγματικότητα (3) Βελτιστοποίηση υπερπαραμέτρων GRU νευρωνικών δικτύων με την υβριδική Μπεϋζιανή-εξελικτική στρατηγική για πρόβλεψη χρήσης πόρων στο Edge computing.

Λέξεις Κλειδιά

Νευρωνικά Δίκτυα, Βαθιά Μάθηση ,Βελτιστοποίηση Υπερπαραμέτρων, Κωδικοποιητής-Αποκωδικοποιητής, Ενορχήστρωση Δικτύου, Πρόβλεψη Χρήσης Πόρων, Πρόβλεψη Φόρτου Δικτύου, Επαυξημένη Πραγματικότητα, Ολογραφία

Abstract

This dissertation presents a variety of solutions that contribute to the optimal management and orchestration of resources that are part of Edge and Cloud frameworks. The dissertation is divided into theoretical and practical parts. In the theoretical part, the architectural features and design requirements of a state-of-the-art framework for supporting augmented reality applications are analyzed in detail. Particular emphasis is placed on network traffic and resource usage prediction using deep learning methodologies in order to conduct network orchestration in an optimal manner. In addition, various hyperparameter optimization techniques are analyzed, in order to ensure the maximum possible efficiency of the prediction mechanism. The practical part includes the implementation of various forecasting mechanisms using deep learning methodologies in the python 3 programming language . Part of the code is at the end of the dissertation, in the dedicated appendix.

This dissertation was carried out within the scope of the activities of the Accordion and Charity research projects, which have received funding from the research and innovation program "Horizon 2020" of the European Union. Part of the dissertation has been published in three peer-reviewed scientific conferences. The titles of these publications are: (1) An Encoder-Decoder Deep Learning Approach for Multistep Service Traffic Prediction (2) Cloud for Holography and Augmented Reality (3) Hypertuning GRU neural networks with a hybrid Bayesian-evolutionary strategy for edge resource usage prediction

Keywords

Neural Networks, Deep Learning, Hyperparameter Optimization, Encoder-Decoder, Network Orchestration, Resource Usage Prediction, Network Traffic Prediction, Augmented Reality, Holography

Ευχαριστίες

Ολοκληρώνοντας τις προπτυχιακές σπουδές μου στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ του Εθνικού Μετσόβιου Πολυτεχνείου, θα ήθελα να ευχαριστήσω την κ. Βαρβαρίγου που μου προσέφερε τη δυνατότητα να πραγματοποιήσω τη διπλωματική μου εργασία σε έναν εξαιρετικά ενδιαφέρον τομέα. Θα ήθελα να ευχαριστήσω θερμά τον κ. Τσερπέ για την πολύτιμη καθοδήγηση που μου έχει προσφέρει. Ένα μεγάλο ευχαριστώ στον δρ Γιάννη Βιόλο για την στήριξη και την καθοδήγηση που τόσο απλόχερα μου παρείχε καθόλη τη διάρκεια εκπόνησής της διπλωματικής εργασίας. Θα ήθελα επιπλέον να ευχαριστήσω τον φίλο και συνάδελφο Στέλιο Τσανάκα για την τόσο παραγωγική συνεργασία μας. Επιπροσθέτως, θα ήθελα να ευχαριστήσω τους καθηγητές κ. Βαρβαρίγο και κ. Παπαβασιλείου που ήταν παρόντες στην παρουσίαση της διπλωματικής μου εργασίας και συνέβαλαν στην σύνθεση της τριμελούς επιτροπής. Τέλος, θα ήθελα να ευχαριστήσω το οικογενειακό μου περιβάλλον για την στήριξη που μου παρείχε καθ'όλη την διάρκεια των σπουδών μου.

Θεόδωρος Θεοδωρόπουλος,
Αθήνα, Φεβρουάριος 2022

Περιεχόμενα

Ευρετήριο Σχημάτων

Ευρετήριο Πινάκων

Δομή διπλωματικής εργασίας

- Το πρώτο Κεφάλαιο λειτουργεί ως μια γενική εισαγωγή στη βασική θεματική που διέπει αυτή τη διπλωματική εργασία.
- Στο δεύτερο Κεφάλαιο γίνεται εκτενής ανάλυση του πλαισίου CHARITY, το οποίο αποτελεί ένα υπερσύγχρονο σύστημα που κάνει χρήση διαφόρων τεχνολογιών βελτιστοποίησης της απόδοσης και ενορχήστρωσης Cloud και Edge πόρων προκειμένου να διασφαλίσει την εύρυθμη λειτουργία πλήθους αρκετά απαιτητικών εφαρμογών. Μέσα από την ανάλυση αυτή επιτυγχάνεται η ανάδειξη των προκλήσεων που υπάρχουν στα υβριδικά Cloud-Edge συστήματα. Το περιεχόμενο αυτού του κεφαλαίου βασίζεται σε αυτή την δημοσίευση [1], στην οποία συμμετέχει ο συγγραφέας της πτυχιακής εργασίας .
- Το τρίτο Κεφάλαιο είναι αφιερωμένο στη πρόβλεψη της χρήσης πόρων και τις διάφορες πτυχές της.
- Το τέταρτο Κεφάλαιο περιλαμβάνει ορισμούς και περιγραφές διαφόρων τύπων επαναλλαμβανόμενων νευρωνικών δικτύων. Χρήση των δικτύων αυτών πραγματοποιείται σε επόμενα Κεφάλαια στα πλαίσια σχετικής πειραματικής αξιολόγησης.
- Το πέμπτο Κεφάλαιο εμπεριέχει διάφορες τεχνικές βελτιστοποίησης υπερπαραμέτρων νευρωνικών δικτύων. Το περιεχόμενο αυτού του κεφαλαίου βασίζεται σε αυτή την δημοσίευση [2], στην οποία συμμετέχει ο συγγραφέας της πτυχιακής εργασίας.
- Το έκτο Κεφάλαιο κεφάλαιο εμπεριέχει την πειραματική διαδικασία που διεξήχθη προκειμένου να αξιολογηθεί η προτεινόμενη αρχιτεκτονική HBES σε σχέση με τις υπάρχουσες μεθόδους βελτιστοποίησης υπερπαραμέτρων.
- Το έβδομο Κεφάλαιο είναι αφιερωμένο στη πρόβλεψη του φόρτου δικτύου.
- Το όγδοο Κεφάλαιο αποτελεί ανάλυση διαφόρων τοπολογιών κωδικοποιητή - αποκωδικοποιητή που βασίζονται στη βαθιά μάθηση. Επιπλέον, ο συγγραφέας της διπλωματικής εργασίας προτείνει μια πρωτότυπη αρχιτεκτονική κωδικοποιητή - αποκωδικοποιητή.
- Το ένατο Κεφάλαιο βασίζεται στην υλοποίηση των προαναφερθέντων μοντέλων πρόβλεψης φόρτου δικτύου και στην εκτενή ανάλυση των αποτελεσμάτων που προέκυψαν μετά από σχετικές δοκιμές. Ο συγγραφέας της διπλωματικής εργασίας θεωρώντας πως τα ευρήματα του Κεφαλαίου αυτού ίσως φανούν χρήσιμα σε κομμάτι της επιστημονικής κοινότητας, τα συμπεριέλαβε σε σχετική δημοσίευση [3].

1. Εισαγωγή

Τα συστήματα κατανεμημένου υπολογισμού που ενορχηστρώνουν και διαχειρίζονται μια δεξαμενή ετερογενών πόρων υπολογισμού, επικοινωνίας και αποθήκευσης είναι μια εξαιρετική λύση για την αντιμετώπιση της συνεχώς αυξανόμενης παραγωγής δεδομένων και αιτημάτων υπηρεσιών [4], [5]. Τα δεδομένα που διατρέχουν τους κόμβους ενός δικτύου σε συνδυασμό με τα διάφορα αιτήματα υπηρεσιών χαρακτηρίζονται ως φόρτος δικτύου. Τα σύγχρονα υπολογιστικά μοντέλα όπως το Cloud computing και το Edge computing έχουν την ικανότητα να υποστηρίξουν απαιτήσεις που σχετίζονται με μεγάλο όγκο δεδομένων και υψηλή απαιτούμενη ταχύτητα στα πλαίσια ορισμένων εφαρμογών. Το Cloud computing έχει αναδειχθεί ως λύση για την αντιμετώπιση του μεγάλου όγκου δεδομένων παρέχοντας πόρους υποδομής με ελαστικό τρόπο, προκειμένου να συμβαδίζει με τις διακυμάνσεις του φόρτου εργασίας [6]. Επιπλέον, το Edge computing μπορεί να αντιμετωπίσει τα μειονεκτήματα των λύσεων που βασίζονται στο Cloud μετακινώντας την υπολογιστική διαδικασία πιο κοντά στην άκρη του δικτύου όπου παράγονται τα δεδομένα, προκειμένου να μειωθεί η καθυστέρηση και το απαιτούμενο εύρος ζώνης μεταξύ κέντρων δεδομένων και αισθητήρων [7].

Για να επιταχυνθεί η υιοθέτηση και να αποκομιστούν τα οφέλη του Edge computing, πρέπει να αξιοποιηθούν τεχνολογίες από διάφορους τομείς για να επιτευχθεί τελικά η ενσωμάτωση Cloud και Edge με σκοπό την δημιουργία ενός υπολογιστικού συνεχούς . Ένας από αυτούς τους τομείς είναι η Βαθιά Μηχανική Μάθηση. Στα πλαίσια αυτής της διπλωματικής εργασίας θα αναλυθούν διάφοροι τρόποι με τους οποίους η Βαθιά Μηχανική Μάθηση μπορεί να συμβάλει στην ενορχήστρωση και την βελτιστοποίηση της απόδοσης ενός συνεχούς που προκύπτει από τη συνένωση Cloud και Edge. Ιδιαίτερη έμφαση δίνεται στη σημασία της πρόβλεψης κατανάλωσης πόρων και φόρτου δικτύου, στις οποίες εστιάζουν τα αντίστοιχα κεφάλαια.

Στα πλαίσια της πρόβλεψης κατανάλωσης πόρων, εξετάζεται η χρήση Gated Recurrent Unit (GRU) δικτύων. Προκειμένου το προς εξέταση GRU μοντέλο να είναι όσο το δυνατόν πιο αποδοτικό γίνεται, πραγματοποιείται μία συγκριτική ανάλυση μεταξύ 3 τεχνικών βελτιστοποίησης υπερπαραμέτρων νευρωνικών δικτύων. Οι τεχνικές αυτές είναι οι καθιερωμένες: εξελικτική στρατηγική και Bayesian βελτιστοποίηση, καθώς και η καινοτόμος Hybrid Bayesian Evolution Strategy (HBES) που κάνει χρήση στοιχείων και των δύο προαναφερθέντων τεχνικών.

Τέλος, στα πλαίσια της πρόβλεψης φόρτου δικτύου προτείνεται μία καινοτόμος υβριδική αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή που κάνει ταυτόχρονη χρήση απλών και αμφίδρομων Long Short Term Memory (LSTM) δικτύων προκειμένου να πραγματοποιήσει πρόβλεψη σε επίπεδο πολλαπλών βημάτων. Η αρχιτεκτονική αυτή, όπως θα αναλυθεί στο αντίστοιχο κεφάλαια καταφέρνει να ξεπεράσει σε αποτελεσματικότητα τις αντίστοιχες μεθόδους πρόβλεψης πολλαπλών βημάτων.

2. Cloud για ολογραφία και επαυξημένη πραγματικότητα

Τα σύγχρονα δίκτυα καλούνται να στεγάσουν εξαιρετικά απαιτητικές εφαρμογές που κάνουν χρήση τεχνολογιών Cloud και Edge. Οι εφαρμογές αυτές που μπορούν να χρησιμοποιηθούν από τους χρήστες μέσω του Διαδικτύου. Επιπλέον, λειτουργούν σε κοινόχρηστους υπολογιστικούς πόρους που κατανέμονται σε πολλαπλές τοποθεσίες. Αυτός ο τύπος εφαρμογών είναι εξαιρετικά επωφελής στο πλαίσιο των υπολογιστικά απαιτητικών εφαρμογών, καθώς επιτρέπει στους χρήστες να έχουν εξ αποστάσεως πρόσβαση στους απαραίτητους υπολογιστικούς πόρους.

Η Εκτεταμένη Πραγματικότητα (XR) είναι μια κατηγορία υπολογιστικά απαιτητικών εφαρμογών των οποίων ο στόχος είναι να ελαχιστοποιήσουν το χάσμα μεταξύ του ψηφιακού και του φυσικού κόσμου. Περιλαμβάνει ένα ευρύ φάσμα εφαρμογών όπως η εικονική πραγματικότητα (VR), η επαυξημένη πραγματικότητα (AR) και η μικτή πραγματικότητα. Οι εφαρμογές XR είναι εξαιρετικά απαιτητικές όσον αφορά τους υπολογιστικούς πόρους και τους πόρους αποθήκευσης, καθώς απαιτούν διάφορα στοιχεία XR, όπως τρισδιάστατα μοντέλα. Στο πλαίσιο του μονολιθικού τρόπου ανάπτυξης εφαρμογών, αυτοί οι πόροι θα έπρεπε να ενσωματωθούν στον αποκλειστικό εξοπλισμό XR, καθιστώντας έτσι αυτή την προσπάθεια απαγορευτικά δαπανηρή ή/και ογκώδη. Προκειμένου να αποφευχθεί αυτό το πρόβλημα, καθιερώθηκε η έννοια της εφαρμογής XR που βασίζεται στο Cloud.

Η δημιουργία εφαρμογών XR, συμπεριλαμβανομένης της Επαυξημένης /Εικονικής /Μικτής Πραγματικότητας και της Ολογραφίας, έχει ως επακόλουθο τη δημιουργία διαφόρων προκλήσεων που η επιστημονική κοινότητα καλείται να ξεπεράσει. Αυτές οι προκλήσεις είναι συνυφασμένες με την ίδια τη δομή αυτού του τύπου εφαρμογών. Ενώ κάθε εφαρμογή παρουσιάζει ένα ξεχωριστό σύνολο απαιτήσεων Ποιότητας Εμπειρίας (Quality of Experience) (QoE) και Ποιότητας Υπηρεσίας (Quality of Service) (QoS), υπάρχουν ορισμένα κοινά χαρακτηριστικά που πρέπει να ληφθούν υπόψη κατά την εξέταση των πλαισίων που ακολουθούν οι εφαρμογές XR. Το πιο σημαντικό από αυτά είναι η ανάγκη για εξαιρετικά χαμηλό latency και εξαιρετικά υψηλό bandwidth. Μελέτες έχουν δείξει ότι για να παρέχεται μια αποδεκτή εμπειρία τελικού χρήστη όσον τις εφαρμογές XR, η συνολική καθυστέρηση θα πρέπει να είναι μικρότερο από 15ms και το εύρος ζώνης θα πρέπει να μπορεί να κλιμακωθεί έως και τα 30 Gbps. Καθιερωμένες τεχνικές βελτιστοποίησης δικτύου, όπως η διαφοροποίηση της κυκλοφορίας του δικτύου σε best effort και simple, δεν μπορούν να

ανταποκριθούν σε τόσο αυξημένες απαιτήσεις. Ακόμη και παρά τις ραγδαίες προόδους των τεχνολογιών 5G, η διαδικασία διευκόλυνσης αυτής της κατηγορίας εφαρμογών εξακολουθεί να παραμένει αρκετά προκλητική. Το Edge Computing μπορεί να βοηθήσει στην ανακούφιση ενός μέρους του φόρτου που προκαλείται από εφαρμογές XR που βασίζονται στο Cloud. Το Edge Computing επιτρέπει τη διεξαγωγή της διαδικασίας επεξεργασίας δεδομένων πιο κοντά είτε στο σημείο όπου καταναλώνονται οι υπηρεσίες είτε στο σημείο όπου παράγονται τα δεδομένα, μειώνοντας έτσι τη συνολική καθυστέρηση μεταξύ άκρων και το απαιτούμενο εύρος ζώνης. Τέλος, οι εφαρμογές XR που βασίζονται στο Cloud τρέχουν σε πολλαπλούς ετερογενείς πόρους. Επομένως, είναι ζωτικής σημασίας να ενσωματωθούν ορισμένες τεχνολογίες διαχείρισης και ενορχήστρωσης που είναι σε θέση να ανταποκριθούν στην πολυπλοκότητα που προκύπτει από εφαρμογές XR που βασίζονται στο Cloud που είναι ευαίσθητες στον λανθάνοντα χρόνο και στο εύρος ζώνης. Λαμβάνοντας υπόψη αυτούς τους παράγοντες, γίνεται προφανές ότι για να διευκολυνθούν οι εφαρμογές XR που βασίζονται σε Cloud και Edge συστήματα, απαιτείται ταυτόχρονη χρήση διαφόρων τεχνολογιών συστημάτων Edge, ενορχήστρωσης και δικτύου.

2.1 Υπηρεσίες Cloud και Edge για τεχνολογίες ολογραφίας και επαυξημένης πραγματικότητας

Σε αυτό το πλαίσιο, το έργο CHARITY καλείται να αντιμετωπίσει αυτές τις προκλήσεις και να αναπτύξει μια σειρά σχετικών περιπτώσεων χρήσης. Το CHARITY φιλοδοξεί να αξιοποιήσει τα οφέλη της έξυπνης, αυτόνομης ενορχήστρωσης πόρων Cloud, Edge και δικτύου, για να δημιουργήσει μια συμβιωτική σχέση μεταξύ υποδομών χαμηλής και υψηλής καθυστέρησης που θα διευκολύνει τις ανάγκες των αναδυόμενων εφαρμογών. Εξετάζοντας τη συνολική ιδέα, το πρωταρχικό όραμα του CHARITY είναι η ανάπτυξη ενός ενοποιημένου πλαισίου που διασφαλίζει έναν πλήρη κύκλο υψηλής διαδραστικής διαχείρισης υπηρεσιών, που εκτείνεται από το CI/CD (Continuous Integration / Continuous Delivery) έως τη διαχείριση του κύκλου ζωής (Life Cycle Management) (LCM) και την ενορχήστρωση.

Το οικοσύστημα του CHARITY αποτελείται από τρεις βασικούς πυλώνες:

1) Το CHARITY θα επικεντρωθεί στο σχεδιασμό, την ανάπτυξη και τη διαχείριση άκρως διαδραστικών υπηρεσιών, υποστηρίζοντας εφαρμογές επόμενης γενιάς, εκπληρώνοντας τις υψηλές απαιτήσεις τους.

2) Η υποδομή και η τεχνολογία Cloud και Edge θα σχεδιαστούν για να επιτυγχάνουν, μεταξύ άλλων, εξοικονόμηση κόστους, αύξηση της ελαστικότητας υπηρεσίας και μείωση των εξαρτήσεων λογισμικού/ υλισμικού. Τόσο οι τεχνικές Τεχνητής Νοημοσύνης (Artificial Intelligence) (AI) όσο και οι έννοιες Zero-touch Network and Slice Life-cycle Management (ZSM) θα διαδραματίσουν κρίσιμο ρόλο.

3) Βελτιστοποίηση της τηλεπικοινωνιακής υποδομής, για τη διασφάλιση της επίτευξης των επιθυμητών Key Performance Indicators (KPIs) και απαιτήσεων εφαρμογών με δυνατότητα AR, VR και Ολογραφίας, π.χ., μέσω της χρήσης κατηγοριών κίνησης δικτύου και ροών με προτεραιότητα.

2.2 Υπερσύγχρονα Use Cases XR εφαρμογών

Παρακάτω, περιγράφονται οι εφαρμογές περιπτώσεων χρήσης (Use Cases) (UC), στις οποίες θα δοκιμαστεί η πλατφόρμα CHARITY και εντοπίζονται οι κύριες προκλήσεις τους. Τα UC οργανώνονται σε τρεις κύριες κατηγορίες, με στόχο την αντιμετώπιση των κυρίων προκλήσεων σε αυτούς τους τομείς, επιτρέποντας τελικά την επικύρωση και την ανάδειξη των καινοτομιών του Edge & Cloud Computing στα πλαίσια του CHARITY.

2.2.1 Ολογραματικές τεχνολογίες σε πραγματικό χρόνο

Η Ολογραματική Συναυλία UC1.1 χρησιμοποιεί ένα ψευδο-ολογραφικό σύστημα προβολής, βασισμένο στην αρχή του Pepper's Ghost [8], που δημιουργεί μια ψευδαίσθηση μουσικών που παίζουν ζωντανά σε μια σκηνή. Διαφορετικά μέλη του συγκροτήματος βρίσκονται σε διαφορετικές τοποθεσίες και συμμετέχουν ουσιαστικά σε μια συναυλία. Το «2D ολόγραμμα» κάθε μουσικού καταγράφεται μέσω μιας βιντεοκάμερας και μεταδίδεται μαζί με τη ροή του ήχου, υποβάλλεται σε περαιτέρω επεξεργασία στο σύννεφο CHARITY, συγχρονίζεται και προβάλλεται σε μια ειδική οθόνη στη σκηνή. Το βίντεο από τους θεατές, που βρίσκεται μπροστά από τη σκηνή, ηχογραφείται και μεταδίδεται στα μέλη του συγκροτήματος ως ανατροφοδότηση. Το UC βασίζεται σε cloud και τοπική επεξεργασία που υποστηρίζεται από τις υπηρεσίες CHARITY και εκμεταλλεύεται σχετικούς πόρους λογισμικού και υλικού που υποστηρίζονται από την πλατφόρμα. Ορισμένες μονάδες και υπηρεσίες ενδέχεται να εκτελούνται απομακρυσμένα στην πλατφόρμα CHARITY, π.χ. μίξη και σύνθεση βίντεο, συμπίεση και απόδοση, ενώ άλλα τοπικά σε υπολογιστές στη σκηνή ή στην τοποθεσία των μουσικών π.χ. την υπηρεσία συγχρονισμού. Η κύρια πρόκληση αυτού του

UC είναι να συγχρονίσει όλες τις ροές βίντεο πριν από την εμφάνιση των μουσικών στη σκηνή. Ιδιαίτερα ο συγχρονισμός ήχου είναι σημαντικός. Επομένως ο διαχωρισμός του ήχου από τα δεδομένα βίντεο και ο χειρισμός του με μεγαλύτερη προτεραιότητα είναι μία από τις επιλογές που πρέπει να εφαρμοστούν. Ο στόχος είναι να επιτευχθεί μια άψογη εμπειρία ήχου και σε περίπτωση ξαφνικών περιορισμών εύρους ζώνης, θυσιάζοντας την ποιότητα του βίντεο για τον ήχο.

Η Ολογραματική Συνάντηση UC1.2 επιτρέπει στον κύριο συμμετέχοντα, που ενεργεί ως ομιλητής, να βρίσκεται σε οποιαδήποτε τοποθεσία και να μεταδίδει το βίντεο και τον ήχο του σε πολλές οθόνες σε διάφορους χώρους ταυτόχρονα. Ο τοπικός υπολογιστής στη θέση του ομιλητή υποστηρίζει τη λήψη μη επεξεργασμένων ροών βίντεο από τις διάφορες τοποθεσίες του κοινού, για να επιτρέψει την οπτική επικοινωνία μεταξύ του ομιλητή και του κοινού σε πραγματικό χρόνο. Αυτό το UC είναι μια απλοποιημένη έκδοση του UC1.1, όπου δεν απαιτείται συγχρονισμός, αλλά η κύρια πρόκληση είναι να μπορούν να υποστηρίζονται ταυτόχρονα πολλαπλές συσκευές προβολής.

Στον Ολογραμματικό Βοηθό UC1.3, ένα τρισδιάστατο (3D) avatar παρουσιάζεται σε μια ολογραμματική τρισδιάστατη οθόνη (H3D) [9], η οποία παρέχει απαντήσεις σε φυσική γλώσσα και μεταγλωττισμένες 3D οπτικές πληροφορίες , σε ερωτήματα που εκφωνούνται από τον χρήστη, μετά την ανάκτηση αποτελεσμάτων από υπηρεσίες Διαδικτύου τρίτων. Οι πληροφορίες και οι υπηρεσίες που προσφέρονται από τον Ολογραμματικό Βοηθό είναι προσβάσιμες μέσω υπηρεσιών που βασίζονται σε σύννεφο μέσω API που παρέχονται από την πλατφόρμα CHARITY ή από υπηρεσίες τρίτων που είναι διαθέσιμες στο διαδίκτυο (π.χ. καιρός, μετοχές ή chatbot). Η οθόνη H3D βασίζεται στην τρισδιάστατη ολογραφία, η οποία χρησιμοποιεί την παρεμβολή της τεχνολογίας φωτός [10] για τη διαμόρφωση του συνεκτικού φωτός και τη δημιουργία ρεαλιστικών οπτικών αναπαραστάσεων εκατομμυρίων τρισδιάστατων σημείων στο χώρο, παρέχοντας έτσι πραγματικό βάθος. Τα οπτικοποιημένα τρισδιάστατα δεδομένα εξάγονται από μια ροή τρισδιάστατου Point Cloud που λαμβάνεται από τις υπηρεσίες του CHARITY. Αυτή η περίπτωση χρήσης απλώς σε τοπικό λογισμικό, υλικό υπολογιστή και μία οθόνη H3D (συμπεριλαμβανομένου του Eye-Tracking), υποστηρίζεται από υπηρεσίες Cloud του CHARITY που φιλοξενούν τη λογική βοηθού, την απόδοση Unity3D, την αναγνώριση ομιλίας και το 3D processing του Point Cloud. h ροή του τρισδιάστατου Point Cloud λαμβάνεται και εμφανίζεται απευθείας, ως τρισδιάστατο ολόγραμμα, στην οθόνη H3D σε πραγματικό χρόνο [11]. Η κύρια πρόκληση αυτού του UC είναι η δημιουργία, συμπίεση / αποσυμπίεση δεδομένων τρισδιάστατου Point Cloud σε πραγματικό χρόνο.

2.2.2 Τεχνολογίες εικονικής εκπαίδευσης

Στην ιατρική εκπαίδευση με χρήση εικονικής πραγματικότητας UC2.1, πολλοί παίκτες εκτελούν ηλεκτρονικά προκαθορισμένα χειρουργικά σενάρια σε περιβάλλον VR, προς μια βελτιωμένη [12] εμπειρία ιατρικής εκπαίδευσης [13], [14]. Το UC εκμεταλλεύεται πόρους του CHARITY για προηγμένη επεξεργασία CPU και GPU για προσομοιώσεις, απόδοση, συμπίεση που απαιτεί χαμηλή καθυστέρηση και αυξημένο εύρος ζώνης, με ιδιαίτερη έμφαση στα ασύρματα HMD με περιορισμένους πόρους, GPU, μπαταρία και δυνατότητα κινητικότητας. Η εφαρμογή αναπτύσσεται μέσω δύο μικροϋπηρεσιών που στεγάζονται σε Edge και Cloud. Της Geometric Algebra inTerpolation Engine (GATE) και της Physics Engine, η οποία είναι υπεύθυνη για τον υπολογισμό, την απόδοση και την κωδικοποίηση των εικόνων που θα μεταδοθούν στο HMD μέσω σήματος. Επιπλέον, η προσαρμογή σε χρόνο εκτέλεσης και η δυναμική βελτιστοποίηση που παρέχει το GATE αξιοποιούνται με βάση τα χαρακτηριστικά του δικτύου [15]. Το ελαφρύ HMD είναι υπεύθυνο για την αποκωδικοποίηση και την προβολή των μεταφερόμενων εικόνων από το Edge και για τη λήψη και τη μεταφορά συμβάντων που σχετίζονται με τον χρήστη (π.χ. θέση ελεγκτών, ενεργοποιητές) στα πλαίσια της εφαρμογής. Η κύρια πρόκληση αυτού του UC είναι ο διαχωρισμός του Physics Engine από τον αγωγό Unity3D, δημιουργώντας έτσι ξεχωριστή υπηρεσία που θα επιτρέψει χαμηλότερους χρόνους λειτουργίας.

Χρησιμοποιώντας την Ξενάγηση Εικονική Πραγματικότητας UC2.2, ο χρήστης μπορεί να απολαύσει διαδραστικές εμπειρίες εικονικής περιήγησης και σκηνές ζωντανής ροής στα πλαίσια της εικονικής πραγματικότητας. Η εφαρμογή μπορεί να χρησιμοποιηθεί για πολλαπλούς σκοπούς, όπως μάθηση, αφήγηση, μάρκετινγκ και χρήσεις που σχετίζονται με την ακίνητη περιουσία. Η εικονική περιήγηση υποστηρίζει βίντεο 360 μοιρών, πανοράματα, τρισδιάστατα μοντέλα, τυπικές εικόνες και βίντεο, καθώς και βασικά τρισδιάστατα πλέγματα. Η εφαρμογή περιλαμβάνει πολλές ενότητες τόσο στο back-end όσο και στο front-end. Οι front-end μονάδες αποτελούνται από μια εφαρμογή Ιστού που επεξεργάζεται σε πραγματικό χρόνο τα βίντεο 360◦ που δημιουργούνται από τον χρήστη και διαχειρίζεται μια εφαρμογή προβολής που επιτρέπει στον χρήστη να προβάλλει και να καταναλώνει το περιεχόμενο που δημιουργείται από την εφαρμογή Ιστού. Το back-end αποτελείται από πολλά στοιχεία που είναι υπεύθυνα για τη φιλοξενία περιεχομένου πολυμέσων, την επεξεργασία εικόνας, την απόδοση τρισδιάστατων μοντέλων, τη μετατροπή μορφής βίντεο και τη ροή βίντεο στο back-office. Αυτά τα εξαρτήματα εντάσσονται ως μικρο-υπηρεσίες στο CHARITY. Η κύρια πρόκληση αυτού του UC είναι η υποστήριξη ταχύτερης μετατροπής και ροής βίντεο μέσω

των προηγμένων δυνατοτήτων επεξεργασίας της πλατφόρμας CHARITY, του χαμηλού χρόνου καθυστέρησης και του αυξημένου εύρους ζώνης. Επιπλέον, ανάλογα με τον τύπο της συσκευής προβολής ή τα χαρακτηριστικά του δικτύου, η υπηρεσία μηχανής 3D πρέπει να είναι ικανή να προσαρμόσει τις διαδικασίες επεξεργασίας και απόδοσης σε διαφορετικές αναλύσεις.

2.2.3 Διαδραστικές τεχνολογίες μεικτής πραγματικότητας

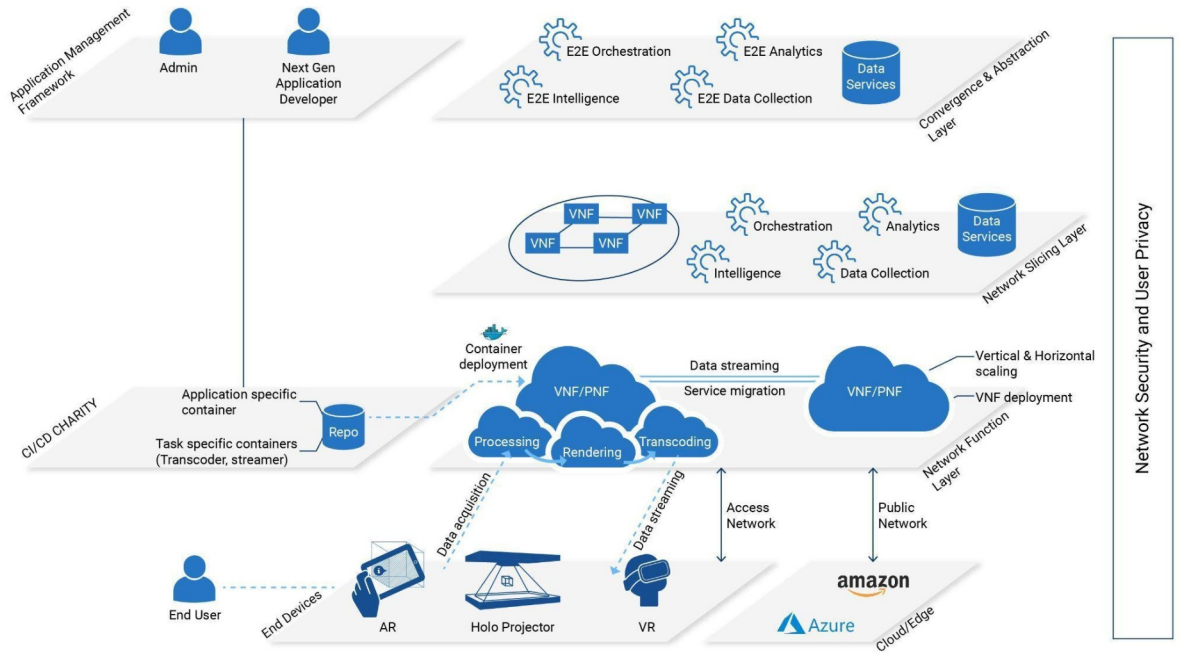
Το Συνεργατικό Παίγνιο UC3.1 παρέχει ένα εξαιρετικά καθηλωτικό παιχνίδι επαυξημένης πραγματικότητας για πολλούς παίκτες. Για να παρέχει στους παίκτες μια επαρκώς διαδραστική εμπειρία, έχει αναπτυχθεί μια αποκλειστική μηχανή πολλών παικτών για το συγχρονισμό όλων των δυναμικών αντικειμένων του παιχνιδιού και των καταστάσεων του χρήστη. Η συνολική λύση απαιτεί η υποδομή να παρέχει βασικά χαρακτηριστικά: πολύ χαμηλή καθυστέρηση δικτύου και αποτελεσματική υπηρεσία ανακάλυψης πόρων, μια αξιόπιστη υποδομή (Edge και Cloud) για την προστασία του διακομιστή παιχνιδιών από πιθανές παραβιάσεις. Το UC3.1 εξερεύνησε την τεχνολογία 3D Point Cloud για να εμπλουτίσει το παιχνίδι. Οι δυνατότητες των ενσωματωμένων καμερών αξιοποιούνται για την παροχή εισόδου και η χρήση των δεδομένων εξόδου για τη μίξη του πραγματικού και του εικονικού περιβάλλοντος. Η υπηρεσία Mesh Collider Generator Service που υποστηρίζεται από το CHARITY χρησιμοποιείται για να επιτρέψει την ακριβή ανακατασκευή της γεωμετρίας του πραγματικού περιβάλλοντος μέσα σε μία περίοδο λειτουργίας του παιχνιδιού. Η κύρια πρόκληση αυτού του UC είναι να βελτιστοποιήσει τη μηχανή πολλών παικτών ελαχιστοποιώντας την ποσότητα των δεδομένων που αποστέλλονται μέσω του δικτύου για να διατηρήσει χαμηλούς χρόνους εξυπηρέτησης μεταξύ πολλών παικτών.

Ο Εκπαιδευτής Επανδρωμένων / Μη Επανδρωμένων Επιχειρήσεων UC3.2 δημιουργεί ένα καθηλωτικό συνεργατικό εκπαιδευτικό περιβάλλον αναδυόμενων τεχνολογιών στα πλαίσια επανδρωμένων και μη επανδρωμένων επιχειρήσεων, ενώ ελαχιστοποιεί την ανάγκη ακριβού εξοπλισμού. Ο βασικός στόχος είναι να προωθηθεί η ανάπτυξη προσομοιωτών εκπαίδευσης στο συνεχές Cloud-Edge. Οι εκπαιδευόμενοι μπορούν ουσιαστικά να συνεργαστούν σε ένα συνθετικό περιβάλλον για να πραγματοποιήσουν συντονισμένες ενέργειες. Μπορούν επίσης να ελέγχουν εξ αποστάσεως μη επανδρωμένα οχήματα, όπως απλά εναέρια drones διάσωσης, για να προσεγγίζουν απρόσιτα εδάφη προκειμένου να αποκτήσουν επίγνωση της κατάστασης σε σενάρια αναζήτησης και διάσωσης. Η ρευστότητα των πόρων που προκύπτει μέσω του συνεχούς στο Cloud-Edge

και της ενορχήστρωσης του δικτύου, αξιοποιείται για να διευκολύνει τη δέσμευση και τη συνεργασία στα πλαίσια πολλαπλών περιπτώσεων προσομοίωσης. Η κύρια πρόκληση σε αυτό το UC είναι η εικονικοποίηση του υπάρχοντος τοπικού μηχανισμού προσομοίωσης.

2.3. Αρχιτεκτονική του συστήματος

Το Σχήμα 1 απεικονίζει τη γενική αρχιτεκτονική του πλαισίου CHARITY. Αποτελείται από τέσσερα κύρια επίπεδα που υποστηρίζονται από ένα Πλαίσιο Διαχείρισης Εφαρμογών (AMF) και έναν αγωγό CI/CD.



Σχήμα 1: Γενική αρχιτεκτονική του πλαισίου CHARITY.

Το επίπεδο υποδομής, στο κάτω μέρος, αποτελείται από τα φυσικά στοιχεία που εμπλέκονται στην υπηρεσία XR, η οποία εκτείνεται από τις συσκευές τελικού χρήστη έως το δίκτυο που μεταφέρει δεδομένα, μέσω του Cloud-Edge συνεχούς που προσφέρει υπολογιστικές δυνατότητες. Λόγω της ανάγκης για αυξημένο εύρος ζώνης και της ευαίσθητης σε καθυστέρηση φύσης των υπηρεσιών XR, το CHARITY στοχεύει στη ταυτόχρονη μόχλευση πολλών παρόχων Cloud ως ένα τρόπο για την δημιουργία του Edge-Cloud συνεχούς, παρέχοντας έτσι επίσης πολύ υψηλό εύρος ζώνης και αξιόπιστη και ντετερμινιστική δικτύωση.

Το δεύτερο επίπεδο, με το όνομα Network Function Layer (NFL), είναι υπεύθυνο για την αφαίρεση της ετερογένειας της υποκείμενης υποδομής και, κατά συνέπεια, την εφαρμογή της έννοιας του Edge-Cloud συνεχούς, ορίζοντας ένα πλαίσιο ενορχήστρωσης που μπορεί να εκτελεί απρόσκοπτα και αποτελεσματικά υπηρεσίες XR. Το NFL παρέχει μια εγγενή πλατφόρμα στο Cloud όπου οι υπηρεσίες XR υλοποιούνται και εκτελούνται ως μικρό-υπηρεσίες, γεγονός που επιτρέπει μεγάλη ευελιξία και αποτελεσματικότητα. Πράγματι, μια υπηρεσία μπορεί εύκολα να μεταφερθεί για να ανταποκρίνεται στις απαιτήσεις της όσον αφορά τα KPIs. Επίσης, μια υπηρεσία XR μπορεί να χωριστεί έτσι ώστε ένα μέρος της να εκτελείται στο Edge για να μειωθεί η καθυστέρηση δικτύου και η χρήση εύρους ζώνης, ενώ το τμήμα της που χαρακτηρίζεται από βαρύ υπολογιστικό φορτίο εκτελείται σε μακρινούς Cloud πόρους.

Δεδομένου του γεγονότος ότι μια υπηρεσία XR μπορεί να χωριστεί σε πολλούς τομείς, το Network Slicing Layer (NSL) είναι υπεύθυνο να συγκεντρώσει όλους αυτούς τους τομείς και να τους συρράψει για να δημιουργήσει μια ενιαία υπηρεσία από άκρο σε άκρο. Ο πρώτος βρόχος αυτοματισμού υπηρεσιών παρέχεται από την NSL. Υπάρχει σε κάθε τομέα που συνθέτει την υπηρεσία XR. Υλοποιείται ακολουθώντας την έννοια OODA (Observe, Orient, Decide, Act), η οποία ειδικεύεται στη συλλογή και την ανάλυση δεδομένων, την ευφυΐα και την ενορχήστρωση του συστήματος. Αυτός ο βρόχος τοπικού αυτοματισμού ακολουθεί την ιδέα του ZSM [16] και έτσι, εφαρμόζει αυτο-ενορχήστρωση, αυτο βελτιστοποίηση και αυτο-ίαση που συμβάλλει στην ικανοποίηση των KPI για κάθε υποτμήμα.

Το Convergence and Abstraction Layer (CAL) είναι ο διαχειριστής και ο ενορχηστρωτής των end-to-end XR υπηρεσιών. Είναι υπεύθυνο για την επιβολή των KPIs σε ολόκληρο το επίπεδο XR υπηρεσιών. Αυτό σημαίνει ότι οι αποφάσεις για τη σύνθεση του υποτμήματος, όπως το ποια υποφέτα θα χρησιμοποιηθεί ή πού θα εκτελεστεί η εκάστοτε υποφέτα, λαμβάνονται σε αυτό το επίπεδο. Για παράδειγμα, έχοντας ένα συγκεκριμένο προϋπολογισμό καθυστέρησης, αυτό το επίπεδο θα χωρίσει αυτόν τον προϋπολογισμό στους κατάλληλους τομείς, έτσι ώστε ο χρόνος της end-to-end καθυστέρησης να είναι χαμηλότερος από αυτόν που ορίζει ο προϋπολογισμός. Ακολουθώντας το πλαίσιο ZSM και για την επιβολή των KPIs, η CAL εφαρμόζει επίσης έναν βρόχο αυτοματισμού end-to-end. Ο κύριος στόχος αυτού του βρόχου είναι να επιβάλει τους KPIs και να εκτελέσει προληπτικές ενέργειες προκειμένου να προστατεύσει την υπηρεσία XR από υποβάθμιση της ποιότητας της. Προσφέρει επίσης μια διεπαφή από την οποία οι προγραμματιστές και οι πάροχοι XR μπορούν να υποβάλουν τα προσχέδιά τους. Αυτά αποτελούνται από αρχεία περιγραφής που καθορίζουν όλα τα δομικά στοιχεία και τη διασύνδεσή τους για να σχηματίσουν μια υπηρεσία XR.

Το επίπεδο ασφάλειας δικτύου και απορρήτου χρήστη (NSUP) χρησιμοποιείται για την ασφάλεια της πλατφόρμας και των υπηρεσιών XR που εκτελούνται σε αυτή. Αυτό συμβάλει στην ασφάλεια των εικόνων του λογισμικού που συνθέτει την XR υπηρεσία και της επικοινωνίας μεταξύ των στοιχείων τους, στην προστασία του απορρήτου των χρηστών μέσω της επεξεργασίας των δεδομένων τους στο Edge και, κατά συνέπεια, της αποφυγής της μεταφοράς ευαίσθητων δεδομένων σε όλο το δίκτυο με σκοπό την επεξεργασία στο Cloud. Το NSUP μπορεί επίσης να παρέχει δυναμική ασφάλεια όπου τα στοιχεία ασφαλείας προστίθενται δυναμικά στην υπηρεσία XR.

Τέλος, ο αγωγός AMF και CI/CD, που παρέχεται από το CHARITY, είναι το σημείο εισόδου των παρόχων και προγραμματιστών XR στην πλατφόρμα CHARITY. Μεταξύ άλλων, το AMF χρησιμοποιείται για τον καθορισμό του σχεδιαγράμματος των υπηρεσιών XR και για την εκκίνηση, τη διακοπή, την τροποποίηση και τη διαμόρφωση των εκτελούμενων υπηρεσιών XR. Ενώ η διεκπεραίωση σε μορφή CI/CD χρησιμοποιείται για να διασφαλιστεί ότι η ανάπτυξη μιας νέας υπηρεσίας XR ή/και μιας νέας έκδοσης μιας υπηρεσίας που εκτελείται δεν θα προκαλέσει υποβάθμιση της ποιότητας για αυτήν την υπηρεσία XR ή οποιαδήποτε άλλη ταυτόχρονη υπηρεσία XR. Περισσότερες λεπτομέρειες για την αρχιτεκτονική CHARITY μπορείτε να βρείτε εδώ [17].

2.4. Ενορχήστρωση Cloud και Edge υποδομών

Οι απομακρυσμένοι πόροι και οι μικροϋπηρεσίες θεωρούνται σε μεγάλο βαθμό ως η κατάλληλη λύση για την αποτελεσματική ανάπτυξη και διαχείριση νέων εφαρμογών στα πλαίσια μιας υβριδική υποδομής Edge και Cloud. Αυτό πιθανότατα θα οδηγήσει σε ένα μοντέλο εφαρμογής Cloud πολλαπλών συστατικών όπου τα στοιχεία ενώ ανήκουν σε ετερογενή περιβάλλοντα μπορούν να διαχειρίζονται από έναν κεντρικού υπολογιστή.

Η πρωταρχική ανάγκη που προκύπτει σε ένα τέτοιο σύστημα είναι η ενορχήστρωση της ανάπτυξης των διαφόρων συστατικών του, έτσι ώστε να ικανοποιούνται όλες οι εξαρτήσεις μεταξύ των συστατικών. Μετά την ανάπτυξη, πρέπει να υπάρχουν μηχανισμοί παρακολούθησης και ανάκτησης για την αντιμετώπιση πιθανών δυσλειτουργιών που ενδέχεται να παρουσιάσουν κάποια εξαρτήματα .

Προηγούμενα πειράματα έχουν αναφέρει διάφορα σενάρια με εφαρμογές που έχουν αναπτυχθεί σε διαφορετικούς παρόχους IaaS και PaaS και δείχνουν ότι σχεδόν το είκοσι τοις εκατό των σεναρίων παρουσίασαν κάποια αποτυχία. Άλλες εργασίες έχουν δείξει ότι όσο μεγαλύτερος είναι ο αριθμός των στοιχείων που σχηματίζουν μια εφαρμογή, τόσο

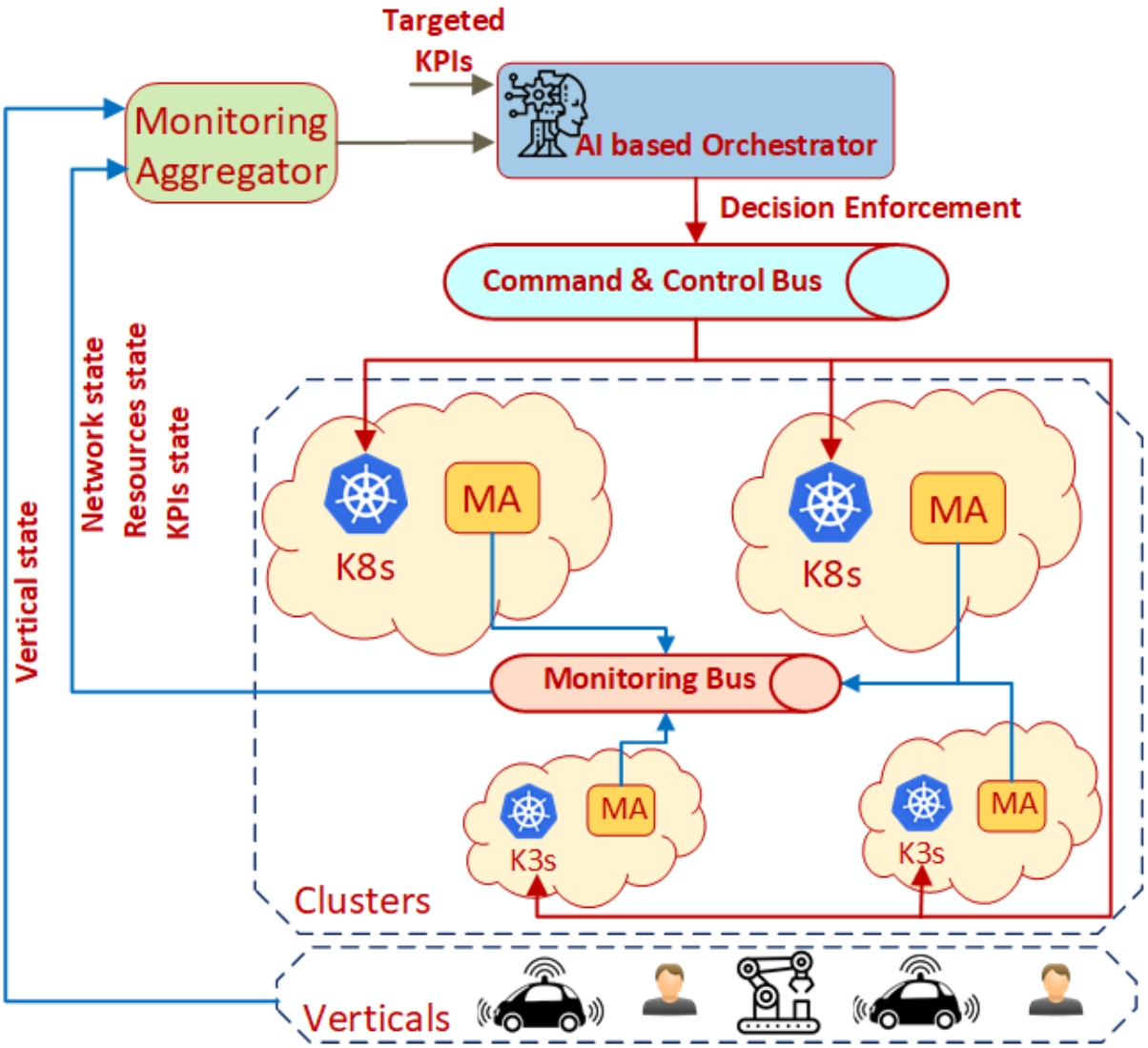
μεγαλύτερη είναι η πιθανότητα να αποτύχει η ανάπτυξή της, λόγω κάποιας πιθανής δυσλειτουργίας ενός εκ των στοιχείων της.

Ωστόσο, η μέχρι στιγμής διαθέσιμη υποστήριξη για λειτουργίες εφαρμογών Cloud και την επαναφορά τους από δυσλειτουργίες δεν είναι ακόμη πλήρως αυτοματοποιημένη. Συνήθως, οι φόρτοι εργασίας επανεκκινούνται χειροκίνητα, ίσως σε ένα νέο μηχάνημα που βασίζεται σε εργαλεία όπως το Chef ή το Puppet, κάτι που απαιτεί σημαντική χειροκίνητη παρέμβαση. Οι αυτοματοποιημένες λύσεις που προσφέρονται από την Amazon, την Google ή το Azure μπορεί να εκτελούν επαρκώς στον έγκαιρο εντοπισμό της αποτυχίας, αλλά τα μέτρα μετριασμού των συνεπειών της περιορίζονται στη διακοπή της οντότητας που παρουσίασε την δυσλειτουργία και στη δημιουργία μιας νέας. Αυτές οι διαδικασίες περιορίζονται σε εξαρτήματα που αναπτύσσονται χρησιμοποιώντας συγκεκριμένες λύσεις και δεν λαμβάνονται υπόψη άλλες εξαρτήσεις εκτός από τους σχετικούς όγκους αποθήκευσης.

2.5 Οι πέντε πυλώνες του CHARITY

Α. Τοποθέτηση υπηρεσιών:

Για την οποία απαιτείται ένα πλαίσιο Ενορχήστρωσης με επίγνωση πόρων (Artificial Intelligence Resource Orchestrator) (AIRO) βασισμένο σε τεχνητή νοημοσύνη. Το πλαίσιο AIRO [18] αξιοποιεί την έννοια του ZSM, την εγγενή χρήση του Cloud και τις τεχνικές Μηχανικής Μάθησης για την αποτελεσματική διαχείριση πόρων δικτύου και υπολογιστών. Η λειτουργία του εν λόγω πλαισίου απεικονίζεται στο Σχήμα 2.



Σχήμα 2: Η λειτουργία του Artificial Intelligence Resource Orchestrator.

Το πλαίσιο AIRO υλοποιείται με την ανάπτυξη ενός μηχανισμού παρακολούθησης και διαχείρισης. Ο μηχανισμός αυτός στεγάζεται στον κύριο κόμβο κάθε συμπλέγματος (cluster) κόμβων προκειμένου να δημιουργηθεί ένας ενιαίος τομέας διαχείρισης.

B. Primitives βασισμένα σε GPU που υποστηρίζουν τοποθέτηση υπηρεσιών που με τη χρήση τεχνητής νοημοσύνης:

Λόγω της συνεχούς φόρτου χρειάζεται ένα σύστημα μηχανικής μάθησης για να κάνει το AIRO πλήρως προσαρμοστικό.

Γ. Αποκεντρωμένη διαχείριση αντιγράφων υπηρεσίας:

Εφόσον μπορούμε να θεωρήσουμε ότι το Edge σύστημα αποτελείται από οντότητες με μια συγκεκριμένη γεωγραφική θέση που αντιπροσωπεύει μια μικρή έως μεσαία δεξαμενή δυνητικά ετερογενών πόρων, απαιτείται ένας αποκεντρωμένος μηχανισμός για τον έλεγχο του αριθμού των αντιγράφων εφαρμογών της ίδιας υπηρεσίας σε μια πλατφόρμα Edge Computing, τηρώντας παράλληλα τις ζητούμενες απαιτήσεις QoE και QoS.

Δ. Παρακολούθηση και πρόβλεψη:

Η παρακολούθηση περιλαμβάνει τη διαδικασία συλλογής, ανάλυσης και χρήσης πληροφοριών συστηματικά, που παρέχει τη συνεχή απεικόνιση και εκτίμηση της κατάστασης και της προόδου μιας εφαρμογής, υπηρεσίας ή υποδομής. Μια τέτοια συνεχής διαδικασία παρακολούθησης παρέχει έναν τρόπο ανάλυσης του περιβάλλοντος για να ελεγχθεί εάν οι εφαρμογές και η υποδομή λειτουργούν όπως προβλέπεται. Πράγματι, η παρακολούθηση του περιβάλλοντος σε πραγματικό χρόνο επιτρέπει, για παράδειγμα, την ελαχιστοποίηση του χρόνου απόκρισης σε περιστατικά (π.χ. τον εντοπισμό και τον μετριασμό των συνεπειών των επιθέσεων στον κυβερνοχώρο).

Στο παρελθόν, αυτή η παρακολούθηση χρησίμευε ουσιαστικά ως υποστήριξη στη διαδικασία λήψης αποφάσεων για μη αυτόματες ενέργειες διαχείρισης υπηρεσιών και υποδομών, αλλά επί του παρόντος, μόλις συμβεί κάτι εκτός της αναμενόμενης συμπεριφοράς, είναι δυνατό να ληφθούν οι κατάλληλες ενέργειες και αποφάσεις. Καθώς προχωράμε σε πιο περίπλοκα και απαιτητικά σενάρια, οι αλγόριθμοι παρακολούθησης και αργότερα, οι αλγόριθμοι πρόβλεψης αποκτούν μια εντελώς νέα σημασία στην ενορχήστρωση και τη διαχείριση του κύκλου ζωής των εφαρμογών επόμενης γενιάς. Αυτοί οι μηχανισμοί εξαρτώνται σε μεγάλο βαθμό από μια ολοκληρωμένη προσέγγιση παρακολούθησης σε πραγματικό χρόνο και από την ποιότητα των μετρήσεων που συλλέγονται.

Ε. Ενορχήστρωση με γνώμονες την ασφάλεια και την διασφάλιση της ιδιωτικότητας:

Καθώς η ενορχήστρωση και ο προγραμματισμός των υπηρεσιών XR αποτελούν τον πυρήνα της αρχιτεκτονικής, η ασφάλεια για την παροχή υπηρεσιών από άκρο σε άκρο γίνεται μια σημαντική πτυχή. Εκτός από την βασική ασφάλεια της εφαρμογής και την εγγενή ασφάλεια που παρέχει το Cloud, επιτυγχάνουμε ασφαλή ενορχήστρωση σε τρία μέρη:

i Ασφαλής εκτέλεση: πειραματιζόμαστε με ασφαλή εκτέλεση λειτουργιών σε αξιόπιστα περιβάλλοντα εκτέλεσης για τη διασφάλιση του απορρήτου των χρηστών που έχουν πρόσβαση σε εφαρμογή που στεγάζεται στο Cloud-Edge συνεχές.

ii Ασφάλεια μικρουπηρεσίων: αναπτύχθηκε για την εύρεση του ελάχιστου συνόλου δυνατοτήτων που χρειάζονται τα containers για την σωστή εκτέλεση των εφαρμογών τους, ελαχιστοποιώντας παράλληλα τις αλληλεπιδράσεις τους με τον πυρήνα του λειτουργικού συστήματος. Αυτό το εργαλείο ενσωματώθηκε στη δομή του ενορχηστρωτή για να εκτελεί τόσο στατική όσο και δυναμική ανάλυση των μικροϋπηρεσιών κατά την εκτέλεση εργασιών.

iii Χρήση προγραμματιζόμενων διακοπτών: Οι συμβατικοί διακόπτες στο δίκτυο αντικαθίστανται από προγραμματιζόμενους διακόπτες. Αξιοποιούμε αυτές τις επερχόμενες εξελίξεις για να σχεδιάσουμε και να αναπτύξουμε λειτουργίες που μπορούν να αναπτυχθούν μέσα στους διακόπτες για τον εντοπισμό τυχόν ανωμαλιών κατά την ενορχήστρωση.

2.6 Ενέργεια, δεδομένα και αποδοτικοί υπολογιστικοί μηχανισμοί για την υποστήριξη δυναμικά προσαρμοζόμενων υπηρεσιών και υπηρεσιών ενημέρωσης δικτύου

Α. Νέες υπηρεσίες δεδομένων για εφαρμογές AR και VR.

Οι εφαρμογές VR και AR UC του CHARITY απαιτούν εξειδικευμένες υπηρεσίες δεδομένων για την επίτευξη των στόχων τους. Η ανάπτυξη αυτών των εργασιών απαιτεί σημαντική προσπάθεια, ενώ η απόδοση των υπάρχουσων λύσεων είναι ανεπαρκής για την κάλυψη των απαιτήσεων των UC. Ως αποτέλεσμα, πρέπει να αναπτυχθούν νέοι αλγόριθμοι. Σε αυτήν την ενότητα, περιγράφουμε εν συντομία τέτοιες υπηρεσίες δεδομένων.

Ο Ολογραφικός Βοηθός UC1.3 απαιτεί την αποτελεσματική μετάδοση μεγάλης ποσότητας τρισδιάστατου Point Cloud από το Cloud στη συσκευή 3D Ολογραφίας. Η συμπίεση του Point Cloud είναι ένα ενεργό ερευνητικό θέμα στη Γραφική Υπολογιστών τα τελευταία χρόνια [19]. Παρόλα αυτά, η απόδοση των υφιστάμενων μεθοδολογιών συμπίεσης Point Cloud μπορεί να είναι ανεπαρκής για να εγγυηθεί τον υψηλό όγκο δεδομένων που απαιτείται από αυτό το UC για την παροχή εμπειρίας υψηλής ποιότητας. Συνεπώς πρέπει να αναπτυχθεί ένας προσαρμοσμένος κωδικοποιητής-αποκωδικοποιητής. Στην υπό διερεύνηση προσέγγιση, το Point Cloud απαρτίζεται από πλήθος voxel. Κάθε voxel αντιπροσωπεύει ένα

σημείο στον τρισδιάστατο χώρο με πρόσθετες πληροφορίες που σχετίζονται με αυτό. Ορισμένες από αυτές τις πληροφορίες αφορούν πληροφορίες ορατότητας, χρώμα, διαφάνεια και ούτω καθεξής. Αυτό επιτρέπει την αυξομείωσης της ποιότητας της ανάλυσης του voxel σύμφωνα με το δίκτυο και τους διαθέσιμους υπολογιστικούς πόρους. Υποθέτοντας ότι η σκηνή δεν αλλάζει γρήγορα κατά τη διάρκεια του χρόνου, αυτός ο όγκος πληροφοριών μπορεί να αντιμετωπιστεί ως τρισδιάστατο βίντεο και μόνο οι διαφορές μεταξύ ενός βήματος και του επόμενου να χρειαστεί να μεταδοθούν. Η κύρια ιδέα είναι να κωδικοποιηθούν τέτοιες διαφορές ακολουθώντας προσεγγίσεις που έχουν προκύψει από το V-PCC [20].

Το Συνεργατικό Παίγνιο Επαυξημένης Πραγματικότητας UC3.1 χρειάζεται έναν καλό συγχρονισμό μεταξύ του υλικού περιβάλλοντος και των ενεργειών των παικτών. Για το σκοπό αυτό χρησιμοποιούνται δύο υπηρεσίες δεδομένων, μια υπηρεσία πλέγματος (MS) που δημιουργεί ένα πλέγμα ξεκινώντας από τα τρισδιάστατα σημεία που εξάγονται από τις εικόνες της κάμερας και μια άλλη υπηρεσία δεδομένων που εκτιμά με ακρίβεια τη θέση και τον προσανατολισμό των χρηστών. Η υπηρεσία αυτή ονομάζεται Localization Service (LS). Το MS επιτρέπει στους παίκτες να αλληλεπιδρούν με το περιβάλλον με αποτελεσματικό και διαδραστικό τρόπο. Σε ορισμένες περιπτώσεις, αυτή η υπηρεσία δεν απαιτείται, για παράδειγμα, για smartphones με δυνατότητες 3D tracking, καθώς αυτά που διαθέτουν Lidar. Δημιουργήθηκαν προσεγγίσεις τοπικής προσαρμογής βάσει εικόνας για να δημιουργηθεί το LS. Ιδιαίτερα έμφαση δόθηκε σε προσεγγίσεις 3D tracking με βάση τη δομή, καθώς μέσω αυτού είναι δυνατή η τρισδιάστατη ανακατασκευή του περιβάλλοντος. Στο συγκεκριμένο πλαίσιο αυτού του UC, η απαιτούμενη ακρίβεια είναι υψηλότερη από τις συνηθισμένες εφαρμογές οπτικού εντοπισμού, αλλά δεν παρουσιάζονται τυπικά προβλήματα όπως πιθανές αλλαγές καιρού ή φωτισμού, επειδή μπορούμε να υποθέσουμε ότι το περιβάλλον του παιχνιδιού δεν αλλάζει πολύ από την πρώτη απεικόνιση. Οι σύγχρονες προσεγγίσεις Βαθιάς Μάθησης από άκρο σε άκρο [21], [22] παρέχουν καλές εκτιμήσεις αλλά δεν γενικεύουν ικανοποιητικά. Είναι απαραίτητο να γίνουν εργασίες για τη βελτίωση της γενίκευσης εκμεταλλευόμενοι την υπολογιστική ικανότητα του Cloud για να μπορούν να συντονιστούν τα δίκτυα εν μέσω των διαφόρων διεργασιών και να εξασφαλιστεί εξαιρετικά ακριβή εκτίμηση θέσης και προσανατολισμού.

Β. Αποτελεσματικοί μηχανισμοί αποθήκευσης και προσωρινής αποθήκευσης.

Στο edge computing, ένας μεγάλος όγκος δεδομένων παράγεται και καταναλώνεται από διάφορες Edge εφαρμογές. Μία από τις βασικές προκλήσεις στην ανάπτυξη εφαρμογών στο Edge είναι η αποτελεσματική κοινή χρήση δεδομένων μεταξύ των πολλαπλών Edge υπολογιστών-πελατών. Η κοινή χρήση δεδομένων μπορεί να πραγματοποιηθεί εντός μεμονωμένων πλαισίων εφαρμογών ή μέσω μιας εξωτερικής υπηρεσίας αποθήκευσης. Η αποθήκευση στο Edge μπορεί να βελτιώσει σημαντικά την πρόσβαση στα δεδομένα, η οποία με τη σειρά της εξυπηρετεί εφαρμογές ευαίσθητες σε καθυστέρηση. Παρά τις πρόσφατες εξελίξεις στην κλάδο της παροχής λύσεων αποθήκευσης στο Edge, απομένουν ακόμη ζητήματα που πρέπει να αντιμετωπιστούν. Μερικά από αυτά τα ζητήματα σχετίζονται με τις μη λειτουργικές απαιτήσεις των εφαρμογών που βασίζονται σε Cloud. Επιπλέον, οι Edge κόμβοι έχουν γενικά περιορισμένους πόρους υπολογισμού, αποθήκευσης, δικτύου ή ισχύος και το κατανεμημένο, δυναμικό και ετερογενές περιβάλλον στο Edge μαζί με τις διαφορετικές απαιτήσεις της εφαρμογής θέτει αρκετές προκλήσεις.

Για να αντιμετωπιστούν αυτά τα ζητήματα, πρέπει να αξιοποιηθεί η βασική υποδομή και να επεκταθούν ή να ενσωματωθούν μερικές από τις πιο εξέχουσες λύσεις λογισμικού όπως το MinIO, το OpenStack Swift και το CEPH με υπηρεσίες αποθήκευσης που βασίζονται σε Cloud. Ωστόσο, χρειάζεται μια πιο περίπλοκη λύση για να αντιμετωπιστεί η εγγενής αναξιοπιστία των Edge συσκευών. Η έρευνα για την αποτελεσματική τοποθέτηση δεδομένων διαδραματίζει εξέχοντα ρόλο στην ανάπτυξη μιας αξιόπιστης λύσης αποθήκευσης στο Edge με την ασφάλεια να εμφανίζεται ως συνεχής ανησυχία όταν ετερογενή συστήματα αποθήκευσης σε κόμβους Edge και Cloud πρέπει να ανταλλάσσουν δεδομένα. Επιπλέον, όσον αφορά τη διαχείριση πόρων, παρουσιάζονται πολλές προκλήσεις σχετικά με την προσαρμογή στα δυναμικά περιβάλλοντα και τη βελτιστοποίηση μεγάλης κλίμακας εξαιτίας της ανάγκης για συνεργασία μεταξύ μεγάλου αριθμού παρόχων Edge πόρων. Η βιβλιογραφία παρουσιάζει πολλαπλές επιλογές σχετικά με αυτά τα θέματα. Οι αποφάσεις σχεδόν σε πραγματικό χρόνο μπορούν να βελτιωθούν αισθητά μετακινώντας τα στοιχεία που εκτελούν τις αναλύσεις πιο κοντά στα δεδομένα. Ως αποτέλεσμα, οι Edge αρχιτεκτονικές μπορούν να μειώσουν τον όγκο των δεδομένων που διασχίζουν το δίκτυο, ελαχιστοποιώντας έτσι την καθυστέρηση και το συνολικό κόστος.

Μεταξύ των πιο σχετικών εργασιών, υπάρχει μια πολυεπίπεδη προσέγγιση για τη διαχείριση της αποθήκευσης δεδομένων και ένας προσαρμοστικός αλγόριθμος που υπολογίζει δυναμικά την αντιστάθμιση μεταξύ της ποιότητας και της ποσότητας δεδομένων που είναι αποθηκευμένα στο Edge και στο Cloud [23]. Όσον αφορά το ερώτημα ποια μέρη

των δεδομένων πρέπει να μεταφορτωθούν στο Cloud, προτείνεται ένα μοντέλο κατανεμημένου συστήματος αποθήκευσης πολλαπλών επιπέδων για Edge υπολογιστές [24], το οποίο βασίζεται σε έναν αλγόριθμο αντικατάστασης πολλαπλών παραγόντων LFU (Last Frequency Used).

Επιπλέον, η προσωρινή αποθήκευση στην άκρη μπορεί να βελτιώσει σημαντικά τη διαθεσιμότητα δεδομένων, την δυνατότητα ανάκτησης και να μειώσει τον απαιτούμενο χρόνο παράδοσης [25] . Ως εκ τούτου, απαιτούνται αποτελεσματικά σχήματα μάθησης για δεδομένα μεγάλου όγκου και υψηλών διαστάσεων, προκειμένου να σχεδιαστούν αποτελεσματικοί προληπτικοί αλγόριθμοι προσωρινής αποθήκευσης. Όσον αφορά την ασφάλεια, η πιο δημοφιλής επιλογή είναι η χρήση blockchains. Οι σκέψεις σχετικά με τη χρήση τεχνολογιών και εργαλείων blockchain για την εφαρμογή ενός αποτελεσματικού συστήματος αποθήκευσης στο Edge υπάρχουν εδώ και αρκετό καιρό [26]. Η πρόθεση είναι να αντιμετωπιστούν θέματα όπως η αξιοπιστία του δικτύου και η κατανομή της αποθήκευσης και του υπολογισμού σε μεγάλο αριθμό κατανεμημένων Edge κόμβων με ασφαλή τρόπο.

Η κατευθυντήρια αρχή στο CHARITY είναι η εφαρμογή ενός υβριδικού πλαισίου αποθήκευσης σε Cloud και Edge, το οποίο είναι κατανεμημένο σε ετερογενείς κόμβους Edge και Cloud με βάση ορισμένες έξυπνες αποφάσεις σχετικά με την τοποθέτηση δεδομένων, την προσωρινή αποθήκευση δεδομένων και τις εκτιμήσεις σχετικά με την απόδοση (QoE) και την ασφάλεια. Επιπλέον δίνεται έμφαση στην επίλυση του προβλήματος της διανομής των δεδομένων και της εκφόρτωση με βάση τις απαιτήσεις των εφαρμογής του CHARITY. Καθώς οι παραδοσιακοί αλγόριθμοι LFU που χρησιμοποιούνται στο Edge λαμβάνουν υπόψη μόνο τη συχνότητα πρόσβασης των δεδομένων, το CHARITY θα δανειστεί ιδέες για την αξιολόγηση της «σημασίας» των δεδομένων από διάφορους τομείς, συμπεριλαμβανομένων των καταστημάτων κατανεμημένων δεδομένων με ανοχή σε σφάλματα, βελτιώνοντας έτσι την απόδοση του αποθηκευτικού χώρου.

Επιπλέον, το CHARITY θα παρέχει βελτιστοποιημένη προ-ανάκτηση και τοποθέτηση δεδομένων μέσω έξυπνων μηχανισμών εισαγωγής που είναι σε θέση να προσδιορίσουν το σωστό χρονικό πλαίσιο κατά το οποίο τα δεδομένα θα πρέπει να προ-ανακληθούν στο Edge, διατηρώντας τα σε προσωρινή μνήμη. Θα χρησιμοποιηθούν μηχανισμοί μηχανικής μάθησης και προγνωστικής ανάλυσης, με στόχο τη δημιουργία ενός πιο συγκεκριμένου μοντέλου πρόβλεψης, βελτιστοποιώντας έτσι τη διαδικασία εκφόρτωσης και αποτρέποντας τη δημιουργία σημείων συμφόρησης και παραβιάσεων στις QoS και QoE απαιτήσεις της πλατφόρμας. Επιπλέον, θα χρησιμοποιηθούν αποτελεσματικά σχήματα που βασίζονται σε μηχανική μάθηση προκειμένου να παρέχεται ένας νέος τρόπος ανάπτυξης της προσωρινής αποθήκευσης στο Edge. Τέλος, η ενσωμάτωση του blockchain και του Edge computing θα

αποφέρει κάποια οφέλη όσον αφορά την ασφάλεια, το απόρρητο και την αυτόματη χρήση πόρων.

Γ. Δικτύωση ευαίσθητη σε καθυστέρηση και ευαίσθητη στο εύρος ζώνης.

Στο CHARITY, στόχος είναι η ανάπτυξη ενός δυναμικού πλαισίου δρομολόγησης πολλαπλών διαδρομών για τη βελτίωση της επικοινωνίας από άκρο σε άκρο στο πλαίσιο των αυστηρών απαιτήσεων των AR, VR και βασισμένων σε ολογραφία εφαρμογών. Για να γίνει αυτό, είναι απαραίτητο να αναπτυχθούν μηχανισμοί που μπορούν να διευκολύνουν τον προγραμματισμό και τη δρομολόγηση της κυκλοφορίας που είναι ευαίσθητη σε καθυστέρηση και/ή ευαίσθητη στη διαθεσιμότητα εύρους ζώνης. Το εξάρτημα που θα είναι υπεύθυνο για την παροχή αυτών των λειτουργιών αναφέρεται ως ο μηχανισμός ευφυούς δρομολόγησης της κυκλοφορίας. Ο μηχανισμός Ευφυούς Δρομολόγησης της Κυκλοφορίας αξιοποιεί πληροφορίες σχετικά με τις διάφορες ροές κυκλοφορίας, την τοπολογία του δικτύου και την κατάσταση του δικτύου, προκειμένου να καθιερώσει λειτουργίες δρομολόγησης και προγραμματισμού κυκλοφορίας με τρόπο που να συμμορφώνεται με τις QoS απαιτήσεις. Οι απαιτούμενες πληροφορίες που σχετίζονται με τις ροές κυκλοφορίας είναι οι αντίστοιχες απαιτήσεις πηγής, προορισμού και QoS. Επιπλέον, ο μηχανισμός Ευφυούς Δρομολόγησης Κυκλοφορίας θα καταναλώνει προβλέψεις φόρτου δικτύου που παρέχονται από έναν αποκλειστικό μηχανισμό Πρόβλεψης Φόρτου Δικτύου. Μια πρώτη έκδοση του μηχανισμού αυτού θα υλοποιηθεί στη συνέχεια αυτής της διπλωματικής εργασίας.

Ο Έξυπνος Μηχανισμός Δρομολόγησης Κυκλοφορίας αξιοποιεί τα πρότυπα Software Defined Networking (SDN) προκειμένου να έχει πρόσβαση σε ζωτικής σημασίας πληροφορίες σχετικά με την κίνηση και την τοπολογία του δικτύου. Ο ελεγκτής SDN μπορεί να χρησιμοποιήσει Northbound APIs για να δημιουργήσει επικοινωνία με το επίπεδο εφαρμογής και Southbound API, όπως το OpenFlow, για να επικοινωνήσει με τις συσκευές προώθησης. Αυτά τα κανάλια επικοινωνίας επιτρέπουν στον ελεγκτή SDN να εξετάσει την κατάσταση του δικτύου και τις πληροφορίες που σχετίζονται με τη ροή και στη συνέχεια να τροποποιήσει τους πίνακες ροής των συσκευών προώθησης ανάλογα. Επιπλέον, ο Έξυπνος Μηχανισμός Δρομολόγησης Κυκλοφορίας είναι σχεδιασμένος ώστε για να αξιοποιεί τη Βαθιά Ενισχυτική Μάθηση (Deep Reinforcement Learning) (DRL) προκειμένου να διεξάγει αυτές τις λειτουργίες με τον βέλτιστο τρόπο που είναι σύμφωνος με τις απαιτήσεις QoS. Ο κεντρικός έλεγχος που παρέχεται από το SDN βελτιώνει σημαντικά την ποιότητα της διαδικασίας δρομολόγησης, επιτρέποντας τη δημιουργία κεντρικών πολιτικών δικτύου και στη συνέχεια τη μεταφορά των πολιτικων αυτών στις συσκευές προώθησης. Η διαμόρφωση

του Χώρου Δράσεων του DRL πράκτορα έχει σχεδιαστεί με τρόπο που είναι σύμφωνος με τα πρότυπα του SDN.

Αν και έχουν γίνει πολυάριθμες επιστημονικές προσπάθειες σχετικά με τη χρήση προτύπων που βασίζονται σε DRL στο πλαίσιο του SDN [27], μόνο μερικά από αυτά έχουν σχεδιαστεί για να εξυπηρετούν τη δρομολόγηση πολλαπλών διαδρομών λαμβάνοντας υπόψη τους περιορισμούς QoS [28]. Το CHARITY στοχεύει να διευρύνει στην υπάρχουσα επιστημονική βιβλιογραφία [29] όσον αφορά την ανάπτυξη δομών που βασίζονται σε DRL με επίγνωση του QoS που υποστηρίζουν δρομολόγηση πολλαπλών διαδρομών. Για το σκοπό αυτό, ο Χώρος Δράσεων θα πρέπει επίσης να μοντελοποιηθεί με τρόπο που να μπορεί να αντικατοπτρίζει σωστά την πολυπλοκότητα της δρομολόγησης πολλαπλών διαδρομών. Επιπλέον, ο Χώρος Καταστάσεων πρέπει να υλοποιηθεί με τρόπο που να περιλαμβάνει τις προβλέψεις κυκλοφορίας. Με αυτόν τον τρόπο, είναι δυνατό να καταστεί δυνατή η δυναμική δημιουργία πολιτικών που λαμβάνουν υπόψη την αναμενόμενη μελλοντική κατάσταση του δικτύου καθώς και την τρέχουσα. Τέλος, ο Έξυπνος Μηχανισμός Δρομολόγησης Κυκλοφορίας θα αξιοποιήσει επίσης τα Νευρωνικά Δίκτυα Γράφων (Graph Neural Networks) (GNN) προκειμένου να βελτιώσει την αποτελεσματικότητα των αλγορίθμων δρομολόγησης που βασίζονται σε DRL [30]. Η χρήση GNN θα επιτρέψει στις δομές του δικτύου να αναπαρασταθούν με πιο ακριβή τρόπο, ενσωματώνοντας σωστά τις περίπλοκες σχέσεις που δημιουργούνται στις δομές που βασίζονται σε γραφήματα.

2.7 Ένα πλαίσιο κατάλληλο για τη διαχείριση των εφαρμογών

Μαζί με την παροχή προηγμένων ενεργοποιητών υπηρεσιών XR, το CHARITY έχει στόχο να καταστήσει αυτές τις βελτιωμένες δυνατότητες όσο το δυνατόν πιο προσιτές και εύχρηστες στους προγραμματιστές εφαρμογών XR, για να υποστηρίξουν βελτιώσεις στον κύκλο ανάπτυξης των εφαρμογών XR, όσον αφορά την ταχύτητα, το κόστος και την αποτελεσματικότητα. Το Application Management Framework (AMF) θα είναι ένα στοιχείο που θα επιτρέπει την πρόσβαση σε αυτές τις δυνατότητες. Η ραχοκοκαλιά του είναι το μοντέλο CI/CD (Continuous Integration/Continuous Delivery), προσαρμοσμένο και ερμηνευμένο σύμφωνα με τις ανάγκες και τις ιδιαιτερότητες του CHARITY, ενσωματωμένο με την ανάπτυξη προσαρμοσμένων εργαλείων. Η δημιουργία σχεδίων τμημάτων του δικτύου είναι μια βασική έννοια του CHARITY και είναι ανάλογη με τον μηχανισμό περιγραφής των υπηρεσιών δικτύου που αναφέρεται στο [31]. Το AMF θα επιτρέψει στους προγραμματιστές

εφαρμογών XR να σχεδιάσουν τα δικά τους αφηρημένα σχέδια τμημάτων δικτύου. Αυτοί οι αφηρημένοι μηχανισμοί περιγραφής θα μετατρέπονται σε συγκεκριμένους μηχανισμούς περιγραφής από ένα στοιχείο του επιπέδου ενορχήστρωσης XR υπηρεσιών, προκειμένου να δημιουργηθούν περιγραφές που είναι έτοιμες να μετατραπούν σε αντικείμενα στην πλατφόρμα CHARITY. Έτσι, το AMF είναι το κύριο σημείο εισόδου για τους προγραμματιστές XR εφαρμογών προκειμένου να ορίσουν και να χειριστούν τις XR υπηρεσίες τους. Οι ενέργειες αυτές μπορεί να περιλαμβάνουν τα ακόλουθα:

• Σχεδιασμός αφηρημένων τμημάτων δικτύου, που προορίζονται ως μείγματα Λειτουργιών Εικονικού Δικτύου και Υπηρεσιών Δικτύου. Το μείγμα περιλαμβάνει τεχνουργήματα που παρέχονται από το CHARITY (ενεργοποιητές XR υπηρεσιών) και εικονικούς συνδέσμους.

• Επικύρωση σχεδιαγραμμάτων τμημάτων δικτύου που δημιουργούνται.

• Εγγραφή και αποθήκευση σε κοινό αποθετήριο (XR Service Blueprint Templates Repository).

• Ενημέρωση των ήδη καταχωρημένων σχεδιαγραμμάτων τμημάτων δικτύου.

Πέρα από τη δημιουργία και τη διαχείριση σχεδιαγραμμάτων, η AMF επιτρέπει στον προγραμματιστή XR εφαρμογών να ενσωματώσει τις εφαρμογές XR στην πλατφόρμα CHARITY μέσω:

• Εγγραφή/Ενσωμάτωση XR εφαρμογής. Αυτό το μέρος περιλαμβάνει τη μεταφόρτωση των στοιχείων της εφαρμογής σε ένα κοινόχρηστο αποθετήριο, συνοδευόμενο από μια κατάλληλη αφηρημένη περιγραφή.

• Ορισμός προτύπων μοντέλων εφαρμογών που περιγράφουν τα διαφορετικά στοιχεία της εφαρμογής, μαζί με την περιγραφή της διασύνδεσης και της διαλειτουργικότητάς τους.

• Επικύρωση των σύνθετων εφαρμογών σε ένα διαχωρισμένο περιβάλλον δοκιμών, σύμφωνα με τις δοκιμές που έχουν οριστεί από τους XR Application Developers.

• Διαχείριση δυναμικών αλλαγών στο μοντέλο της εφαρμογής κατά την εκτέλεση της εφαρμογής, συμπεριλαμβανομένων ενημερώσεων σε ήδη εκτελούμενες μικροϋπηρεσίες. Αυτό συμπεριλαμβάνει και τη δυνατότητα προσθήκης νέων μικροϋπηρεσιών ή παροπλισμού των ήδη εκτελούμενων, διατηρώντας την απαιτούμενη συνέπεια με τα αποθετήρια αφηρημένων μοντέλων εφαρμογής.

Κάθε περιγραφόμενο τεχνούργημα, που θα πρέπει να επικυρώνεται εσωτερικά μέσω δοκιμών, πιθανώς μαζί με εγγενή εξαρτήματα του CHARITY. Η επικύρωση θα πρέπει να γίνεται σε περιβάλλον δοκιμής. Παραδείγματος χάρη:

• Μέσω δοκιμής ενός συστατικού (εάν παρέχεται από τους XR Application Developers).

• Μέσω δοκιμής ενσωμάτωσης που παρέχεται από τους NextGen Application Developers, η οποία εκτελείται με στοιχεία CHARITY (mocks ή full).

• Σάρωση ασφαλείας.

Για τη διαχείριση των XR εφαρμογών, τα στοιχεία του επιπέδου AMF θα χρησιμοποιήσουν δύο προσεγγίσεις: χαλαρά συζευγμένες διεπαφές με το επίπεδο ενορχήστρωσης XR υπηρεσιών για την υλοποίηση ενός μηχανισμού δημοσίευσης/εγγραφής και ορισμένους κοινόχρηστους χώρους αποθήκευσης για την αποθήκευση τεχνουργημάτων που δημιουργούνται από το AMF και αντίστροφα.

3. Πρόβλεψη χρήσης πόρων σε Edge συστήματα

Κατά την τελευταία δεκαετία, υπήρξε μια αυξανόμενη ανάγκη να έρθουν η επεξεργασία και τα δεδομένα πιο κοντά στις συσκευές όπου παράγονται [32]. Αυτές οι συσκευές μπορεί να περιλαμβάνουν έξυπνα αντικείμενα, κινητά τηλέφωνα, πύλες δικτύου και αισθητήρες Internet of Things (IoT). Αυτό το κατανεμημένο υπολογιστικό μοντέλο, που ορίζεται ως Edge, στοχεύει στο να δημιουργήσει αποκεντρωμένες τοπολογίες και να επιτρέψει τη μετεγκατάσταση διαφόρων υπολογιστικών και αποθηκευτικών πόρων πιο κοντά στην άκρη του δικτύου. Με αυτόν τον τρόπο, αναμένεται να παρέχει παράδοση υπηρεσιών και εξαργύρωση περιεχομένου σε καλύτερους χρόνους απόκρισης και ταχύτητες μεταφοράς.

Κατά την εξέταση της φύσης και των απαιτήσεων των σύγχρονων εφαρμογών γίνεται σαφές ότι η εκτέλεσή τους σε αποτελεσματικά ενορχηστρωμένους κόμβους επεξεργασίας είναι μείζονος σημασίας προκειμένου να πληρούνται τα πρότυπα Ποιότητας Υπηρεσίας (QoS) που ορίζει ο κλάδος. Πιο συγκεκριμένα, η αύξηση των συσκευών IoT στην άκρη του δικτύου έχει ως αποτέλεσμα την παραγωγή τεράστιων ποσοτήτων δεδομένων και φόρτου εργασίας [33]. Αυτή η πρωτόγνωρη κατάσταση γέννησε την ανάγκη για μια ευφυή και προσαρμοστική διαχείριση των υπολογιστικών πόρων [34], οι οποίοι είναι σε θέση να παρέχουν υψηλή απόδοση και χαμηλή καθυστέρηση με τον περιορισμό ότι οι κόμβοι επεξεργασίας στην άκρη είναι περιορισμένοι σε αριθμό.

Υπηρεσίες ενορχήστρωσης στα πλαίσια του Edge Computing, όπως η εκφόρτωση εργασιών και η προσαρμοστική κατανομή πόρων μπορούν να κάνουν καλύτερη χρήση των κόμβων επεξεργασίας με ένα μηχανισμό πρόγνωσης της χρήσης των πόρων. Η κύρια περιοχή εστίασης είναι η ένταση με την οποία λειτουργούν η CPU, η μνήμη RAM, το εύρος ζώνης και ο δίσκος. Αυτές οι μετρήσεις έχουν τιμές που παρουσιάζουν διασταυρούμενη συσχέτισης, καθιστώντας λογική επιλογή την προσέγγιση με χρήση χρονοσειρών [35]. Ένα μοντέλο πολλαπλής παλινδρόμησης επαναλαμβανόμενου νευρωνικού δικτύου (RNN) [36]που αξιοποιεί τα χαρακτηριστικά χρονοσειρών μέσω των Gated Recurrent Units (GRU) [37] μπορεί να είναι μια εξέχουσα λύση για την πρόβλεψη των μετρήσεων πόρων.

Τα προαναφερθέντα γεγονότα αποτέλεσαν κίνητρο για τη δημιουργία ενός μοντέλου GRU-RNN πολλαπλής παλινδρόμησης που προβλέπει με ενοποιημένο τρόπο τη χρήση πόρων των Edge κόμβων επεξεργασίας, αξιοποιώντας έτσι τα χαρακτηριστικά των χρονοσειρών τους.

3.1 Σχετικές μελέτες στον κλάδο της πρόβλεψης χρήσης πόρων.

Πρόσφατες μελέτες έχουν δείξει ότι η πρόβλεψη χρήσης πόρων σε ένα υπολογιστικό σύστημα Edge και Cloud μπορεί να είναι βασική προϋπόθεση για την αποτελεσματική διαχείριση πόρων και την διασφάλιση του QoS για τους χρήστες [38]. Τα περισσότερα από αυτά επικεντρώνονται σε στατιστικά μοντέλα, μοντέλα Μηχανικής Μάθησης και Βαθιάς Μάθησης μοντέλα. Μερικά από αυτά αξιοποιούν τα επαναλαμβανόμενα νευρωνικά δίκτυα (Recurrent Neural Networks) (RNN) και συγκεκριμένα το δίκτυα LSTM (Long Short Term Memory), διαμορφώνοντας τις μετρήσεις χρήσης πόρων ως ακολουθίες δεδομένων. Αλλά κανένα από αυτά δεν επικεντρώνεται στα πλεονεκτήματα των GRUs (Gated Recurrent Units) σε σύγκριση με τα LSTM δίκτυα.

Ο Autoregressive Integrated Moving Average (ARIMA) είναι ένα αρκετά δημοφιλές στατιστικό μοντέλο για την πρόβλεψη χρονοσειρών που βασίζεται στην αυτόματη παλινδρόμηση και σταθμίζει μια σειρά από καθυστερημένες παρατηρήσεις με βάση το πόσο πρόσφατες είναι. Το ARIMA έχει χρησιμοποιηθεί για την αποφυγή υποπαροχής ή υπερπαροχής πόρων σε κέντρα δεδομένων . Έγινε χρήση του ARIMA στην πειραματική μας σύγκριση που παρέχεται από τα statsmodels της python [39].

Η Μηχανική Μάθηση προσεγγίζει τη δημιουργία μοντέλων με βάση ιστορικά δεδομένα προκειμένου να κάνει προβλέψεις σε νέες περιπτώσεις. Η Μηχανική Μάθηση χρησιμοποιείται ευρέως σε Edge και Cloud computing για τρεις σκοπούς: (α) χαρακτηρισμό και πρόβλεψη φόρτου εργασίας, (β) τοποθέτηση εξαρτημάτων και ενοποίηση συστήματος και (γ) ελαστικότητα και αποκατάσταση λειτουργικότητας εφαρμογής [40]. Ενώ υπάρχουν πολλά κλασικά μοντέλα ML διαθέσιμα στο κοινό, επιλέχθηκε για τα πειράματά να γίνει χρήση του XGBoost [41] επειδή είναι δημοφιλές για τη νίκη στο Kaggle και σε άλλους διακεκριμένους διαγωνισμούς Μηχανικής Μάθησης [42]. Το XGBoost χρησιμοποιεί ως επί το πλείστον δέντρα αποφάσεων ενισχυμένα με κλίση και είναι διαθέσιμο ως βιβλιοθήκη λογισμικού ανοιχτού κώδικα.

Ο περιορισμός των προσεγγίσεων Μηχανικής Μάθησης είναι ότι κάθε φορά που αλλάζει η υποδομή Edge ή Cloud, η συμπεριφορά του χρήστη ή η εφαρμογή, τα νέα μοντέλα θα πρέπει να εκπαιδεύονται από την αρχή με ανθρώπινη βοήθεια. Η αυτόματη μηχανική μάθηση επιτυγχάνεται με έναν αυτοματοποιημένο τρόπο καθοδήγησης της διαδικασίας εκμάθησης των μοντέλων, μεγιστοποιώντας την απόδοση, ελαχιστοποιώντας το υπολογιστικό κόστος και χωρίς ανθρώπινη συμμετοχή. Στον τομέα του Cloud Computing, το AUCROP [43] επιτυγχάνει την αυτοματοποιημένη χρήση κλασικών αλγορίθμων ML. Επιπλέον, ένα αυτοματοποιημένο μετα-μοντέλο Μηχανικής Μάθησης γενικής χρήσης για

προεπεξεργασία δεδομένων, παλινδρόμηση και συντονισμό υπερπαραμέτρων μέσω της Bayesian βελτιστοποίησης είναι το Auto-sklearn παρουσιάζεται στο [44].

Τα RNNs είναι μια κατηγορία μοντέλων βαθιάς Μάθησης ικανών για πρόβλεψη χρονοσειρών. Τα RNNs και συγκεκριμένα τα LSTMs έχουν χρησιμοποιηθεί με επιτυχία στο παρελθόν για την πρόβλεψη χρήσης πόρων ξεπερνώντας το μοντέλο ARIMA στην πρόβλεψη χρονοσειρών της χρήσης CPU [38]. Επιπλέον έχει γίνει χρήση GA-LSTM με σκοπό τη πρόβλεψη χρήσης πόρων.

Το GRU και το LSTM έχουν πολλά κοινά. Ωστόσο, το GRU έχει απλούστερη δομή και καλύτερη ευελιξία. Μια πειραματική σύγκριση μεταξύ αυτών των δύο παραλλαγών του RNN είχε ζητηθεί από τους συμμετέχοντες στην προηγούμενη παρουσίασή μας στο συνέδριο [45]. Επεκτείναμε επίσης την έρευνα στον κλάδο της βελτιστοποίησης υπερπαραμέτρων συνδυάζοντας την εξελικτική στρατηγική με τη Bayesian βελτιστοποίηση.

3.2 Η αναγκαιότητα πρόβλεψης χρήσης πόρων στα Edge Συστήματα

Η διαχείριση και η ενορχήστρωση των υπολογιστικών Edge υποδομών μπορεί να βελτιωθεί με την πρόβλεψη των μετρήσεων χρήσης πόρων που λαμβάνουν χώρα με ένα μοντέλο RNN. Κυρίως αυτές οι μετρήσεις είναι η CPU, η RAM, το εύρος ζώνης και η είσοδος/έξοδος του δίσκου. Το Edge computing χαρακτηρίζεται από τη δυναμική συμπεριφορά και την ετερογένεια των Edge κόμβων επεξεργασίας σε συνδυασμό με περιορισμούς που εισάγονται από τις Συμφωνίες Επιπέδου Υπηρεσιών (Service Level Agreements) (SLAs).

Η δυναμική συμπεριφορά των κόμβων ακμών οφείλεται στη διακύμανση των αιτημάτων εφαρμογής και του φόρτου εργασίας. Ο αριθμός των αιτημάτων ανά χρονικό διάστημα αλλάζει κατά τη διάρκεια των ημερών και πολλά περιοδικά φαινόμενα μπορεί να τον επηρεάσουν. Η ετερογένεια σημαίνει ότι οι κόμβοι ακμών έχουν ως επί το πλείστον διαφορετικά χαρακτηριστικά ως προς το μέγεθος της μνήμης και την υπολογιστική ισχύ. Μπορούμε να αναλογιστούμε αυτήν την ετερογένεια λαμβάνοντας υπόψη τις διαφορετικές γεύσεις του Raspberry Pis που είναι διαθέσιμες, συνυπάρχουν και συνεργάζονται σε μια υποδομή αιχμής.

Τα SLAs θέτουν τους περιορισμούς απόδοσης για τους Edge κόμβους όσον αφορά τη διαθεσιμότητα, την απόδοση και τους διαφορετικούς τύπους καθυστερήσεων. Οι πάροχοι αιχμής αγωνίζονται για να μην παραβιαστούν τα SLAs. Έτσι, πρέπει να λαμβάνονται

έγκαιρες και βέλτιστες αποφάσεις σχετικά με την εκφόρτωση εργασιών και την κατανομή πόρων.

3.2.1 Προσαρμοστική κατανομή πόρων

Η απόδοση των εφαρμογών εξαρτάται στενά από τους διαθέσιμους πόρους στις συσκευές τελικού χρήστη και στους Edge κόμβους. Προκειμένου να ξεπεραστούν οι υπολογιστικοί περιορισμοί, είναι δυνατή η δέσμευση επιπρόσθετων υπολογιστικών Edge πόρων, οι οποίοι χειρίζονται σωστά τον φόρτο εργασίας που δημιουργείται από τις διάφορες εργασίες. Υπάρχουν δύο τρόποι με τους οποίους μπορεί να επιτευχθεί αυτή η λειτουργικότητα. Το πρώτο αναφέρεται ως οριζόντια αυξομείωση και είναι η διαδικασία απόκτησης πρόσθετων υπολογιστικών κόμβων ενδεχομένως από διαφορετικούς κεντρικούς υπολογιστές. Η δεύτερη ονομάζεται κατακόρυφη αυξομείωση και είναι η διαδικασία αύξησης της υπολογιστικής ικανότητας των ήδη υπάρχοντων κόμβων.

Στην περίπτωση της οριζόντιας αυξομείωσης, είναι ζωτικής σημασίας να έχουμε μια ήδη υπάρχουσα εκτίμηση της σχετικά βραχυπρόθεσμης αναμενόμενης χρήσης των πόρων, επειδή η δημιουργία μιας εικονικής μηχανής απαιτεί ένα χρονικό διάστημα αρκετών λεπτών. Αυτό το ατυχές χρονικό όριο προκαλεί ένα αρκετά σημαντικό πλήγμα στην ικανότητα του δικτύου να αντιδρά έγκαιρα σε περίπτωση ξαφνικής αύξησης στη χρήση του δικτύου.

3.2.2 Έξυπνη εκφόρτωση εργασιών

Η εκφόρτωση εργασιών αναφέρεται στη διαδικασία επιλογής συγκεκριμένων πόρων του δικτύου για τη διαχείριση διαφόρων εργασιών, ανάλογα με τις απαιτήσεις τους. Αυτή η διαδικασία βασίζεται στη μάλλον απλή αρχή της σωστής κατανομής του φόρτου εργασίας μεταξύ των διαθέσιμων υπολογιστικών και αποθηκευτικών πόρων προκειμένου να επιτευχθεί καλύτερος χρόνος απόκρισης. Ο μηχανισμός απόφασης εκφόρτωσης εργασιών, αφού εξετάσει τη συνεχή χρήση πόρων κάθε Edge κόμβου που λειτουργεί, θα αποφασίσει ποιος από αυτούς θα λάβει την επόμενη εργασία. Ο κύριος σκοπός της διαδικασίας επιλογής Edge κόμβου είναι να αποφευχθεί η επιλογή ενός Edge κόμβου, ο οποίος ήδη λειτουργεί κοντά στη μέγιστη δυνατότητα του. Επιπλέον, δεν μπορεί να μην παρατηρήσει κανείς ότι οι λειτουργίες της έξυπνης εκφόρτωσης εργασιών και της προσαρμοστικής κατανομής πόρων είναι στενά αλληλένδετες. Μέσω της διασφάλισης ότι οι διάφοροι Edge κόμβοι δεν λειτουργούν κοντά στη μέγιστη δυνατή τους ικανότητα, το χρονικό πλαίσιο που το δίκτυο μπορεί να ανταποκριθεί σε πιθανή, ξαφνική έκρηξη κίνησης διευρύνεται σημαντικά.

3.2.3 Προληπτική ανοχή σφαλμάτων

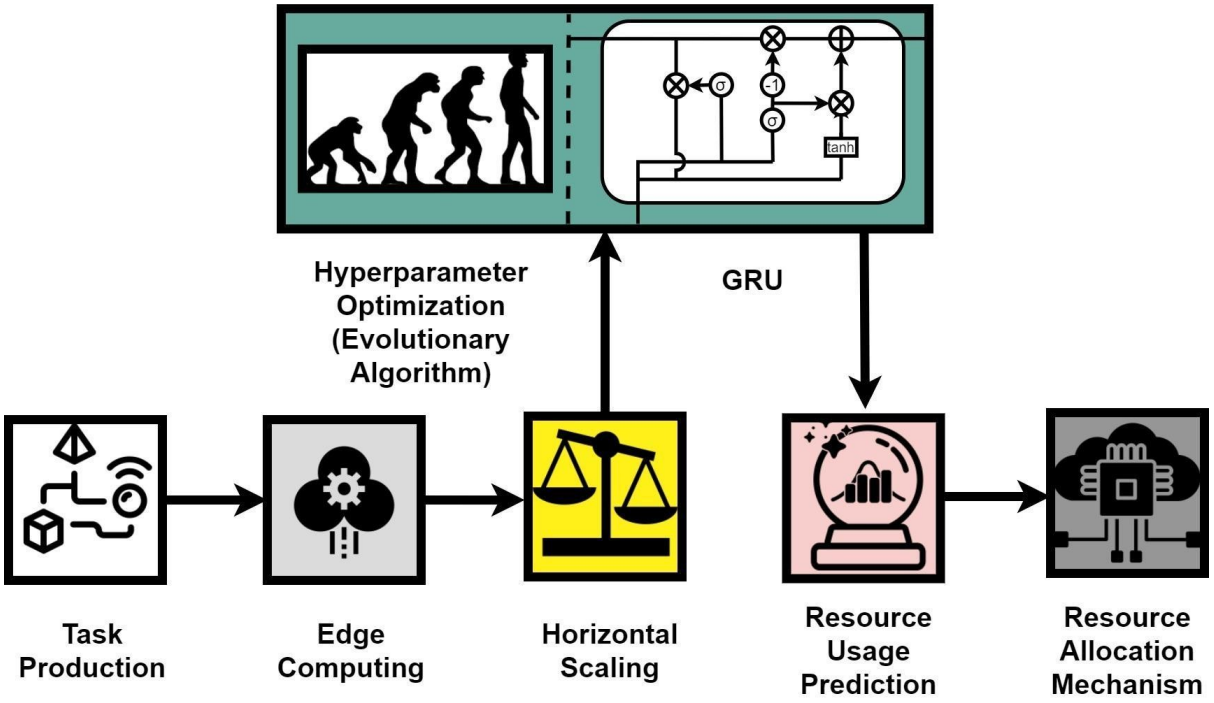
Δεδομένου ότι το modus operandi του Edge Computing αποτελείται από έναν τεράστιο αριθμό υπολογιστικών κόμβων που λειτουργούν ταυτόχρονα, είναι εξαιρετικά σημαντικό να θεωρείται η δυσλειτουργία εξαρτημάτων ως κάτι το αναπόφευκτο. Με αυτόν τον τρόπο είναι δυνατό να διασφαλιστεί ότι το δίκτυο θα συνεχίσει να λειτουργεί χωρίς διακοπή όταν ένα ή περισσότερα από τα στοιχεία του αποτυγχάνουν. Ο κύριος τρόπος για να διασφαλιστεί ότι ένα δίκτυο θα μπορεί να συνεχίσει να εκτελεί τη λειτουργία του ακόμη και μετά από μια κρίσιμη αποτυχία είναι με τη χρήση εφεδρικών στοιχείων, τα οποία αντικαθιστούν αυτόματα τα αποτυχημένα, με τρόπο που εγγυάται μη απώλεια της υπηρεσίας.

Όπως εξηγήθηκε στην προηγούμενη υποενότητα, η δημιουργία μιας νέας εικονικής μηχανής απαιτεί μια ορισμένη περίοδο αναμονής, η οποία θα είχε σοβαρές επιπτώσεις στην αποτελεσματικότητα του δικτύου. Έτσι, η ανοχή σφαλμάτων μπορεί να επωφεληθεί από τη λήψη προληπτικών αποφάσεων. Σε κάθε δεδομένη στιγμή, το δίκτυο θα πρέπει να περιέχει έναν συγκεκριμένο αριθμό υπολογιστικών κόμβων, οι οποίοι παραμένουν σε αδράνεια έως ότου ένα από τα στοιχεία που ήδη λειτουργούν ορθά πάψει να λειτουργεί σωστά. Με τη χρήση αλγορίθμων Μηχανικής Μάθησης είναι δυνατή η εξαγωγή πληροφοριών σχετικά με τα πρότυπα χρήσης πόρων, των επιρρεπών στην δυσλειτουργία μηχανήματα και το χρονικό πλαίσιο στο οποίο εμφανίζονται τα σφάλματα. Αυτό επιτρέπει την προληπτική ανοχή σφαλμάτων για να διασφαλιστεί ότι οι λειτουργίες του δικτύου θα συνεχίσουν να λειτουργούν αδιάκοπα και ότι το συνολικό κόστος θα περιοριστεί στο ελάχιστο.

4. Χρήση επαναλαμβανόμενων νευρωνικών δικτύων

Ένα GRU-RNN μοντέλο προτείνεται ως ένα ακριβές και αποτελεσματικό μοντέλο πρόβλεψης που ικανοποιεί τις ιδιαιτερότητες των μετρήσεων χρήσης πόρων αιχμής. Επειδή οι μετρικές πόρων όπως η CPU, η RAM, ο δίσκος και το εύρος ζώνης διαμορφώνουν ακολουθίες με υψηλή ομοιότητα, το RNN μοντέλο είναι μια κατάλληλη προσέγγιση. Το RNN μοντέλο συνδυάζει τα πλεονεκτήματα της Βαθιάς Μάθησης με τα χαρακτηριστικά της πρόβλεψης χρονοσειρών. Υπάρχουν διάφοροι τύποι αρχιτεκτονικών RNN. Στη προκειμένη περίπτωση, γίνεται χρήση των GRUs επειδή μπορούν να μάθουν μακροπρόθεσμες χρονικές εξαρτήσεις, είναι υπολογιστικά αποδοτικά και έχουν καλή απόδοση σε μικρότερα σύνολα δεδομένων.

Η ροή εργασιών του εργαλείου πρόβλεψης της χρήσης πόρων σε ένα περιβάλλον Edge Computing απεικονίζεται στο σχήμα 3.



Σχήμα 3: Η ροή εργασιών του εργαλείου πρόβλεψης της χρήσης πόρων σε ένα περιβάλλον Edge Computing.

Σε πρώτο στάδιο, οι Edge συσκευές δημιουργούν εργασίες που εκφορτώνονται μερικώς ή πλήρως στην Edge υποδομή. Η ενορχήστρωση των Edge υπολογιστικών πόρων

περιλαμβάνει έναν έξυπνο μηχανισμό οριζόντιας αυξομείωσης που υπολογίζει μια σχεδόν βέλτιστη αρχιτεκτονική GRU χρησιμοποιώντας τον αλγόριθμο HBES.

Το GRU μοντέλο πρόβλεψης παρέχει την πρόβλεψη χρήσης πόρων που μπορεί να χρησιμοποιηθεί για προσαρμοστική κατανομή πόρων ή εκφόρτωση εργασιών. Στις επόμενες υποενότητες, θα γίνει περιγραφή της θεωρίας πίσω από τα βασικά μέρη του GRU μοντέλου για την πρόβλεψη χρήσης πόρων.

4.1 Επαναλαμβανόμενα νευρωνικά δίκτυα

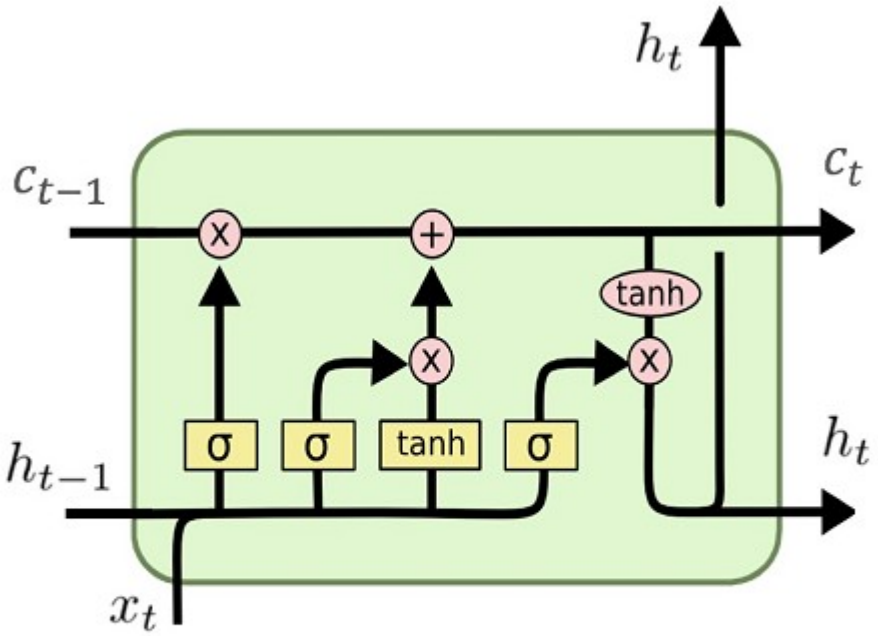
Τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks) (ANN) μπορούν να οριστούν ως μέσα προσέγγισης συναρτήσεων που μπορούν να αντιστοιχίζουν αναπαραστάσεις δεδομένων χαμηλότερου επιπέδου σε αναπαραστάσεις δεδομένων υψηλότερης διαστατικότητας. Τα RNNs είναι ένας τύπος ANN, που αντιλαμβάνεται τη δυναμική χρονική συμπεριφορά, συλλαμβάνοντας ακολουθίες δεδομένων και διατηρώντας την προηγούμενη κατάσταση εισόδου.

Από αρχιτεκτονικής απόψεως, το RNN μοντέλο βασίζεται σε διάφορους κόμβους που μοιάζουν με νευρώνες οργανωμένους σε διαδοχικά επίπεδα, όπου κάθε κόμβος συνδέεται με κόμβους του επόμενου διαδοχικού στρώματος, διατηρώντας επαναλαμβανόμενες συνδέσεις. Η εφαρμογή αυτής της συγκεκριμένης ιδέας επιτρέπει στις πληροφορίες που προκύπτουν από προηγούμενες εισροές δεδομένων να επηρεάσουν τις μελλοντικές εξόδους, καθιστώντας έτσι το RNN μια σταθερή επιλογή για μοντελοποίηση χρονοσειρών, λαμβάνοντας υπόψην τις συμφραζόμενες πληροφορίες. Παρακολουθώντας τις Edge υπολογιστικές υποδομές, συλλέγονται διαδοχικά δεδομένα και προβλέπονται οι μελλοντικές μετρήσεις χρήσης πόρων από το RNN με βάση τις τρέχουσες και τις προηγούμενες τιμές τους.

4.1.1 Long Short Term Memory

Η αρχιτεκτονική δικτύου LSTM δημιουργήθηκε για να αντιμετωπίσει το πρόβλημα της εξαφάνισης των κλίσεων στα RNNs. Όταν αντιμετωπίζουμε ένα πρόβλημα δεδομένων χρονοσειρών, όπως η πρόβλεψη της χρήσης πόρων, μπορούμε να λάβουμε πολύ πιο χρήσιμες πληροφορίες αν κοιτάξουμε τα ιστορικά δεδομένα της χρήσης των μηχανημάτων μας, αντί να κοιτάξουμε απλώς την τρέχουσα κατάστασή τους. Με αυτόν τον τρόπο, μπορούμε να κατανοήσουμε καλύτερα έννοιες όπως η τάση, οι οποίες εξηγούνται μόνο στα πλαίσια της παρόδου του του χρόνου. Τα LSTM, ακριβώς όπως τα απλά RNN,

χρησιμοποιούν την Κρυφή κατάσταση για να συνδέσουν διαδοχικούς κόμβους, ώστε να επιτρέπουν την καλύτερη αξιοποίηση των χρονικών δεδομένων. Ωστόσο, χρησιμοποιείται επίσης μια ατομική κατάσταση, η οποία αποτελεί άλλη σύνδεση μεταξύ των κόμβων. Κάθε μονάδα LSTM μπορεί να διαβάσει από την ατομική κατάσταση, να τη διαμορφώσει ή να την επαναφέρει μέσω της χρήσης πυλών. Η αρχιτεκτονική των μονάδων LSTM απεικονίζεται στο Σχήμα 4.



Σχήμα 4

LSTM
(Long-Short Term Memory)

Υπάρχουν συνολικά τρεις πύλες, καθεμία από τις οποίες κάνει χρήση μίας σιγμοειδούς συνάρτησης. Αυτό διασφαλίζει ότι το μοντέλο παραμένει διαφοροποιήσιμο, καθώς το σιγμοειδές επίπεδο προσφέρει ομαλές καμπύλες στην περιοχή από 0 έως 1. Κάθε μία από τις πύλες παίρνει ως είσοδο την είσοδο του συστήματος καθώς και την κρυφή κατάσταση του προηγούμενου χρονικού βήματος. Εκτός από τις πύλες, ένα διάνυσμα **C** είναι υπεύθυνο για τη μεταφορά των υποψήφιων πληροφοριών που μπορούν να προστεθούν στην ατομική κατάσταση. Το **C** χρησιμοποιεί ένα στρώμα tanh, το οποίο είναι υπεύθυνο για τον περιορισμό του φαινομένου της εξαφάνισης της κλίσης. Έτσι, οι πληροφορίες της εκάστοτε LSTM μονάδας μπορούν να διατηρηθούν περισσότερο χωρίς να εξαφανιστούν. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό είναι διατηρώντας τις διαβαθμίσεις μηδενικές στο κέντρο, μεταξύ των τιμών -1 και 1.

Η πύλη εισόδου διαχειρίζεται τα εισερχόμενα δεδομένα και ελέγχει εάν η μνήμη της μονάδας πρέπει να ενημερωθεί. Εφαρμόζεται στο **C** και το αποτέλεσμα προστίθεται στη συνέχεια στην ατομική κατάσταση. Η σιγμοειδοποίηση της πύλης χρησιμοποιείται είτε για να μετριάσει είτε να ενισχύσει την επίδραση που θα πρέπει να έχουν οι νέες πληροφορίες στην ατομική κατάσταση.

Η πύλη λήθης είναι η οντότητα που είναι υπεύθυνη για την επιλογή των πληροφοριών που θεωρούνται λιγότερο σημαντικές και την αφαίρεση τους από την ατομική κατάσταση. Επιπλέον, η χρήση της σιγμοειδούς συνάρτησης, παράγει μια κλιμακούμενη έξοδο για κάθε τιμή που αποθηκεύεται στην ατομική κατάσταση.

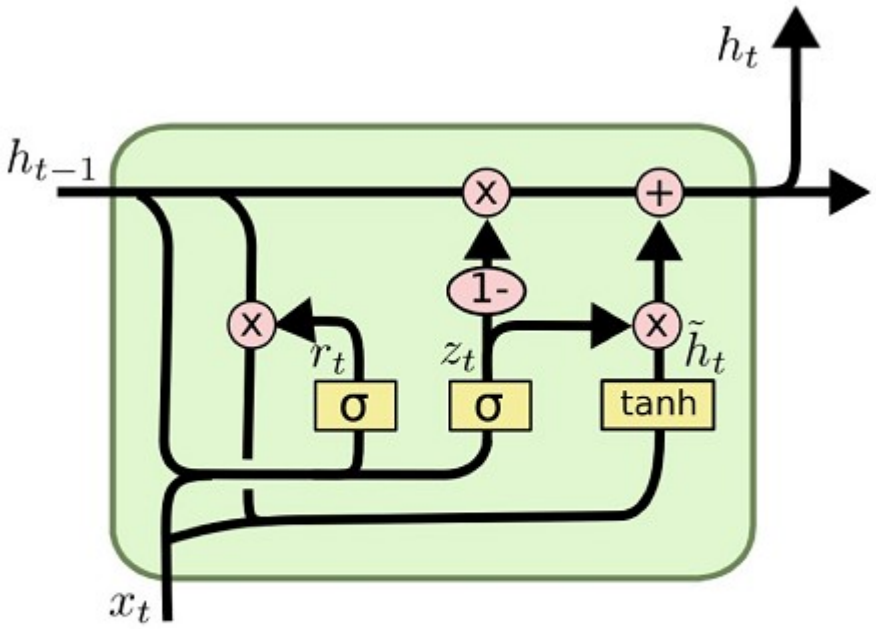
Τέλος, η πύλη εξόδου είναι το τελικό επίπεδο πριν από την παραγωγή της νέας κρυφής κατάστασης. Χρησιμοποιεί τη σιγμοειδή συνάρτηση ως φίλτρο που θα εφαρμοστεί στην ατομική κατάσταση αφού περάσει πρώτα από ένα επίπεδο tanh. Αφού ολοκληρωθεί αυτή η διαδικασία, τόσο η κρυφή κατάσταση όσο και η ατομική κατάσταση συνθέτουν την έξοδο του κελιού LSTM που θα εισαχθεί στο επόμενο χρονικό βήμα.

4.1.2 Gated Recurrent Units

Διάφορες αρχιτεκτονικές που σχετίζονται με πύλες έχουν χρησιμοποιηθεί προκειμένου να αντιμετωπιστεί το πρόβλημα της εξαφάνισης της κλίσης που είναι μία από τις κύριες προκλήσεις στα RNNs. Μέσω της χρήσης πυλών, το εκάστοτε νευρωνικό δίκτυο είναι σε θέση να διατηρεί τις σημαντικές πληροφορίες και να τις τροφοδοτεί με επιτυχία στα επόμενα χρονικά βήματα. Τα δύο πιο αξιοσημείωτα είναι τα δίκτυα Long Short Term Memory (LSTM) και τα δίκτυα Gated Recurrent Units (GRU) [46]. Τα GRUs είναι παρόμοια με τα LSTMs. Και τα δύο καταφέρνουν να αποτρέψουν το πρόβλημα της εξαφάνισης της κλίσης χρησιμοποιώντας δομές πύλης. Αυτό που τα ξεχωρίζει είναι το γεγονός ότι τα GRUs συνδυάζουν την πύλη λήθης και την πύλη εισόδου για να σχηματίσουν μια ενιαία πύλη ενημέρωσης. Μειώνοντας τον αριθμό των εμπλεκόμενων πυλών, τα GRUs καταφέρνουν να αποτελούνται από λιγότερες πολύπλοκες δομές και επομένως να είναι πιο υπολογιστικά αποδοτικά σε σύγκριση με τα LSTMs ενώ ταυτόχρονα καταφέρνουν να έχει εξίσου καλή απόδοση.

Τα δίκτυα GRU εμπεριέχουν και αυτά τον μηχανισμό κρυφής κατάστασης που συνδέει μια μονάδα του δικτύου με την επόμενη, επιτρέποντας έτσι την εκδήλωση δυναμικής χρονικής συμπεριφοράς με παρόμοιο τρόπο. Κάθε μονάδα GRU είναι ενδεικτική ενός συγκεκριμένου βήματος χρόνου που διευκολύνει τη μεταφορά σημαντικών πληροφοριών μέσω του χρονικού συνεχούς. Επιπλέον, περιέχει δύο διακριτές δομές πύλης. Η πρώτη

αναφέρεται ως πύλη επαναφοράς ενώ η δεύτερη ως πύλη ενημέρωσης. Και τα δύο φέρουν σιγμοειδή επίπεδα που παρέχουν ομαλές καμπύλες στη ζώνη 0 προς 1, διασφαλίζοντας έτσι ότι το μοντέλο θα παραμείνει διαφοροποιήσιμο. Με τη συμπίεση των τιμών μεταξύ 0 και 1, η σιγμοειδής ενεργοποίηση βοηθά επίσης το δίκτυο να μάθει ποια δεδομένα είναι σημαντικά ή όχι και στη συνέχεια να τα διατηρήσει ή να τα ξεχάσει. Η λειτουργικότητα των δικτύων GRU πραγματοποιείται με τη μορφή των παρακάτω βημάτων. Όπως εξηγήθηκε πριν, κάθε GRU χρησιμοποιεί μια πύλη επαναφοράς και μια πύλη ενημέρωσης. Κάθε μία από αυτές τις πύλες έχει δύο πίνακες με βάρη. Ο πρώτος αντιστοιχεί στην είσοδο ενώ ο δεύτερος αντιστοιχεί στην κρυφή κατάσταση. Η αρχιτεκτονική των μονάδων GRU απεικονίζεται στο Σχήμα 5.



Σχήμα 5 GRU (Gated Recurrent Unit)

Η πύλη επαναφοράς της GRU είναι υπεύθυνη για να αποφασίσει ποιές από τις προηγούμενες πληροφορίες θα ξεχαστούν. Όπως και στην περίπτωση των LSTM, το πρώτο βήμα είναι ο πολλαπλασιασμός της εισόδου και της κρυφής κατάστασης με τα αντίστοιχα βάρη τους. Το άθροισμα των αποτελεσμάτων του πολλαπλασιασμού στη συνέχεια διέρχεται από ένα σιγμοειδές επίπεδο.

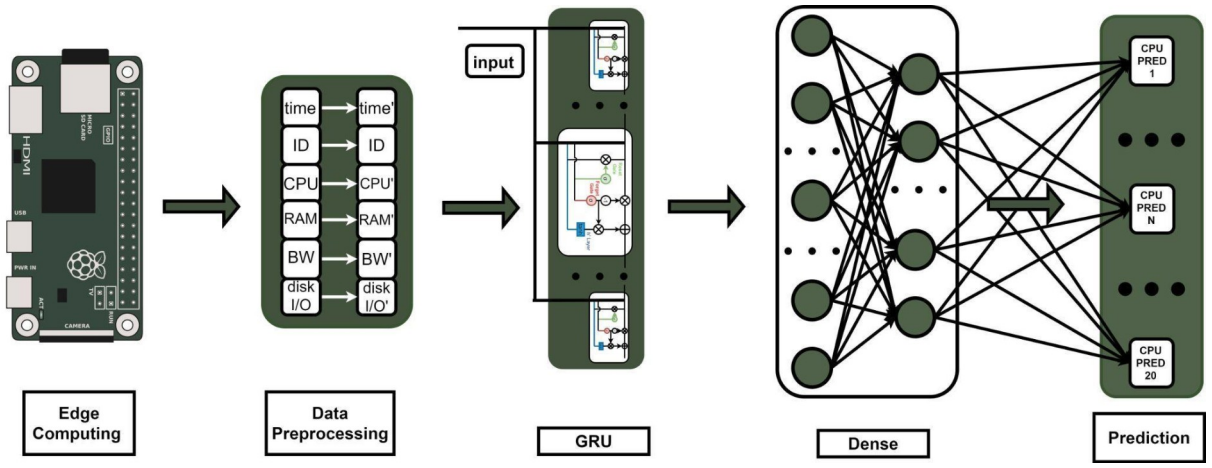
Η πύλη ενημέρωσης είναι υπεύθυνη για τον καθορισμό του ποιές από τις πληροφορίες που συγκεντρώθηκαν κατά τα προηγούμενα χρονικά βήματα πρέπει να διαβιβαστούν για μελλοντική χρήση. Από αυτή την άποψη, η συμπεριφορά της είναι αρκετά παρόμοια με αυτή της πύλης επαναφοράς. Το πρώτο βήμα για να υλοποιήσουμε την πύλη

ενημέρωσης απαιτεί τον πολλαπλασιασμό της εισόδου και της κρυφής κατάστασης με τα αντίστοιχα βάρη τους. Στη συνέχεια, τα αποτελέσματα των πολλαπλασιασμών προστίθενται και περνούν μέσα από ένα σιγμοειδές επίπεδο. Η έξοδος της πύλης ενημέρωσης θα αναφέρεται ως **u**.

Το επόμενο βήμα είναι να δημιουργηθεί μια υποψήφια νέα κρυφή κατάσταση. Ομοίως με τις πύλες επαναφοράς και ενημέρωσης, εμπλέκονται επίσης δύο πίνακες βαρών. Το πρώτο αντιστοιχεί στην είσοδο και το δεύτερο αντιστοιχεί στην κρυφή κατάσταση. Το πρώτο βήμα προς τη δημιουργία μιας υποψήφιας νέας κρυφής κατάστασης είναι ο πολλαπλασιασμός της εισόδου με τα αντίστοιχα βάρη της. Το δεύτερο βήμα είναι να υπολογίσετε το γινόμενο Hadamard με τρόπο στοιχειακού μεταξύ της κρυφής κατάστασης και της εξόδου της πύλης επαναφοράς. Αυτή η διαδικασία είναι απαραίτητη για να καθοριστεί το ποιές από τις πληροφορίες που συγκεντρώθηκαν κατά τη διάρκεια των προηγούμενων χρονικών βημάτων θα αφαιρεθούν. Το γινόμενο Hadamard στη συνέχεια πολλαπλασιάζεται με τα βάρη της κρυφής κατάστασης. Στη συνέχεια, τα αποτελέσματα των δύο πολλαπλασιασμών αθροίζονται. Το άθροισμα διέρχεται μέσω ενός επιπέδου tanh, το οποίο ελαχιστοποιεί τις επιπτώσεις του φαινομένου της εξαφάνισης της κλίσης. Αυτό επιτυγχάνεται με την κατανομή των κλίσεων με επαρκή τρόπο, εντός ενός μηδενικού κέντρου εύρους. Έτσι, επιτρέπει στις πληροφορίες να διατηρούνται περισσότερο χωρίς να εξαφανίζονται. Το προϊόν των μέχρι τώρα λειτουργιών είναι η υποψήφια νέα κρυφή κατάσταση που θα αναφέρεται ως **h**.

Για να δημιουργηθεί η νέα κρυφή κατάσταση, το πρώτο βήμα που απαιτείται είναι να εκτελεστεί πολλαπλασιασμός βάσει στοιχείων μεταξύ της εξόδου της πύλης ενημέρωσης και της κρυφής κατάστασης. Το δεύτερο είναι να εκτελεστεί πολλαπλασιασμός βάσει στοιχείων μεταξύ του **h** και του προϊόντος που θα προκύψει από την πράξη **1−u**. Η ενημερωμένη κρυφή κατάσταση είναι το άθροισμα των δύο πράξεων πολλαπλασιασμού. Η ενημερωμένη κρυφή κατάσταση μεταφέρεται στην επόμενη μονάδα GRU, η οποία αντιστοιχεί στο επόμενο χρονικό βήμα.

Το Σχήμα 6 απεικονίζει ένα GRU που έχει ενσωματωθεί στη διαδικασία πρόβλεψης χρήσης πόρων.



Σχήμα 6: GRU που έχει ενσωματωθεί στη διαδικασία πρόβλεψης χρήσης πόρων.

5. Βελτιστοποίηση υπερπαραμέτρων

Η παροχή μιας συστηματικής μεθοδολογίας για την κατασκευή μοντέλων RNN με αυτόματο τρόπο χρησιμοποιώντας ιστορικά δεδομένα αποτελεί την πρόκληση που πραγματεύεται αυτή η ενότητα. Η υπάρχουσες λύσεις προϋποθέτουν την δαπάνη αρκετών ωρών για την εύρεση μιας αποδεκτής αρχιτεκτονικής κάθε φορά που αλλάζουν οι εφαρμογές, η συμπεριφορά του χρήστη ή η Edge υποδομή. Συνεπώς, οι διαθέσιμες μέθοδοι αυτοματισμού Μηχανικής Μάθησης εξακολουθούν να έχουν σημαντικές ελλείψεις.

5.1 Εξελικτική στρατηγική

Η Εξελικτική Στρατηγική (Evolutionary Strategy) (ES) [47] ανήκει στην κατηγορία των εξελικτικών αλγορίθμων που είναι προσεγγίσεις μετα-ευριστικής βελτιστοποίησης με βάση τον πληθυσμό, εμπνευσμένες από τις αρχές της βιολογικής εξέλιξης. Η διατύπωση του ES βασίζεται σε διαδοχικές επαναλήψεις μετάλλαξης και επιλογής σε έναν πληθυσμό υποψήφιων λύσεων. Οι υποψήφιες λύσεις, που ονομάζονται επίσης άτομα, αρχικοποιούνται σε τυχαίες θέσεις σε ένα χώρο ν-διαστάσεων και κινούνται προς θέσεις που ελαχιστοποιούν μια αντικειμενική συνάρτηση. Αυτές οι διαστάσεις είναι οι αριθμητικές υπερπαράμετροι GRU-RNN που πρέπει να βελτιστοποιηθούν.

Για τις ανάγκες του για τις ανάγκες βελτιστοποίησης των υπερπαραμέτων του GRU-RNN, χρησιμοποιήθηκε το Mean Squared Error (MSE) του υποψήφιου RNN. Σε κάθε επανάληψη, ένας πληθυσμός από RNNs εκπαιδεύεται, αξιολογείται με βάση το MSE και τα πιο ακριβή από αυτά μεταλλάσσονται στην επόμενη επανάληψη. Η μετάλλαξη είναι μια στοχαστική διαδικασία που βασίζεται σε μια κανονική κατανομή που εισάγει παραλλαγές στα άτομα με το χαμηλότερο MSE σε κάθε επανάληψη. Στην αρχή, η εξερεύνηση για διαφορετικές υποψήφιες λύσεις είναι εντατική, κάνοντας ισχυρότερες μεταλλάξεις προς νέες περιοχές του χώρου αναζήτησης. Σε κάθε επανάληψη, το εύρος της εξερεύνησης μειώνεται και η εκμετάλλευση των επιλεγμένων ατόμων γίνεται πιο εντατική. Αυτό σημαίνει ότι η μετάλλαξη εισάγει έντονες παραλλαγές στις πρώτες επαναλήψεις και το εύρος των παραλλαγών μειώνεται καθώς η εξελικτική διαδικασία προχωρά προκειμένου να συγκλίνει σε μια σχεδόν βέλτιστη αρχιτεκτονική RNN.

Η βελτιστοποίηση υπερπαραμέτρων μοντέλων μηχανικής μάθησης με ES σε αντίθεση με άλλους εξελικτικούς αλγόριθμους, όπως ο Γενετικός Αλγόριθμος (GA), έχει το

πλεονέκτημα ότι δεν συνδυάζει εκ νέου διαφορετικές τοπολογίες ANN που μπορεί να έχουν σημαντικές αποκλίσεις στους φαινότυπους τους. Αυτό συμβαίνει επειδή η διασταύρωση στους GA έχει τη δυσκολία ότι οι γονείς μπορεί να έχουν διαφορετικές αρχιτεκτονικές που δεν μπορούν να ενοποιηθούν στους απογόνους τους. Ένα τυπικό παράδειγμα είναι εάν ο ένας γονέας είναι ένα LSTM-RNN 2 επιπέδων με 6 πυκνά στρώματα και ο δεύτερος γονέας είναι ένα GRU-RNN 2 επιπέδων με 4 πυκνά στρώματα. Οι φαινότυποι των LSTM και GRU δεν μπορούν να ανασυνδυαστούν ομαλά. Από την άλλη πλευρά, το ES βασίζεται αποκλειστικά σε ενδεχόμενες επιλογές και μεταλλάξεις που διεκπεραιώνουν ομαλά την εξελικτική διαδικασία. Συγκεκριμένα, οι λειτουργίες μετάλλαξης εισάγουν παραλλαγές στους εναπομείναντες υποψηφίους παρέχοντας την ευκαιρία να εξερευνήσουν λύσεις γειτόνων που μπορεί να οδηγήσουν σε αυξημένη καταλληλότητα.

5.2 Bayesian βελτιστοποίηση

Η Bayesian Βελτιστοποίηση (Bayesian Optimization) (BO) [48] χρησιμοποιείται ευρέως για την εκτίμηση υπερπαραμέτρων σε μοντέλα Βαθιάς Μάθησης. Είναι μια προφανής επιλογή για τη διαδικασία αναζήτησης στον διαστατικό χώρο προκειμένου να βρεθούν οι κοντινές στις βέλτιστες υπερπαράμετροι του GRU-RNN. Ο BO εξάγει επαναληπτικά νέες παρατηρήσεις του χώρου αναζήτησης με μια συνάρτηση απόκτησης και εκτιμά την αντικειμενική συνάρτηση με μια υποκατάστατη συνάρτηση. Όσο αυξάνεται ο αριθμός των παρατηρήσεων του BO τόσο μεγαλύτερη είναι πιθανότητα για την εύρεση της καθολικής βέλτιστης θέσης. Ωστόσο, θα πρέπει να λάβουμε υπόψη ότι ο αριθμός των παρατηρήσεων είναι πεπερασμένος και υπολογιστικά δαπανηρός, επομένως η διαδικασία έξυπνης αναζήτησης θα πρέπει να επιλέγει σημεία που μεγιστοποιούν την πιθανότητα εύρεσης μιας νέας βέλτιστης λύσης μετά από μια αντιστάθμιση στα πλαίσια του μοντέλου εξερεύνησης και εκμετάλλευσης. Η υποκατάστατη συνάρτηση προσεγγίζει την αντικειμενική συνάρτηση και ενημερώνεται κάθε φορά που η αντικειμενική συνάρτηση αξιολογείται με βάση τα νέα υποψήφια σημεία. Η συνάρτηση απόκτησης αποφασίζει πού θα ληφθεί το επόμενο δείγμα στην επαναληπτική διαδικασία του BO, βρίσκοντας τα σημεία που μεγιστοποιούν την αναμενόμενη βελτίωση. Η αναμενόμενη βελτίωση είναι συνάρτηση δύο παραγόντων. Ο πρώτος υπολογίζει τις περιοχές που η υποκατάστατη συνάρτηση έχει βέλτιστα σημεία και ο δεύτερος εκτιμά τις περιοχές με υψηλή αβεβαιότητα πρόβλεψης που δεν έχουν διερευνηθεί ακόμη αποτελεσματικά.

5.3 Υβριδική εξελικτική στρατηγική με Bayesian βελτιστοποίηση

Η βελτιστοποίηση υπερπαραμέτρων για ένα RNN είναι μια σημαντική πρόκληση καθώς περιλαμβάνει σημαντικές αρχιτεκτονικές αποφάσεις για μια σχεδόν βέλτιστη

τοπολογία. Το Hybrid Evolution Strategy with Bayesian Optimization (HBES) αποτελεί μια καινοτόμο, ολιστική και ενοποιημένη προσέγγιση για βελτιστοποίηση υπερπαραμέτρων που συγχωνεύει τις μεθοδολογίες ES και BO. Το ES είναι υπεύθυνο για την εξέλιξη ενός πληθυσμού RNN με βάση τις αριθμητικές υπερπαραμέτρους του και κάθε μεμονωμένο RNN υπολογίζει τις ονομαστικές του υπερπαραμέτρους με το BO όπως περιγράφεται από τον καρακάτων αλγόριθμο.

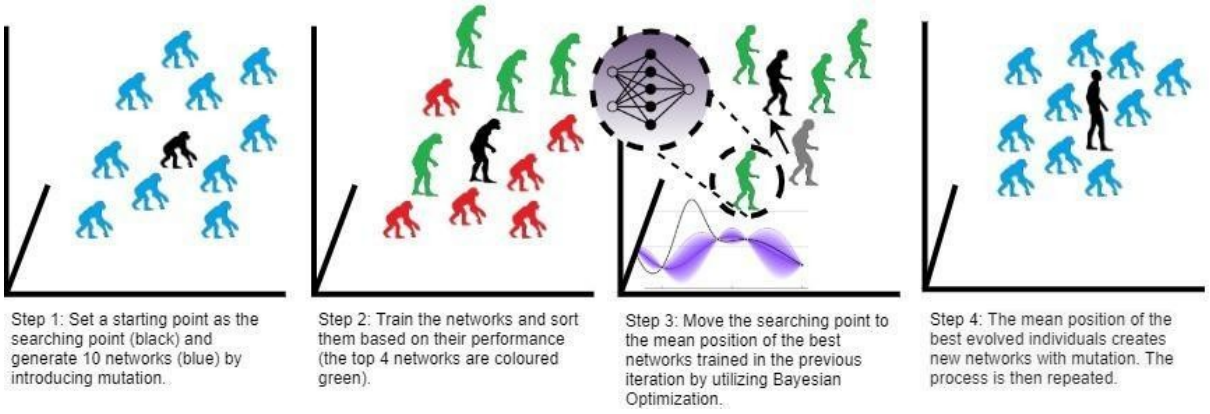
Βήμα 1: Αρχικοποίηση της Εξελικτικής Στρατηγικής
Θέτουμε το αρχικό σημείο αναζήτησης του αλγορίθμου. Συνήθως επιλέγουμε το [0.5, 0.5, …, 0.5], έχοντας μεταφέρει τις πιθανές τιμές υπερπαραμέτρων στο διάστημα [0,1]
Βήμα 2: Για κάθε δίκτυο που ανήκει στον πληθυσμό μας:
i) Προσθέτουμε τυχαίο θόρυβο στο σημείο αναζήτησης
ii) Αναιρούμε την κλίμακα του [0,1] που είχαμε επιβάλει στις τιμές των υπερπαραμέτρων ώστε να έχουμε τις πραγματικές τιμές που θα χρησιμοποιήσει το δίκτυο μας
iii) Μπεϋζιανή Βελτιστοποίηση με Γκαουσιανές Διαδικασίες

1 Εφαρμογή πρότερης Γκαουσιανής Διαδικασίας (Gaussian Process prior) στην f
2 Παρατηρούμε την f σε n_0 σημεία σύμφωνα με έναν αρχικό πειραματικό σχεδιασμό
3 Αρχικοποιούμε $n=n_0$
4 Επαναλαμβάνουμε όσο $n \leq N$:
    a Ενημερώνουμε την μεταγενέστερη κατανομή πιθανότητας (posterior probability distribution) στην f χρησιμοποιώντας όλα τα διαθέσιμα δεδομένα
    b Ορίζουμε το x_n ως τη μεγιστοποίηση της συνάρτησης απόκτησης (acquisition function) στο x
    c Παρατηρούμε το $y_n=f(x_n,x_i(t+1),v_i(t+1))$
    d Θέτουμε $n \leftarrow n+1$

Βήμα 3: Ταξινομούμε τα αποτελέσματα και την αντίστοιχη λίστα υπερπαραμέτρων
Βήμα 4: Βρίσκουμε το νέο σημείο αναζήτησης του αλγορίθμου, υπολογίζοντας το μέσο όρο των συντεταγμένων των κορυφαίων k δικτύων
Βήμα 5: Πηγαίνουμε στο βήμα 2 μέχρι να ολοκληρωθεί ο απαιτούμενος αριθμός εποχών/επαναλήψεων

Οι αριθμητικές υπερπαράμετροι είναι ο αριθμός των επαναλαμβανόμενων επιπέδων και των επιπέδων τροφοδοσίας, ο αριθμός των νευρώνων για κάθε στρώμα, η αναδρομή, οι εποχές, το μέγεθος παρτίδας, το ποσοστό εγκατάλειψης και ο ρυθμός εκμάθησης. Οι

ονομαστικές υπερπαράμετροι είναι ο τύπος του RNN, οι συναρτήσεις ενεργοποίησης και οι βελτιστοποιητές. Η αποκτηθείσα γνώση των ονομαστικών υπερπαραμέτρων είναι καθολική σε όλο τον πληθυσμό και ενημερώνεται από όλα τα άτομα κατά τη διάρκεια των γενεών. Ο απώτερος στόχος του HBES είναι μέσω της διαδικασίας της Bayesian εξέλιξης να καταλήξει σε ένα σχεδόν βέλτιστο και εκπαιδευμένο RNN που μπορεί να προβλέψει έγκαιρα και με ακρίβεια τη χρήση των πόρων που θα παρατηρηθεί στα επόμενα χρονικά βήματα. Τα τέσσερα κύρια βήματα HBES στα πλαίσια μιας επανάληψης της εξελικτικής διαδικασίας απεικονίζονται στο Σχήμα 7.



Σχήμα 7: Τα τέσσερα κύρια βήματα του HBES στα πλαίσια μιας επανάληψης της εξελικτικής διαδικασίας.

.

6. Πειραματική αξιολόγηση μέρος Α

Η πειραματική αξιολόγηση πραγματοποιήθηκε σε μια πραγματική υποδομή, όπου για Edge κόμβοι επεξεργασίας χρησιμοποιήθηκαν Raspberry Pi3 με τετραπύρηνο ARM Cortex-A53 64 bit στα 1,4 GHz, φορτωμένα με λειτουργικό σύστημα Raspbian που είναι μια έκδοση του Debian Linux. Το σύνολο των δεδομένων κατασκευάστηκε από ένα εργαλείο παρακολούθησης που υλοποιήθηκε στην Python 3 με τις βιβλιοθήκες psutil [49] και GPUtil [50]. Έγινε παρακολούθηση της χρήσης της CPU, της μνήμης RAM, του δίσκου και του εύρους ζώνης σε πραγματικό χρόνο, σε ένα δευτερόλεπτο χρονικό διάστημα.

Η προς εξέταση εφαρμογή ήταν μια ταξινόμηση κειμένου επεξεργασίας φυσικής γλώσσας. Η περίπτωση χρήσης βασίστηκε στο να γίνει η ταξινόμηση κειμένου σε περιβάλλον Edge computing, τοπικά, κοντά στους κατόχους του κειμένου και όχι σε υποδομές Cloud Computing εξαιτίας ζητημάτων απορρήτου. Ο λόγος για αυτήν την επιλογή αυτή είναι ότι οι κάτοχοι του κειμένου δεν συμφώνησαν να μεταφερθούν και να υποβληθούν σε επεξεργασία τα κείμενά τους σε απομακρυσμένους διακομιστές. Για να προσπελαστεί η εφαρμογή από απόσταση και να λάβουμε τα σύνολα δεδομένων χρήσης πόρων χρησιμοποιήσαμε το πρωτόκολλο SSH, αλλά δεν είχαμε τα δικαιώματα πρόσβασης στα επεξεργασμένα κείμενα.

6.1 Υλοποίηση μοντέλου και τα διάφορα Πλαίσια σύγκρισης

Το μοντέλο HBES και το μοντέλο παλινδρόμησης πολλαπλών εξόδων GRU (HBES-GRU) υλοποιούνται στην Python 3 χρησιμοποιώντας τα πλαίσια NumPy, pandas, statistics, Scikit-learn, SciPy, Scikit-Optimize, TensorFlow 2 και Keras. Το περιβάλλον που χρησιμοποιήσαμε είναι το σημειωματάριο Jupyter του Google Colaboratory. Συγκρίναμε το HBES-GRU, με μια προσέγγιση βασικής χρονοσειράς, το μετα-μοντέλο Μηχανικής Μάθησης για την πρόβλεψη χρήσης πόρων AUCROP, το Auto-sklearn, το XGBoost, ένα μοντέλο LSTM με γενετικό αλγόριθμο (GA-LSTM) και το Keras-Tuner.

6.2 Αποτελέσματα, αξιολόγησης και συζήτηση

Στην πειραματική αξιολόγηση, ξεκινήσαμε με την ανάλυση χρονοσειρών και βρήκαμε θετικές συσχετίσεις για καθυστερήσεις σε ένα εύρος τιμών από 1 έως 22. Αυτό επιβεβαιώνει

την ισχυρή ιδιότητα αυτο-ομοιότητας που έχουν οι ακολουθίες μετρήσεων χρήσης πόρων. Στη συνέχεια, χρησιμοποιώντας το μοντέλο πρόβλεψης ARIMA, αξιολογήσαμε τις προβλέψεις μετρήσεων πόρων. Για παράδειγμα, το CPU RMSE ήταν 18.474. Συγκρίνοντας τα αποτελέσματα των στατιστικών μοντέλων με τις προσεγγίσεις Μηχανικής Μάθησης και Βαθιάς Μάθησης, διαπιστώσαμε ότι η τελευταία παρουσίασε μια βελτίωση που ξεπερνά το 20% όσον αφορά το RMSE στις περισσότερες περιπτώσεις. Από αυτή την άποψη, αποφασίσαμε να εστιάσουμε στα μοντέλα Μηχανικής Μάθησης και Βαθιάς Μάθησης.

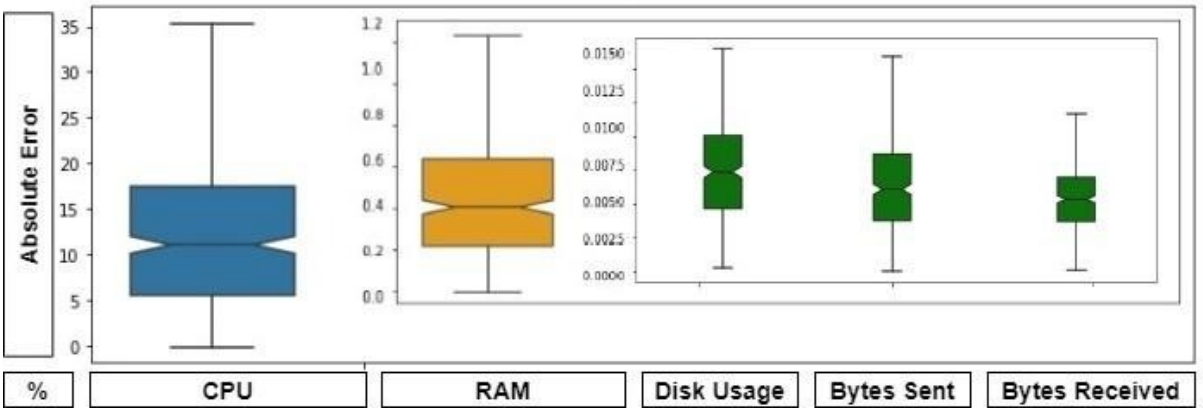
TABLE I
COMPARISON OF SINGLE-OUTPUT & MULTI-OUTPUT PREDICTION METHODS OF RESOURCE USAGE METRICS.

Method	RMSE	MAE	CPU-1 (%)		RAM-1 (%)		Infer. Time	
			RMSE	MAE	RMSE	MAE	Single	Batch
HBES-GRU	0.0641	0.0276	15.918	12.815	1.694	0.580	0.033	0.038
GA-LSTM	0.0674	0.0338	16.099	12.838	1.746	0.917	0.020	0.024
Keras-Tuner	0.0785	0.0377	16.291	13.290	2.631	0.818	0.042	0.042
AUCROP	0.0814	0.0414	17.235	14.009	2.480	1.482	0.004	0.011
XGBoost	0.1139	0.0599	16.457	13.569	1.515	0.472	0.060	0.010
Auto-sklearn	0.1055	0.0243	52.659	17.856	1.546	0.526	0.263	0.572

Πίνακας 1.

Στον Πίνακα 1 συνοψίζονται τα πειραματικά αποτελέσματα. Οι δύο πρώτες στήλες παρέχουν το RMSE και το MAE, που καταγράφηκαν μεταξύ όλων των τιμών δοκιμής των συσκευών και των μετρήσεων πόρων. Για το RMSE, που δίνει επιπλέον ποινή σε προβλέψεις με σημαντικά λάθη, μπορούμε να δούμε ότι η HBES-GRU είχε την καλύτερη απόδοση. Στη στήλη με τίτλο MAE, βλέπουμε ότι τα δύο καλύτερα μοντέλα είναι το Auto-sklearn και το HBES-GRU. Οι μετρικές λάθους τους είναι αρκετά παρόμοιες και έχουν σημαντικά καλύτερη απόδοση σε σύγκριση με τα άλλα μοντέλα.

Οι στήλες CPU-1 και RAM-1 αντιπροσωπεύουν τα RMSE και MAE για τον Edge κόμβο επεξεργασίας που είχε τις περισσότερες λανθασμένες προβλέψεις στην υποδομή. Επιπλέον, το Σχήμα 8 απεικονίζει το 25ο και το 75ο εκατοστημόριο, τη διάμεσο, το ελάχιστο και το μέγιστο των μετρήσεων της τιμής σφάλματος. Αυτές οι μετρήσεις περιλαμβάνουν την CPU, τη μνήμη RAM, τη χρήση δίσκου και το εύρος ζώνης ως προς τα byte που αποστέλλονται και λαμβάνονται .



Σχήμα 8: Το 25ο και το 75ο εκατοστημόριο, η διάμεσος, το ελάχιστο και το μέγιστο των μετρήσεων της τιμής σφάλματος.

Όσον αφορά τη χρήση του δίσκου και το εύρος ζώνης, τα σφάλματα πρόβλεψης ήταν ασήμαντα. Αυτό δεν οφείλεται μόνο στην ικανότητα του HBES-GRU να παρέχει ακριβείς προβλέψεις, αλλά στη μικρή διακύμανση των τιμών τους. Η διακύμανση της CPU είναι πολύ εντονότερη από της RAM και το HBES-GRU αποτυπώνει με καλύτερο τρόπο τις αλλαγές σε σύγκριση με τα άλλα μοντέλα. Το XGBoost έχει καλύτερη απόδοση από το HBES-GRU στη μνήμη RAM. Αυτό δικαιολογείται από τη δομή που έχει το XGBoost. Το XGBoost μπορεί να δημιουργήσει συγκεκριμένα δέντρα αποφάσεων για τα υπολείμματα της μνήμης RAM και να αποτυπώσει σχεδόν βέλτιστα την αργή συμπεριφορά αλλαγής της.

Τέλος, βλέπουμε τους χρόνους εκτέλεσης των μοντέλων προκειμένου να κάνουν μία μόνο πρόβλεψη ή μια παρτίδα από εκατό προβλέψεις. Οι μετρήσεις καταγράφηκαν σε δευτερόλεπτα. Όλοι οι χρόνοι εκτέλεσης, εκτός από το Auto-sklearn, είναι στην περιοχή από 11 έως 60 msec. Αυτοί οι χρόνοι εκτέλεσης καθιστούν την πρόβλεψη χρήσης πόρων μια γρήγορη διαδικασία, που είναι ικανή να ενσωματωθεί σε εφαρμογές που χρονικά ευαίσθητες.

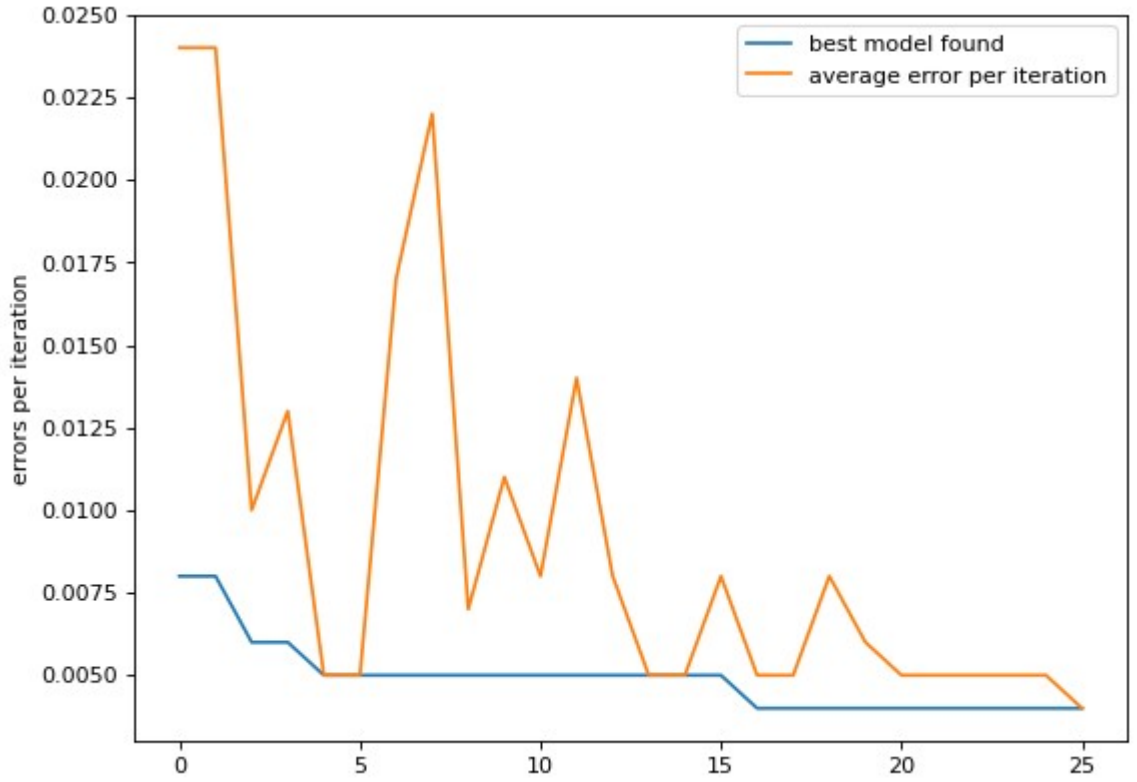
Σε αυτήν την έρευνα, δεν συγκρίναμε τους χρόνους εκπαίδευσης επειδή θέλαμε να κάνουμε μια εξαντλητική έξυπνη αναζήτηση στο χώρο των υποθέσεων και να δούμε τα όρια ακρίβειας που επιτυγχάνουν τα μοντέλα. Τα πειράματα γίνανε σε διάφορα χρονικά πλαίσια. Οι μετρήσεις στον Πίνακα 1 αφορούν χρονικό ορίζοντα 8 λεπτών. Επιλέξαμε να απεικονίσουμε αυτόν τον χρονικό ορίζοντα επειδή είναι κοντά στον χρόνο που απαιτείται για την δημιουργία ενός VM.

Βλέπουμε ότι ακόμα κι αν οι GRUs είναι απλούστερες στη δομή τους σε σύγκριση με τα LSTMs και δεν διαθέτουν την πύλη εξόδου, έχουν καλύτερη απόδοση. Αυτό το συμπέρασμα επιβεβαιώνεται από τη βιβλιογραφία στην περίπτωση μικρών στο μέγεθος συνόλων δεδομένων που χαρακτηρίζονται από χαμηλή συχνότητας απόκτησης .

Τρία ακόμη σημαντικά συμπεράσματα που έχουμε είναι τα εξής: Στις περισσότερες μετρήσεις, τα μοντέλα βαθιάς μάθησης (HBES-GRU, GA-LSTM, Keras- Tuner) έχουν καλύτερη απόδοση από τα μοντέλα μηχανικής μάθησης(AUCROP, XGBoost, Auto-sklearn). Στη συνέχεια, οι εξελικτικοί αλγόριθμοι για βελτιστοποίηση υπερπαραμέτων (HBES-GRU και GA-LSTM) έχουν καλύτερη απόδοση σε σύγκριση με την απλή Bayesian βελτιστοποίηση (Keras-Tuner). Τέλος, βλέπουμε μια σημαντική βελτίωση χρησιμοποιώντας την υβριδική προσέγγιση της Bayesian βελτιστοποίησης και της στρατηγικής εξέλιξης σε σύγκριση με τους απλούς γενετικούς αλγόριθμους.

6.3 Σύγκλιση της Υβριδικής εξελικτικής στρατηγικής με Bayesian βελτιστοποίηση

Η σύγκλιση καθώς και η θέση του καθολικού βέλτιστου σημείου είναι δύο από τα πιο σημαντικά θέματα στον τομέα των εξελικτικών αλγορίθμων. Η σύγκλιση σημαίνει ότι καθώς ο πληθυσμός εξελίσσεται, τα άτομα πλησιάζουν στη βέλτιστη λύση συρρικνώνοντας την απόκλισή τους. Αλλά δεν μπορούμε να είμαστε σίγουροι εάν τα σημεία σύγκλισης στο χώρο του γονότυπου είναι καθολικό ή τοπικό ελάχιστο. Για το λόγο αυτό, ο αλγόριθμος HBES στην αρχή της διαδικασίας εξέλιξης παράγει μια ισχυρή διακύμανση στη μετάλλαξη που συρρικνώνεται κατά τις επαναλήψεις. Ταυτόχρονα, είναι ο καλύτερος γονότυπος που χρησιμοποιείται σε όλες τις επαναλήψεις. Η σύγκλιση του HBES απεικονίζεται στο Σχήμα 9. Βλέπουμε ότι στην αρχή ο μέσος κύκλος σφαλμάτων πληθυσμού κυμαίνεται έντονα. Σε ορισμένες επαναλήψεις παγιδεύεται στα τοπικά ελάχιστα, όπως για παράδειγμα στις επαναλήψεις έξι έως έντεκα. Σε ορισμένες άλλες επαναλήψεις βρίσκεται σε περιοχές που σχηματίζουν οροπέδιο όπως βλέπουμε στις επαναλήψεις είκοσι έως είκοσι πέντε. Αλλά, τελικά χρησιμοποιώντας τη μετάλλαξη, τα άτομα ξεφεύγουν από τις περιοχές του οροπεδίου και τα τοπικά ελάχιστα και κινούνται προς τις πλησιέστερες βέλτιστες περιοχές. Αυτές οι πλησίον των βέλτιστων σημείων περιοχές στο χώρο του γονότυπου μεταφράζονται στις πλησιέστερες προς τις βέλτιστες αρχιτεκτονικές GRU-RNN στον χώρο των φαινοτύπων. Αυτές οι αρχιτεκτονικές GRU-RNN παρέχουν τις πιο ακριβείς προβλέψεις χρήσης πόρων για χρήση CPU, RAM, δίσκου και εύρους ζώνης σε μια υποδομή Edge computing.



Σχήμα 9: Η σύγκλιση του HBES.

6.4 Συμπεράσματα

Σε αυτή τη διπλωματική εργασία, έγινε θεωρητική ανάλυση της χρησιμότητας της πρόβλεψης χρήσης πόρων για μια έξυπνη ενορχήστρωση σε μια υπολογιστική Edge υποδομή. Συγκεκριμένα, παρουσιάστηκε το πώς η προσαρμοστική κατανομή πόρων και η εκφόρτωση εργασιών μπορούν να αξιοποιήσουν τις πληροφορίες σχετικά με τη χρήση πόρων. Έγινε πειραματική ανάλυση σχετικά με τη χρήση του GRU-RNN για τη μοντελοποίηση χρήσης πόρων. Έγινε παρουσίαση ενός αλγόριθμου για βελτιστοποίηση υπερπαραμέτρων που συνδυάζει την εξελικτική στρατηγική με τη Bayesian βελτιστοποίηση και που ξεπερνά τους ευρέως χρησιμοποιούμενους hypertuners όπως το Keras-Tuner και άλλα μοντέλα μηχανικής μάθησης τελευταίας τεχνολογίας.

7. Πρόβλεψη φόρτου δικτύου

Στα συστήματα Edge computing και Cloud computing, η μοντελοποίηση και η πρόβλεψη της δυναμικής κίνησης υπηρεσιών είναι εξαιρετικά χρήσιμη για την αντιμετώπιση του μεγάλου όγκου αιτημάτων δεδομένων και υπηρεσιών προκειμένου να διασφαλιστεί η βέλτιστη ενορχήστρωση των δαπανηρών πόρων υποδομής. Ο όρος φόρτος δικτύου έχει δύο διαφορετικές ερμηνείες στην υπολογιστική βιβλιογραφία, οι οποίες είναι ως επί το πλείστον συνδεδεμένες. Πρώτον, ο φόρτος μπορεί να ερμηνευθεί ως η ποσότητα των δεδομένων που κινούνται στις υποδομές του δικτύου. Η δεύτερη ερμηνεία αναφέρεται στον αριθμό των αιτημάτων χρήστη ή σύνδεσης στις υπηρεσίες. Τα μεταδεδομένα που αντιστοιχούν και στις δύο ερμηνείες του φόρτου μπορούν να συλλεχθούν στο επίπεδο μεταφοράς του Πρωτοκόλλου Ελέγχου Μεταφοράς/Πρωτοκόλλου Διαδικτύου (TCP/IP) χρησιμοποιώντας ένα διαγνωστικό εργαλείο δικτύου τύπου traceroute.

Για περισσότερο από μία δεκαετία, η πρόβλεψη του φόρτου του δικτύου χρησιμοποιείται για να εκτιμηθεί η ποσότητα των πόρων Cloud ή Edge που απαιτούνται από τις διάφορες υπηρεσίες δεδομένων [51]. Αυτή η προσέγγιση αποκτά νόημα στο πλαίσιο της δυναμικής διαχείρισης πόρων με δυνατότητα προληπτικής αυξομείωσης του αριθμού τους ανά τακτά χρονικά διαστήματα. Η υπόθεση είναι ότι εάν προβλέψουμε την ποσότητα των δεδομένων και τον αριθμό των αιτημάτων, μπορούμε να αντιγράψουμε αντίστοιχα τις εικονικές μηχανές και τις λειτουργίες δικτύου των διαφόρων υπηρεσιών. Σε αυτή την ενότητα, δεν ερευνάται και δεν αξιολογείται ο μηχανισμός Cloud που κάνει την ενορχήστρωση. Συνήθως γίνεται χρήση ευρέως αποδεκτών εργαλείων όπως το Kubernetes [52] για την ανάπτυξη, την αυξομείωση του αριθμού των πόρων και τη διαχείριση των υπηρεσιών δεδομένων. Αντίθετα, προτείνεται ένα πρωτότυπο μοντέλο πρόβλεψης φόρτου δικτύου που ενεργοποιεί την αυξομείωση του αριθμού και την διαχείριση των πόρων.

Σε γενικές γραμμές, η ακριβής πρόβλεψη του μέλλοντος είναι ένα αρκετά δύσκολο εγχείρημα. Ευτυχώς, ο φόρτος από TCP παρουσιάζει ισχυρή αυτοσυσχέτιση [53], που καθιστά διάφορες στατιστικές μεθόδους και μεθόδους χρονοσειρών ικανές να εκτελέσουν μοντελοποίηση και πρόβλεψη. Η αυτοσυσχέτιση καταγράφει τη σχέση μεταξύ της τρέχουσας τιμής μιας μέτρησης και των προηγούμενων τιμών της. Πολλές αναλύσεις δικτύου έχουν δείξει ότι η κίνηση TCP χαρακτηρίζεται από επαναλαμβανόμενα μοτίβα κατά την πάροδο του χρόνου. Όσον αφορά τα διάφορα μοντέλα πρόβλεψης που χρησιμοποιούνται, για

περισσότερο από μία δεκαετία οι μηχανικοί δικτύων χρησιμοποιούν στατιστικά μοντέλα όπως Poisson, Autoregressive–Moving-Average (ARMA) και Autoregressive Integrated Moving Average (ARIMA). Η εμφάνιση της Βαθιάς Μάθησης έχει αλλάξει δραστικά το τοπίο της ανάλυσης δεδομένων και της λήψης αποφάσεων. Συγκεκριμένα, στην περίπτωση της πρόβλεψης χρονοσειρών, τα επαναλαμβανόμενα νευρωνικά δίκτυα (Recursive Neural Networks) (RNN) συχνά ξεπερνούν σημαντικά τα παραδοσιακά στατιστικά μοντέλα πρόβλεψης. Το πρώτο στάδιο αυτής της έρευνας είναι αφιερωμένο στην απάντηση στο ερώτημα ότι τα RNN είναι μια καλύτερη επιλογή σε σύγκριση με μοντέλα στατιστικών χρονοσειρών για την πρόβλεψη της κυκλοφορίας δικτύου των υπηρεσιών δεδομένων.
Τα κοινά μοντέλα χρονοσειρών και τα απλά RNN έχουν σχεδιαστεί για να παρέχουν προβλέψεις μόνο ενός βήματος. Τα μοντέλα πρόβλεψης πολλαπλών βημάτων εξάγουν μια ακολουθία τιμών διαδοχικών χρονικών βημάτων. Η πρόβλεψη πολλαπλών βημάτων σχετικά με τις χρονοσειρές κίνησης είναι εξαιρετικά σημαντική επειδή μπορεί να χρησιμοποιηθεί προκειμένου να επιτευχθεί καλύτερη ευαισθησία του προγραμματισμού πόρων σε σύγκριση με τις μεθοδολογίες πρόβλεψης ενός βήματος. Ο μηχανισμός ενορχήστρωσης πόρων μπορεί να εφαρμόσει μια πιο εξελιγμένη προσαρμογή σε πραγματικό χρόνο των εντατικών ροών εργασιών που βασίζονται σε δεδομένα χρησιμοποιώντας πληροφορίες σχετικά με το φόρτο σε μορφή πολλαπλών βημάτων [54] επειδή κάθε εικονική συσκευή και λειτουργία υπηρεσίας έχει διαφορετικό χρόνο ανάπτυξης. Ο κωδικοποιητής-αποκωδικοποιητής μπορεί να χρησιμοποιηθεί για πρόβλεψη χρονοσειρών σε μορφή πολλαπλών βημάτων. Ο κωδικοποιητής-αποκωδικοποιητής είναι μια σύνθετη αρχιτεκτονική Μηχανικής Μάθησης που αποτελείται από δύο τεχνητά νευρωνικά δίκτυα (ANN) που αλληλεπιδρούν μέσω λανθάνουσας μεταβλητής και πραγματοποιούν προβλέψεις ακολοθιών. Η μοντελοποίηση και η δυνατότητα εφαρμογής των αρχιτεκτονικών κωδικοποιητή-αποκωδικοποιητή για την πρόβλεψη της κυκλοφορίας υπηρεσιών δεν είχε προταθεί στη βιβλιογραφία και είναι μια σημαντική καινοτομία της έρευνάς γύρω από την οποία δομήθηκε η συγκεκριμένη διπλωματική εργασία.

Οι τέσσερις κύριες συνεισφορές της έρευνας που πραγματοποιήθηκε είναι:

i Θεωρητική και πειραματική σύγκριση καθιερωμένων μοντέλων στατιστικών χρονοσειρών με προσεγγίσεις βαθιάς μάθησης για πρόβλεψη κίνησης υπηρεσιών.

ii Προτείνεται χρήση μοντέλων κωδικοποιητή-αποκωδικοποιητή για πρόβλεψη φόρτου υπηρεσιών σε μορφή πολλαπλών βημάτων.

iii Ανάλυση και αξιολόγηση πολλαπλών τοπολογιών νευρωνικών δικτύων για τον κωδικοποιητή-αποκωδικοποιητή.

iv Δημιουργία μιας πρωτότυπης αρχιτεκτονικής υβριδικού κωδικοποιητή-αποκωδικοποιητή που κάνει ταυτόχρονη χρήση απλών και αμφίδρομων LSTM.

7.1 Σχετικές μελέτες στον κλάδο της πρόβλεψης φόρτου δικτύου.

Η πρόβλεψη της ποσότητας των δεδομένων που μεταδίδονται μέσω του δικτύου και του αριθμού των αιτημάτων υπηρεσίας σε διαδοχικά χρονικά βήματα είναι εξαιρετικά χρήσιμη για την εφαρμογή ενός βέλτιστου σχεδίου διαχείρισης των πόρων της υποδομής στην οποία εκτελούνται διάφορες υπηρεσίες δεδομένων [55]. Η πρόβλεψη φόρτου υπηρεσιών έχει μεγάλη σημασία για την εξισορρόπηση του φορτίου και την κατανομή των πόρων προκειμένου να εκπληρωθούν οι απαιτήσεις QoS. Με την ταχεία ανάπτυξη των κέντρων δεδομένων, η μεγάλης κλίμακας πρόβλεψη κίνησης δεδομένων των δικτύων απαιτεί πιο προχωρημένες μεθόδους για την αντιμετώπιση των πολύπλοκων ιδιοτήτων της εξάρτησης υψηλής διασταλτικότητας, της μεγάλης εμβέλειας και της μη γραμμικότητας που συχνά παρουσιάζονται σε τέτοιου τύπου προβλήματα.

Για πολλά χρόνια έχουν χρησιμοποιηθεί διαφορετικές μέθοδοι για τη μοντελοποίηση και την πρόβλεψη του φόρτου των υπηρεσιών. Στην αρχή, χρησιμοποιήθηκαν στατιστικά μοντέλα, όπως οι διαδικασίες Poisson, αλλά παρουσίασαν τον περιορισμό ότι δεν καταγράφουν το χαρακτηριστικό αυτο-ομοιότητας [56] των τιμών της ακολουθίας. Στη συνέχεια, μοντέλα χρονοσειρών όπως το Autoregressive–Moving–Average (ARMA) και οι παραλλαγές τους Autoregressive Integrated Moving Average (ARIMA) και Seasonal ARIMA (SARIMA) [57] χρησιμοποιήθηκαν για την πρόβλεψη του φόρτου δικτύων και κατάφεραν να ελαχιστοποιήσουν το κόστος λειτουργίας λαμβάνοντας υπόψη δύο τύποι κόστους:

i) Το κόστος πόρων Cloud που προκύπτει όταν εκτελείται μη ορθή παροχή πόρων λόγω υπερεκτίμησης της έντασης του φόρτου και ii) Το κόστος υποβάθμισης QoS που προκύπτει όταν η ένταση του φόρτου υποεκτιμάται, με αποτέλεσμα να κατανέμονται λιγότεροι πόροι από αυτούς που πραγματικά απαιτούνται και έτσι να τίθεται σε κίνδυνο η ικανοποίηση QoS και QoE απαιτήσεων.

Με την έλευση της Μηχανικής Μάθησης, πολλά μοντέλα λήψης αποφάσεων μετά από πειραματική σύγκριση και επανασχεδιασμό αντικαταστάθηκαν τελικά από ANN. Οι πρώτες μελέτες έδειξαν ότι το ARIMA αποδίδει καλύτερα από το απλό feed-forward ANN [58]. Ο λόγος είναι ότι το απλό feed-forward ANN δεν έχει σχεδιαστεί για διαδοχικές εργασίες.

Επιτρέπει στις πληροφορίες να ταξιδεύουν μονόδρομα και δεν μπορεί να συλλάβει τα περιοδικά μοτίβα και τα μοτίβα αυτοσυσχέτισης που χαρακτηρίζουν τη διαμόρφωση του φόρτου του δικτύου. Τα RNN είναι μια διαφορετική κατηγορία ANN που μοντελοποιεί τη χρονική αλληλουχία δεδομένων έτσι ώστε κάθε παρατήρηση να εξαρτάται από τις προηγούμενες που εκτελούνται και προς τις δύο κατευθύνσεις χάρη σε βρόχους στο δίκτυό τους. Οι πληροφορίες που προέρχονται από παρελθοντικές εισόδους ανατροφοδοτούνται στο δίκτυο παρέχοντας ένα είδος μνήμης των παρελθοντικών εισόδων προκειμένου να προβλεφθούν οι μελλοντικές. Πολύπλοκα μοντέλα RNN που αξιοποιούν τη χρονική συμπεριφορά των κέντρων δεδομένων έχουν χρησιμοποιηθεί με επιτυχία για πρόβλεψη φόρτου μιας υπηρεσίας.

Πολλές υπηρεσίες μεταφοράς, αποθήκευσης και επεξεργασίας δεδομένων χαρακτηρίζονται από χρονικές εξαρτήσεις μικρής και μεγάλης εμβέλειας, καθιστώντας την πρόβλεψη πολλαπλών βημάτων μια εξέχουσα λύση [59]. Η πρόβλεψη πολλαπλών βημάτων με χρήση RNN με πρόβλεψη που εμπεριέχει πολλά χρονικά βήματα έχει εφαρμοστεί για την πρόβλεψη χρονοσειρών φόρτου δικτύου τύπου IoT [60]. Αυτή η προσέγγιση βασίζεται στην υπόθεση ότι για κάθε βήμα της πρόβλεψης η έξοδος του RNN χρησιμοποιείται από την είσοδο για να γίνει η πρόβλεψη του επόμενου βήματος. Ένας περιορισμός είναι ότι αυτή η προσέγγιση δεν έχει σχεδιαστεί για πρόβλεψη αλληλουχίας και ως αποτέλεσμα τείνει να συσσωρεύει σφάλματα μεταξύ των βημάτων.

Μια αρχιτεκτονική που βασίζεται στη λογική “από ακολουθία σε ακολουθία” (seq2seq) μπορεί να συλλάβει τις χρονικές εξαρτήσεις και να παρέχει προβλέψεις για διαφορετικά χρονικά βήματα. Μια εξέχουσα προσέγγιση για το seq2seq είναι ο κωδικοποιητής-αποκωδικοποιητής [61] που αποτελείται από ένα νευρωνικό δίκτυο που χαρτογραφεί την ακολουθία εισόδου των παρελθοντικών βημάτων σε ένα ενδιάμεσο διάνυσμα και τον αποκωδικοποιητή που χαρτογραφεί το ενδιάμεσο διάνυσμα σε μια πρόβλεψη που έχει τη μορφή ακολουθίας. Οι κωδικοποιητές-αποκωδικοποιητές έχουν χρησιμοποιηθεί σε πολλά πεδία για πρόβλεψη σε μορφή πολλαπλών βημάτων, αλλά δεν έχουν χρησιμοποιηθεί στην πρόβλεψη φόρτου υπηρεσιών. Ειδικά οι προβλέψεις πολλαπλών βημάτων στους τομείς της μεταφοράς [62] και της χωροχρονικής κινητικότητας [63] έχουν πολλά κοινά όσον αφορά τη διατύπωση προοπτικών προβλημάτων και τη δομή δεδομένων με την κίνηση των υπηρεσιών. Ένας κωδικοποιητής-αποκωδικοποιητής με κωδικοποιητή νευρωνικού δικτύου συνέλιξης (Convolution Neural Network) (CNN) και αποκωδικοποιητή RNN έχει χρησιμοποιηθεί για διαδικασίες πρόβλεψης [64].

Για να συνοψίσουμε τα κίνητρα αυτής της έρευνας, είδαμε ότι η πρόβλεψη της κυκλοφορίας υπηρεσιών είναι μια βασική πρόκληση προκειμένου να επιτευχθεί η βέλτιστη διαχείριση των πόρων στους οποίους λειτουργούν διάφορες υπηρεσίες μεγάλων δεδομένων. Οι υπάρχουσες λύσεις στατιστικών χρονοσειρών έχουν ξεπεραστεί από τις προσεγγίσεις Μηχανικής Μάθησης. Ωστόσο, οι προσεγγίσεις Μηχανικής Μάθησης δεν έχουν επεκταθεί επαρκώς για να παρέχουν πρόβλεψη σε μορφή πολλαπλών βημάτων. Τα μοντέλα κωδικοποιητή-αποκωδικοποιητή είναι μια εξέχουσα προσέγγιση για την πρόβλεψη πολλαπλών βημάτων. Αυτή είναι η πρώτη εργασία που εξετάζει τη χρήση κωδικοποιητών-αποκωδικοποιητών για πρόβλεψη φόρτου υπηρεσιών σε μορή πολλαπλών βημάτων.

Επιπλέον, εξετάζονται τέσσερις διαφορετικοί τύποι κωδικοποιητή-αποκωδικοποιητή: α) Στοιβαγμένα (Stacked) LSTMS, β) CNN-LSTMS, γ) Αμφίδρομα (Bidirectional) LSTM και δ) το καινοτόμο Υβριδικό LSTM. Επιπλέον, αξίζει να σημειωθεί πως κάνουμε χρήση της Bayesian βελτιστοποίησης προκειμένου να δημιουργήσουμε σχεδόν βέλτιστες τοπολογίες για τις διάφορες αρχιτεκτονικές κωδικοποιητή-αποκωδικοποιητή. Τέλος, παρέχουμε προβλέψεις για τρεις μετρήσεις φόρτου: α) Τον αριθμό των αιτημάτων, β) τον όγκο των δεδομένων που μεταδίδονται και γ) τη διάρκεια των περιόδων σύνδεσης υπηρεσίας.

7.2 Πρόβλεψη φόρτου δικτύου σε μορφή πολλαπλών βημάτων

Ένα από τα βασικά θέματα που πραγματεύεται αυτή η διπλωματική εργασία είναι η πρόβλεψη φόρτου υπηρεσιών σε μορφή πολλαπλών βημάτων. Κάθε χρονικό βήμα της πρόβλεψης έχει διαμορφωθεί ώστε να διαρκεί πέντε λεπτά και κάθε πρόβλεψη έχει σχεδιαστεί ώστε να περιέχει πέντε χρονικά βήματα. Με αυτόν τον τρόπο, μπορούμε να έχουμε μια πρόβλεψη με χρονικό ορίζοντα αυτόν των 25 λεπτών με διαφορετικές λεπτομέρειες και ακρίβεια για κάθε χρονικό βήμα σχετικά με τον αριθμό των αιτημάτων των υπηρεσιών, την ποσότητα των δεδομένων που μεταδίδονται και τη συνολική διάρκεια των περιόδων σύνδεσης των υπηρεσιών. Η διακύμανση του φόρτου αλλάζει συχνά και είναι σύνηθες φαινόμενο να βλέπουμε ξαφνικές αυξομειώσεις. Σε περίπτωση που χρησιμοποιήσουμε ένα μόνο χρονικό βήμα των 25 λεπτών, θα χάναμε πολύτιμες πληροφορίες σχετικά με το τι συμβαίνει κατά τη διάρκεια αυτών των μεγάλων χρονικών περιόδων. Τα διάφορα μοντέλα στατιστικής πρόβλεψης παρέχουν αποτελέσματα με ακρίβεια που είναι υψηλή στο αρχικό βήμα και στη συνέχεια μειώνεται σταθερά όσο προχωράμε προς τα τελευταία βήματα. Αυτό συμβαίνει λόγω του γεγονότος ότι το μοντέλο πρόβλεψης στα τελευταία βήματα έχει συγκεντρωτικά σφάλματα όσον αφορά την πρόβλεψη από τα προηγούμενα βήματα. Θα δούμε στο κεφάλαιο της πειραματικής αξιολόγησης ότι η

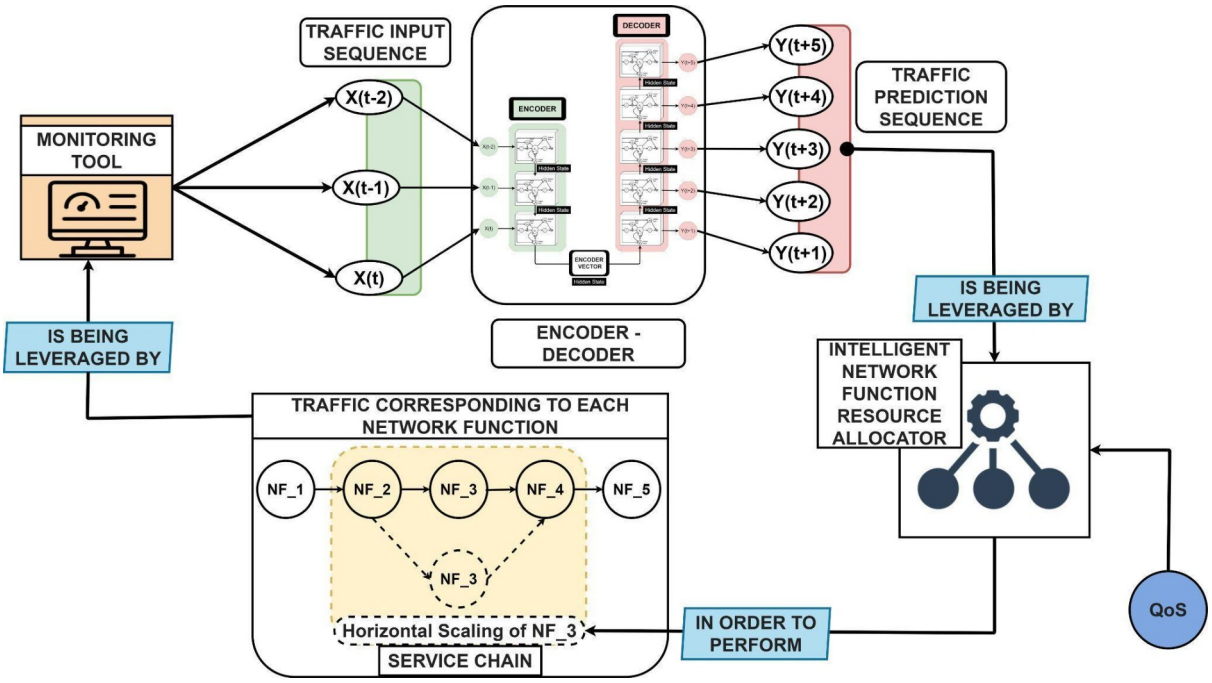
προσέγγιση τύπου αλληλουχία σε ακολουθία με τον κωδικοποιητή - αποκωδικοποιητή διατηρεί καλή ακρίβεια και στα πέντε διαδοχικά χρονικά βήματα.

Η είσοδος κάθε φορά αποτελείται από ένα είδος μεταβλητής που αντιστοιχεί σε ένα χαρακτηριστικό των δεδομένων. Αυτό έγινε προκειμένου να διασφαλσιτεί η γενικότητα της μεθόδου που χρησιμοποιείται. Η ταυτόχρονη χρήση πολλαπλών χαρακτηριστικών εισόδου μπορεί να αυξήσει την ακρίβεια της πρόβλεψης, αλλά δεν είναι σίγουρο ότι το σύνολο αυτών των χαρακτηριστικών θα είναι πάντοτε διαθέσιμα κατά την περίοδο λειτουργίας των υπηρεσιών. Επιπλέον, εάν η μέθοδος μη μεταβλητής πρόβλεψης έχει καλά αποτελέσματα, άλλοι ερευνητές μπορούν να συμπεριλάβουν περισσότερα χαρακτηριστικά και να εφαρμόσουν μια πολυμεταβλητή προσέγγιση. Οι παρατηρήσεις δεδομένων που χρησιμοποιήσαμε προέρχονται από υλοποιήσεις του TCP, το οποίο είναι ένα εξαιρετικά αξιόπιστο πρωτόκολλο βασισμένο σε IP για επικοινωνία μεταξύ υπηρεσιών εφαρμογών και υπολογιστικών συσκευών. Το TCP εγγυάται την ακεραιότητα των δεδομένων και αποτρέπει την απώλεια δεδομένων, την καταστροφή ή την παράδοση εκτός παραγγελίας. Χρησιμοποιεί επίσης την αρίθμηση ακολουθίας πακέτων και τα πακέτα επιβεβαίωσης για την επιτυχή παράδοση δεδομένων. Οι υποδομές cloud και edge computing βασίζονται σε μεγάλο βαθμό στη σουίτα πρωτοκόλλων TCP/IP [65], καθιστώντας έτσι την ανίχνευση του TCP μια αξιόπιστη πηγή πληροφοριών σχετικά με τις υπηρεσίες μεγάλων δεδομένων και τον προοπτικό φόρτο εργασίας τους.

Η πρόβλεψη κίνησης υπηρεσιών είναι σημαντική για τη βέλτιστη διαχείριση α) Υπολογιστικών, β) Αποθηκευτικών Πόρων και γ) Πόρων δικτύου πάνω από τους οποίους εκτελούνται οι υπηρεσίες μεγάλων δεδομένων. Αυτοί οι πόροι περιλαμβάνουν κόμβους αφιερωμένους στην επεξεργασία δεδομένων, τη συμπίεση, την αναφορά και την οπτικοποίηση και λειτουργίες δικτύου όπως τείχη προστασίας, συστήματα πρόληψης εισβολής και μετάφραση διευθύνσεων δικτύου [66]. Η πρόβλεψη κυκλοφορίας υπηρεσιών αξιοποιείται για να βελτιστοποιηθεί η προληπτική ανάπτυξη πόρων ως απόκριση στις αλλαγές στα αιτήματα υπηρεσίας και στον φόρτο εργασίας, εφαρμόζοντας μια ακολουθία διαδικασιών για το πώς/πού να αποθηκεύονται, να επεξεργάζονται και να μεταδίδονται δεδομένα σε διάφορα περιβάλλοντα εκτέλεσης.

Το Σχήμα 10 απεικονίζει ένα σύγχρονο σενάριο αλυσίδας υπηρεσιών. Το εργαλείο παρακολούθησης κυκλοφορίας παρέχει τις τρέχουσες τιμές κυκλοφορίας στον κωδικοποιητή-αποκωδικοποιητή, ο οποίος στη συνέχεια εξάγει την ακολουθία πρόβλεψης κυκλοφορίας. Η ακολουθία πρόβλεψης κυκλοφορίας αξιοποιείται από το Intelligent Network Function Resource Allocation για την παροχή των απαραίτητων πόρων εν κινήσει, διατηρώντας έτσι την εκπλήρωση των απαιτήσεων QoS σε αποδεκτά επίπεδα. Ο

μηχανισμός κατανομής πόρων Intelligent Network Function Resource Allocator εκτελεί αυξομείωση του αριθμού των πόρων και κατανέμοντας τους δυναμικά ώστε να είναι αρκετοί για να διαχειριστούν τις ροές δεδομένων των επόμενων χρονικών περιόδων. Κάθε πόρος έχει διαφορετικό χρόνο ανάπτυξης. Για παράδειγμα, μια μετάφραση διεύθυνσης δικτύου χρειάζεται λιγότερο από πέντε λεπτά, αλλά μια εικονική μηχανή νέφους απαιτεί 10-15 λεπτά ή ακόμα περισσότερα σε ορισμένες περιπτώσεις για ανάπτυξη, ανάλογα με τις υπηρεσίες δεδομένων που εκτελεί. Τα έξυπνα στοιχεία κατανομής πόρων και οι μηχανισμοί πρόβλεψης μπορούν να εκτελούνται τοπικά στους κόμβους επεξεργασίας ακμών. Σε αυτήν την έρευνα, δεν εξετάζουμε τον έξυπνο μηχανισμό κατανομής πόρων, αλλά εστιάζουμε στην πρόβλεψη της κυκλοφορίας και στην ακρίβεια των προβλέψεων κατά τη διάρκεια διαφορετικών χρονικών βημάτων.



Σχήμα 10: Ένα σύγχρονο σενάριο αλυσίδας υπηρεσιών.

7.3 Μοντελοποίηση και πρόβλεψη φόρτου δικτύου με χρήση χρονοσειρών

Η έκφραση του φόρτου υπηρεσιών μπορεί να μοντελοποιήθει ως χρονοσειρά. Ο φόρτος υπηρεσιών μπορεί να αναλυθεί με τη χρήση διερευνητικών και προγνωστικών μεθόδων. Έπειτα από μια διερευνητική ανάλυση, προκύπτουν επαναλαμβανόμενα μοτίβα φόρτου. Με βάση τα αποτελέσματα αυτά, πραγματοποιείται συσχέτιση μεταξύ των τιμών φόρτου και αποσύνθεση χρονοσειρών. Στην προγνωστική ανάλυση, προβλέπουμε την τιμή του επόμενου βήματος με βάση τις τιμές που συλλέχθηκαν κατά τα προηγούμενα χρονικά βήματα.

Η αποσύνθεση χρονοσειρών περιλαμβάνει τη μέση τιμή των παρατηρήσεων, την τάση που αντιπροσωπεύει την αυξανόμενη ή φθίνουσα συμπεριφορά των τιμών, την εποχικότητα που εκφράζει τον επαναλαμβανόμενο βραχύ κύκλο στις τιμές και τα υπολείμματα που είναι ένα σύνολο από τυχαία παραλλαγές. Ο φόρτος υπηρεσιών χαρακτηρίζεται από εποχικότητα, επειδή η συμπεριφορά των χρηστών χαρακτηρίζεται από τακτές και προβλέψιμες αλλαγές που επαναλαμβάνονται κάθε μέρα ή εβδομάδα. Σε μακροσκοπική ανάλυση, οι περισσότερες υπηρεσίες έχουν μια αυξητική τάση όσον αφορά τις απαιτήσεις στα πλαίσια όγκου δεδομένων, η οποία εκδηλώνεται κατά την καταγραφή των χρονοσειρών που προκύπτουν από δεδομένα καθημερινής χρήσης.

Ο ακρογωνιαίος λίθος ανάλυσης χρονοσειρών είναι η σταθερότητα. Σταθερότητα σημαίνει ότι οι στατιστικές ιδιότητες των παρατηρήσεων δεν αλλάζουν με την πάροδο του χρόνου. Έτσι, οι χρονοσειρές με τάσεις ή με εποχικότητα, δηλώνονται ως μη σταθερές. Προκειμένου να μοντελοποιήσουμε μια ακολουθία παρατηρήσεων ως χρονοσειρές, θα πρέπει να τη μετατρέψουμε σε σταθερή με μια διαφορική πράξη και στη συνέχεια να αφαιρέσουμε την τάση και τις παρενέργειες της εποχικότητας.

Η αυτοσυσχέτιση απεικονίζει τις σχέσεις μεταξύ των τωρινών παρατηρήσεων και των παρατηρήσεων που συλλέχθηκαν σε προηγούμενα χρονικά βήματα, ενώ η μερική αυτοσυσχέτιση απορρίπτει τις σχέσεις των παρατηρήσεων που περιγράφουν τις άμεσες σχέσεις μεταξύ των παρατηρήσεων και των καθυστερήσεων τους. Η θετική αυτοσυσχέτιση μπορεί να θεωρηθεί ως μια τάση για την υπηρεσία δεδομένων να διατηρήσει τις ίδιες τιμές φόρτου από τη μια χρονική περίοδο στην άλλη. Στην πειραματική αξιολόγηση, θα δούμε διαγράμματα της αποσύνθεσης χρονοσειρών και της αυτοσυσχέτισης.

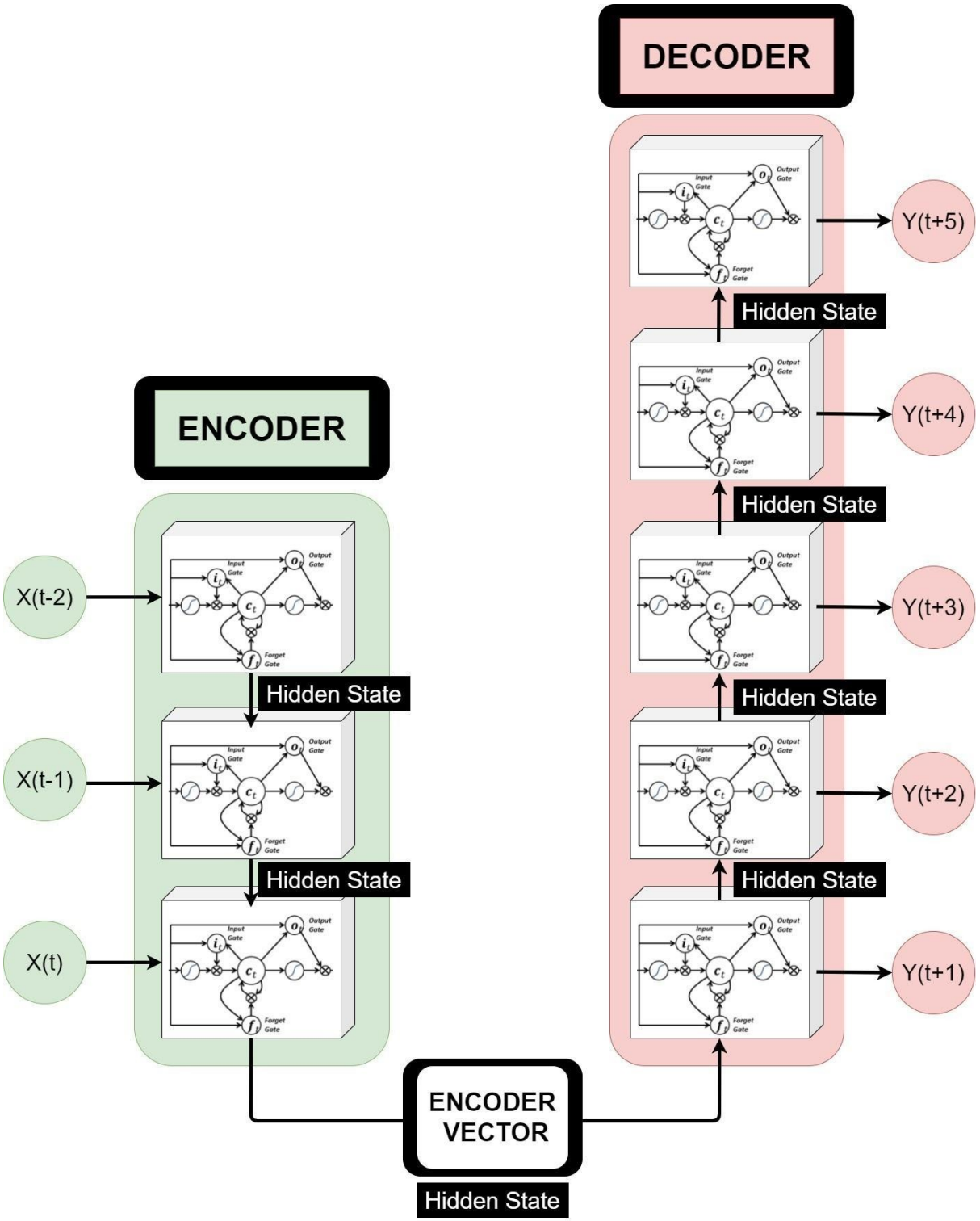
Το Auto-Regressive Moving-Average (ARMA) είναι ένα μοντέλο πρόβλεψης χρονοσειρών που αποτελείται από το τμήμα Auto-Regressive (AR) που περιλαμβάνει την παλινδρόμηση της μεταβλητής στις δικές της τιμές καθυστέρησης και το τμήμα Moving-Average (MA) που μοντελοποιεί το σφάλμα ως ένα γραμμικό συνδυασμό σφαλμάτων που εμφανίζονται ταυτόχρονα και σε διάφορες χρονικές στιγμές. Το μοντέλο ARMA χαρακτηρίζεται από δύο τιμές: τον αριθμό του αυτοπαλινδρομικού όρου **p** και τον αριθμό των σφαλμάτων πρόβλεψης με καθυστέρηση στην εξίσωση πρόβλεψης, **q**.

Το ARMA χρησιμοποιείται σε σταθερές χρονοσειρές. Εάν η χρονοσειρά δεν είναι σταθερή, εφαρμόζουμε την πράξη διαφοροποίησης λαμβάνοντας τη διαφορά μεταξύ των τιμών δεδομένων και των προηγούμενων τιμών. Ο αριθμός των φορών που πρέπει να διαφοροποιηθεί η αρχική χρονοσειρά για να επιτευχθεί σταθερότητα που συμβολίζεται με τον όρο **d**.

Το μοντέλο πρόβλεψης Autoregressive Integrated Moving Average (ARIMA) βασίζεται στο ARMA και κάνει και αυτό χρήση του **d** προκειμένου να κάνει τις διάφορες χρονοσειρές σταθερές. Για να καθοριστεί η βέλτιστη σειρά ARMA και ARIMA των όρων **p**, **q** και **d**, μπορούμε να προσδιορίσουμε τη σειρά διαφοροποίησης του **d** με δοκιμές μηδενικής υπόθεσης Kwiatkowski–Phillips–Schmidt–Shin [21], Augmented Dickey-Fuller ή Phillips–Perron.

8. Αρχιτεκτονικές Κωδικοποιητή- Αποκωδικοποιητή για τη πρόβλεψη φόρτου δικτύου σε μορφή πολλαπλών βημάτων

Αυτό που κάνει τα μοντέλα κωδικοποιητή-αποκωδικοποιητή ιδανικό υποψήφιο για πρόβλεψη τύπου seq2seq είναι η εγγενής ικανότητά τους να χαρτογραφούν αλληλουχίες διαφορετικού μήκους μεταξύ τους. Αυτή η λειτουργικότητα είναι το αποτέλεσμα της αρχιτεκτονικής του μοντέλου. Ο κωδικοποιητής παίρνει την ακολουθία εισόδου και αναπαριστά τις πληροφορίες ως μεταβλητές. Ο αποκωδικοποιητής ρυθμίζεται στις τελικές καταστάσεις του κωδικοποιητή και εκπαιδεύεται κατάλληλα ώστε να δημιουργεί εξόδους με βάση τις πληροφορίες που συλλέγει ο κωδικοποιητής. Στο Σχήμα 11 απεικονίζεται μια τυπική τοπολογία κωδικοποιητή-αποκωδικοποιητή.



Σχήμα 11: Μια τυπική τοπολογία κωδικοποιητή-αποκωδικοποιητή

8.1 Κωδικοποιητές

1) Κωδικοποιητής απλών LSTM: Για τη δημιουργία του αρχιτεκτονικής κωδικοποιητή της χρησιμοποιήθηκε ένα μοντέλο LSTM μονής κατεύθυνσης. Αυτό το μοντέλο λαμβάνει ως είσοδο τις τιμές που αντιστοιχούν στα τελευταία 3 χρονικά βήματα και παράγει ένα διάνυσμα εξόδου N στοιχείων που αποτελεί μια αρχική αναπαράσταση της ακολουθίας εισόδου. Το μέγεθος του N αντιστοιχεί στον αριθμό των μονάδων LSTM που χρησιμοποιούνται.

2) Κωδικοποιητής αμφίδρομων LSTM: Σε αντίθεση με τον προηγούμενο κωδικοποιητή, ο κωδικοποιητής αμφίδρομων LSTM αποτελείται από ένα μοντέλο αμφίδρομων LSTMs. Αυτό το μοντέλο λαμβάνει ως είσοδο τις τιμές που αντιστοιχούν στα τελευταία 3 χρονικά βήματα και παράγει ένα διάνυσμα εξόδου N στοιχείου που αποτελεί μια αρχική αναπαράσταση της ακολουθίας εισόδου. Το μέγεθος του N αντιστοιχεί στον αριθμό των μονάδων αμφίδρομων LSTMs που χρησιμοποιούνται.

3) Κωδικοποιητής CNN-LSTM: Τα νευρωνικά δίκτυα συνέλιξης (Convolution Neural Networks) (CNN) δεν έχουν σχεδιαστεί για να δέχονται είσοδο σε μορφή ακολουθιών. Ωστόσο, ένα μονοδιάστατο επίπεδο CNN είναι ικανό να λαμβάνει χρονοσειρές ως είσοδο και στη συνέχεια να μαθαίνει τα κύρια χαρακτηριστικά τους. Επιπλέον, τόσο τα CNN όσο και τα LSTM αναμένουν μια τρισδιάστατη είσοδο. Όσον αφορά τα CNN, αυτό το σχεδιαστικό χαρακτηριστικό έχει διαμορφωθεί για να μπορεί να λαμβάνει τα τρία διαφορετικά κανάλια που στα χρώματα κόκκινο, πράσινο και μπλε . Τα LSTM από την άλλη πλευρά απαιτούν μια τρισδιάστατη είσοδο που αντιστοιχεί στον α) αριθμό δειγμάτων, β) αριθμό χρονικών βημάτων προς εξέταση και γ) αριθμό χαρακτηριστικών. Χρησιμοποιούνται δύο μονοδιάστατα επίπεδα συνέλιξης. Το πρώτο διαβάζει την ακολουθία εισόδου και προβάλλει το αποτέλεσμα σε έναν χάρτη χαρακτηριστικών. Το δεύτερο λαμβάνει ως είσοδο την έξοδο του πρώτου και εκτελεί την ίδια λειτουργία για να ενισχύσει όποια τυχόν σημαντικά χαρακτηριστικά εμπεριέχονται στον αρχικό χάρτη χαρακτηριστικών. Στη συνέχεια χρησιμοποιείται ένα επίπεδο max-pooling προκειμένου να συγκεντρωθούν χαρακτηριστικά από τους χάρτες που δημιουργήθηκαν από τα δύο προηγούμενα επίπεδα. Τέλος, χρησιμοποιείται ένα επίπεδο επίπεδο για να αναδιαμορφώσει την έξοδο του κωδικοποιητή στο επιθυμητό σχήμα, το οποίο μπορεί στη συνέχεια να επεξεργαστεί ο αποκωδικοποιητής.

4) Υβριδικό μοντέλο απλών και αμφίδρομων LSTM: Αυτό το νέο αρχιτεκτονικό μοντέλο είναι το προϊόν της χρήσης αμφίδρομων και απλών LSTM αντί μόνο του ενός εκ των δύο. Το επίπεδο εισόδου είναι ένα αμφίδρομο LSTM. Στη συνέχεια, ένα επίπεδο LSTM μονής κατεύθυνσης στοιβάζεται πάνω από το διπλής κατεύθυνσης. Το αμφίδρομο στρώμα θα παρέχει μια έξοδο κρυφής κατάστασης για κάθε χρονικό βήμα σε τρισδιάστατη μορφή, η οποία στη συνέχεια χρησιμοποιείται ως είσοδος από το στρώμα μονής κατεύθυνσης. Η βασική ιδέα πίσω από αυτήν την αρχιτεκτονική επιλογή είναι το γεγονός ότι με την εισαγωγή ετερογενών επιπέδων το μοντέλο θα μπορεί να εκμεταλλευτεί τις χρονικές συσχετίσεις που υπάρχουν στις διάφορες χρονοσειρές με πιο εξελιγμένο τρόπο σε σύγκριση με τα υπόλοιπα μοντέλα. Επιπλέον, το γεγονός ότι χρησιμοποιούνται πολλαπλά επίπεδα επιτρέπει την αναπαράσταση των χαρακτηριστικών της ακολουθίας εισόδου με πιο εύρωστο τρόπο. Η ίδια σχεδιαστική λογική εφαρμόζεται και επίπεδο αποκωδικοποιητή προκειμένου να αντικατοπτρίζεται η μορφολογία του κωδικοποιητή. Αντί για το βασικό μοντέλο LSTM που χρησιμοποιήθηκε στους αποκωδικοποιητές των μοντέλων που αναλύθηκαν προηγουμένως, το υβριδικό μοντέλο χρησιμοποιεί ένα στρώμα διπλής κατεύθυνσης στοιβαγμένο πάνω σε ένα στρώμα μονής κατεύθυνσης. Αυτή η δομική συμμετρία επιτρέπει στον αποκωδικοποιητή να ανακατασκευάσει ορθά τα υποκείμενα χρονικά μοτίβα της ακολουθίας εισόδου.

8.2 Αποκωδικοποιητής

Ο αποκωδικοποιητής υλοποιείται με τη χρήση ενός μοντέλου LSTM. Κάθε μονάδα που αποτελεί μέρος του αποκωδικοποιητή αναμένεται να δώσει μια τιμή για κάθε ένα από τα 5 μελλοντικά χρονικά βήματα που εξετάζονται. Για να γίνει αυτό, χρησιμοποιείται ένα επίπεδο Repeat-Vector. Επιπλέον, είναι απαραίτητο να ενσωματωθούν δύο επιπλέον στρώματα. Τα επίπεδα αυτά είναι το επίπεδο ερμηνείας και το επίπεδο εξόδου. Το επίπεδο ερμηνείας είναι ουσιαστικά ένα πλήρως συνδεδεμένο επίπεδο και Ο σκοπός του είναι να ερμηνεύσει ορθά κάθε χρονικό βήμα στην ακολουθία εξόδου του αποκωδικοποιητή και να στείλει το προϊόν που προκύπτει στο επίπεδο εξόδου. Αυτή η συγκεκριμένη μεθοδολογία έχει ως αποτέλεσμα μια πρόβλεψη ενός βήματος στην ακολουθία εξόδου. Δεδομένου ότι είναι επιθυμητή η πρόβλεψη των επόμενων 5 βημάτων, είναι απαραίτητο να τυλιχτεί τόσο το επίπεδο ερμηνείας όσο και το επίπεδο εξόδου μέσα σε ένα περιτύλιγμα Time2Vec. Με αυτόν τον τρόπο, όλα τα στοιχεία της εξόδου που παρέχεται από τον αποκωδικοποιητή θα υποβληθούν σε επεξεργασία από το ίδιο πλήρως συνδεδεμένο επίπεδο εξόδου.

8.3 Βελτιστοποίηση υπερπαραμέτρων

Για να βρεθούν οι βέλτιστες τιμές για τις διάφορες υπερπαραμέτρους χρησιμοποιήθηκε η Bayesian βελτιστοποίηση που παρέχεται από τη μονάδα Keras-Tuner. Οι συγκεκριμένες παράμετροι που εξετάστηκαν ήταν ο αριθμός των εποχών, ο ρυθμός εκμάθησης, ο αριθμός των μονάδων στο επίπεδο εισόδου, ο αριθμός των μονάδων στο επίπεδο LSTM του αποκωδικοποιητή και πιο συγκεκριμένα ο αριθμός των μονάδων για τα επίπεδα LSTM και αμφίδρομων LSTM που αποτελούν το Υβριδική αρχιτεκτονική κωδικοποιητή - αποκωδικοποιητή.

9. Πειραματική Αξιολόγηση μέρος Β

9.1 Υλοποίηση Μοντέλου, Πλαίσια και Σύνολο Δεδομένων

Οι τέσσερις κωδικοποιητές-αποκωδικοποιητές, τα μοντέλα ARMA και ARIMA υλοποιούνται στην Python 3 χρησιμοποιώντας τα πλαίσια NumPy, pandas, statistics, Scikit-learn, SciPy, Scikit-Optimize, TensorFlow 2 και το API Keras υψηλότερου επιπέδου. Το περιβάλλον που χρησιμοποιήσαμε είναι το σημειωματάριο Jupyter του Google Colaboratory. Ο πηγαίος κώδικας των πειραμάτων είναι διαθέσιμος στο τέλος της διπλωματικής εργασίας στο παράρτημα. Τα πειράματα πραγματοποιήθηκαν σε ένα πραγματικό σύνολο δεδομένων με ίχνη TCP υπηρεσιών δεδομένων και ομαδοποιήθηκαν σε βήματα διάρκειας πέντε λεπτών.

9.2 Μετρικές Αξιολόγησης

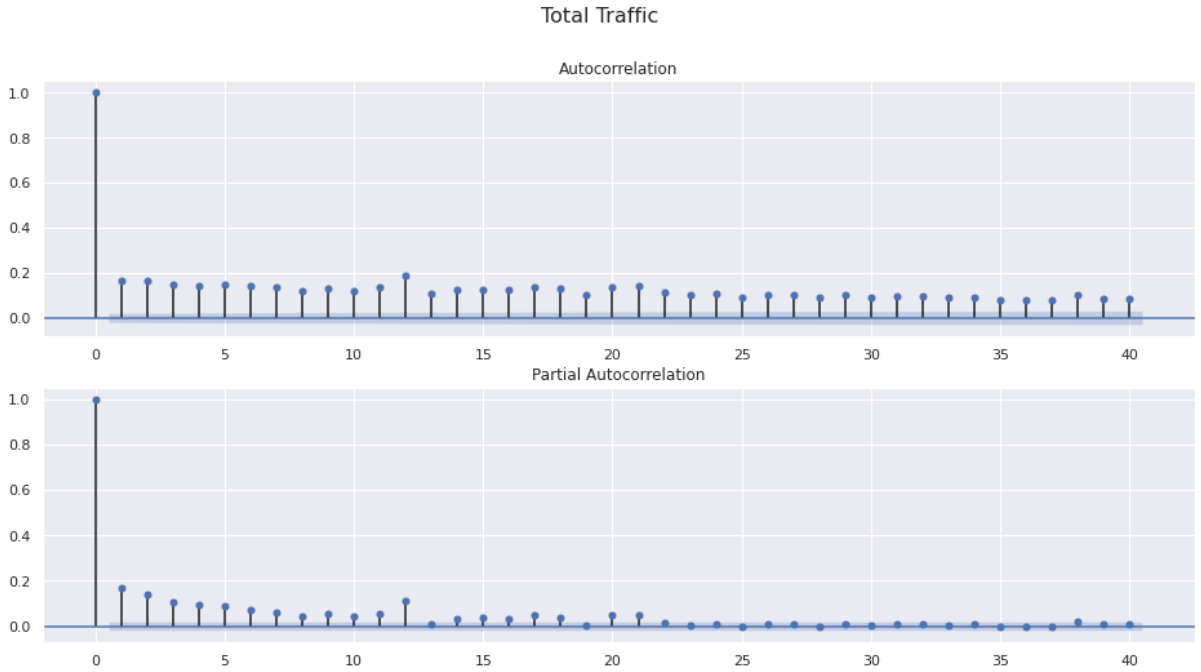
Για να αξιολογηθεί η ακρίβεια του προτεινόμενου μοντέλου, χρησιμοποιήθηκαν μετρήσεις σφάλματος και μετρήσεις χρόνου. Στις μετρήσεις σφάλματος ανήκουν το μέσο απόλυτο σφάλμα (MAE) και το μέσο τετραγωνικό σφάλμα (RMSE). Το MAE εκφράζει τη μέση απόλυτη διαφορά μεταξύ των τιμών στόχου και των προβλεπόμενων τιμών. Το μέσο τετραγωνικό σφάλμα εκφράζει την τυπική απόκλιση των σφαλμάτων δίνοντας έμφαση στην διασπορά των σφαλμάτων στον χώρο τιμών. Το MAE προτιμάται όταν όλα τα σφάλματα έχουν την ίδια σημασία, ενώ το RMSE όταν πρέπει να τιμωρούμε τα μεγάλα σφάλματα ακόμα και αν είναι λίγα. Στα πειράματά που πραγματοποιήθηκαν, ο όγκος των μεταδιδόμενων δεδομένων εκφράζεται σε megabytes και η διάρκεια των υπηρεσιών σε δευτερόλεπτα.

Στους πίνακες 3 εώς 5, γίνεται αξιολόγηση για κάθε χρονικό βήμα ξεχωριστά ώστε να καθοριστεί η ικανότητα πρόβλεψης των μοντέλων σε κάθε συγκεκριμένο χρονικό βήμα.

9.3 Διερευνητική Ανάλυση Δεδομένων

Λόγω της μορφολογίας των δεδομένων φόρτου και του αριθμού των αιτημάτων, εμφανίζεται μια συνεχής ισχυρή συσχέτιση μεταξύ των τιμών καθυστέρησης. Αυτό

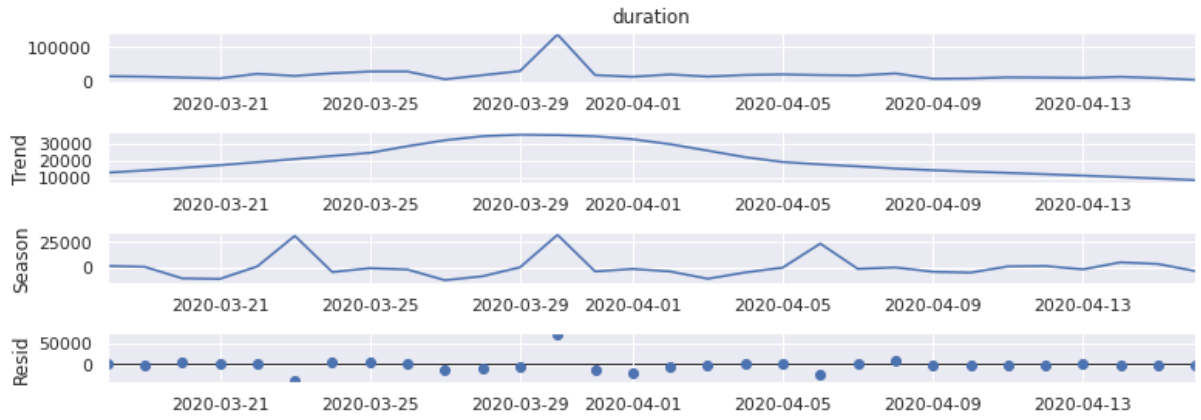
απεικονίζεται στα Σχήματα 12 και 13 όπου ο άξονας x των γραφημάτων ACF και PACF υποδεικνύει την καθυστέρηση βάση της οποίας υπολογίζεται ο συντελεστής αυτοσυσχέτισης και ο συντελεστής μερικής αυτοσυσχέτισης. Ο άξονας y δείχνει την τιμή της συσχέτισης (μεταξύ -1 και 1). Συνεπώς πρέπει να χρησιμοποιηθούν πολύπλοκα μοντέλα προκειμένου να ενσωματωθεί ορθά το moving average με βάση τις ισχυρά συσχετισμένες καθυστερήσεις που προέρχονται από το διάγραμμα αυτοσυσχέτισης, λόγω της υψηλής τιμής του moving average. Για το λόγο αυτό, ο αριθμός των αιτημάτων και ο όγκος των δεδομένων ορίζονται καλύτερα από τα διαγράμματα μερικής αυτοσυσχέτισης που αφαιρούν τις έμμεσες συσχετίσεις.



Σχήμα 12: Αυτοσυσχέτιση και μερική αυτοσυσχέτιση όγκου δεδομένων.

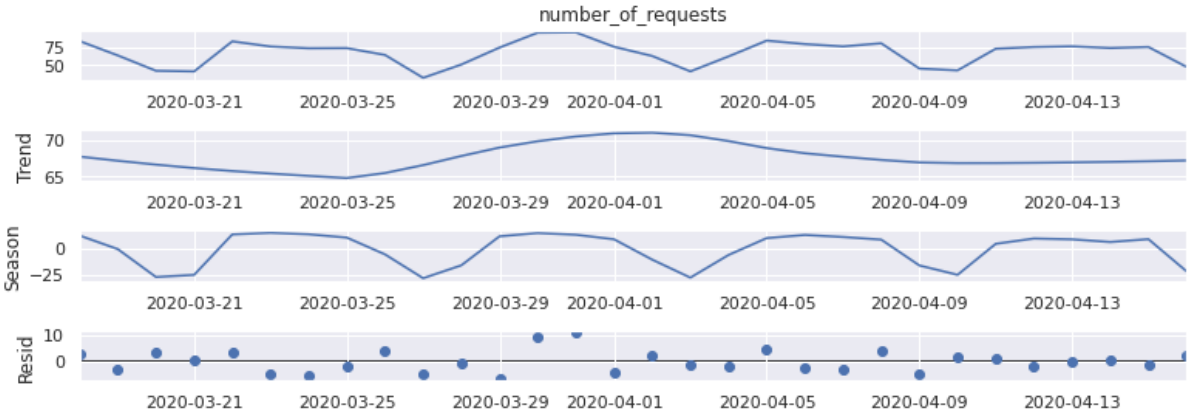
Σχήμα 13: Αυτοσυσχέτιση και μερική αυτοσυσχέτιση αριθμού αιτημάτων.

Από την γραφική παράσταση μερικής αυτοσυσχέτισης του όγκου δεδομένων, παρατηρούμε ότι οι συνολικές τιμές επισκεψιμότητας σχηματίζουν ισχυρές εξαρτήσεις σε εύρος 2-3 τιμών και στη συνέχεια τείνουν να παρουσιάζουν αποκλίσεις στη συμπεριφορά τους. Δεδομένης αυτής της παρατήρησης, είναι ασφαλές να συμπεράνουμε ότι σε ένα μοντέλο πρόβλεψης είναι σημαντικό να λαμβάνονται υπόψη οι τιμές που είναι έως και 10-15 λεπτά πριν από το τρέχον χρονικό βήμα. Με παρόμοιο τρόπο, η γραφική παράσταση μερικής αυτοσυσχέτισης του αριθμού των αιτημάτων δείχνει ότι σχηματίζονται ισχυρές σχέσεις μερικής αυτοσυσχέτισης για τιμές που είναι ίσες με 1, 2 και 3. Συμπεραίνουμε έτσι ότι κάθε τρέχουσα τιμή εξαρτάται σε σημαντικό βαθμό από τα προηγούμενα χρονικά βήματα, και πως για να επιτύχουμε μεγαλύτερη ακρίβεια πρόβλεψης, πρέπει να συμπεριλάβουμε τις τιμές για τα τελευταία 3 χρονικά βήματα. Επιπλέον, παρατηρώντας τα διαγράμματα συμπεραίνουμε ότι δεν υπάρχει εποχικότητα στα δεδομένα, καθώς η κατανομή των τιμών είναι τυχαία.



Σχήμα 14: Περιοδικότητας διάρκειας συνεδρίας.

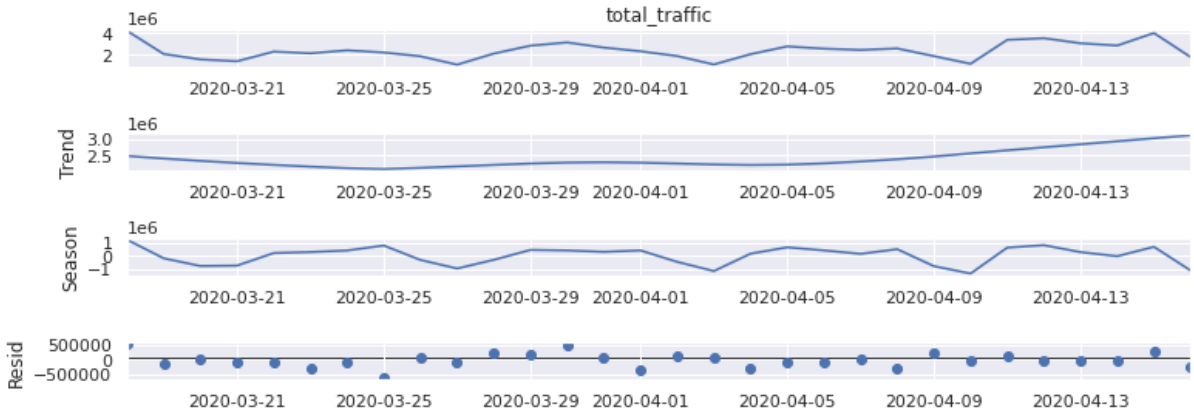
Μετά την εξέταση των γραφημάτων αυτοσυσχέτισης και μερικής αυτοσυσχέτισης φαίνεται να μην υπάρχει εποχικότητα όσον αφορά τις τιμές δεδομένων. Προκειμένου να κατανοηθούν καλύτερα τα πρότυπα συμπεριφοράς των δεδομένων σε πολλαπλά χρονικά βήματα, είναι επιτακτική ανάγκη να αλλάξουμε τη συχνότητα παρατήρησης περιοδικών φαινομένων, προκειμένου να εστιάσουμε αρχικά σε συχνότητες επιπέδου ημέρας, που θεωρητικά είναι το επίπεδο στο οποίο τείνουν να εκδηλώνονται τα διάφορα περιοδικά φαινόμενα σχετικά με τα δεδομένα φόρτου δικτύου. Από τα αποτελέσματα των σχημάτων 14, 15 και 16 εμφανίζεται περιοδικότητα στον όγκο των δεδομένων και στον αριθμό των αιτημάτων.



Σχήμα 15: Περιοδικότητα αριθμού αιτημάτων.

Τα μοτίβα περιοδικότητας έχουν κάποιες διακυμάνσεις τιμών σε σχέση με κάθε περίοδο σήματος, μπορούμε να συμπεράνουμε με ασφάλεια πως υπάρχει ημερήσια περιοδικότητα

στις τιμές των δεδομένων. Αντίθετα, κατά την εξέταση της συχνότητας σε επίπεδο ώρας, οι τιμές σχηματίζουν μεγάλες διακυμάνσεις, χωρίς τα δεδομένα να συγκλίνουν σε ένα σταθερό περιοδικό φαινόμενο. Τέλος, παρατηρείται ότι για τη συχνότητα επιπέδου ημέρας οι τιμές του θορύβου στο σήμα όγκο δεδομένων αποτυπώνονται περισσότερο με πιο σαφή τρόπο.



Σχήμα 16: Περιοδικότητα όγκου δεδομένων

9.4 Αξιολόγηση αποτελεσμάτων πρόβλεψης

TABLE I
COMPARISON FOR THE FIRST SINGLE-STEP PREDICTION STATISTICAL TIME SERIES, DEEP LEARNING AND ENCODER-DECODER METHODS.

Method	Requests		Traffic		Duration		Training	Inference
	RMSE	MAE	RMSE	MAE	RMSE	MAE	Time	Time
ARMA	23.149	16.493	1430557	937158	42303	19242	14.290	0.841
ARIMA	23.199	16.554	1424586	889310	**41251**	14026	**13.665**	**0.297**
LSTM 1step	**22.344**	**15.936**	**804957**	**610643**	42531	10497	38.458	2.680
LSTM vec	26.205	18.398	831477	645565	22372	10957	52.138	2.622
ED LSTM	26.450	18.977	836682	639757	44654	11390	136.256	2.684
ED CNN-LSTM	26.602	18.998	838714	649777	42499	**10029**	88.254	2.540
ED Bid-LSTM	26.734	19.108	844648	646605	42504	10040	167.308	2.759
Hybrid	26.052	18.688	806952	645182	42396	10172	318.303	2.632

Πίνακας 2.

Ο Πίνακας 2 συνοψίζει τη σύγκριση της πρόβλεψης σε επίπεδο ενός σταδίου. Για την πρόβλεψη ενός βήματος, το μοντέλο LSTM έχει καλύτερη ακρίβεια αιτημάτων αιτημάτων και όγκου δεδομένων. Κάνοντας πειράματα με μοντέλα πολλαπλών βημάτων είδαμε ότι οι κωδικοποιητές-αποκωδικοποιητές ξεπερνούν το μοντέλο ARIMA. Έτσι, απεικονίζονται στους πίνακες 3 εώς 5 μόνο τις διάφορες αρχιτεκτονικές κωδικοποιητή - αποκωδικοποιητή.

TABLE II
COMPARISON OF THE MULTI-STEP PREDICTION METHODS FOR THE NUMBER OF REQUEST.

Requests	1st step		2nd step		3rd step		4th step		5th step	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
LSTM Vec	22.794	**15.704**	25.808	**17.885**	26.637	**18.956**	27.539	**19.378**	28.249	**20.066**
ED LSTM	22.675	16.147	25.807	18.422	27.119	19.454	27.901	20.157	29.071	20.705
ED CNN-LSTM	23.348	16.584	25.877	18.311	26.906	19.282	28.006	20.063	28.873	20.749
ED Bid-LSTM	22.827	16.071	26.241	18.615	27.191	19.275	28.311	20.401	29.102	21.177
ED Hybrid	**22.656**	16.173	**25.495**	18.145	**26.489**	19.098	**27.504**	19.754	**28.114**	20.268

TABLE III
COMPARISON OF THE MULTI-STEP PREDICTION METHODS FOR THE TRAFFIC (TRANSMITTED DATA).

Traffic	1st step		2nd step		3rd step		4th step		5th step	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
LSTM Vec	781968	605345	798364	628584	832546	652228	846999	**660414**	862397	**669140**
ED LSTM	770830	**581789**	801711	**614400**	845053	**649910**	876312	672496	889506	680190
ED CNN-LSTM	801861	615874	815760	631545	843934	657737	864654	671775	867361	671956
ED Bid-LSTM	785167	595622	824090	632736	855272	658113	878195	673668	880516	672885
ED Hybrid	**757915**	601998	**788857**	632948	**812605**	651148	**831969**	666148	**843414**	673666

TABLE IV
COMPARISON OF THE MULTI-STEP PREDICTION METHODS FOR THE SESSION DURATION.

Duration	1st step		2nd step		3rd step		4th step		5th step	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
LSTM Vec	43604	11091	43085	10710	43105	10924	43105	10924	43437	11078
ED LSTM	44816	11377	44473	11354	44518	11407	44723	11398	44740	11416
ED CNN-LSTM	43232	10591	42702	10139	42409	**10018**	**42142**	**9697**	**42012**	**9701**
ED Bid-LSTM	**42407**	**10125**	42702	10152	42594	10125	42434	9983	42380	9814
ED Hybrid	42437	10179	**42324**	**10126**	**42363**	10212	42429	10161	42425	10184

Πίνακας 5.

Κατά την προσπάθεια εκτέλεσης πρόβλεψης πολλαπλών βημάτων, αναμένεται ότι οι μετρήσεις RMSE και το MAE επηρεάζονται σε μεγάλο βαθμό από το πόσο μακριά στο μέλλον προσπαθούμε να προβλέψουμε. Με άλλα λόγια, τα RMSE και MAE του πρώτου χρονικού βήματος αναμένεται να είναι χαμηλότερα σε σύγκριση με αυτά του πέμπτου χρονικού βήματος. Αυτές οι εκτιμήσεις ικανοποιούνται μετά την εξέταση των προβλέψεων πολλαπλών βημάτων που παράγονται από τα μοντέλα κωδικοποιητή-αποκωδικοποιητή σε σχέση με έναν αριθμό αιτημάτων και του όγκου δεδομένων, όπως μπορούμε να δούμε στον Πίνακα 3 και τον Πίνακα 4 αντίστοιχα.

Όσον αφορά την πρόβλεψη διάρκειας σε όλα τα μοντέλα που εφαρμόζονται, φαίνεται να υπάρχει ένα σαφές μοτίβο όπως μπορούμε να δούμε στον Πίνακα 5. Τα αποτελέσματα RMSE και MAE στα διάφορα χρονικά βήματα είναι σχετικά τα ίδια. Αυτό το φαινόμενο προκαλείται από το γεγονός ότι οι χρονοσειρές διάρκειας φέρουν μηδενική αυτοσυσχέτιση, καθιστώντας έτσι τα διάφορα μοντέλα ανίκανα να πραγματοποιήσουν ορθές προβλέψεις βασισμένες σε χρονικά μοτίβα. Αντίθετα, τα μοντέλα αναγκάζονται να παράγουν προβλέψεις με βάση τη θεμελιώδη ταλάντωση της χρονοσειράς διάρκειας προκειμένου να ελαχιστοποιηθεί η συνάρτηση απώλειας κατά τη φάση προσαρμογής.

Το γεγονός ότι το υβριδικό μοντέλο χρησιμοποιεί μεγαλύτερο αριθμό στρωμάτων του επιτρέπει να αξιοποιεί καλύτερα τα χαρακτηριστικά του σήματος σε σύγκριση με τα άλλα μοντέλα. Αυτό το φαινόμενο ενισχύεται από το γεγονός ότι το υβριδικό μοντέλο αποτελείται από ετερογενή στρώματα (αμφίδρομα και απλά) τα οποία επιτρέπουν την εκμετάλλευση των χρονικών μοτίβων με πιο ορθό τρόπο.

Αυτός ο ισχυρισμός υποστηρίζεται από το γεγονός ότι το υβριδικό μοντέλο παρήγαγε τις καλύτερες βαθμολογίες RMSE όσον αφορά τον όγκο των δεδομένων και τον αριθμό των αιτημάτων. Από την άλλη πλευρά, οι καλύτερες βαθμολογίες MAE σε σχέση με τον όγκο δεδομένων και τον αριθμό των αιτημάτων παρήχθησαν από διάφορα μοντέλα βασισμένα στο LSTM των οποίων η απλούστερη και πιο ρηχή αρχιτεκτονική τους επέτρεψε να συντονιστούν στη θεμελιώδη ταλάντωση των αντίστοιχων χρονοσειρών. Ωστόσο, το υβριδικό μοντέλο ήταν σε θέση να ακολουθήσει το σήμα με μεγαλύτερη ακρίβεια, έχοντας τη δυνατότητα να παράγει προβλέψεις πιο κοντά στις πραγματικές τιμές.

9.5 Συμπεράσματα

Έγινε πρόβλεψη φόρτου δικτύου σε επίπεδο πολλαπλών βημάτων. Η αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή ξεπερνά άλλες στατιστικές μεθόδους όσον αφορά το RMSE για όλα τα χρονικά βήματα και τις μετρήσεις κίνησης. Προτάθηκε επίσης μια καινοτόμος αρχιτεκτονική υβριδικού κωδικοποιητή-αποκωδικοποιητή με απλά και αμφίδρομα επίπεδα LSTMs που στα περισσότερα πειράματα παρουσίασαν τα καλύτερα αποτελέσματα. Ο κύριος περιορισμός της προτεινόμενης προσέγγισης είναι η επιπλέον επιβάρυνση των Edge υποδομών εξαιτίας της παρακολούθησης και της λήψης αποφάσεων που απαιτούνται.

Η πρόβλεψη φόρτου δικτύου πολλαπλών βημάτων μπορεί να αξιοποιηθεί από τις σύγχρονες υποδομές Edge και Cloud computing για την εκπλήρωση των διαφόρων QoS απαιτήσεων των υπηρεσιών και τη βελτιστοποίηση της διαχείρισης των πόρων υπολογισμού, αποθήκευσης και δικτύου.

Ευρετήριο όρων

GRU	Gated Recurrent Unit	Επαναλαμβανόμενη Μονάδα που κάνει χρήση Πυλών
LSTM	Long Short Term Memory	Μακρά Βραχεία Μνήμη
HBES	Hybrid Bayesian Evolution Strategy	Υβριδική Bayesian Εξελικτική Στρατηγική
QoS	Quality of Service	Ποιότητα Υπηρεσίας
QoE	Quality of Experience	Ποιότητα Εμπειρίας
XR	Extended Reality	Εκτεταμένη Πραγματικότητα
AR	Augmented Reality	Επαυξημένη Πραγματικότητα
VR	Virtual Reality	Εικονική Πραγματικότητα
CI/CD	Continuous Integration / Continuous Delivery	Συνεχής Ενσωμάτωση / Συνεχής Παράδοση
LCM	Life Cycle Management	Διαχείριση Κύκλου Ζωής
AI	Artificial Intelligence	Τεχνητή Νοημοσύνη
ZSM	Zero-touch Network and Slice Life-cycle Management	Διαχείριση Φετών Κύκλου Ζωής σε Δίκτυο Μηδενικής Επαφής
KPI	Key Performance Indicators	Βασικοί Δείκτες Επίδοσης
UC	Use Case	Περίπτωση Χρήσης
H3D	Holographic 3 Dimensional	Τρισδιάστατο Ολογραφικό
CPU	Central Processing Unit	Κεντρική Μονάδα Επεξεργασίας
RAM	Random Access Memory	Μνήμη τυχαίας Προσπέλασης
GPU	Graphics Processing Unit	Μονάδα Επεξεργασίας Γραφικών
CHARITY	Cloud for Holography and Augmented RealITY	Cloud για Ολογραφία και Επαυξημένη Πραγματικότητα

GATE	Geometric Algebra inTerpolation Engine	Μηχανή Παρεμβολής Γραμμικής Άλγεβρας
AMF	Application Management Framework	Πλαίσιο Διαχείρισης Εφαρμογής
NFL	Network Function Layer	Επίπεδο Λειτουργιών Δικτύου
CAL	Convergence and Abstraction Layer	Επίδεδο Αφαιρετικότητας και Σύγκλισης
NSUP	Network Security User Privacy	Ασφάλεια Δικτύου Ιδιωτικότητα Χρήστη
IaaS	Infrastructure as a Service	Υποδομή ως Υπηρεσία
PaaS	Platform as a Service	Πλατφόρμα ως Υπηρεσία
AIRO	Artificial Intelligence Resource Orchestrator	Ενορχηστρωτής Πόρων που βασίζεται στη Τεχνητή Νοημοσύνη
LFU	Last Frequency Used	που Χρησιμοποιήθηκε Τελευταίο
SDN	Software Defined Networking	Δικτύωση που βασίζεται στο λογισμικό
DRL	Deep Reinforcement Learning	Βαθιά Ενισχυτική Μάθηση
GNN	Graph Neural Networks	Νευρωνικά Δίκτυα Γράφων
IoT	Internet of Things	Διαδίκτυο των Πραγμάτων
RNN	Recurrent Neural Networks	Επαναλαμβανόμενα Νευρωνικά Δίκτυα
ARIMA	AutoRegressive Integrated Moving Average	
ARMA	AutoRegressive Moving Average	
SLA	Service Level Agreements	Συμφωνία Επιπέδου Υπηρεσίας
MSE	Mean Squared Error	Μέσο Τετραγωνικό Σφάλμα
MAE	Mean Absolute Error	Μέσο Απόλυτο Σφάλμα
GA	Genetic Algorithm	Γενετικός Αλγόριθμος
ES	Evolution Strategy	Εξελικτική Στρατηγική
BO	Bayesian Optimization	Bayesian Βελτιστοποίηση

TCP/IP	Transfer Control Protocol / Internet Protocol	Πρωτοκόλλου Ελέγχου Μεταφοράς / Πρωτοκόλλου Διαδικτύου
TCP	Transfer Control Protocol	Πρωτοκόλλου Ελέγχου Μεταφοράς
SARIMA	AutoRegressive Moving Average	
ANN	Artificial Neural Networks	Τεχνητά Νευρωνικά Δίκτυα
CNN	Convolution Neural Network	Νευρωνικά Δίκτυα Συνέλιξης
IP	Internet Protocol	Πρωτοκόλλου Διαδικτύου
seq2seq	sequence to sequence	ακολουθία σε ακολουθία

Αναφορές

[1] A. Makris *et al.*, “Cloud for Holography and Augmented Reality,” in *2021 IEEE 10th International Conference on Cloud Networking (CloudNet)*, 2021, pp. 118–126. doi: 10.1109/CloudNet53349.2021.9657125.

[2] J. Violos, S. Tsanakas, T. Theodoropoulos, A. Leivadeas, K. Tserpes, and T. Varvarigou, “Hypertuming GRU Neural Networks for Edge Resource Usage Prediction,” in *2021 IEEE Symposium on Computers and Communications (ISCC)*, 2021, pp. 1–8. doi: 10.1109/ISCC53001.2021.9631548.

[3] T. Theodoropoulos, “An-Encoder-Decoder-Deep-Learning-Approach-for-Multistep-Service-Traffic-Prediction .” May 2021. Accessed: May 22, 2021. [Online]. Available: https://github.com/theodorosthd/An-Encoder-Decoder-Deep-Learning-Approach-for-Multistep-Service-Traffic-Prediction

[4] B. Tang *et al.*, “Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities,” *IEEE Trans. Ind. Inform.*, vol. 13, no. 5, pp. 2140–2150, Oct. 2017, doi: 10.1109/TII.2017.2679740.

[5] A. Yassine, S. Singh, M. S. Hossain, and G. Muhammad, “IoT big data analytics for smart homes with fog and cloud computing,” *Future Gener. Comput. Syst.*, vol. 91, pp. 563–573, Feb. 2019, doi: 10.1016/j.future.2018.08.040.

[6] B. M. Balachandran and S. Prasad, “Challenges and Benefits of Deploying Big Data Analytics in the Cloud for Business Intelligence,” *Procedia Comput. Sci.*, vol. 112, pp. 1112–1122, Jan. 2017, doi: 10.1016/j.procs.2017.08.138.

[7] J. Violos *et al.*, “User Behavior and Application Modeling in Decentralized Edge Cloud Infrastructures,” in *Economics of Grids, Clouds, Systems, and Services*, 2017, pp. 193–203.

[8] Wikipedia contributors, “Pepper’s ghost — Wikipedia, The Free Encyclopedia.” [Online]. Available: https://en.wikipedia.org/wiki/Pepper\%27s_ghost

[9] R. Häussler *et al.*, “Large real-time holographic 3D displays: enabling components and results,” *Appl Opt*, vol. 56, no. 13, pp. F45–F52, May 2017, doi: 10.1364/AO.56.000F45.

[10] S. A. Benton and Jr. Bove V. Michael, *Holographic Imaging*. USA: Wiley-Interscience, 2008.

[11] E. Zschau, R. Missbach, A. Schwerdtner, and H. Stolle, “Generation, encoding, and presentation of content on holographic displays in real time,” in *Three-Dimensional Imaging, Visualization, and Display 2010 and Display Technologies and Applications for Defense, Security, and Avionics IV*, 2010, vol. 7690, pp. 118–130. doi: 10.1117/12.851015.

[12] M. Hassandra *et al.*, “A virtual reality app for physical and cognitive training of older people with mild cognitive impairment: mixed methods feasibility study,” *JMIR Serious Games*, vol. 9, no. 1, p. e24170, 2021.

[13] G. Papagiannakis *et al.*, “MAGES 3.0: Tying the knot of medical VR,” in *ACM SIGGRAPH 2020 Immersive Pavilion*, 2020, pp. 1–2.

[14] P. Zikas *et al.*, “Covid-19-VR Strikes Back: innovative medical VR training,” in *ACM SIGGRAPH 2021 Immersive Pavilion*, 2021, pp. 1–2.

[15] M. Kamarianakis, N. Lydatakis, and G. Papagiannakis, “Never `Drop the Ball’ in the Operating Room: An Efficient Hand-Based VR HMD Controller Interpolation Algorithm, for Collaborative, Networked Virtual Environments,” in *Advances in Computer Graphics*, Cham, 2021, pp. 694–704.

[16] ETSI GS ZSM 002, “Zero-touch network and Service Management (ZSM): Reference Architecture,” European Telecommunications Standards Institute (ETSI), Aug. 2019.
[17] T. Taleb and et al., “Towards Supporting XR Services: Architecture and Enablers,” *Submitt. IEEE IOT J.*.
[18] A. Boudi, M. Bagaa, P. Pöyhönen, T. Taleb, and H. Flinck, “AI-Based Resource Management in Beyond 5G Cloud Native Environment,” *IEEE Netw.*, vol. 35, no. 2, pp. 128–135, 2021.
[19] C. Cao, M. Preda, and T. Zaharia, “3D Point Cloud Compression: A Survey,” in *The 24th International Conference on 3D Web Technology*, New York, NY, USA, 2019, pp. 1–9. doi: 10.1145/3329714.3338130.
[20] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, “An overview of ongoing point cloud compression standardization activities: video-based (V-PCC) and geometry-based (G-PCC),” *APSIPA Trans. Signal Inf. Process.*, vol. 9, Apr. 2020, doi: 10.1017/ATSIP.2020.12.
[21] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA, Dec. 2015, pp. 2938–2946. doi: 10.1109/ICCV.2015.336.
[22] A. Valada, N. Radwan, and W. Burgard, “Deep Auxiliary Learning for Visual Localization and Odometry,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6939–6946. doi: 10.1109/ICRA.2018.8462979.
[23] I. Lujic, V. De Maio, and I. Brandic, “Efficient edge storage management based on near real-time forecasts,” in *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, 2017, pp. 21–30.
[24] J. Xing, H. Dai, and Z. Yu, “A distributed multi-level model with dynamic replacement for the storage of smart edge computing,” *J. Syst. Archit.*, vol. 83, pp. 1–11, 2018.
[25] Y. Huang, X. Song, F. Ye, Y. Yang, and X. Li, “Fair and efficient caching algorithms and strategies for peer data sharing in pervasive edge computing environments,” *IEEE Trans. Mob. Comput.*, vol. 19, no. 4, pp. 852–864, 2019.
[26] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, “Integrated blockchain and edge computing systems: A survey, some research issues and challenges,” *IEEE Commun. Surv. Tutor.*, vol. 21, no. 2, pp. 1508–1532, 2019.
[27] C.-C. Fang, C. Cheng, Z. Tang, and C. Li, “Research on Routing Algorithm Based on Reinforcement Learning in SDN,” *J. Phys. Conf. Ser.*, 2019.
[28] M. K. Awad, M. H. H. Ahmed, A. F. Almutairi, and I. Ahmad, “Machine Learning-Based Multipath Routing for Software Defined Networks,” *J. Netw. Syst. Manag.*, vol. 29, pp. 1–30, 2021.
[29] J. Rischke, P. Sossalla, H. Salah, F. H. P. Fitzek, and M. Reisslein, “QR-SDN: Towards Reinforcement Learning States, Actions, and Rewards for Direct Flow Routing in Software-Defined Networks,” *IEEE Access*, vol. 8, pp. 174773–174791, 2020, doi: 10.1109/ACCESS.2020.3025432.
[30] P. Almasan, J. Suárez-Varela, A. Badia-Sampera, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio, “Deep Reinforcement Learning meets Graph Neural Networks: exploring a routing optimization use case.” 2020.
[31] ETSI, “Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Network Service Templates Specification (ETSI GS NFV-IFA 014 V4.2.1).” ETSI, 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/014/04.02.01_60/gs_NFV-IFA014v040201p.pdf
[32] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, “Edge computing: A survey,” *Future Gener. Comput. Syst.*, vol. 97, pp. 219–235, Aug. 2019, doi:

10.1016/j.future.2019.02.050.
[33] W. Yu *et al.*, “A Survey on the Edge Computing for the Internet of Things,” *IEEE Access*, vol. 6, pp. 6900–6919, 2018, doi: 10.1109/ACCESS.2017.2778504.
[34] X. Liu, J. Yu, J. Wang, and Y. Gao, “Resource Allocation With Edge Computing in IoT Networks via Machine Learning,” *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3415–3426, Apr. 2020, doi: 10.1109/JIOT.2020.2970110.
[35] F. Nisar and B. Ahmed, “Resource Utilization in Data Center by Applying ARIMA Approach,” in *Intelligent Technologies and Applications*, Singapore, 2020. doi: 10.1007/978-981-15-5232-8_64.
[36] B. Shiva Prakash, K. V. Sanjeev, R. Prakash, and K. Chandrasekaran, “A Survey on Recurrent Neural Network Architectures for Sequential Learning,” in *Soft Computing for Problem Solving*, Singapore, 2019, pp. 57–66. doi: 10.1007/978-981-13-1595-4_5.
[37] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu, “Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions,” *Procedia Comput. Sci.*, vol. 131, pp. 895–903, Jan. 2018, doi: 10.1016/j.procs.2018.04.298.
[38] M. Masdari and A. Khoshnevis, “A survey and classification of the workload forecasting methods in cloud computing,” *Clust. Comput.*, vol. 23, no. 4, pp. 2399–2424, Dec. 2020, doi: 10.1007/s10586-019-03010-3.
[39] S. Seabold and J. Perktold, “Statsmodels: Econometric and Statistical Modeling with Python,” *Proc. 9th Python Sci. Conf.*, vol. 2010, Jan. 2010.
[40] T. L. Duc, R. G. Leiva, P. Casari, and P.-O. Östberg, “Machine Learning Methods for Reliable Resource Provisioning in Edge-Cloud Computing: A Survey,” *ACM Comput. Surv.*, vol. 52, no. 5, p. 94:1-94:39, Sep. 2019, doi: 10.1145/3341145.
[41] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 785–794, Aug. 2016, doi: 10.1145/2939672.2939785.
[42] D. Nielsen, “Tree Boosting With XGBoost - Why Does XGBoost Win ‘Every’ Machine Learning Competition?,” 2016, Accessed: Feb. 24, 2021. [Online]. Available: https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2433761
[43] J. Violos, E. Psomakelis, K. Tserpes, F. Aisopos, and T. Varvarigou, “Leveraging User Mobility and Mobile App Services Behavior for Optimal Edge Resource Utilization,” in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, New York, NY, USA, May 2019, pp. 7–12. doi: 10.1145/3312614.3312620.
[44] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, “Auto-sklearn: Efficient and Robust Automated Machine Learning,” in *Automated Machine Learning: Methods, Systems, Challenges*, Cham: Springer International Publishing, 2019. doi: 10.1007/978-3-030-05318-5_6.
[45] J. Violos, E. Psomakelis, D. Danopoulos, S. Tsanakas, and T. Varvarigou, “Using LSTM Neural Networks as Resource Utilization Predictors: The Case of Training Deep Learning Models on the Edge,” in *Economics of Grids, Clouds, Systems, and Services*, Cham, 2020, pp. 67–74. doi: 10.1007/978-3-030-63058-4_6.
[46] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” *ArXiv14123555 Cs*, Dec. 2014, Accessed: Jan. 20, 2021. [Online]. Available: http://arxiv.org/abs/1412.3555
[47] N. Hansen, D. Arnold, and A. Auger, *Evolution Strategies*. 2015. doi: 10.1007/978-3-662-43505-2_44.
[48] P. I. Frazier, “A Tutorial on Bayesian Optimization,” *ArXiv180702811 Cs Math Stat*, Jul. 2018, Accessed: Feb. 24, 2021. [Online]. Available: http://arxiv.org/abs/1807.02811
[49] G. Rodola, “giampaolo/psutil, https://github.com/giampaolo/psutil.” Jun. 2020. Accessed: Jun. 05, 2020. [Online]. Available: https://github.com/giampaolo/psutil
[50] A. K. Mortensen, “anderskm/gputil, https://github.com/anderskm/gputil.” Jun. 2020.

Accessed: Jun. 05, 2020. [Online]. Available: https://github.com/anderskm/gputil
[51] R. G. Garroppo, S. Giordano, M. Pagano, and G. Procissi, “On traffic prediction for resource allocation: A Chebyshev bound based allocation scheme,” *Comput. Commun.*, vol. 31, no. 16, pp. 3741–3751, Oct. 2008, doi: 10.1016/j.comcom.2008.05.019.
[52] E. Kim, K. Lee, and C. Yoo, “On the Resource Management of Kubernetes,” in *2021 International Conference on Information Networking (ICOIN)*, Jan. 2021, pp. 154–158. doi: 10.1109/ICOIN50884.2021.9333977.
[53] D. R. Figueiredo, B. Liu, V. Misra, and D. Towsley, “On the autocorrelation structure of TCP traffic,” *Comput. Netw.*, vol. 40, no. 3, pp. 339–361, Oct. 2002, doi: 10.1016/S1389-1286(02)00299-2.
[54] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu, and W. Zhang, “Interactive Temporal Recurrent Convolution Network for Traffic Prediction in Data Centers,” *IEEE Access*, vol. 6, pp. 5276–5289, 2018, doi: 10.1109/ACCESS.2017.2787696.
[55] M. Adel Serhani, H. T. El-Kassabi, K. Shuaib, A. N. Navaz, B. Benatallah, and A. Beheshti, “Self-adapting cloud services orchestration for fulfilling intensive sensory data-driven IoT workflows,” *Future Gener. Comput. Syst.*, vol. 108, pp. 583–597, Jul. 2020, doi: 10.1016/j.future.2020.02.066.
[56] V. Paxson and S. Floyd, “Wide area traffic: the failure of Poisson modeling,” *IEEEACM Trans. Netw.*, vol. 3, no. 3, pp. 226–244, Jun. 1995, doi: 10.1109/90.392383.
[57] V. Eramo, T. Catena, F. G. Lavacca, and F. di Giorgio, “Study and Investigation of SARIMA-based Traffic Prediction Models for the Resource Allocation in NFV networks with Elastic Optical Interconnection,” in *2020 22nd International Conference on Transparent Optical Networks (ICTON)*, Jul. 2020, pp. 1–4. doi: 10.1109/ICTON51198.2020.9203070.
[58] P. Sekwatlakwatla, M. Mphahlele, and T. Zuva, “Traffic flow prediction in cloud computing,” in *2016 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Nov. 2016, pp. 123–128. doi: 10.1109/ICACCE.2016.8073735.
[59] F. Pilka and M. Oravec, “Multi-step ahead prediction using neural networks,” in *Proceedings ELMAR-2011*, Sep. 2011, pp. 269–272.
[60] A. R. Abdellah, O. A. K. Mahmood, A. Paramonov, and A. Koucheryavy, “IoT traffic prediction using multi-step ahead prediction with neural network,” in *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Oct. 2019, pp. 1–4. doi: 10.1109/ICUMT48472.2019.8970675.
[61] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, “Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 1672–1678. doi: 10.1109/IVS.2018.8500658.
[62] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, “Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies,” *Transp. Res. Part C Emerg. Technol.*, vol. 105, pp. 297–322, Aug. 2019, doi: 10.1016/j.trc.2019.05.039.
[63] X. Wang, X. Guan, J. Cao, N. Zhang, and H. Wu, “Forecast network-wide traffic states for multiple steps ahead: A deep learning approach considering dynamic non-local spatial correlation and non-stationary temporal dependency,” *Transp. Res. Part C Emerg. Technol.*, vol. 119, p. 102763, Oct. 2020, doi: 10.1016/j.trc.2020.102763.
[64] Z. Wang, X. Su, and Z. Ding, “Long-Term Traffic Prediction Based on LSTM Encoder-Decoder Architecture,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–11, 2020, doi: 10.1109/TITS.2020.2995546.
[65] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, “The Characteristics of Cloud

Computing," in *2010 39th International Conference on Parallel Processing Workshops*, Sep. 2010, pp. 275–279. doi: 10.1109/ICPPW.2010.45.
[66] A. Nadjaran Toosi, J. Son, Q. Chi, and R. Buyya, "ElasticSFC: Auto-scaling techniques for elastic service function chaining in network functions virtualization-based clouds," *J. Syst. Softw.*, vol. 152, pp. 108–119, Jun. 2019, doi: 10.1016/j.jss.2019.02.052.

Παράρτημα Κώδικα

LSTM

```
#!/usr/bin/env python
# coding: utf-8
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from numpy import zeros, newaxis
import keras
import tensorflow as tf
get_ipython().run_line_magic('matplotlib', 'inline')
plt.rcParams.update({'figure.figsize':(7,5), 'figure.dpi':100})
get_ipython().system('pip install seaborn')
import seaborn as sns
from math import sqrt
from numpy import split
from numpy import array
from pandas import read_csv
from sklearn.metrics import mean_squared_error
from matplotlib import pyplot
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers import LSTM
from keras.layers import TimeDistributed
from keras.layers import RepeatVector
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from tensorflow_addons.layers import MultiHeadAttention
from keras.layers import Bidirectional
from keras.optimizers import Adam
from sklearn.metrics import mean_absolute_error
import seaborn as sns
from keras.layers import Dense, Dropout, Flatten
import time




df=pd.read_csv("expanded-lbl.csv",sep=",",header=None)




for x in range(40):
    df.at[x,0]=0
    df.at[x,1]=0




arr = [0 for j in range(8420)]
cnt=0
```

```
j=0



for x in range(0,336760):
    if((x)%40==0):
        arr[j]=cnt
        cnt=0
        j=j+1

    cnt=cnt+df.iloc[x,5]



def split_dataset(data):
    train, test = data[0:5892], data[5892:8418]

    return train, test



# create sequences
def split_sequence(seq, steps, out):
    X, Y = list(), list()
    for i in range(len(seq)):
        end = i + steps
        outi = end + out
        if outi > len(seq)-1:
            break
        seqx, seqy = seq[i:end], seq[end:outi]
        X.append(seqx)
        Y.append(seqy)
    return np.array(X), np.array(Y)



train, test = split_dataset(arr)

# number of time steps
steps = 3
out = 5
features=1

# split into samples
X_train, Y_train = split_sequence(train, steps, out)
X_test, Y_test = split_sequence(test, steps, out)
X_train = X_train.reshape((Y_train.shape[0], X_train.shape[1], features))
```

```
# define model
model = Sequential()
model.add(LSTM(180, activation='relu', input_shape=(steps, features)))
model.add(Dense(1))
adam = Adam(lr=0.0001)
model.compile(optimizer=adam, loss='mse')




# fit model
model.fit(X_train, Y_train, epochs=66, verbose=0)




preds=[]

for i in range(2518):
    tmp = np.array(X_test[i])
    tmp = prox.reshape((1, 3, n_features))

    yhat = model.predict(tmp, verbose=0)
    preds.append(yhat)




preds=np.array(preds)
preds=preds.reshape(2518,5)
Y_test=np.array(y_test)
Y_test.reshape(2518,5)




def column(matrix, i):
    return [row[i] for row in matrix]

a_preds=column(preds,4)
a_test=column(y_test,4)




rms = mean_squared_error(a_preds, a_test, squared=False)
print(rms)
```

```
mea=mean_absolute_error(a_preds, a_test)
print(mea)
```

LSTM_ED:

```
#!/usr/bin/env python
# coding: utf-8



import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from numpy import zeros, newaxis
import keras
import tensorflow as tf
get_ipython().run_line_magic('matplotlib', 'inline')
plt.rcParams.update({'figure.figsize':(7,5), 'figure.dpi':100})
get_ipython().system('pip install seaborn')
import seaborn as sns
from math import sqrt
from numpy import split
from numpy import array
from pandas import read_csv
from sklearn.metrics import mean_squared_error
from matplotlib import pyplot
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers import LSTM
from keras.layers import TimeDistributed
from keras.layers import RepeatVector
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from tensorflow_addons.layers import MultiHeadAttention
from keras.layers import Bidirectional
from keras.optimizers import Adam
from sklearn.metrics import mean_absolute_error
import seaborn as sns
from keras.layers import Dense, Dropout, Flatten
import time



df=pd.read_csv("expanded-lbl.csv",sep=",",header=None)
```

```
for x in range(40):
    df.at[x,0]=0
    df.at[x,1]=0




arr = [0 for j in range(8420)]
cnt=0
j=0




for x in range(0,336760):
    if((x)%40==0):
        arr[j]=cnt
        cnt=0
        j=j+1

    cnt=cnt+df.iloc[x,5]




def split_dataset(data):
    train, test = data[0:5892], data[5892:8418]

    return train, test




# create sequences
def split_sequence(seq, steps, out):
    X, Y = list(), list()
    for i in range(len(seq)):
        end = i + steps
        outi = end + out
        if outi > len(seq)-1:
            break
        seqx, seqy = seq[i:end], seq[end:outi]
        X.append(seqx)
        Y.append(seqy)
    return np.array(X), np.array(Y)
```

```
train, test = split_dataset(arr)

# number of time steps
steps = 3
out = 5
features=1

# split into samples
X_train, Y_train = split_sequence(train, steps, out)
X_test, Y_test = split_sequence(test, steps, out)
X_train = X_train.reshape((Y_train.shape[0], X_train.shape[1], features))




# define model
model = Sequential()
model.add(LSTM(180, activation='relu', input_shape=(steps, features)))
model.add(Dense(1))
adam = Adam(lr=0.0001)
model.compile(optimizer=adam, loss='mse')




# fit model
model.fit(X_train, Y_train, epochs=66, verbose=0)




preds=[]

for i in range(2518):
    tmp = np.array(X_test[i])
    tmp = prox.reshape((1, 3, n_features))

    yhat = model.predict(tmp, verbose=0)
    preds.append(yhat)




preds=np.array(preds)
preds=preds.reshape(2518,5)
Y_test=np.array(y_test)
Y_test.reshape(2518,5)
```

```
def column(matrix, i):
    return [row[i] for row in matrix]

a_preds=column(preds,4)
a_test=column(y_test,4)



rms = mean_squared_error(a_preds, a_test, squared=False)
print(rms)
mea=mean_absolute_error(a_preds, a_test)
print(mea)
```

BIDIRECTIONAL LSTM_ED:

```
#!/usr/bin/env python
# coding: utf-8



import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from numpy import zeros, newaxis
import keras
import tensorflow as tf
get_ipython().run_line_magic('matplotlib', 'inline')
plt.rcParams.update({'figure.figsize':(7,5), 'figure.dpi':100})
get_ipython().system('pip install seaborn')
import seaborn as sns
from math import sqrt
from numpy import split
from numpy import array
from pandas import read_csv
from sklearn.metrics import mean_squared_error
from matplotlib import pyplot
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers import LSTM
from keras.layers import TimeDistributed
from keras.layers import RepeatVector
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from tensorflow_addons.layers import MultiHeadAttention
from keras.layers import Bidirectional
from keras.optimizers import Adam
from sklearn.metrics import mean_absolute_error
```

```
import seaborn as sns
from keras.layers import Dense, Dropout, Flatten
import time



#load dataset
df=pd.read_csv("expanded-lbl.csv",sep=",",header=None)



for x in range(40):
    df.at[x,0]=0
    df.at[x,1]=0



arr = [0 for j in range(8420)]
cnt=0
j=0



#create sums for selected inputs / columns (5: traffic , 7: number of requests) - each sum
corresponds to one time-step
for x in range(0,336760):
    if((x)%40==0):
        arr[j]=cnt
        cnt=0
        j=j+1

    cnt=cnt+df.iloc[x,5]



# split dataset for train - test
def split_dataset(data):
    train, test = data[0:5892], data[5892:8418]

    return train, test



# create sequences
def split_sequence(seq, steps, out):
```

```
    X, Y = list(), list()
    for i in range(len(seq)):
        end = i + steps
        outi = end + out
        if outi > len(seq)-1:
            break
        seqx, seqy = seq[i:end], seq[end:outi]
        X.append(seqx)
        Y.append(seqy)
    return np.array(X), np.array(Y)



train, test = split_dataset(arr)

# number of time steps -input
steps = 3
# number of time steps -output
out = 5
features=1

# split into samples
X_train, Y_train = split_sequence(train, steps, out)
X_test, Y_test = split_sequence(test, steps, out)
X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], features))
Y_train = Y_train.reshape((Y_train.shape[0], Y_train.shape[1], 1))



model = Sequential()
model.add(Bidirectional(LSTM(180, activation='relu', input_shape=(steps, features))))
model.add(RepeatVector(out))
model.add(Bidirectional(LSTM(180, activation='relu',return_sequences=True)))
model.add(TimeDistributed(Dense(64, activation='relu')))
model.add(TimeDistributed(Dense(1)))
adam = Adam(lr=0.001)
model.compile(loss='mse', optimizer=adam)



# fit model
model.fit(X_train, Y_train, epochs=66, verbose=0)



# produce predictions
preds=[]
```

```
for i in range(2518):
    tmp = np.array(X_test[i])
    tmp = prox.reshape((1, 3, n_features))

    yhat = model.predict(tmp, verbose=0)
    preds.append(yhat)



# resize predictions to be compliant with the format
preds=np.array(preds)
preds=preds.reshape(2518,5)
Y_test=np.array(Y_test)
Y_test.reshape(2518,5)



# produce time-step specific predictions
def column(matrix, i):
    return [row[i] for row in matrix]
# set the specific time-step to be examined (0 : 4)
a_preds=column(preds,0)
a_test=column(y_test,0)



#print RMSE
rms = mean_squared_error(preds, Y_test, squared=False)
print(rms)
#print MAE
mea=mean_absolute_error(preds, Y_test)
print(mea)



#print time-step specific RMSE
rms = mean_squared_error(a_preds, a_test, squared=False)
print(rms)

#print time-step specific MAE
mea=mean_absolute_error(a_preds, a_test)
print(mea)
```

CNN-LSTM_ED:

```
#!/usr/bin/env python
```

```
# coding: utf-8

# In[56]:


import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from numpy import zeros, newaxis
import keras
import tensorflow as tf
get_ipython().run_line_magic('matplotlib', 'inline')
plt.rcParams.update({'figure.figsize':(7,5), 'figure.dpi':100})
get_ipython().system('pip install seaborn')
import seaborn as sns
from math import sqrt
from numpy import split
from numpy import array
from pandas import read_csv
from sklearn.metrics import mean_squared_error
from matplotlib import pyplot
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers import LSTM
from keras.layers import TimeDistributed
from keras.layers import RepeatVector
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from tensorflow_addons.layers import MultiHeadAttention
from keras.layers import Bidirectional
from keras.optimizers import Adam
from sklearn.metrics import mean_absolute_error
import seaborn as sns
from keras.layers import Dense, Dropout, Flatten
import time




#load dataset
df=pd.read_csv("expanded-lbl.csv",sep=",",header=None)




for x in range(40):
    df.at[x,0]=0
    df.at[x,1]=0
```

```
# In[59]:


arr = [0 for j in range(8420)]
cnt=0
j=0




#create sums for selected inputs / columns (5: traffic , 7: number of requests) - each sum
corresponds to one time-step
for x in range(0,336760):
    if((x)%40==0):
        arr[j]=cnt
        cnt=0
        j=j+1

    cnt=cnt+df.iloc[x,5]




# split dataset for train - test
def split_dataset(data):
    train, test = data[0:5892], data[5892:8418]

    return train, test




# create sequences
def split_sequence(seq, steps, out):
    X, Y = list(), list()
    for i in range(len(seq)):
        end = i + steps
        outi = end + out
        if outi > len(seq)-1:
            break
        seqx, seqy = seq[i:end], seq[end:outi]
        X.append(seqx)
        Y.append(seqy)
    return np.array(X), np.array(Y)




train, test = split_dataset(arr)

# number of time steps -input
```

```
steps = 3
# number of time steps -output
out = 5
features=1

# split into samples
X_train, Y_train = split_sequence(train, steps, out)
X_test, Y_test = split_sequence(test, steps, out)
X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], features))
Y_train = Y_train.reshape((Y_train.shape[0], Y_train.shape[1], 1))




# define model
model = Sequential()
model.add(Conv1D(filters=32, kernel_size=1, activation='relu',
input_shape=(steps,features)))
model.add(Conv1D(filters=32, kernel_size=2, activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(RepeatVector(out))
model.add(LSTM(200, activation='relu', return_sequences=True))
model.add(TimeDistributed(Dense(100, activation='relu')))
model.add(TimeDistributed(Dense(1)))
model.compile(loss='mse', optimizer='adam')




# fit model
model.fit(X_train, Y_train, epochs=50, verbose=0)




# produce predictions
preds=[]

for i in range(2518):
    tmp = np.array(X_test[i])
    tmp = prox.reshape((1, 3, n_features))

    yhat = model.predict(tmp, verbose=0)
    preds.append(yhat)




# resize predictions to be compliant with the format
preds=np.array(preds)
```

```
preds=preds.reshape(2518,5)
Y_test=np.array(Y_test)
Y_test.reshape(2518,5)



# produce time-step specific predictions
def column(matrix, i):
    return [row[i] for row in matrix]
# set the specific time-step to be examined (0 : 4)
a_preds=column(preds,0)
a_test=column(y_test,0)



#print RMSE
rms = mean_squared_error(preds, Y_test, squared=False)
print(rms)
#print MAE
mea=mean_absolute_error(preds, Y_test)
print(mea)



#print time-step specific RMSE
rms = mean_squared_error(a_preds, a_test, squared=False)
print(rms)

#print time-step specific MAE
mea=mean_absolute_error(a_preds, a_test)
print(mea)
```

HYBRID_ED:

```
#!/usr/bin/env python
# coding: utf-8



import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from numpy import zeros, newaxis
import keras
import tensorflow as tf
get_ipython().run_line_magic('matplotlib', 'inline')
plt.rcParams.update({'figure.figsize':(7,5), 'figure.dpi':100})
get_ipython().system('pip install seaborn')
```

```
import seaborn as sns
from math import sqrt
from numpy import split
from numpy import array
from pandas import read_csv
from sklearn.metrics import mean_squared_error
from matplotlib import pyplot
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers import LSTM
from keras.layers import TimeDistributed
from keras.layers import RepeatVector
from keras.layers.convolutional import Conv1D
from keras.layers.convolutional import MaxPooling1D
from tensorflow_addons.layers import MultiHeadAttention
from keras.layers import Bidirectional
from keras.optimizers import Adam
from sklearn.metrics import mean_absolute_error
import seaborn as sns
from keras.layers import Dense, Dropout, Flatten
import time



#load dataset
df=pd.read_csv("expanded-lbl.csv",sep=",",header=None)



#create sums for selected inputs / columns (5: traffic , 7: number of requests) - each sum
corresponds to one time-step
for x in range(0,336760):
    if((x)%40==0):
        arr[j]=cnt
        cnt=0
        j=j+1

    cnt=cnt+df.iloc[x,5]



# split dataset for train - test
def split_dataset(data):
    train, test = data[0:5892], data[5892:8418]

    return train, test
```

```
# create sequences
def split_sequence(seq, steps, out):
    X, Y = list(), list()
    for i in range(len(seq)):
        end = i + steps
        outi = end + out
        if outi > len(seq)-1:
            break
        seqx, seqy = seq[i:end], seq[end:outi]
        X.append(seqx)
        Y.append(seqy)
    return np.array(X), np.array(Y)



train, test = split_dataset(arr)

# number of time steps -input
steps = 3
# number of time steps -output
out = 5
features=1

# split into samples
X_train, Y_train = split_sequence(train, steps, out)
X_test, Y_test = split_sequence(test, steps, out)
X_train = X_train.reshape((Y_train.shape[0], X_train.shape[1], features))
Y_train = Y_train.reshape((Y_train.shape[0], Y_train.shape[1], 1))



model = Sequential()
model.add(Bidirectional(LSTM(256, activation='relu', input_shape=(n_steps,
n_features),return_sequences=True)))
model.add(LSTM(128, activation='relu'))
model.add(RepeatVector(n_out))
model.add(LSTM(256, activation='relu', return_sequences=True))
model.add(Bidirectional(LSTM(128, activation='relu',return_sequences=True)))
model.add(TimeDistributed(Dense(64, activation='relu')))
model.add(TimeDistributed(Dense(1)))
adam = Adam(lr=0.001)
model.compile(loss='mse', optimizer=adam)
```

```
# fit model
model.fit(X_train, Y_train, epochs=66, verbose=0)




for x in range(40):
    df.at[x,0]=0
    df.at[x,1]=0




arr = [0 for j in range(8420)]
cnt=0
j=0




# produce predictions
preds=[]

for i in range(2518):
    tmp = np.array(X_test[i])
    tmp = prox.reshape((1, 3, n_features))

    yhat = model.predict(tmp, verbose=0)
    preds.append(yhat)




# resize predictions to be compliant with the format
preds=np.array(preds)
preds=preds.reshape(2518,5)
Y_test=np.array(Y_test)
Y_test.reshape(2518,5)




# produce time-step specific predictions
def column(matrix, i):
    return [row[i] for row in matrix]
# set the specific time-step to be examined (0 : 4)
a_preds=column(preds,0)
a_test=column(y_test,0)
```

```
#print RMSE
rms = mean_squared_error(preds, Y_test, squared=False)
print(rms)
#print MAE
mea=mean_absolute_error(preds, Y_test)
print(mea)


#print time-step specific RMSE
rms = mean_squared_error(a_preds, a_test, squared=False)
print(rms)

#print time-step specific MAE
mea=mean_absolute_error(a_preds, a_test)
print(mea)
```