



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
Εθνικόν και Καποδιστριακόν  
Πανεπιστήμιον Αθηνών  
— ΙΔΡΥΘΕΝ ΤΟ 1837 —

**Τμήμα Φυσικής**  
**Μεταπτυχιακό Δίπλωμα Ειδίκευσης**  
**Ηλεκτρονικού Αυτοματισμού**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Υλοποίηση Ροών Μηχανικής Μάθησης για την Πρόβλεψη**  
**Κίνησης σε Δίκτυα 5ης Γενιάς**

Ευάγγελος Πετρόπουλος

ΑΜ:7110132100213

**Επιβλέπουσα:** Τζανακάκη Άννα, Αναπληρώτρια Καθηγήτρια

**ΑΘΗΝΑ**

**ΙΟΥΛΙΟΣ 2023**

## Περίληψη

Κύριος στόχος της συγκεκριμένης μεταπτυχιακής εργασίας είναι η μελέτη, υλοποίηση και βελτιστοποίηση μοντέλων μηχανικής μάθησης σε δίκτυα κινητών επικοινωνιών 5<sup>ης</sup> γενεάς (5G), ώστε να επιτευχθεί η βέλτιστη διαχείριση των υπολογιστικών και δικτυακών πόρων των σχετικών υποδομών. Η βέλτιστη αυτή διαχείριση πόρων στηρίζεται στην παρατήρηση χρήσης τους και την εφαρμογή τεχνικών μηχανικής μάθησης προκειμένου να προβλεφθούν οι μελλοντικές ανάγκες της υποδομής σε αυτούς τους πόρους. Σε αυτό το πλαίσιο αρχικά γίνεται μια σύντομη αναφορά στην αρχιτεκτονική και τα πλεονεκτήματα ενός δικτύου κινητών επικοινωνιών 5G, ενώ περιγράφονται και τα συστήματα παρακολούθησης/επιτήρησης των δικτύων 5G. Στη συνέχεια παρατίθεται μια εισαγωγή του πεδίου της μηχανικής μάθησης καθώς και της γλώσσας προγραμματισμού *python* που αξιοποιείται για την εφαρμογή μοντέλων μηχανικής μάθησης. Η εργασία συμπεριλαμβάνει επίσης την παρουσίαση ενός σύντομου και παραστατικού οδηγού για την υλοποίηση ενός ενεργού δικτύου κινητών επικοινωνιών 5G που επιτηρείται από ένα σύστημα παρακολούθησης δεδομένων. Πιο συγκεκριμένα το δίκτυο αυτό παρακολουθείται ως προς την κατανάλωση υπολογιστικών και δικτυακών πόρων μέσω των μετρικών (*metrics*) *node\_cpu\_seconds\_total* και *node\_network\_receive\_bytes\_total*, οι οποίες αντιστοιχούν στο συνολικό χρόνο που καταναλώνει ο επεξεργαστής (CPU) για *interrupts* λόγω του Δικτύου 5G και στα συνολικά *bytes* που λαμβάνονται στο UPF στοιχείο του Δικτύου 5G από τους σταθμούς βάσης (gNB). Οι μετρικές αυτές συλλέγονται με τη μορφή χρονοσειρών και αποθηκεύονται σε μια βάση δεδομένων. Η ανάκτηση των χρονοσειρών των δύο μετρικών από την βάση δεδομένων, πραγματοποιείται με κώδικα που υλοποιήθηκε σε γλώσσα προγραμματισμού *python*. Η ίδια γλώσσα προγραμματισμού χρησιμοποιείται και για την υλοποίηση και βελτιστοποίηση ως προς την ακρίβεια πρόβλεψης των μοντέλων μηχανικής μάθησης Gated Recurrent Unit Network (GRU), Long Short-Term Memory Network (LSTM), Convolutional Neural Network (CNN) και Multi-layer Perceptron (MLP), τα οποία προβλέπουν την εξέλιξη των δύο μετρικών στο χρόνο. Επίσης η γλώσσα προγραμματισμού *python* αξιοποιείται και για την ανάπτυξη μιας γραφικής διεπαφής χρήστη που επιτρέπει την εύκολη ανάκτηση των δεδομένων χρονοσειρών από την βάση δεδομένων, την βελτιστοποίηση των μοντέλων καθώς και την εικονικοποίηση των αποτελεσμάτων. Τέλος παρουσιάζεται η ερευνητική διαδικασία βελτιστοποίησης των μοντέλων μηχανικής μάθησης GRU, LSTM, CNN και MLP ώστε αυτά να προβλέπουν με την μεγαλύτερη δυνατή ακρίβεια, την εξέλιξη στον χρόνο των *node\_cpu\_seconds\_total* και *node\_network\_receive\_bytes\_total* μετρικών. Πιο συγκεκριμένα σε κάθε χρονοσειρά μιας μετρικής εφαρμόζεται ένα μοντέλο μηχανικής μάθησης, για το οποίο μέσω μιας διαδικασίας βελτιστοποίησης βρίσκονται οι βέλτιστες παράμετροι (υπερπαραμέτροι) του μοντέλου, ώστε αυτό να προβλέπει με όσο το δυνατόν μεγαλύτερη ακρίβεια την εξέλιξη της χρονοσειράς στον χρόνο. Η διαδικασία αυτή επαναλαμβάνεται για τα τέσσερα μοντέλα και τις δύο μετρικές που αναφέρθηκαν παραπάνω. Τα αποτελέσματα της παρούσας μελέτης οδηγούν στην εύρεση του βέλτιστου μοντέλου μηχανικής μάθησης, το οποίο είναι ικανό να προβλέπει με μεγάλη ακρίβεια την εξέλιξη των χρονοσειρών των δύο προαναφερθέντων μετρικών, που αντιπροσωπεύουν την δικτυακή κίνηση του συστήματος 5G. Χρησιμοποιώντας ως κριτήριο αξιολόγησης την τιμή της *loss* συνάρτησης, βέλτιστο μοντέλο μηχανικής μάθησης για την μετρική *node\_cpu\_seconds\_total* αναδείχθηκε το MLP ενώ για την μετρική *node\_network\_receive\_bytes\_total* αναδείχθηκε το GRU.

## **Λέξεις κλειδιά**

Δίκτυο 5<sup>ης</sup> γενεάς, 5G Core Network, New Generation Radio Access Network, Prometheus, free5GC, UERANSIM, Μηχανική Μάθηση, Βαθιά Μάθηση, Τεχνητά Νευρωνικά Δίκτυα, Python, Anvil

## Summary

The primary objective of this master thesis is to study, implement, and optimize machine learning models in 5th generation (5G) mobile communication networks in order to achieve optimal management of computational and network resources in relevant infrastructures. This resource management is based on observing the resources' usage and applying machine learning techniques to predict the future needs of the infrastructure for these resources. In this context, the thesis begins with a brief introduction to the architecture and advantages of a 5G mobile communication network, as well as the monitoring/surveillance systems for 5G networks. Then, the field of machine learning and the Python programming language, which is used for implementing machine learning models, are introduced. Moreover, this thesis includes a presentation of a concise and illustrative guide for implementing an active 5G mobile communication network monitored by a data monitoring system. This network is monitored in terms of the consumption of computational and network resources using the *node\_cpu\_seconds\_total* and *node\_network\_receive\_bytes\_total* metrics. These metrics represent the total time the CPU consumes for interrupts due to the 5G Network and the total bytes received at the 5G Network's UPF (User Plane Function) element from base stations (gNB), respectively. The data from these metrics are collected as time series and stored in a database. Python programming language is used to retrieve the time series data from the database, implement and optimize the accuracy of machine learning models Gated Recurrent Unit Network (GRU), Long Short-Term Memory Network (LSTM), Convolutional Neural Network (CNN), and Multi-layer Perceptron (MLP). These models predict the evolution of the two metrics over time. Also, python programming language is used to develop a graphic user interface that facilitates easy data retrieval from the database, machine learning model optimization and visualization of the results. Finally, the research process for optimizing the GRU, LSTM, CNN, and MLP machine learning models to predict the evolution of the *node\_cpu\_seconds\_total* and *node\_network\_receive\_bytes\_total* metrics with the highest possible accuracy, is presented. The results of the present study lead to the identification of the optimal machine learning model, capable of accurately predicting the evolution of these metrics, which represent the network traffic of the 5G system. Using the loss function evaluation criterion, MLP emerges as the optimal machine learning model for the *node\_cpu\_seconds\_total* metric, while the GRU emerges as the optimal model for the *node\_network\_receive\_bytes\_total* metric.

## Keywords

5th Generation Network, 5G Core Network, New Generation Radio Access Network, Prometheus, free5GC, UERANSIM, Machine Learning, Deep Learning, Artificial Neural Network, Python, Anvil

*Αφιερωμένο στην αγαπημένη μου γιαγιά*

## Ευχαριστίες

Κατ' αρχάς επιθυμώ να ευχαριστήσω τους επιβλέποντες καθηγητές αυτής της πτυχιακής εργασίας κ. Άννα Τζανακάκη και κ. Μάρκο Αναστασόπουλο, οι οποίοι μέσω της ανάθεσης του συγκεκριμένου θέματος μου έδωσαν την δυνατότητα ώστε να εντρυφήσω σε ένα τόσο επίκαιρο και ενδιαφέρον αντικείμενο.

Ακόμη, ευχαριστώ θερμά τον υποψήφιο διδάκτορα κ. Αλέξανδρο Μανωλόπουλο, διότι καθ' όλη την διάρκεια της μεταπτυχιακής εργασίας με βοήθησε σε ό,τι και αν χρειάστηκα, παρόλο τον φόρτο εργασίας του.

Θα ήθελα επίσης να ευχαριστήσω ιδιαιτέρως τον μέντορά μου, όπως τον αποκαλώ, και διδάκτορα του τμήματος Φυσικής του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών κ. Χαρίλη Τσαγκαράκη ο οποίος με εισήγαγε στον κόσμο της επιστήμης που ονομάζεται Φυσική και με έκανε να τον αγαπήσω, ενώ ταυτόχρονα με δίδαξε την αξία της επιμονής και της υπομονής.

Επιπλέον, θέλω να απευθύνω ένα πολύ μεγάλο ευχαριστώ προς τους φίλους μου και την κοπέλα μου Δανάη, οι οποίοι είναι πάντα δίπλα μου και πρόθυμοι να ακούσουν τα προβλήματά μου, να με συμβουλευσουν και να με βοηθήσουν στις χαρές και στις λύπες.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη στην γιαγιά μου Γεωργία, την μητέρα μου Χριστίνα, τον αδερφό μου Τζώρτζη και τον πατριό μου Αλέξανδρο, διότι από πολύ μικρή ηλικία βρίσκονται στο πλευρό μου, με στηρίζουν σε όλα τα θέματα και έχουν διαδραματίσει καίριο ρόλο στη διαμόρφωση της προσωπικότητάς μου.

## Πίνακας Περιεχομένων

1 Εισαγωγή.....	11
2 Δίκτυα 5 <sup>ης</sup> Γενεάς (5G) .....	14
2.1 Εισαγωγή στην νέα τεχνολογία δικτύων κινητής τηλεπικοινωνίας 5G (5 <sup>ης</sup> γενεάς) .....	14
2.2 Ιστορική Αναδρομή Τεχνολογιών Κινητής Τηλεπικοινωνίας (ή Κυψελωτών Δικτύων) .....	14
2.3 Οι Βασικοί Παράγοντες για την Χρησιμοποίηση της Τεχνολογίας Δικτύων Κινητής Τηλεφωνίας 5G (5 <sup>ης</sup> γενεάς) Σύμφωνα με τις Απαιτήσεις της Εποχής .....	15
2.4 Τεχνολογίες που Συνέβαλαν στην Ανάπτυξη της Τεχνολογίας Δικτύων Κινητής Τηλεφωνίας 5G (5 <sup>ης</sup> Γενεάς) .....	16
2.5 Δίκτυο Πυρήνα 5ης Γενεάς .....	16
2.5.1 Αρχιτεκτονική του Δικτύου Πυρήνα 5ης Γενεάς.....	16
2.5.2 Βασικές Λειτουργίες της Αρχιτεκτονικής του Δικτύου Πυρήνα 5ης Γενεάς.....	17
2.6 Δίκτυο Ραδιοπρόσβασης NG-RAN .....	19
2.6.1 Αρχιτεκτονική του Δικτύου Ραδιοπρόσβασης NG-RAN .....	19
2.6.2 Λειτουργικά Χαρακτηριστικά της Αρχιτεκτονικής του Δικτύου Ραδιοπρόσβασης NG-RAN .....	20
3 Συστήματα Παρακολούθησης και Χρονοσειρές Μετρικών.....	23
3.1 Εισαγωγή στα Συστήματα Παρακολούθησης.....	23
3.2 Χρονοσειρές Μετρικών .....	23
3.2.1 Ορισμός Χρονοσειρών Μετρικών .....	23
3.2.2 Κατηγορίες Χρονοσειρών Μετρικών .....	24
3.2.3 Αποθήκευση Χρονοσειρών Μετρικών .....	25
3.2.4 Ανάλυση Χρονοσειρών Μετρικών .....	26
3.2.5 Τύποι Ανάλυσης Χρονοσειρών Μετρικών .....	26
3.3 Εισαγωγή στο Prometheus .....	27
3.3.1 Αποθήκευση Δεδομένων Χρονοσειρών Μετρικών στο Prometheus .....	27
4 Μηχανική Μάθηση με Υλοποίηση σε Γλώσσα Προγραμματισμού Python .....	30
4.1 Ιστορική Αναδρομή στο Πεδίο της Τεχνητής Νοημοσύνης .....	31
4.2 Μοντέλα Μηχανικής Μάθησης .....	31
4.3 Μοντέλα Τεχνητών Νευρωνικών Δικτύων.....	33
4.3.1 Νευρώνες Τεχνητών Νευρωνικών Δικτύων .....	34
4.3.2 Τύποι Τεχνητών Νευρωνικών Δικτύων .....	35
4.4 Μοντέλα Βαθιάς Μάθησης.....	36
4.5 Python .....	37

4.5.1 Pandas .....	38
4.5.2 Matplotlib .....	38
4.5.3 Plotly .....	38
4.5.4 Numpy .....	38
4.5.5 Datetime .....	38
4.5.6 Scikit-learn ή sklearn .....	39
4.5.6 Keras.....	39
4.5.7 Optuna .....	39
4.5.8 InfluxDB.....	39
4.5.9 Anvil .....	39
5 Ανάλυση και Αποτελέσματα Βελτιστοποίησης Μοντέλων Μηχανικής Μάθησης για την Πρόβλεψη Κίνησης Δικτύων 5 <sup>ης</sup> Γενεάς .....	40
5.1 Περιβάλλον Υλοποίησης και Βελτιστοποίησης Μοντέλων Μηχανικής Μάθησης για την Πρόβλεψη Κίνησης Δικτύων 5 <sup>ης</sup> Γενεάς .....	40
5.1.1 Προεγκατεστημένα Λογισμικά .....	40
5.1.2 Υλοποίηση Κώδικα.....	43
5.1.3 Υλοποίηση Γραφικής Διεπαφής Χρήστη Μέσω Anvil Framework.....	50
5.2 Βελτιστοποίηση Μοντέλων Μηχανικής Μάθησης.....	52
5.2.1 Περιγραφή Ερευνητικής Διαδικασίας Βελτιστοποίησης.....	52
5.2.2 Αποτελέσματα Ερευνητικής Διαδικασίας Βελτιστοποίησης .....	55
6 Επίλογος.....	61
Βιβλιογραφία .....	65



## Πίνακας Εικόνων

Εικόνα 1: : Αρχιτεκτονική ενός δικτύου 5ης Γενεάς .....	14
Εικόνα 2: Η service-based αρχιτεκτονική των λειτουργιών ενός 5G Core Network [1] .....	17
Εικόνα 3: Οι βασικές λειτουργίες ενός 5G Core Network και οι διασυνδέσεις τους με τις συσκευές και το New Generation Radio Access Network (3GPP Radio Network) [1] .....	19
Εικόνα 4: Η διασύνδεση των ραδιοσταθμών βάσης του New Generation Radio Access Network μεταξύ τους καθώς και των ραδιοσταθμών βάσης του New Generation Access Network με το 5G Core Network [1] .....	21
Εικόνα 5: Η διασύνδεση του 5G Core Network με τις συσκευές και το New Generation Radio Access Network (3GPP Radio Network) [1] .....	22
Εικόνα 6: Παράδειγμα χρονοσειράς που αναπαριστά το αριθμό των ετήσιων πωλήσεων στην Ηνωμένες Πολιτείες της Αμερικής [3] .....	24
Εικόνα 7: Η γενική δομή μετρήσεων χρονοσειρών στη βάση δεδομένων που συνδέεται το Prometheus [2] .....	28
Εικόνα 8: Αρχιτεκτονική Τεχνητών Νευρωνικών Δικτύων .....	34
Εικόνα 9: Αρχιτεκτονική Νευρώνα Τεχνητών Νευρωνικών Δικτύων .....	35
Εικόνα 10: Βιβλιοθήκες της γλώσσας προγραμματισμού Python που χρησιμοποιήθηκαν .....	43
Εικόνα 11: Συνάρτηση ανάκτησης των δύο μετρικών από το 5G δίκτυο .....	44
Εικόνα 12: Η αρχή της Objective συνάρτησης που ορίζονται οι υπερπαραμέτροι και οι δυνατές τους τιμές .....	45
Εικόνα 13: Η υπόλοιπη της Objective συνάρτησης .....	45
Εικόνα 14: Η συνάρτηση παραγωγής του μοντέλου με τις βέλτιστες υπερπαραμέτρους .....	46
Εικόνα 15: Συνάρτηση προεπεξεργασίας και πρόβλεψης εξέλιξης δεδομένων χρονοσειρών – παραγωγής διαγραμμάτων (1) .....	47
Εικόνα 16: Συνάρτηση προεπεξεργασίας και πρόβλεψης εξέλιξης δεδομένων χρονοσειρών – παραγωγής διαγραμμάτων (2) .....	48
Εικόνα 17: Συνάρτηση προεπεξεργασίας και πρόβλεψης εξέλιξης δεδομένων χρονοσειρών – παραγωγής διαγραμμάτων (3) .....	49
Εικόνα 18: Συνάρτηση προεπεξεργασίας και πρόβλεψης εξέλιξης δεδομένων χρονοσειρών – παραγωγής διαγραμμάτων (4) .....	49
Εικόνα 19: Γραφική Διεπαφή Χρήστη στο Anvil .....	50
Εικόνα 20: Κώδικας στον Anvil εξυπηρετητή για την σύνδεση γραφικών στοιχείων με τις συναρτήσεις 52	
Εικόνα 21: Επιλογή στην drop-down λίστες της γραφικής διεπαφής χρήστη για την ανάκτηση δεδομένων χρονοσειρών από την βάση δεδομένων των μετρικών node_cpu_seconds_total node_network_receive_bytes_total των τελευταίων 90 λεπτών .....	52
Εικόνα 22: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης LSTM της μετρικής node_cpu_seconds_total .....	53
Εικόνα 23: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης LSTM της μετρικής node_cpu_seconds_total .....	53
Εικόνα 24: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης MLP της μετρικής node_cpu_seconds_total .....	54
Εικόνα 25: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης CNN της μετρικής node_cpu_seconds_total .....	54

Εικόνα 26: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης GRU της μετρικής node_network_receive_bytes_total.....	54
Εικόνα 27: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης LSTM της μετρικής node_network_receive_bytes_total.....	54
Εικόνα 28: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης MLP της μετρικής node_network_receive_bytes_total.....	54
Εικόνα 29: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης CNN της μετρικής node_network_receive_bytes_total.....	55
Εικόνα 30: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίησης της ποιότητας του μοντέλου GRU για την μετρική node_cpu_seconds_total.....	55
Εικόνα 31: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίησης της ποιότητας του μοντέλου LSTM για την μετρική node_cpu_seconds_total.....	56
Εικόνα 32: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίησης της ποιότητας του μοντέλου MLP για την μετρική node_cpu_seconds_total.....	56
Εικόνα 33: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίησης της ποιότητας του μοντέλου CNN για την μετρική node_cpu_seconds_total.....	57
Εικόνα 34: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίησης της ποιότητας του μοντέλου GRU για την μετρική node_network_receive_bytes_total.....	57
Εικόνα 35: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίησης της ποιότητας του μοντέλου LSTM για την μετρική node_network_receive_bytes_total.....	58
Εικόνα 36: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίησης της ποιότητας του μοντέλου MLP για την μετρική node_network_receive_bytes_total.....	58
Εικόνα 37: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίησης της ποιότητας του μοντέλου CNN για την μετρική node_network_receive_bytes_total.....	59
Εικόνα 38: Οι βέλτιστες υπερπαραμέτροι καθώς και η τιμή της loss συνάρτησης για κάθε μοντέλο μηχανικής μάθησης που εφαρμόζεται στα δεδομένα της μετρικής node_cpu_seconds_total.....	59
Εικόνα 39: Οι βέλτιστες υπερπαραμέτροι καθώς και η τιμή της loss συνάρτησης για κάθε μοντέλο μηχανικής μάθησης που εφαρμόζεται στα δεδομένα της μετρικής node_network_receive_bytes_total.....	59

## 1 Εισαγωγή

Με την εισαγωγή και εφαρμογή της τεχνολογίας δικτύων κινητής τηλεπικοινωνίας 5G (5<sup>ης</sup> Γενεάς) ξεκίνησε μια νέα εποχή για τα δίκτυα κινητής επικοινωνίας, τα οποία πλέον λόγω των προηγμένων δυνατοτήτων που προσφέρουν στους χρήστες, καλούνται να διαχειριστούν πάρα πολλές συνδέσεις κινητών συσκευών, μεγάλο αριθμό δεδομένων καθώς και την συχνή απαίτηση πολλών εφαρμογών για μηδαμινές καθυστερήσεις. Ωστόσο, πέρα των απαιτήσεων αυτών, αυξάνεται συνεχώς η πολυπλοκότητα και η ετερογένεια των δικτύων κινητής τηλεπικοινωνίας 5G (5<sup>ης</sup> Γενεάς) καθώς υπάρχει προσπάθεια κεντροποιημένης διαχείρισης των δικτύων επικοινωνιών, με αποτέλεσμα η εξασφάλιση βέλτιστης απόδοσης και αποτελεσματικής διαχείρισης υπολογιστικών ή δικτυακών πόρων να είναι μια σημαντική πρόκληση [2]. Λύση σε αυτό το πρόβλημα τα τελευταία χρόνια προσπαθεί να δοθεί μέσω τεχνικών μηχανικής μάθησης (machine learning methods), οι οποίες αποτελούν ένα πολλά υποσχόμενο εργαλείο για την αντιμετώπιση των προκλήσεων αυτών καθώς και την βελτιστοποίηση διαφόρων πτυχών των δικτύων κινητής τηλεπικοινωνίας 5G (5<sup>ης</sup> Γενεάς), όπως η διαχείριση υπολογιστικών ή δικτυακών πόρων [28].

Η μηχανική μάθηση είναι ένα υποπεδίο της τεχνητής νοημοσύνης (Artificial), που ουσιαστικά έχει ως στόχο να ωθήσει τα υπολογιστικά συστήματα να μαθαίνουν από τα δεδομένα έτσι ώστε να μπορούν στην συνέχεια να λάβουν αποφάσεις ανάλογα τα ερεθίσματα που δέχονται χωρίς να είναι ρητώς προγραμματισμένα. Στις μέρες μας παρά τις μελέτες που έχουν πραγματοποιηθεί για την και αξιοποίηση τεχνικών μηχανικής μάθησης σε συνδυασμό με τις αρχιτεκτονικές δικτύων κινητής επικοινωνίας 5G (5<sup>ης</sup> Γενεάς), δεν έχει προκύψει κάποια πλήρης προτυποποίηση. Για αυτό τον λόγο στο πεδίο της ενσωμάτωσης τεχνικών μηχανικής μάθησης σε αρχιτεκτονικές δικτύων κινητής επικοινωνίας 5G (5<sup>ης</sup> Γενεάς) έχει πλέον ξεκινήσει από την International Telecommunication Union (ITU) [28] εκτεταμένη έρευνα με στόχο να προκύψει κάποιο πρότυπο, καθώς τα πλεονεκτήματα από αυτή την ενσωμάτωση τόσο από την μεριά των χρηστών όσο και των παρόχων δικτύων φαίνεται να είναι αξιοσημείωτα.

Πιο συγκεκριμένα ένα μέρος των μελετών έχει αναλωθεί σε τεχνικές μηχανικής μάθησης που προβλέπουν την δικτυακή κίνηση μέσω της κατανάλωσης υπολογιστικών ή δικτυακών πόρων σε δίκτυα κινητής επικοινωνίας 5G (5<sup>ης</sup> Γενεάς) [27]. Οι τεχνικές αυτές είναι σε θέση να υποστηρίξουν την βελτιστοποίηση δρομολόγησης των δεδομένων και την πρόβλεψη μελλοντικών αναγκών σε υπολογιστικούς ή δικτυακούς πόρους, ώστε να αποσυμφορηθεί το δίκτυο εξισορροπώντας αποτελεσματικά το φορτίο και επίσης να αποφευχθούν μελλοντικά σφάλματα ή πιθανές αστοχίες στο δίκτυο εξαιτίας έλλειψης υπολογιστικών ή δικτυακών πόρων [1]. Ωστόσο παρά τις μελέτες που έχουν προηγηθεί υπάρχει αρκετός χώρος ακόμα για την βελτιστοποίηση των τεχνικών μηχανικής μάθησης, αναζητώντας τις βέλτιστες παραμέτρους στα διάφορα μοντέλα μηχανικής μάθησης, έτσι ώστε αυτά να προβλέπουν με όσο το δυνατόν μεγαλύτερη ακρίβεια την δικτυακή κίνηση, μέσω χρονοσειρών διαφόρων μετρικών που σχετίζονται με την κατανάλωση υπολογιστικών ή δικτυακών πόρων σε δίκτυα κινητής επικοινωνίας 5G (5<sup>ης</sup> Γενεάς).

Η παρούσα μεταπτυχιακή εργασία λοιπόν, επικεντρώνεται στην περιγραφή υλοποίησης μοντέλων μηχανικής μάθησης καθώς και της βελτιστοποίησης τους, έτσι ώστε να μπορούν τα μοντέλα αυτά να αξιοποιηθούν για την πρόβλεψη δικτυακής κίνησης σε δίκτυα κινητής τηλεπικοινωνίας 5G (5<sup>ης</sup> γενεάς) με μεγάλη ακρίβεια. Τα αποτελέσματα αυτής της εργασίας θα συμβάλλουν στην αποτελεσματικότερη μελλοντική διαχείριση των υπολογιστικών ή δικτυακών πόρων του δικτύου, ώστε να αποφευχθούν σφάλματα που μπορεί να οδηγήσουν σε κατάρρευση του. Αναλυτικότερα, δημιουργήθηκε ένα

περιβάλλον που αποτελείται από ένα υπολογιστικό σύστημα με ένα ενεργό δίκτυο κινητής τηλεπικοινωνίας 5G (5<sup>ης</sup> γενεάς), από το οποίο συλλέγονται χρονοσειρές διάφορων μετρικών (metrics) για να παρακολουθείται η δικτυακή κίνηση μέσω της κατανάλωσης υπολογιστικών ή δικτυακών πόρων εξαιτίας λειτουργιών του δικτύου 5G (5<sup>ης</sup> γενεάς), όπως η αποστολή πακέτων δεδομένων προς μια συνδεδεμένη στο δίκτυο συσκευή. Στην συνέχεια στις χρονοσειρές των μετρικών αυτών εφαρμόζονται μοντέλα μηχανικής μάθησης με διάφορους συνδυασμούς παραμέτρων, έτσι ώστε να εντοπιστεί το μοντέλο και οι παράμετροί του, που θα προσφέρει για την εκάστοτε μετρική την μεγαλύτερη ακρίβεια πρόβλεψης της εξέλιξης των χρονοσειρών τους στον χρόνο.

Εμβαθύνοντας στην δομή της εργασίας, αρχικά κρίθηκε απαραίτητη η ανάλυση της αρχιτεκτονικής και των λόγων που ωθούν την χρησιμοποίηση ενός δικτύου κινητής τηλεπικοινωνίας 5G (5<sup>ης</sup> γενεάς) (Κεφάλαιο 2). Συνεχίζοντας, ορίστηκαν θεωρητικά οι χρονοσειρές και αφού κατηγοριοποιήθηκαν, παρουσιάστηκαν τρόποι αποθήκευσης και ανάλυσης αυτών (Κεφάλαιο 3). Αμέσως μετά, πραγματοποιήθηκε μια σύντομη περιγραφή των συστημάτων παρακολούθησης ή επιτήρησης (Κεφάλαιο 3), τα οποία ουσιαστικά συλλέγουν μετρικές χρονοσειρών από υπολογιστικά συστήματα και τις αποθηκεύουν επίσης σε ειδικές βάσεις δεδομένων χρονοσειρών, όπως τονίζεται. Προχωρώντας στην εργασία (Κεφάλαιο 4) γίνεται μια πλήρης ανάλυση των πεδίων της τεχνητής νοημοσύνης, της μηχανικής μάθησης, των τεχνητών νευρωνικών δικτύων αλλά και της βαθιάς μάθησης, η οποία ουσιαστικά αποτελεί υποπεδίο της μηχανικής μάθησης για την εκπαίδευση τεχνητών νευρωνικών δικτύων. Εν συνεχεία παρουσιάζεται η γλώσσα προγραμματισμού *python* και οι βιβλιοθήκες της (Κεφάλαιο 4.5), λόγω των οποίων έχει ταυτιστεί με το πεδίο της μηχανικής μάθησης. Κατόπιν, ακολουθεί ένας σύντομος οδηγός για την πραγμάτωση ενός δικτύου κινητής τηλεπικοινωνίας 5G (5<sup>ης</sup> γενεάς) σε ένα υπολογιστικό σύστημα, το οποίο επιτηρείται από ένα εργαλείο παρακολούθησης που συλλέγει χρονοσειρές διάφορων μετρικών σχετικών με την δικτυακή κίνηση, όπως η κατανάλωση υπολογιστικών ή δικτυακών πόρων από το δίκτυο κινητής τηλεπικοινωνίας 5G (5<sup>ης</sup> γενεάς) και ύστερα τις αποθηκεύει σε μια βάση δεδομένων (Κεφάλαιο 5.1.1). Επίσης, στον οδηγό αυτόν, παρουσιάζεται ο υλοποιημένος στην γλώσσα προγραμματισμού *python* κώδικας για την ανάκτηση των δεδομένων χρονοσειρών των μετρικών *node\_cpu\_seconds\_total* και *node\_network\_receive\_bytes\_total* από την προαναφερθέντα βάση δεδομένων, οι οποίες αντιστοιχούν στο συνολικό χρόνο που καταναλώνει ο επεξεργαστής (CPU) για *interrupts* λόγω του Δικτύου 5G και στα συνολικά bytes που λαμβάνονται στο UPF στοιχείο του Δικτύου 5G από τους σταθμούς βάσης (gNB) (Κεφάλαιο 5.1.2). Έπειτα αναλύεται ο κώδικας που εφαρμόστηκε για την βελτιστοποίηση των Gated Recurrent Unit Network (GRU), Long Short-Term Memory Network (LSTM), Convolutional Neural Network (CNN) και Multi-layer Perceptron (MLP) μοντέλων μηχανικής μάθησης που χρησιμοποιήθηκαν για να προβλέπουν την εξέλιξη των δύο προαναφερθέντων μετρικών στο χρόνο (Κεφάλαιο 5.1.2). Πέρα από αυτά, περιγράφεται ακόμη και η γραφική διεπαφή χρήστη που παράχθηκε έτσι ώστε εύκολα να ανακτηθούν τα δεδομένα χρονοσειρών των δύο μετρικών από την βάση δεδομένων, να εφαρμοστούν-βελτιστοποιηθούν τα μοντέλα μηχανικής μάθησης καθώς και να εικονικοποιηθούν τα αποτελέσματα (Κεφάλαιο 5.1.3). Ακολουθως, σκιαγραφείται η ερευνητική διαδικασία, κατά την οποία βελτιστοποιούνται οι υπερπαραμέτροι των GRU, LSTM, CNN και MLP μοντέλων μηχανικής μάθησης, ώστε αυτά να προβλέπουν με όσο το δυνατόν μεγαλύτερη ακρίβεια την εξέλιξη των *node\_cpu\_seconds\_total* και *node\_network\_receive\_bytes\_total* μετρικών στον χρόνο. (Κεφάλαιο 5.2.1). Τα αποτελέσματά της διαδικασίας βελτιστοποίησης παρουσιάζονται συγκεντρωτικά και από αυτά εξάγονται τα συμπεράσματα (Κεφάλαιο 5.2.2), που είναι η εύρεση του μοντέλου μηχανικής μάθησης καθώς και οι τιμές των παραμέτρων αυτού, το οποίο προβλέπει με την μεγαλύτερη δυνατή ακρίβεια την εξέλιξη στον χρόνο των *node\_cpu\_seconds\_total* και *node\_network\_receive\_bytes\_total*

μετρικών που αντιπροσωπεύουν έμμεσα την δικτυακή κίνηση του δικτύου κινητής τηλεπικοινωνίας 5G (5<sup>ης</sup> γενεάς). Καταληκτικά, στον επίλογο της εργασίας (Κεφάλαιο 6) πραγματοποιείται μια σύνοψη αυτής.



δικτύου κινητής τηλεπικοινωνίας LTE (4G) το έτος 2008, οπότε παρατηρήθηκε εξαιτίας αυτής η ευρεία υιοθέτηση του Mobile Broad Band (MBB), μέσω του οποίου κατέστη εφικτή η χρήση βίντεο και δεδομένων στο all-IP δίκτυο. Τελευταία είναι η μετάβαση στην τεχνολογία δικτύων κινητής τηλεπικοινωνίας 5G όπου παρατηρείται μια ριζική αναθεώρηση-αναβάθμιση των τηλεπικοινωνιών, σε αντιδιαστολή με τη μετάβαση από την τεχνολογία της κινητής τηλεπικοινωνίας που ονομάζεται 2G σε αυτή με το όνομα 3G ή την μετάβαση από την τεχνολογία της κινητής τηλεπικοινωνίας που ονομάζεται 3G σε αυτή με το όνομα 4G. Σε αντίθεση με τις προηγούμενες γενεές δικτύων το 5G δεν εστιάζει στο να έχει πελάτες μόνο απλούς χρήστες αλλά κυρίως τις βιομηχανίες, ενσωματώνοντας στο δίκτυο τις απαιτήσεις τους [2]. Επιπλέον, τα δύο μέρη του δικτύου 5G έχουν σχεδιαστεί με τέτοιο τρόπο ώστε οι διάφορες υπηρεσίες που προσφέρονται, όπως η πρόσβαση σε κάποιο ιστότοπο, η μεταφορά δεδομένων, τα cloud services, οι εφαρμογές δικτύου, καθώς και η διαχείριση όλων αυτών στο δίκτυο, να γίνεται με απλοποιημένες, αυτοματοποιημένες μεθόδους, ενώ κρίθηκε απαραίτητο να προβλέπεται η μελλοντική ανάγκη νέων υπηρεσιών και η εύκολη προσθήκη τους στο δίκτυο. Ακόμη, στο εξής μέσω του 5G θα καλυφθεί και η ανάγκη για machine to machine επικοινωνία, γεγονός που θα διαφοροποιήσει τον τρόπο ανάπτυξης των υπηρεσιών. Να τονιστεί πως σε κάθε γενιά είναι εμφανής η αύξηση στο εύρος ζώνης και στις ταχύτητες που παρέχονται στους καταναλωτές-πελάτες.

### 2.3 Οι Βασικοί Παράγοντες για την Χρησιμοποίηση της Τεχνολογίας Δικτύων Κινητής Τηλεφωνίας 5G (5<sup>ης</sup> γενεάς) Σύμφωνα με τις Απαιτήσεις της Εποχής

Η τεχνολογία δικτύων κινητής τηλεφωνίας 5G (5<sup>ης</sup> γενεάς) αναπτύχθηκε με γνώμονα τις ανάγκες των βιομηχανιών κυρίως, αλλά και αιτημάτων από τους χρήστες για νέες υπηρεσίες που είχαν μείνει ανεκπλήρωτες από τις προηγούμενες γενεές δικτύων κινητής τηλεφωνίας. Μερικές από αυτές τις ανάγκες που ικανοποιήθηκαν και είναι βασικοί παράγοντες χρησιμοποίησης της τεχνολογίας δικτύων κινητής τηλεφωνίας 5G (5<sup>ης</sup> γενεάς) περιγράφονται παρακάτω [2]:

- Βελτιωμένο Mobile Broadband (eMBB) δηλαδή διευρυμένο ρυθμό ροής δεδομένων για μεταφορά μεγαλύτερου όγκου δεδομένων.
- Machine Type Communications (mMTC) οι οποίες κάνουν δυνατή την machine to machine επικοινωνία (M2M) καθώς και την επικοινωνία εφαρμογών που ανήκουν στον τομέα του Internet of Things (IoT), χωρίς ωστόσο την επιβάρυνση των υπόλοιπων υπηρεσιών.
- Απαίτηση υψηλής αξιοπιστίας και χαμηλής καθυστέρησης επικοινωνία (URLLC) για τις εφαρμογές που είναι ευαίσθητες στον χρόνο.
- Αύξηση της χωρητικότητας του δικτύου με πολύ λιγότερο κόστος σε σχέση με την αντίστοιχη αύξηση της χωρητικότητας σε ένα δίκτυο κινητής τηλεπικοινωνίας που χρησιμοποιεί την τεχνολογία 4G/LTE.
- Αυτοματοποίηση κάποιων διαδικασιών, όπως αύξηση χωρητικότητας, αναβάθμιση λογισμικού και χρήση στατιστικών για την βελτιστοποίηση του συστήματος.
- Ενοποίηση των κινητών και σταθερών υπηρεσιών που προσφέρουν οι πάροχοι στους πελάτες όσον αφορά τις υποδομές που χρησιμοποιούνται και τις διαδικασίες λειτουργικότητας, γεγονός που αποσκοπεί στην μείωση του κόστους.



## 2.4 Τεχνολογίες που Συνέβαλαν στην Ανάπτυξη της Τεχνολογίας Δικτύων Κινητής Τηλεφωνίας 5G (5<sup>ης</sup> Γενεάς)

Η ανάπτυξη ορισμένων τεχνολογιών τα τελευταία χρόνια βοήθησε στην δημιουργία του 5G, αξιοποιώντας διάφορα από τα χαρακτηριστικά των τεχνολογιών αυτών. Μερικές κύριες τεχνολογίες που προώθησαν την ανάπτυξη του 5G είναι οι εξής [2]:

1. Εικονικοποίηση (Virtualization)
2. Cloud native αρχιτεκτονική
3. Containerization
4. Μικροϋπηρεσίες (Microservices)
5. Αυτοματοποίηση (Automization)

## 2.5 Δίκτυο Πυρήνα 5ης Γενεάς

### 2.5.1 Αρχιτεκτονική του Δικτύου Πυρήνα 5ης Γενεάς

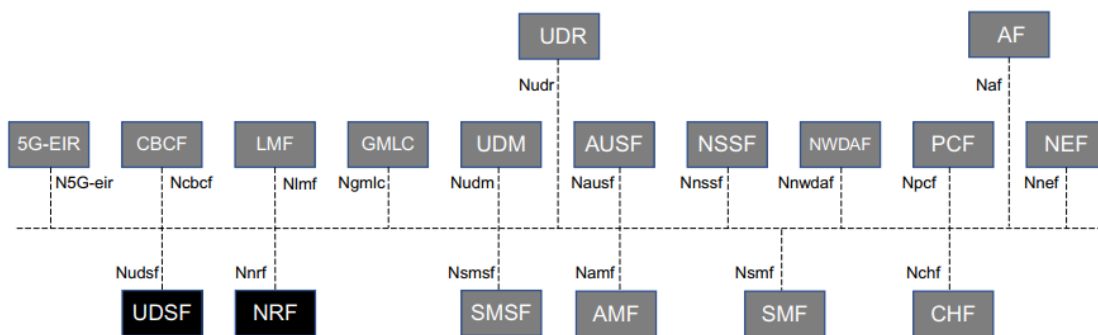
Το Δίκτυο Πυρήνα 5ης γενεάς (5G Core Network) είναι ο πυρήνας ενός τηλεπικοινωνιακού δικτύου κινητής τηλεφωνίας τεχνολογίας 5G. Δημιουργεί αξιόπιστη και ασφαλή σύνδεση στο δίκτυο για τις διάφορες συσκευές, ενώ ταυτόχρονα παρέχει πρόσβαση και στις υπηρεσίες του δικτύου.

Σε σύγκριση με την αρχιτεκτονική του Core Network της προηγούμενης γενεάς που ονομάζεται EPC, αυτή του 5G Core Network, παρόλο που επεξεργάζεται τα δεδομένα και συνδέεται με τα 3GPP Radio Access Networks με πανομοιότυπο τρόπο, διαφέρει αρκετά στη διαδικασία του signaling, καθώς και στον τρόπο διαφόρων λειτουργιών (λειτουργία ονομάζεται ένα κομμάτι λογισμικού του 5G Core Network που προσφέρει κάποιες υπηρεσίες στο δίκτυο) του Core Network, οι οποίες λειτουργίες χαρακτηρίζονται από λογική και έχουν την δυνατότητα να πραγματοποιούν την διαδικασία του signaling. Όσον αφορά τον τρόπο διασύνδεσης διαφόρων λειτουργιών του Core Network, πλέον στο 5G Core Network αυτή γίνεται μέσω Service-Based διεπαφών κάνοντας έτσι τις υπηρεσίες της κάθε λειτουργίας διαθέσιμες σε όλες τις υπόλοιπες λειτουργίες του Core Network, σε αντίθεση με την αρχιτεκτονική του EPC όπου χρησιμοποιούνται οι άμεσες point-to-point διεπαφές μεταξύ των διαφόρων λειτουργιών του Core Network που κρίνεται απαραίτητο να επικοινωνούν μεταξύ τους. Αναλυτικότερα, η κάθε λειτουργία του 5G Core Network μέσω της διεπαφής της διασυνδέεται στην κοινή Service-Based αρχιτεκτονική κάνοντας διαθέσιμες τις υπηρεσίες της σε όποιες λειτουργίες είναι συνδεδεμένες σε αυτή. Να τονιστεί πως η Service-Based αρχιτεκτονική έχει εφαρμογή μόνο για διαδικασίες signaling και όχι μεταφοράς δεδομένων, ενώ καθιστά και πιο εύκολη την μελλοντική επέκταση του 5G Core Network σε σχέση με την κλασσική point-to-point αρχιτεκτονική. Ακόμη για κάθε αλληλεπίδραση μεταξύ δύο λειτουργιών του Core Network μια από αυτές καλείται 'Service Consumer' και χρησιμοποιεί τις υπηρεσίες της άλλης η οποία καλείται 'Service Producer' καθώς προσφέρει την υπηρεσία της. Η μέθοδος επικοινωνίας που χρησιμοποιείται για το 5G Core Network και πιο συγκεκριμένα η Service-Based αρχιτεκτονική που έχει χρησιμοποιηθεί σ' αυτό ονομάζεται 'HTTP Representational State Transfer (REST) paradigm' και είναι ουσιαστικά ένα σύνολο κανόνων για το πώς επικοινωνούν διαφορετικά λογισμικά ή, στην περίπτωση του 5G Core Network, οι λειτουργίες (που είναι ένα κομμάτι λογισμικού) σε αρχιτεκτονική Network. Η σύνταξη της μεθόδου αυτής προέρχεται από το HTTP web πρωτόκολλο ενώ βασίζεται και στο μοντέλο του Resource Modeling, όπου ένα καταναμημένο λογισμικό μπορεί να διευθυνσιοδοτηθεί μέσω του



Uniform Resource Identifier (URI) (Το URI παρέχει μοναδικές ακολουθίες χαρακτήρων για την ταυτοποίηση πόρων). Τέλος, για την εύρυθμη λειτουργία της Service-Based αρχιτεκτονική υπάρχει μία λειτουργία με την μορφή λογισμικού στο 5G Core Network που ονομάζεται Network Repository Function (NRF) και η οποία καταγράφει όλες τις διαθέσιμες λειτουργίες του Core Network καθώς και τις υπηρεσίες που αυτές προσφέρουν, βοηθώντας με αυτό τον τρόπο όλες τις λειτουργίες του Core Network που δρουν σαν 'Service Consumer' να εντοπίζουν και να επικοινωνούν με την αντίστοιχη λειτουργία του Core Network που δρα ως 'Service Producer' και θέλουν να χρησιμοποιήσουν τις υπηρεσίες του. Η λειτουργία NRF βασίζεται στο μοντέλο του Service Discovery και έτσι λοιπόν κάθε λειτουργία του Core Network θα πρέπει καταχωρεί στο NRF πως οι υπηρεσίες του είναι διαθέσιμες [2].

Ένα εξαιρετικό λογισμικό ανοιχτού κώδικα που αναπαριστά ένα 5G Core Network ονομάζεται free5GC.



Εικόνα 2: Η service-based αρχιτεκτονική των λειτουργιών ενός 5G Core Network [1]

## 2.5.2 Βασικές Λειτουργίες της Αρχιτεκτονικής του Δικτύου Πυρήνα 5ης Γενεάς

Η βασική χρήση της αρχιτεκτονικής του δικτύου Πυρήνα 5ης γενεάς (5G Core Network) περιλαμβάνει την δημιουργία συνεδριών με ασφαλή τρόπο, καθώς και την προώθηση απαιτούμενων δεδομένων και υπηρεσιών από και προς κινητές συσκευές. Οι βασικές του λειτουργίες, που δεν είναι τίποτε άλλο από ένα κομμάτι λογισμικού συνδεδεμένο στην κοινή Service-Based αρχιτεκτονική, οι οποίες παρέχουν τις υπηρεσίες τους είναι οι εξής [2]:

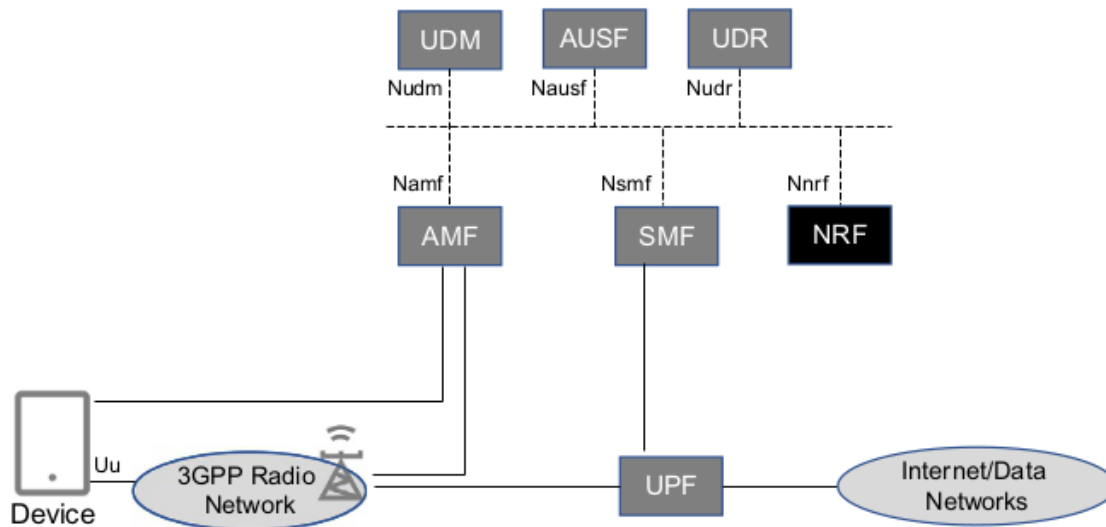
- Η λειτουργία AMF ή αναλυτικά 'Access and Mobility Management Function'. Αυτή αλληλοεπιδρά με το NG-RAN και τις συσκευές μέσω μιας διαδικασίας signaling με την βοήθεια των διεπαφών N2 και N1 αντίστοιχα. Παρόλο που οι υπόλοιπες λειτουργίες του 5G Core Network επικοινωνούν μεταξύ τους μέσω της κοινής Service-Based αρχιτεκτονικής η λειτουργία AMF εμπλέκεται σε πολλές διαδικασίες signaling. Επιπλέον δυνατότητες της AMF είναι η κρυπτογραφημένες διαδικασίες signaling, διαδικασίες εγγραφής, επαλήθευσης και κινητικότητας.
- Η λειτουργία SMF ή αναλυτικά 'Session Management Function'. Αυτή διαχειρίζεται τις συνεδρίες οποιασδήποτε συσκευής. Στις αρμοδιότητες της ανήκουν η δημιουργία, η τροποποίηση, ο τερματισμός και η κατανομή IP διευθύνσεων για κάθε συνεδρία, ενώ επικοινωνεί με τις συσκευές έμμεσα προωθώντας και λαμβάνοντας μηνύματα διάμεσου της λειτουργίας AMF. Επιπλέον, η λειτουργία SMF συνδέεται μέσω της κοινής Service-Based αρχιτεκτονικής με τις υπόλοιπες λειτουργίες του 5G Core Network. Εξαίρεση αποτελεί η point-

to-point διεπαφή N4 που διαθέτει με τις λειτουργίες UPF τις οποίες επιθεωρεί και κατευθύνει την κίνηση των δεδομένων που γίνονται μέσω αυτών, ενώ ταυτόχρονα συλλέγει πληροφορίες χρεώσεων για τις υπηρεσίες που προσφέρθηκαν στους χρήστες.

- Η λειτουργία UPF ή αναλυτικά 'User Plane Function'. Αυτή έχει ως βασική της διεργασία την επεξεργασία και προώθηση δεδομένων. Επίσης συνδέεται με εξωτερικά IP Networks και λειτουργεί ως ένα σημείο με σταθερή διεύθυνση IP ως προς τις συσκευές που συνδέονται στα εξωτερικά δίκτυα, συγκαλύπτοντας έτσι την κινητικότητα καθώς η συσκευή κινείται στον χώρο, μια και τα πακέτα που έχουν ως IP διεύθυνση παράδοσης αυτής της συσκευής θα δρομολογούνται πρώτα στην λειτουργία UPF που εξυπηρετεί την συσκευή και έπειτα στην συσκευή. Επιπροσθέτως, παράγει αναφορές κίνησης δεδομένων που διοχετεύει στην λειτουργία SMF και έχει την δικαιοδοσία να τοποθετεί μια σήμανση στα διάφορα πακέτα η οποία θα καθορίζει την προτεραιότητα του κάθε πακέτου, σε περίπτωση που υπάρξει κίνηση σε κάποιο δίκτυο στο οποίο πιθανώς θα προωθηθούν τα πακέτα, ενώ έχει την δυνατότητα να επιθεωρήσει το περιεχόμενο των πακέτων δεδομένων που αποστέλλονται για να εφαρμόσει την κατάλληλη πολιτική.

Μια ακόμη σημαντική δυνατότητα της λειτουργίας UPF είναι η ικανότητα να αποθηκεύει προσωρινά τα δεδομένα που αποστέλλονται σε μία συσκευή που είναι σε αδρανή κατάσταση και στην συνέχεια να αναγκάζει την συσκευή να επιστρέψει σε κατάσταση σύνδεσης, ώστε στην συνέχεια να λάβει τα δεδομένα που προορίζονταν για αυτήν.

- Η λειτουργία UDR ή αναλυτικά 'Unified Data Repository'. Αυτή ουσιαστικά λειτουργεί ως βάση δεδομένων για τους διάφορους τύπους δεδομένων που υπάρχουν και κινούνται στο 5G Core Network, όπως τα δεδομένα συνδρομητών, καθώς και τα δεδομένα διαφόρων πολιτικών που ακολουθούνται. Τέλος, η λειτουργία UDR είναι μια υπηρεσία που μπορεί να χρησιμοποιηθεί και από άλλες λειτουργίες του 5G Core Network.
- Η λειτουργία UDM ή αναλυτικά 'Unified Data Management Function'. Αυτή διαχειρίζεται τα δεδομένα συνδρομητών που είναι αποθηκευμένα στην βάση δεδομένων της λειτουργίας UDR. Πιο συγκεκριμένα παράγει τα κατάλληλα δεδομένα ταυτοποίησης αξιοποιώντας τα δεδομένα συνδρομητών, δίνοντας έτσι την ανάλογη πρόσβαση στις υπηρεσίες σύμφωνα με την συνδρομή του κάθε χρήστη. Τέλος επειδή στο 5G Core Network υπάρχουν μεγάλες πιθανότητες για την ύπαρξη πολλών AMF και SMF, η λειτουργία UDM είναι ενημερωμένη συνεχώς για τις συγκεκριμένες λειτουργίες AMF και SMF που προσφέρουν τις υπηρεσίες τους σε μία συσκευή.
- Η λειτουργία AUSF ή αναλυτικά 'Authentication Server Function'. Αυτή παρέχει την υπηρεσία ταυτοποίησης μιας συσκευής χρησιμοποιώντας τα απαραίτητα δεδομένα ταυτοποίησης που προσφέρει η λειτουργία UDM και είναι αποθηκευμένα στην βάση δεδομένων που παρέχει η λειτουργία UDR. Επιπροσθέτως, από την AUSF δημιουργούνται κρυπτογραφημένα πακέτα δεδομένων έτσι ώστε ρυθμιστούν διάφορες παράμετροι σε μια συσκευή.
- Η λειτουργία NRF ή αναλυτικά 'Network Repository Function'. Αυτή, όπως προαναφέρθηκε είναι μια υπηρεσία Service Discovery που καταγράφει όλες τις διαθέσιμες λειτουργίες του 5G Core Network, όπως και τις υπηρεσίες που αυτές προσφέρουν, βοηθώντας με αυτό τον τρόπο όλες τις λειτουργίες του 5G Core Network να επικοινωνούν μεταξύ τους και να ανταλλάσσουν τις υπηρεσίες τους.



Εικόνα 3: Οι βασικές λειτουργίες ενός 5G Core Network και οι διασυνδέσεις τους με τις συσκευές και το New Generation Radio Access Network (3GPP Radio Network) [1]

## 2.6 Δίκτυο Ραδιοπρόσβασης NG-RAN

### 2.6.1 Αρχιτεκτονική του Δικτύου Ραδιοπρόσβασης NG-RAN

Το δίκτυο Ραδιοπρόσβασης (New Generation Radio Access Network ή NG-RAN ή αλλιώς New Radio) είναι το δεύτερο μέρος ενός δικτύου κινητής τηλεπικοινωνίας 5G, το οποίο αποτελείται από πολλαπλούς ραδιοσταθμούς βάσης, ο καθένας από τους οποίους εξυπηρετεί την λήψη και την μετάδοση δεδομένων από συνήθως τρία ‘κελιά’ στα οποία προσφέρει τις υπηρεσίες του. Με τον όρο ‘κελί’ νοείται ένα τμήμα μιας μεγάλης γεωγραφικά περιοχής που καλύπτει ένα δίκτυο κινητής τηλεπικοινωνίας. Το μέγεθος και σχήμα ενός κελιού εξαρτάται από διάφορους παράγοντες όπως η ισχύς των ραδιοσταθμών βάσης και των συσκευών που είναι συνδεδεμένες στο δίκτυο κινητής τηλεπικοινωνίας 5G, τις συχνότητες που χρησιμοποιούνται αλλά και τις κεραίες μετάδοσης και λήψης των ραδιοσταθμών βάσης και των συσκευών και, τέλος, των πιθανών εμποδίων που θα υπάρχουν στο περιβάλλον μετάδοσης. Επιπροσθέτως ένα δομικό χαρακτηριστικό των κελιών είναι η ταυτόχρονη μετάδοση ίδιων συχνοτήτων σε διαφορετικά κελιά χωρίς να υπάρχουν παρεμβολές, αυξάνοντας έτσι την χωρητικότητα και κάλυψη του δικτύου. Έτσι λοιπόν οι ραδιοσταθμοί πρέπει να τοποθετούνται σε συγκεκριμένα σημεία ώστε να πληρούν τα κριτήρια που αναλύθηκαν πιο πάνω. Το NG-RAN χρησιμοποιεί αρκετές τεχνολογικές ιδέες από την προηγούμενη γενεά Radio Access Network που ονομάζεται LTE οι οποίες ωστόσο έχουν εξελιχθεί. Μια από αυτές τις ιδέες είναι η χρήση της τεχνολογίας διαμόρφωσης OFDM προσαρμοσμένη στις απαιτήσεις του 5G δικτύου. Αναλυτικότερα, με την τεχνολογία διαμόρφωσης OFDM αυτό που συμβαίνει είναι πως το διαθέσιμο φάσμα χωρίζεται σε κανάλια, στο κάθε ένα από τα οποία αντιστοιχεί ένας φορέας (κύμα με συγκεκριμένη συχνότητα υψηλής τιμής), με αποτέλεσμα η χωρητικότητα της εκάστοτε συσκευής να ρυθμίζεται τόσο στο πεδίο του χρόνου όσο και της συχνότητας. Επίσης το συγκεκριμένο σύστημα διαμόρφωσης είναι πολύ αποτελεσματικό στο φαινόμενο αλλοίωσης του σήματος εξαιτίας πολλαπλών διαδρομών του τελευταίου. Τέλος το NG-RAN έχει την δυνατότητα για εύρος ζώνης συχνοτήτων περίπου 400 MHz σε ένα φορέα, σε αντίθεση με το LTE που είναι αντίστοιχα περίπου 20 MHz, και επίσης στο NG-RAN υπάρχουν πολλές διαφορετικές επιλογές ελάχιστης απόστασης

μεταξύ συχνοτήτων δύο φορέων στο πεδίο της συχνότητας ενώ στο LTE υπάρχει μόνο μία επιλογή. Γενικά είναι θεμιτή η χρησιμοποίηση πολλών φορέων ταυτόχρονα για την καλύτερη αξιοποίηση του φάσματος. Καταληκτικά το NG-RAN είναι πιο ευέλικτο για μελλοντικές προσθήκες δυνατοτήτων και έχει μειωμένες ενεργειακές καταναλώσεις, αθέμιτες παρεμβολές, επιτρέποντας ταυτόχρονα την αποστολή δεδομένων με πολύ μικρή απόκριση και την αποστολή εντολών ελέγχου με τέτοιο τρόπο ώστε να βελτιστοποιείται η χρήση των πόρων [2].

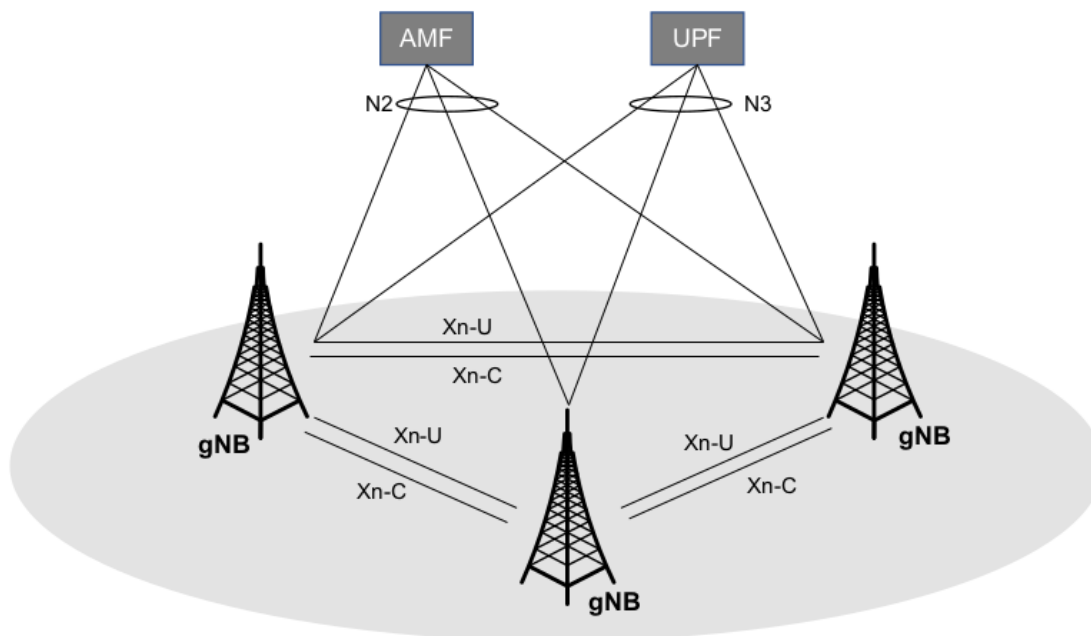
Ένα εξαιρετικό λογισμικό ανοιχτού κώδικα που αναπαριστά ένα αναπαριστά ένα NG-RAN και έχει την δυνατότητα δημιουργίας εικονικών συσκευών-συνδρομητών συνδεδεμένων σε αυτό ονομάζεται UERANSIM.

#### 2.6.2 Λειτουργικά Χαρακτηριστικά της Αρχιτεκτονικής του Δικτύου Ραδιοπρόσβασης NG-RAN

Όπως προαναφέρθηκε, το δίκτυο Ραδιοπρόσβασης (New Generation Radio Access Network ή NG-RAN ή αλλιώς New Radio) αποτελείται από πολλαπλούς ραδιοσταθμούς βάσης που είναι συνδεδεμένοι μεταξύ τους, αλλά και με το 5G Core Network. Σύμφωνα με την βιβλιογραφία ένας ραδιοσταθμός βάσης που είναι συνδεδεμένος με το 5G Core Network ονομάζεται 'gNB'. Οι ραδιοσταθμοί βάσης συνδέονται μεταξύ τους χρησιμοποιώντας τις διεπαφές Xn-C και Xn-U μέσω των οποίων πραγματοποιούνται οι διαδικασίες του signaling και μεταφοράς δεδομένων αντίστοιχα. Επιπλέον οι ραδιοσταθμοί βάσης συνδέονται με μία ή περισσότερες λειτουργίες AMF και UPF του 5G Core Network χρησιμοποιώντας τις διεπαφές N2 και N3, μέσω των οποίων πραγματοποιούνται οι διαδικασίες του signaling και μεταφοράς δεδομένων αντίστοιχα.

Όσον αφορά την μεταφορά δεδομένων μεταξύ των ραδιοσταθμών βάσης και μεταξύ των ραδιοσταθμών βάσης και της λειτουργίας UPF του 5G Core Network που διαχειρίζεται τα δεδομένα, αυτή πραγματοποιείται με δικτύωση IP και ακολουθείται η ίδια στοίβα πρωτοκόλλου επικοινωνίας και για τις δύο διεπαφές μεταφοράς δεδομένων Xn-U και N3. Ξεκινώντας από τα πάνω μέρος της στοίβας πρωτοκόλλου επικοινωνίας, αυτή μεταφέρει τα IP πακέτα δεδομένων χρησιμοποιώντας το GTP-U πρωτόκολλο το οποίο με την σειρά του μεταφέρεται με την βοήθεια μιας τυπικής UDP/IP στοίβας η οποία εκτελείται πάνω από δύο ακόμα στρώματα πρωτοκόλλου που όμως δεν έχουν καθοριστεί σαφώς από την βιβλιογραφία του 5G Core Network. Από την άλλη μεριά, η διαδικασία του signaling μεταξύ των ραδιοσταθμών βάσης και μεταξύ των ραδιοσταθμών βάσης και της λειτουργίας AMF του 5G Core Network που διαχειρίζεται την διαδικασία του signaling πραγματοποιείται επίσης με δικτύωση IP, αλλά σε αυτή την περίπτωση τα ανώτερα στρώματα της στοίβας πρωτοκόλλου επικοινωνίας διαφέρουν σε σχέση με την στοίβα πρωτοκόλλου επικοινωνίας για την μεταφορά δεδομένων που περιγράφηκε προηγουμένως. Στην περίπτωση αυτή υπάρχουν δύο διαφορετικές στοίβες πρωτοκόλλου επικοινωνίας, μία για την διαδικασία signaling μεταξύ των ραδιοσταθμών βάσης μέσω της διεπαφής Xn-U και μια για την διαδικασία signaling μεταξύ των ραδιοσταθμών βάσης και της λειτουργίας AMF του 5G Core Network μέσω της διεπαφής N2. Και στις δύο αυτές στοίβες στο δεύτερο στρώμα της στοίβας πρωτοκόλλου επικοινωνίας αντί για το πρωτόκολλο UDP χρησιμοποιείται το πρωτόκολλο SCTP αφού, πέρα από την πιστοποιημένη, προσφέρει και πιο ασφαλή μεταφορά μηνυμάτων. Επιπροσθέτως στο ανώτερο στρώμα της στοίβας πρωτοκόλλου επικοινωνίας που εφαρμόζεται στην διεπαφή N2 αντί για το πρωτόκολλο GTP-U που χρησιμοποιείται στην στοίβα πρωτοκόλλου επικοινωνίας για την μεταφορά δεδομένων, εφαρμόζεται το πρωτόκολλο NG-AP το οποίο διαχειρίζεται διαδικασίες signaling για την κινητικότητα μεταφέροντας NAS μηνύματα μεταξύ

των συσκευών και του 5G Core Network, καθώς και διαδικασίες signaling για την επαναφορά μια συσκευής από κατάσταση αδράνειας σε ενεργή κατάσταση. Αλλαγές υπάρχουν επίσης και στο ανώτερο στρώμα της στοίβας πρωτοκόλλου επικοινωνίας που εφαρμόζεται στην διεπαφή Xn-U αφού, αντί για το πρωτόκολλο GTP-U που χρησιμοποιείται στην στοίβα πρωτοκόλλου επικοινωνίας για την μεταφορά δεδομένων, εφαρμόζεται το πρωτόκολλο Xn-AP το οποίο διαχειρίζεται διαδικασίες signaling για την κινητικότητα, καθώς και λειτουργίες για την εφαρμογή του σχεδίου που ονομάζεται Dual Connectivity και έχει ως σκοπό τον συνδυασμό δύο τεχνολογιών Radio Access Network όπως αυτή του NG-RAN (5G) και του LTE (4G) [2].

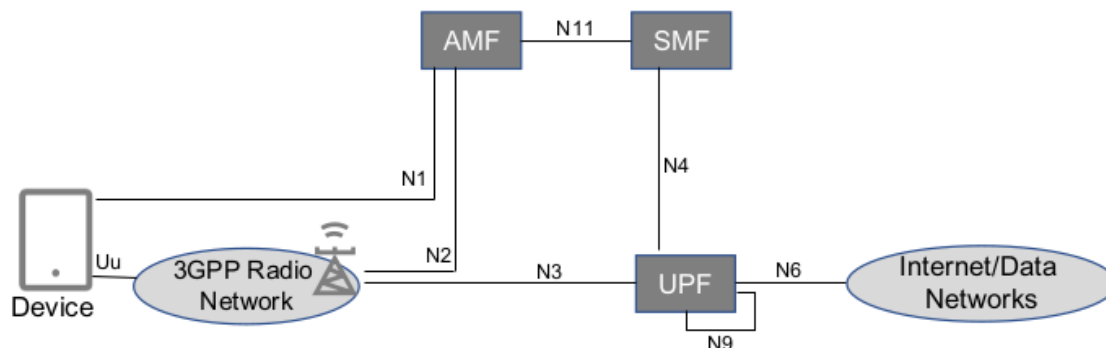


Εικόνα 4: Η διασύνδεση των ραδιοσταθμών βάσης του New Generation Radio Access Network μεταξύ τους καθώς και των ραδιοσταθμών βάσης του New Generation Access Network με το 5G Core Network [1]

Ακόμη πέρα από τις διεπαφές που περιγράφηκαν υπάρχει και η διεπαφή NR αέρα για την επικοινωνία μεταξύ συσκευών και ραδιοσταθμών βάσης, η οποία διαθέτει την δική της ξεχωριστή στοίβα πρωτοκόλλου επικοινωνίας. Αρχίζοντας από το κατώτερο στρώμα, αυτή απαρτίζεται από το λεγόμενο 'Physical layer' ή πιο σύντομα PHY το οποίο έχει ως κύριο έργο του την μεταφορά bit δεδομένων (που έχουν προκύψει από διαμόρφωση και πολυπλεξία) μεταξύ συσκευών και ραδιοσταθμών βάσης. Έπειτα, το αμέσως επόμενο στρώμα της στοίβας πρωτοκόλλου επικοινωνίας ονομάζεται 'Medium Access Control layer' ή πιο σύντομα MAC το οποίο διευθετεί την μεταφορά τόσο διαδικασιών signaling όσο και δεδομένων χρήστη, αφού διαχωρίζεται σε πολλαπλά κανάλια τα οποία στην συνέχεια πολυπλέκονται και στέλνονται στο 'Physical layer' για αποστολή. Επιπλέον προβαίνει σε διορθώσεις και επανεκπομπές των δεδομένων σε περίπτωση σφάλματος ή απώλειας. Το επόμενο στρώμα λέγεται 'Radio Link Control' ή πιο σύντομα RLC και παρέχει τους τρεις τρόπους αποστολής οι οποίοι είναι οι εξής: 1) Transparent mode: αποστολή πακέτων δεδομένων χωρίς επιβεβαίωση από τον δέκτη αν λήφθηκαν ή όχι τα πακέτα, 2) Unacknowledged mode: τα δεδομένα αποστέλλονται όπως στον Transparent mode με την διαφορά πως πριν από την αποστολή τα πακέτα δεδομένων χωρίζονται

σε τμήματα και συνενώνονται ξανά στην πλευρά του δέκτη και 3) Acknowledged mode: αποστολή πακέτων δεδομένων με επιβεβαίωση από τον δέκτη αν λήφθηκαν ή όχι τα πακέτα. Στην συνέχεια το στρώμα που ακολουθεί ονομάζεται 'Packet Data Convergence Protocol' ή πιο σύντομα PDCCP και κύριες λειτουργίες του είναι η κρυπτογράφηση δεδομένων και πληροφοριών signaling, η συμπίεση δεδομένων για βελτιστοποίηση της διαδικασίας αποστολής, καθώς και η δημιουργία αιτήματος σε περίπτωση που τα πακέτα δεδομένων έρθουν με λάθος σειρά. Τέλος, στο ανώτερο επίπεδο για τις διαδικασίες το στρώμα καλείται 'Radio Resource Control' ή αλλιώς σύντομα RRC και εξυπηρετεί διάφορες διαδικασίες signaling μεταξύ συσκευών και του NG-RAN, αλλά και μεταφοράς NAS μηνυμάτων μεταξύ του 5G Core Network και των συσκευών η οποία θεωρητικά γίνεται από την λογική διεπαφή N1 σύμφωνα με την αντίστοιχη βιβλιογραφία για την αρχιτεκτονική του 5G Core Network αλλά πρακτικά μέσω των σε σειρά διεπαφών N2 και NR αέρα.

Αντίθετα για την μεταφορά δεδομένων, το ανώτερο στρώμα ονομάζεται 'Service Data Adaption Protocol' ή αλλιώς σύντομα SDAP και οι κύριες λειτουργίες του είναι η χαρτογράφηση των διαφόρων πακέτων δεδομένων που λαμβάνονται ανάλογα με την σήμανση προτεραιότητας που έχουν για να εξυπηρετηθούν. Αντίστοιχα πρέπει να τοποθετηθεί σήμανσης προτεραιότητας στα πακέτα που πρόκειται να αποσταλούν μέσω της N3 διεπαφής στο UPF του 5G Core Network [2].



Εικόνα 5: Η διασύνδεση του 5G Core Network με τις συσκευές και το New Generation Radio Access Network (3GPP Radio Network) [1]

## 3 Συστήματα Παρακολούθησης και Χρονοσειρές Μετρικών

### 3.1 Εισαγωγή στα Συστήματα Παρακολούθησης

Οι σύγχρονες υπολογιστικές υποδομές ή συστήματα είναι αρκετά πολύπλοκα διότι αξιοποιούν τεχνολογίες όπως Εικονικοποίηση (Virtualization), Cloud native αρχιτεκτονική, Containerization, Μικροϋπηρεσίες (Microservices). Για την εύρυθμη λειτουργία των πολύπλοκων αυτών υποδομών είναι απαραίτητη πλέον η προσθήκη σε αυτές ενός συστήματος (λογισμικού) παρακολούθησης της υποδομής μέσω του οποίου συλλέγονται δεδομένα ώστε να επιτηρείται σφαιρικά αλλά και εκπαιδεύονται μοντέλα μηχανικής μάθησης για να προβλέπουν την κίνηση των δεδομένων και την κατανάλωση υπολογιστικών πόρων, των διάφορων συστατικών του συστήματος, όπως συσκευές και εφαρμογές. Έπειτα, πραγματοποιώντας επεξεργασία των δεδομένων που συλλέγονται και εφαρμόζοντας σε αυτά κάποια μοντέλα μηχανικής μάθησης είναι δυνατή η βελτιστοποίηση και η πρόβλεψη αντίστοιχα, της κατανάλωσης υπολογιστικών πόρων και της ροής των δεδομένων του υπολογιστικού συστήματος. Έτσι λοιπόν επειδή και η υπολογιστική υποδομή ενός δικτύου κινητής τηλεφωνίας 5G (5<sup>ης</sup> γενεάς) βασίζεται στις τεχνολογίες που προαναφέρθηκαν στην αρχή της παραγράφου, γεγονός που την κάνει σύνθετη, κρίνεται απαραίτητη η χρησιμοποίηση στην υποδομή αυτή ενός συστήματος παρακολούθησης για την βελτιστοποίηση και πρόβλεψη των διάφορων λειτουργιών της υποδομής του δικτύου κινητής τηλεφωνίας 5G. Έτσι λοιπόν, στην συνέχεια του κεφαλαίου αυτού αρχικά θα γίνει μια εισαγωγή στην έννοια της χρονοσειράς μιας μετρικής, τις κατηγορίες της καθώς και τρόπους ανάλυσης και αποθήκευσης της. Έπειτα θα ακολουθήσει μια σύντομη περιγραφή του συστήματος παρακολούθησης Prometheus, τον τρόπο που αυτό αποθηκεύει τις χρονοσειρές διάφορων μετρικών που συλλέγει και τέλος την βάση δεδομένων που τις αποθηκεύει [3].

### 3.2 Χρονοσειρές Μετρικών

#### 3.2.1 Ορισμός Χρονοσειρών Μετρικών

Χρονοσειρά ορίζεται μια ακολουθία δεδομένων μιας μετρήσιμης ποσότητας (μετρικής) ενός συστήματος, που έχουν ληφθεί σε συγκεκριμένες χρονικές στιγμές, οι οποίες συνήθως ισαπέχουν μεταξύ τους καθώς προχωρά ο χρόνος και ιεραρχούνται χρονολογικά.

Ακόμη, χρησιμοποιώντας ένα λίγο διαφορετικό ορισμό η χρονοσειρά μιας μετρικής εκφράζει την εξέλιξη στον χρόνο μιας μετρήσιμης ποσότητας κατά τη διάρκεια συνήθως ίσων διαδοχικών χρονικών περιόδων [4] [5].

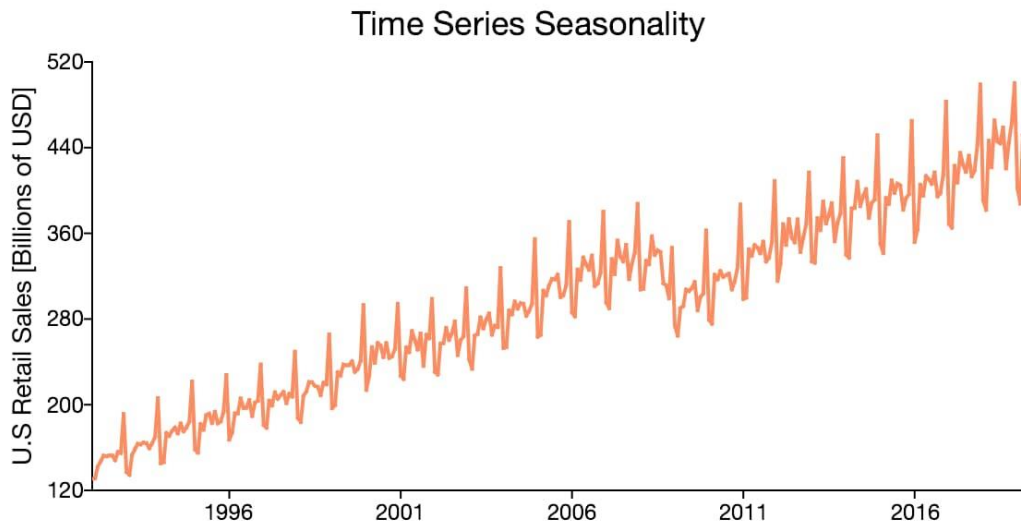
Η δομή της χρονοσειράς μιας μετρικής σε ένα γενικό πλαίσιο απαρτίζεται:

- 1) από μία σειρά αριθμητικών τιμών (value) δηλαδή των δεδομένων,
- 2) μία χρονική σήμανση (timestamp),
- 3) ένα όνομα (μετρική) για την χρονοσειρά (time series name or metric) και
- 4) ένα σύνολο ετικετών (labels) που χαρακτηρίζουν την εκάστοτε χρονοσειρά μιας μετρικής.

Παραδείγματα χρονοσειρών διάφορων μετρικών είναι τα εξής [6]:



- α) η ημερήσια θερμοκρασία,
- β) οι ωριαίες μετρήσεις του δείκτη του χρηματιστηρίου Αθηνών,
- γ) το ετήσιο ΑΕΠ,
- δ) η μηνιαία ανεργία.



Εικόνα 6: Παράδειγμα χρονοσειράς που αναπαριστά το αριθμό των ετήσιων πωλήσεων στην Ηνωμένες Πολιτείες της Αμερικής [3]

### 3.2.2 Κατηγορίες Χρονοσειρών Μετρικών

Ένας τρόπος διάκρισης των χρονοσειρών μετρικών είναι σύμφωνα με το εύρος τιμών που παίρνει ο χρόνος. Έτσι, λαμβάνοντας υπόψη την διάκριση αυτή, οι χρονοσειρές χωρίζονται σε δύο κατηγορίες: 1) χρονοσειρές συνεχούς χρόνου και 2) χρονοσειρές διακριτού χρόνου [4]:

- Χρονοσειρές συνεχούς χρόνου (continuous time series) ονομάζονται οι χρονοσειρές στις οποίες η μέτρηση των τιμών τους είναι δυνατή σε οποιαδήποτε χρονική στιγμή. Σαν παράδειγμα δύναται να θεωρηθεί η θερμοκρασία η οποία καταγράφεται οποιαδήποτε χρονική στιγμή σε μία μέρα.
- Χρονοσειρές διακριτού χρόνου (discrete time series) ονομάζονται οι χρονοσειρές στις οποίες οι μετρήσεις συλλέγονται σε συγκεκριμένα, διακριτά χρονικά διαστήματα που ισαπέχουν μεταξύ τους. Σαν παράδειγμα δύναται να θεωρηθεί η χρονοσειρά που λαμβάνεται από την τιμή των μετοχών στο χρηματιστήριο. Επιπλέον, όπως είναι λογικό, οι χρονοσειρές διακριτού χρόνου είναι εφικτό να προσομοιάσουν σε πολύ καλό βαθμό τις χρονοσειρές συνεχούς χρόνου αν το διάστημα δειγματοληψίας είναι αρκετό.

Επιπροσθέτως οι χρονοσειρές διακριτού χρόνου χωρίζονται και σε: 1) στιγμιαία καταγεγραμμένες και 2) συσσωρευμένες χρονοσειρές [4]:



- Στιγμιαία καταγεγραμμένες (instantaneously recorded time series) ονομάζονται οι διακριτού χρόνου χρονοσειρές στις οποίες υπάρχει η δυνατότητα να συλλεχθούν μετρήσεις σε οποιαδήποτε χρονική στιγμή παρόλο που η συλλογή των μετρήσεων πραγματοποιείται σε συγκεκριμένες χρονικές στιγμές. Σαν παράδειγμα δύναται να θεωρηθεί πάλι η τιμή των μετοχών στο χρηματιστήριο που είναι εφικτό να καταγραφεί οποιαδήποτε χρονική στιγμή σε μία μέρα, αλλά αντίθετα καταγράφεται δύο φορές μέσα στην μέρα.
- Συσσωρευμένες χρονοσειρές (accumulated time series) ονομάζονται οι διακριτού χρόνου χρονοσειρές στις οποίες δεν υπάρχει η δυνατότητα να συλλεχθούν μετρήσεις οποιαδήποτε χρονική στιγμή. Σαν παράδειγμα δύναται να θεωρηθεί η μόλυνση από έναν ιό στον ανθρώπινο οργανισμό και τα σεισμικά γεγονότα τα οποία μπορεί να πραγματοποιηθούν σε τυχαία χρονική στιγμή ωστόσο η μέτρησή τους πραγματοποιείται έμμεσα. Καταληκτικά οι μετρήσεις αθροίζονται ανά τακτά χρονικά διαστήματα.

### 3.2.3 Αποθήκευση Χρονοσειρών Μετρικών

Οι χρονοσειρές, όπως έγινε κατανοητό από τον ορισμό τους, παρατηρούν την εξέλιξη στον χρόνο μετρήσιμων ποσοτήτων (μετρικών) σε ένα σύστημα και γι' αυτό τον λόγο κάθε νέα μέτρηση της μετρήσιμης ποσότητας, καθώς προχωρά ο χρόνος, αποθηκεύεται στην βάση δεδομένων ως νέα εισαγωγή και όχι ως ενημέρωση της προηγούμενης χρονικά τιμής της μετρήσιμης ποσότητας, με αποτέλεσμα τα δεδομένα αυτά από την χρονοσειρά της μετρήσιμης ποσότητας (μετρικής) να αποτελούν μια πολύ σημαντική πηγή, μέσω της οποίας μπορούν να γίνουν αναλύσεις ώστε να διαγνωστούν προβλήματα και να βελτιστοποιηθούν παράμετροι. Με αυτό τον τρόπο μπορεί να πραγματοποιηθεί εκτεταμένη ανάλυση για το πώς άλλαξε μια μετρήσιμη ποσότητα στο παρελθόν, να επιτηρηθεί το πώς αλλάζει μια μετρήσιμη ποσότητα στο παρόν και να προβλεφθεί πώς η ίδια θα μεταβληθεί στο μέλλον. Ωστόσο, όσον αφορά την αποθήκευση των μετρήσεων των χρονοσειρών μετρικών, συναντάται το φαινόμενο ότι τα δεδομένα των χρονοσειρών συσσωρεύονται με πολύ ταχύ ρυθμό, με αποτέλεσμα να υπάρχουν δυσκολίες τόσο στην καταγραφή ή αποθήκευση, αφού δημιουργείται μεγάλος όγκος δεδομένων, αλλά και στην γρήγορη ανάκληση των δεδομένων χρονοσειρών για ανάλυση. Έτσι δημιουργήθηκαν οι βάσεις δεδομένων χρονοσειρών (Time Series Data Bases – TSDB) επειδή οι παραδοσιακές ή οι σχεσιακές ή οι NoSQL βάσεις δεδομένων δεν εξυπηρετούσαν ικανοποιητικά τις ανάγκες των χρηστών για την διαχείριση δεδομένων χρονοσειρών διαφόρων μετρικών. Οι βάσεις δεδομένων χρονοσειρών χρησιμοποιούνται επειδή έχουν τα εξής πλεονεκτήματα σε σχέση με τις υπόλοιπες βάσεις δεδομένων [3]:

- **Διαχείριση μεγάλου όγκου δεδομένων και γρήγορη ανάκληση δεδομένων**

Όπως προαναφέρθηκε, καθώς περνά ο χρόνος τα δεδομένα χρονοσειρών συσσωρεύονται γρήγορα, παράγοντας έτσι μεγάλο όγκο δεδομένων προς αποθήκευση. Ως γνωστόν οι παραδοσιακές, σχεσιακές και οι NoSQL βάσεις δεδομένων δεν έχουν σχεδιαστεί για την ταυτόχρονη διαχείριση τόσο μεγάλου όγκου δεδομένων και την γρήγορη ανάκληση

δεδομένων με αποτέλεσμα να μην δεν αποδίδουν ικανοποιητικά για στην διαχείριση χρονοσειρών.

Από την άλλη πλευρά, οι βάσεις δεδομένων χρονοσειρών διαχειρίζονται μεγάλους όγκους δεδομένων πολύ αποδοτικά, συμπιέζοντας με βέλτιστο τρόπο τα δεδομένα και δίνοντας την δυνατότητα για πολύ γρήγορη ανάκληση και επεξεργασία δεδομένων.

- **Πρόσθετα εργαλεία για την διευκόλυνση ανάλυσης δεδομένων**

Σε αντίθεση με τις υπόλοιπες βάσεις δεδομένων, οι βάσεις δεδομένων χρονοσειρών περιέχουν πρόσθετα έξυπνα εργαλεία και συναρτήσεις για την διευκόλυνση της ανάλυσης δεδομένων χρονοσειρών μετρικών διότι, ως γνωστόν, τα δεδομένα χρονοσειρών αυξάνουν συνεχώς και γρήγορα με τον χρόνο και ίσως δυσκολέψουν την επεξεργασία από τον χρήστη χωρίς αυτά τα έξυπνα εργαλεία και συναρτήσεις.

### 3.2.4 Ανάλυση Χρονοσειρών Μετρικών

Με την ανάλυση δεδομένων γενικά γίνεται έλεγχος, ξεσκαρτάρισμα, ταξινόμηση και χρήση μαθηματικών μεθόδων στα δεδομένα, έτσι ώστε να συγκεντρωθούν τα απαραίτητα στοιχεία μέσω των οποίων θα γίνει εφικτό να παρθεί κάποια απόφαση πάνω σε κάποιο ζήτημα. Αντίστοιχα, με την εφαρμογή μεθόδων μηχανικής μάθησης στις χρονοσειρές μετρικών επιδιώκεται η πρόβλεψη των μελλοντικών τιμών της χρονοσειράς μιας μετρικής, με βάση μόνο την πιο πρόσφατη και τις παρελθούσες τιμές αυτής καθαυτής της χρονοσειράς της μετρικής αυτής και αγνοώντας τυχόν άλλες μεταβλητές από τις οποίες μπορεί να επηρεάζεται ο σχηματισμός των τιμών της. Έτσι λοιπόν, σαν παράδειγμα σε ένα πείραμα, στα οποίο διαφοροποιούνται ορισμένες παράμετροι σε κάποια χρονικά σημεία και χρησιμοποιώντας χρονοσειρές για την καταγραφή δεδομένων διαφόρων μετρήσιμων ποσοτήτων (μετρικών), είναι δυνατό να βρεθούν οι παράγοντες που επηρεάζουν το σύστημα και το οδηγούν σε μια συγκεκριμένη κατεύθυνση όσον αφορά τις τιμές των δεδομένων των χρονοσειρών που μετρούνται [4].

### 3.2.5 Τύποι Ανάλυσης Χρονοσειρών Μετρικών

Επίσης επειδή σήμερα η ανάλυση χρονοσειρών μετρικών προσφέρει πλήθως πληροφοριών για την επίλυση και πρόβλεψη προβλημάτων καθώς και για τον έλεγχο οποιουδήποτε συστήματος έχουν αναπτυχθεί οι εξής μέθοδοι ή τύποι ανάλυσης χρονοσειρών μετρικών:

Περιγραφική ανάλυση: Μελετάται η τάση των τιμών μιας χρονοσειράς με την χρήση εργαλείων απεικόνισης και επεξεργασίας.

Εκπαιδευτική ανάλυση: Μελετάται η συσχέτιση δύο χρονοσειρών και αναζητάται μια σχέση της μεταξύ τους εξάρτησης.

Παραμβατική ανάλυση: Ελέγχει αν ένα γεγονός δύναται να προξενήσει μεταβολή στις τιμές της χρονοσειράς.

Προβλεπτική ανάλυση: Μέσω αυτής προβλέπονται οι μελλοντικές τιμές μιας χρονοσειράς με βάση την πιο πρόσφατη και τις παρελθούσες τιμές της ίδιας καθαυτής της χρονοσειράς εφαρμόζοντας μεθόδους μηχανικής μάθησης.

Φασματική ανάλυση: Μέσω αυτής κατηγοριοποιούνται οι περιοδικές συνιστώσες μιας χρονοσειράς [3] [4] [5].

### 3.3 Εισαγωγή στο Prometheus

Το Prometheus είναι ένα λογισμικό ανοιχτού κώδικα το οποίο παρακολουθεί κυρίως, ενώ έχει την δυνατότητα να στείλει ειδοποιήσεις αν ξεπεραστούν κάποια όρια από μεγέθη που επιτηρούνται με μετρήσεις αριθμητικών χρονοσειρών μετρικών (οι μετρήσεις αριθμητικών χρονοσειρών εξάγονται μέσω του μοντέλου HTTP pull) σε ένα υπολογιστικό μηχάνημα, app, service ή ακόμα μικροϋπηρεσίες οι οποίες πιθανώς εκτελούνται σε ένα σύστημα.

Ακόμη μέσω του Prometheus υπάρχει η δυνατότητα αποθήκευσης των μεγεθών που επιτηρούνται με μετρήσεις αριθμητικών χρονοσειρών μετρικών σε κάποια βάση δεδομένων και εν συνεχεία να πραγματοποιηθεί ανάλυση και επεξεργασία των δεδομένων μέσω κλασσικών και μηχανικής μάθησης μεθόδων για να εξαχθούν χρήσιμα συμπεράσματα τόσο για το παρόν όσο και για το μέλλον. Επίσης το Prometheus διαθέτει μια πανίσχυρη γλώσσα ερωτημάτων για την επεξεργασία των δεδομένων των χρονοσειρών διαφόρων μετρικών και επίσης δίνει την δυνατότητα διοχέτευσης των μετρήσεων αριθμητικών χρονοσειρών σε εργαλεία εικονικοποίησης για την καλύτερη κατανόηση και ανάλυση των δεδομένων. Οι σχεδιαστές του Prometheus για την δημιουργία του χρησιμοποίησαν την γλώσσα προγραμματισμού Go και η διάρθρωση που εφάρμοσαν για την υλοποίηση του λογισμικού του Prometheus καλείται δομοστοιχειωτή. Επιπλέον για την διευκόλυνση του χρήστη αναπτύχθηκαν κάποια εκτελέσιμα αρχεία, έτσι ώστε να απλοποιηθεί η όλη διαδικασία της εγκατάστασης. Εγκυκλοπαιδικά το Prometheus δημιουργήθηκε από την SoundCloud το 2012 ενώ το 2016 απορρίφθηκε από το Ίδρυμα Cloud Native Computing. Σήμερα χαίρει μεγάλης αναγνώρισης, με αποτέλεσμα την χρησιμοποίηση από πολλές εταιρείες, οργανισμούς, προγραμματιστές μέχρι και απλούς χρήστες [3].

#### 3.3.1 Αποθήκευση Δεδομένων Χρονοσειρών Μετρικών στο Prometheus

##### 3.3.1.1 Μοντέλο Αποθήκευσης Δεδομένων Χρονοσειρών Μετρικών στο Prometheus

Οι συλλεγόμενες μετρήσεις των μεγεθών ή μετρικών όπως λέγονται που παρακολουθούνται στα διάφορα apps, services από το Prometheus, αποθηκεύονται ουσιαστικά ως χρονοσειρές στην βάση δεδομένων που συνδέεται το Prometheus. Η γενική δομή της χρονοσειράς μιας μετρικής στην βάση δεδομένων είναι η εξής: 1) μία τιμή που ονομάζεται sample value και είναι τύπου float 64, 2) ένα σύνολο ετικετών ή σε αγγλική ορολογία labels ζευγαριών key-value που καθορίζουν μια μετρική (metric) του συστήματος, 3) μια χρονική σήμανση ή σε αγγλική ορολογία timestamp σύμφωνα με το μοντέλο unix και 4) ένα πεδίο όπου θα καθορίζεται ο μετρικός τύπος περιγραφής των μετρήσεων χρονοσειρών. Σχηματικά η χρονοσειρά μιας μετρικής έχει την εξής μορφή: <metric name>{<label name>=<label value>, ...}<timestamp><sample value>. Για παράδειγμα, επειδή στο Prometheus η συλλογή μετρήσεων χρονοσειρών διάφορων μετρικών από ένα τελικό σημείο (endpoint) που αντιστοιχεί στην θύρα (port) ενός app ή service ονομάζεται instance και συνήθως αντιστοιχεί σε μία διαδικασία (process), ενώ ένα σύνολο με instances που στοχεύουν στην παροχή της ίδιας υπηρεσίας (service) καλείται εργασία (job), το Prometheus για να διευκολύνει τον χρήστη να

προβάλλει τα δεδομένα του με μεγαλύτερη επιλεκτικότητα και οργάνωση δημιουργεί ετικέτες (labels) με το όνομα 'εργασία' και 'instance' [3].

## Prometheus Metrics

Metric name	Labels	Timestamp	Sample Value
...			
http_requests_total{status="200",method="GET"}		@1434317560938	94355
http_requests_total{status="200",method="GET"}		@1434317561287	94934
http_requests_total{status="200",method="GET"}		@1434317562344	96483
http_requests_total{status="404",method="GET"}		@1434317560938	38473
http_requests_total{status="404",method="GET"}		@1434317561249	38544
http_requests_total{status="404",method="GET"}		@1434317562588	38663
http_requests_total{status="200",method="POST"}		@1434317560885	4748
http_requests_total{status="200",method="POST"}		@1434317561483	4795
http_requests_total{status="200",method="POST"}		@1434317562589	4833
http_requests_total{status="404",method="POST"}		@1434317560939	122
...			

Εικόνα 7: Η γενική δομή μετρήσεων χρονοσειρών στη βάση δεδομένων που συνδέεται το Prometheus [2]

### 3.3.1.2 Μετρικοί Τύποι για την Περιγραφή Μετρήσεων Χρονοσειρών Μετρικών στο Prometheus

Οι μετρικοί τύποι περιγραφής των χρονοσειρών διαφόρων μετρικών (metrics) του Prometheus είναι συγκεκριμένα οι εξής [3]:

1. Counter, ο οποίος μετρά πόσες φορές συνέβη ένα γεγονός π.χ. αριθμός των αιτημάτων. Άρα αυξάνεται η τιμή του Counter μόνο όταν περνά ο χρόνος.
2. Gauge, ο οποίος μετρά την τιμή μιας παραμέτρου η οποία μπορεί να αυξομειώνεται ανά τακτά χρονικά διαστήματα.
3. Histogram, ο οποίος αθροίζει τις μετρήσεις ανά περιοχή τιμών ενώ υπάρχει η δυνατότητα να τις αθροίσει κάποιος συνολικά.
4. Summaries, ο οποίος έχει τις δυνατότητες του Histogram αλλά δίνει και ποσοστιαίες μετρήσεις.

### 3.3.1.3 Βάση Δεδομένων Χρονοσειρών Μετρικών InfluxDB

Το Prometheus από προεπιλογή, κατά την εγκατάστασή του, δημιουργεί σε ένα μέρος του τοπικού χώρου ή δίσκου αποθήκευσης, του υπολογιστικού συστήματος στο οποίο εγκαθίσταται, μια ειδική βάση δεδομένων χρονοσειρών (Time Series Data Bases – TSDB) που διατηρεί δεδομένα 15 ημερών. Για να αρθεί ο περιορισμός αυτός, το Prometheus διαθέτει μία διεπαφή που μπορεί να συνδεθεί με άλλες βάσεις δεδομένων έτσι ώστε να μπορεί απευθείας να αποθηκεύει, να διαχειρίζεται, να ανακτά και να ερωτά γρήγορα δεδομένων ή μετρήσεων χρονοσειρών μετρικών, απεριόριστου χρονικού διαστήματος, που θα λαμβάνει από τους διάφορους στόχους του. Μια βάση δεδομένων ανοιχτού κώδικα που συνδέεται μέσω αυτής

της διεπαφής είναι η InfluxDB είτε στην πρώτη (2012) είτε στην δεύτερη (2020) έκδοση της, η οποία προσφέρεται από την InfluxData τόσο για τοπική εγκατάσταση όσο και στο cloud. Η πρώτη έκδοση είναι γραμμένη στην γλώσσα προγραμματισμού Go ενώ η δεύτερη έκδοση είναι γραμμένη στην Go και την Rust. Η πρώτη έκδοση χρησιμοποιεί την ισχυρή InfluxQL γλώσσα ερωτημάτων που μοιάζει με την SQL, ενώ η δεύτερη χρησιμοποιεί τόσο την InfluxQL όσο και την Flux που είναι πιο ευέλικτη και πιο εκφραστική. Επίσης και οι δύο εκδόσεις προσφέρουν ένα γραφικό περιβάλλον για την διαχείριση της βάσης και την απεικόνιση των μετρήσεων σε ταμπλό. Επιπλέον προσφέρεται και μια διεπαφή έτσι ώστε να διαχειρίζεται η βάση μέσω της γλώσσας προγραμματισμού python, που είναι ευρέως διαδομένη, αλλά και άλλων γλωσσών. Όπως όλες οι σύγχρονες βάσεις δεδομένων χρονοσειρών (Time Series Data Bases – TSDB) έτσι και στην InfluxDB αποθηκεύονται τα παρακάτω στοιχεία για κάθε χρονοσειρά μετρικής [7] [8] [9]:

- Μία σειρά αριθμητικών τιμών (value ή αλλιώς field στην InfluxDB), δηλαδή των δεδομένων.
- Κάθε μία από τις αριθμητικές τιμές συνδυάζεται με μία χρονική σήμανση (timestamp).
- Ένα όνομα (name ή αλλιώς measurement στην InfluxDB) και ένα σύνολο ετικετών (labels ή αλλιώς tags στην InfluxDB) που χαρακτηρίζουν τις αριθμητικές τιμές (value ή field) οι οποίες ανήκουν σε μια συγκεκριμένη χρονοσειρά που αντιστοιχεί σε μια μετρική του συστήματος και ξεχωρίζουν από τις αριθμητικές τιμές (value ή field) που ανήκουν σε άλλες χρονοσειρές ή μετρικές που αντιστοιχούν σε άλλες μετρικές του συστήματος.

## 4 Μηχανική Μάθηση με Υλοποίηση σε Γλώσσα Προγραμματισμού Python

Η Μηχανική Μάθηση (Machine Learning) είναι ένα υποπεδίο του ευρύτερου πεδίου της Τεχνητής Νοημοσύνης (Artificial Intelligence). Τι είναι όμως η Τεχνητή Νοημοσύνη και τι είναι η Μηχανική Μάθηση:

- Τεχνητή Νοημοσύνη: είναι ο κλάδος της επιστήμης που ασχολείται με την κατασκευή έξυπνων υπολογιστικών προγραμμάτων και μηχανών τα οποία προσομοιώνουν την ανθρώπινη ευφυΐα. Μέσω αυτών θα μπορούν να αναλυθούν και να παρθούν αποφάσεις χωρίς ανθρώπινη παρέμβαση αλλά με ανθρώπινη λογική. Οι μέθοδοι που θα χρησιμοποιηθούν δεν προσομοιώνουν απαραίτητα τις βιολογικές μεθόδους που παρατηρούνται στον άνθρωπο.
- Μηχανική Μάθηση: είναι ο κλάδος της επιστήμης που εφευρίσκει μοντέλα ή αλγορίθμους ώστε τα υπολογιστικά συστήματα να λαμβάνουν αποφάσεις χωρίς να είναι απολύτως προγραμματισμένα. Τα μοντέλα αυτά αξιοποιούν κάποια δεδομένα που τους τροφοδοτούνται ώστε να εκπαιδευτούν αποκτώντας εμπειρία στο συγκεκριμένο τύπο δεδομένων, με αποτέλεσμα να μπορούν στην συνέχεια να χειρίζονται νέες καταστάσεις ή να παίρνουν αποφάσεις που σχετίζονται με το συγκεκριμένο τύπο δεδομένων προσομοιώνοντας την ανθρώπινη λογική.

Μια υποκατηγορία της Μηχανικής Μάθησης είναι η Βαθιά Μάθηση (Deep Learning) η οποία αξιοποιεί τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks). Πιο αναλυτικά:

- Τεχνητά Νευρωνικά Δίκτυα: περιέχουν υπολογιστικές μεθόδους ή μοντέλα που τα δεδομένα επεξεργάζονται με παρόμοιο τρόπο όπως ο ανθρώπινος εγκέφαλος, προσομοιώνοντας την βιολογική μέθοδο των νευρωνικών δικτύων που παρατηρούνται στον άνθρωπο. Τα δίκτυα αυτά αποτελούνται από τους λεγόμενους νευρώνες (neurons) ή κόμβους (nodes) οι οποίοι οργανώνονται σε στρώματα (layers) προσομοιώνοντας τους νευρώνες του ανθρώπινου Βιολογικού Νευρωνικού Δικτύου και τα βάρη που ενώνουν νευρώνες δύο διαφορετικών στρωμάτων και προσομοιώνουν τις συνάψεις του ανθρώπινου Βιολογικού Νευρωνικού Δικτύου. Μέσω των επεξεργασμένων δεδομένων που εξέρχονται από το δίκτυο μπορούν να εξαχθούν συμπεράσματα που θα οδηγούν στην λύση πολύπλοκων προβλημάτων και την λήψη κάποιας απόφασης.
- Βαθιά Μάθηση: περιέχει κάποια συγκεκριμένα μοντέλα ή αλγορίθμους που χρησιμοποιούν Τεχνητά Νευρωνικά Δίκτυα τα οποία μέσω μιας επαναληπτικής διαδικασίας «εκπαίδευσης» τροποποιούν τα Βάρη μεταξύ των νευρώνων διαφορετικών στρωμάτων, ώστε στο τέλος αυτής να βελτιστοποιηθούν τα Βάρη και καταληκτικά από την έξοδο του δικτύου να εξαχθούν τα επιθυμητά συμπεράσματα που οδηγούν στην λήψη αποφάσεων.

Μέσω της μηχανικής μάθησης τις τελευταίες δεκαετίες έχουν εφευρεθεί τα τελευταία χρόνια κάποιες αποτελεσματικές μέθοδοι για την κατηγοριοποίηση (classification), συσχέτιση (association), ομαδοποίηση (clustering) και παλινδρόμηση (regression) δεδομένων καθώς και την πρόβλεψη της εξέλιξης των χρονοσειρών. Η τελευταία κατηγορία μεθόδων για να εξαγάγει συμπεράσματα για την εξέλιξη μιας χρονοσειράς στον μέλλον αξιοποιεί δεδομένα χρονοσειρών που συλλέγονται μέσω συστημάτων παρακολούθησης [10]. Μια αξιοσημείωτη εφαρμογή της τελευταίας κατηγορίας είναι η πρόβλεψη της

εξέλιξης των χρονοσειρών διαφόρων μετρικών ενός δικτύου κινητής τηλεπικοινωνίας 5G (5<sup>ης</sup> γενεάς) που έμμεσα επιδεικνύουν την δικτυακή κίνηση.

Επιπλέον στην ανάπτυξη της μηχανικής μάθησης συνέβαλε ιδιαίτερα η γλώσσα προγραμματισμού *python*, η οποία διαθέτει διάφορα πακέτα και βιβλιοθήκες που διευκολύνουν και επιταχύνουν την κατασκευή και εφαρμογή των μοντέλων καθώς και την απεικόνιση των αποτελεσμάτων.

Στην συνέχεια του κεφαλαίου παρουσιάζονται αναλυτικά η μηχανική μάθηση, η βαθιά μάθηση, τα νευρωνικά δίκτυα καθώς και οι δυνατότητες της γλώσσας προγραμματισμού *python*.

#### 4.1 Ιστορική Αναδρομή στο Πεδίο της Τεχνητής Νοημοσύνης

Από τις αρχές της δεκαετίας του 1950 μέχρι τα τέλη της δεκαετίας του 1980 επικρατούσε η προσέγγιση της Συμβολικής Τεχνητής Νοημοσύνης στο πεδίο της Τεχνητής Νοημοσύνης. Η προσέγγιση αυτή είχε ταυτιστεί με την πεποίθηση πολλών ερευνητών της περιόδου εκείνης πως αν δοθεί ένας επαρκής αριθμός από σαφείς κανόνες στα υπολογιστικά συστήματα αυτά θα μπορούν να διαχειριστούν δεδομένα. Ωστόσο αποδείχθηκε πως κάτι τέτοιο ήταν αρκετό δύσκολο και εν τέλη ανέφικτο για την διαχείριση δεδομένων σε πολύπλοκα προβλήματα.

Έτσι στις αρχές της δεκαετίας του 1990 εμφανίστηκε στο πεδίο της Τεχνητής Νοημοσύνης η προσέγγιση της Μηχανικής Μάθησης, η οποία χρησιμοποίησε πολλές επιστημονικές μεθόδους όπως η στατιστική και σε συνδυασμό με τον μεγάλο αριθμό που πλέον υπήρχαν καθώς και την υπολογιστική ισχύ που συνεχώς αυξάνονταν έγινε γρήγορα η πιο επιτυχημένη προσέγγιση στον κλάδο της Τεχνητής Νοημοσύνης. Επιπλέον αναπτύχθηκαν πάρα πολλές εφαρμογές Μηχανικής Μάθησης οι οποίες ενσωματώθηκαν στην Βιομηχανία και επιτάχυναν την παραγωγικότητα.

Τέλος στις αρχές της δεκαετίας του 2000 εμφανίστηκε στο πεδίο της Τεχνητής Νοημοσύνης και υποκλάδος της Μηχανικής Μάθησης με το όνομα Βαθιά Μάθηση, ο οποίος έχει προσφέρει πάρα πολλές εφαρμογές σε διάφορους επιστημονικούς κλάδους, στην βιομηχανία αλλά και στην καθημερινότητα των απλών πολιτών [10].

#### 4.2 Μοντέλα Μηχανικής Μάθησης

Όπως προαναφέρθηκε ένα μοντέλο Μηχανικής Μάθησης χρησιμοποιεί κάποια δεδομένα συγκεκριμένου τύπου για να εκπαιδευτεί και στην συνέχεια αξιοποιώντας τις γνώσεις που έχει πάρει, να προβλέψει που μπορούν να κατηγοριοποιηθούν, ομαδοποιηθούν, συσχετιστούν τα δεδομένα που δίνονται στην είσοδο ή πως εξελίσσονται οι χρονοσειρές που δίνονται στην είσοδο. Πιο αναλυτικά τα πιο σημαντικά τα μοντέλα Μηχανικής Μάθησης διαχωρίζονται στις εξής κατηγορίες:

##### 1. **Supervised Machine Learning (Επιβλεπόμενη Μηχανική Μάθηση)**

Στην κατηγορία αυτή εκπαιδεύεται το μοντέλο εισάγοντας τα λεγόμενα δεδομένα με ετικέτα (*label*), τα οποία συνίστανται από ζευγάρια τιμών εισόδου και επιθυμητών τιμών εξόδου από το μοντέλο. Μετά την εκπαίδευση του μοντέλου εισάγονται σε αυτό δοκιμαστικές τιμές εισόδου ώστε να προβλέψει τις τιμές εξόδου και να αξιολογηθεί η αποτελεσματικότητά του. Αν είναι αποτελεσματικό χρησιμοποιείται αλλιώς δοκιμάζονται άλλα μοντέλα. Επίσης η συγκεκριμένη κατηγορία αποτελείται από τις ακόλουθες υποκατηγορίες:

##### a. **Classification (Κατηγοριοποίησης)**



Μέσω αυτής της υποκατηγορίας λύνονται προβλήματα όπου τα δεδομένα εισόδου πρέπει να κατηγοριοποιηθούν σε ένα πεπερασμένο σύνολο κατηγοριών ή κλάσεων όπως λέγονται

b. **Regression (Παλινδρόμησης)**

Μέσω αυτής της υποκατηγορίας λύνονται προβλήματα όπου τα δεδομένα εισόδου και εξόδου έχουν κάποιου είδους γραμμική εξάρτηση.

Μερικές εφαρμογές της κατηγορίας αυτής είναι η ιατρική διάγνωση, ανίχνευση εξαπάτησης, αναγνώριση φωνής και εικόνas καθώς και πρόβλεψη δεδομένων χρονοσειρών.

2. **Unsupervised Machine Learning (Μη Επιβλεπόμενη Μηχανική Μάθηση)**

Στην κατηγορία αυτή εκπαιδεύεται το μοντέλο εισάγοντας τα απλά δεδομένα χωρίς ετικέτα (label), τα οποία συνίστανται μόνο από τιμές εισόδου. Κατά την εκπαίδευση το μοντέλο ομαδοποιήσει ή κατηγοριοποιεί τις μη ταξινομημένες δεδομένα που του δίνονται σύμφωνα με τις ομοιότητες, τα μοτίβα και τις διαφορές μεταξύ αυτών. Μετά την εκπαίδευση του μοντέλου εισάγονται σε αυτό δοκιμαστικές τιμές εισόδου ώστε να προβλέψει την ομάδα ή κατηγορία που ανήκουν αυτές. Ακόμη η συγκεκριμένη κατηγορία αποτελείται και αυτή από τις ακόλουθες υποκατηγορίες:

a. **Clustering (Ομαδοποίηση)**

Μέσω αυτής της υποκατηγορίας λύνονται προβλήματα που απαιτείται να κατηγοριοποιηθούν τα δεδομένα ενός συμπλέγματος σε ομάδες. Τα δεδομένα κάθε ομάδας εμφανίζουν ομοιότητες μεταξύ τους.

b. **Association (Συσχέτιση)**

Μέσω αυτής της υποκατηγορίας λύνονται προβλήματα όπου χρειάζεται να βρεθούν συσχετίσεις και εξαρτήσεις μεταξύ των μεταβλητών των δεδομένων εισόδου.

Μερικές εφαρμογές της κατηγορίας αυτής είναι ο εντοπισμός σφαλμάτων, λογοκλοπής καθώς και την εξαγωγή πληροφορίας από βάσεις δεδομένων.

3. **Reinforcement Machine Learning (Ενισχυτική Μηχανική Μάθηση)**

Στην κατηγορία αυτή τα μοντέλα αξιοποιούν feedback-based διαδικασίες μέσω των οποίων λαμβάνουν κάποιες αποφάσεις ή προβαίνουν σε συγκεκριμένες ενέργειες για τις οποίες αξιολογούνται. Έτσι είτε «επιβραβεύονται» είτε «τιμωρούνται» για τις αποφάσεις τους, βελτιώνοντας με αυτό τον τρόπο της συμπεριφορά τους και ως επακόλουθο την απόδοσή τους. Στόχος των μοντέλων αυτών είναι η μεγιστοποίηση των «επιβραβεύσεων» τους γεγονός που σημαίνει πως το μοντέλο έχει εκπαιδευτεί πάρα πολύ καλά. Και αυτή κατηγορία συντελείται από τις εξής υποκατηγορίες:

a. **Positive Reinforcement Learning (Θετικά Ενισχυτική Μηχανική Μάθηση)**

Στην υποκατηγορία αυτή εισάγονται στα μοντέλα ερεθίσματα ή σήματα, τα οποία θα οδηγήσουν στην αύξηση της πιθανότητας επανάληψης μια επιθυμητής απόφασης ή ενέργειας του μοντέλου.

b. **Negative Reinforcement Learning (Αρνητικά Ενισχυτική Μηχανική Μάθηση)**

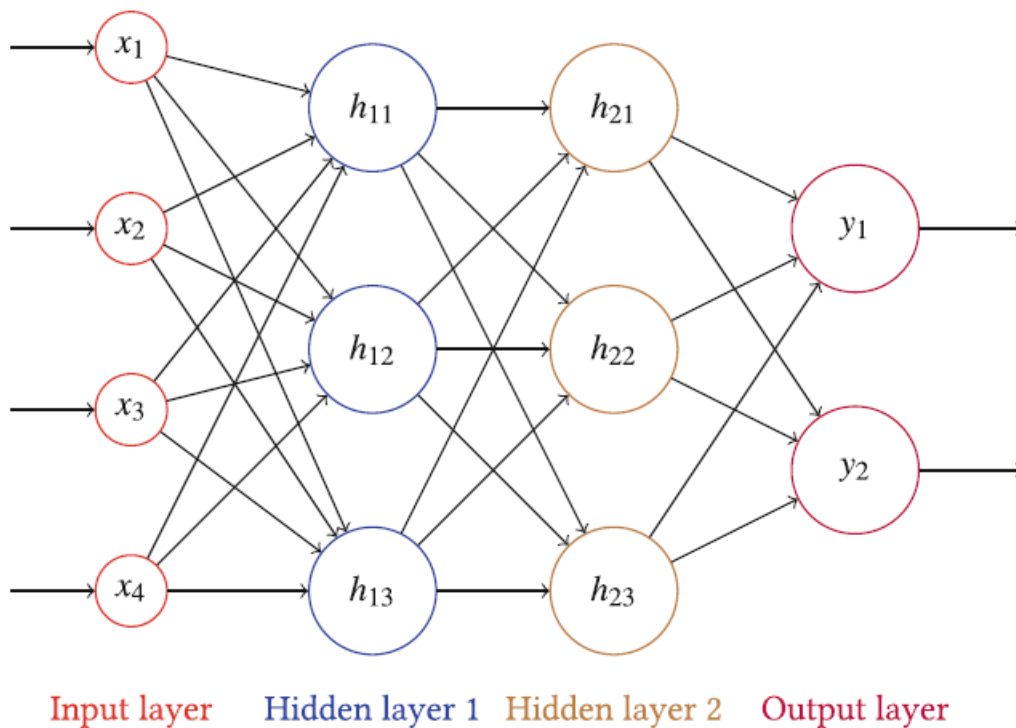


Στην υποκατηγορία αυτή αφαιρούνται από τα μοντέλα ερεθίσματα ή σήματα, τα οποία θα οδηγήσουν στην αύξηση της πιθανότητας επανάληψης μια επιθυμητής απόφασης ή ενέργειας του μοντέλου.

Μερικές εφαρμογές της κατηγορίας αυτής είναι τα ηλεκτρονικά παιχνίδια, η ρομποτική και η διαχείριση πόρων [11].

#### 4.3 Μοντέλα Τεχνητών Νευρωνικών Δικτύων

Η έννοια των Τεχνητών Νευρωνικών Δικτύων προέχεται από τους McCulloch και Pitts οι οποίοι παρουσίασαν ένα Τεχνητό Νευρωνικό Δίκτυο το οποίο μοιάζει σε μεγάλο βαθμό με το Βιολογικό Νευρωνικό Δίκτυο που υπάρχει στον ανθρώπινο οργανισμό, ωστόσο αποτελείται από πολύ πιο απλούς νευρώνες που ουσιαστικά αποτελούν τις επεξεργαστικές μονάδες του δικτύου και ονομάζονται αλλιώς κόμβοι. Οι νευρώνες σε ένα Τεχνητό Νευρωνικό Δίκτυο οργανώνονται σε στρώματα (layers) και πιο συγκεκριμένα ένα δίκτυο αποτελείται από το στρώμα εισόδου (input layer), το στρώμα εξόδου (output layer) καθώς και τα ενδιάμεσα στρώματα που ονομάζονται κρυφά στρώματα (hidden layers). Κάθε στρώμα μπορεί να περιέχει διαφορετικό αριθμό νευρών, όπως επίσης κάθε στρώμα μπορεί να αποτελείται νευρώνες με διαφορετικό μηχανισμό αλλά εντός του ίδιου στρώματος υπάρχουν νευρώνες του ίδιου μηχανισμού. Επίσης οι νευρώνες ή κόμβοι ενός στρώματος συνδέονται με τους νευρώνες του επόμενου και του προηγούμενου στρώματος μέσω των λεγόμενων Βαρών (weights) τα οποία αντιστοιχιστούν στις βιολογικές συνάψεις. Τα δεδομένα που εισέρχονται στο στρώμα εισόδου στην συνέχεια περνούν σειριακά από στρώμα σε στρώμα πραγματοποιώντας κατάλληλες αναπαραστάσεις των δεδομένων έτσι ώστε με βάση την έξοδο του τελευταίου στρώματος να στην ληφθούν οι κατάλληλες αποφάσεις. Τα δίκτυα αυτά είναι ανθεκτικά στον θόρυβο των δεδομένων εισόδου και μπορούν ακόμη να παράξουν ορθολογικό αποτέλεσμα ακόμα και με όχι τόσο καλής ποιότητας δεδομένα εισόδου. Καταληκτικά η λειτουργία ενός Τεχνητού Νευρωνικού Δικτύου εξαρτάται από την αρχιτεκτονική του, δηλαδή αριθμό νευρώνων και στρωμάτων καθώς και του μηχανισμού των νευρώνων και τις τιμές των βαρών [10].



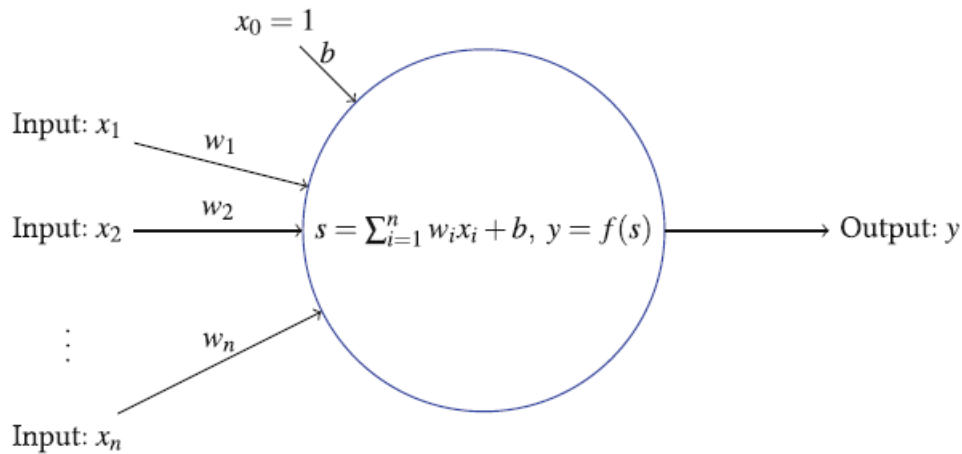
Εικόνα 8: Αρχιτεκτονική Τεχνητών Νευρωνικών Δικτύων

#### 4.3.1 Νευρώνες Τεχνητών Νευρωνικών Δικτύων

Υπάρχουν πολλά είδη νευρώνων, ωστόσο όλα αυτά τα είδη έχουν κάποια κοινά χαρακτηριστικά:

1. Το διάνυσμα του σήματος εισόδου του νευρώνα  $X = (x_1, x_2, \dots, x_n)$ .
2. Το διάνυσμα βάρους  $W = (w_1, w_2, \dots, w_n)$  σύμφωνα με το οποίο καθορίζεται κατά πόσο λαμβάνεται υπόψιν το σήμα εισόδου.
3. Η σταθερά  $b$  που υποδηλώνει bias.
4. Το σταθμισμένο άθροισμα  $s = \sum_{k=1}^n w_k x_k + b$  του σήματος εισόδου σύμφωνα με το διάνυσμα βάρους και την σταθερά  $b$ .
5. Η έξοδος  $y = f(s)$  του επεξεργασμένου σήματος μέσω της συνάρτησης  $f(s)$  που ονομάζεται συνάρτηση ενεργοποίησης (activation function) και ουσιαστικά είναι ο τρόπος που ο νευρώνας λαμβάνει μια απόφαση.

Οι πιο γνωστές συναρτήσεις ενεργοποίησης είναι η  $\text{ReLU}(s) = \max(s, 0)$  και η  $\tanh(s)$ . [10]



Εικόνα 9: Αρχιτεκτονική Νευρώνων Τεχνητών Νευρωνικών Δικτύων

#### 4.3.2 Τύποι Τεχνητών Νευρωνικών Δικτύων

Γενικότερα υπάρχουν πάρα πολλοί τύποι Τεχνητών Νευρωνικών Δικτύων που κύριο χαρακτηριστικό διαφοροποίησης τους, είναι η λειτουργικότητα και η αρχιτεκτονική τους όπως προαναφέρθηκε. Ενδεικτικά 3 από τους κυριότερους τύπους Τεχνητών Νευρωνικών Δικτύων είναι:

1. **Multi-layer Perceptron:** το οποίο αποτελείται από πολλούς νευρώνες και πολλά κρυφά στρώματα. Είναι ένα αμφίδρομο δίκτυο και χρησιμοποιεί τον συνδυασμό των αλγορίθμων Backpropagation και Gradient Descent για να εκπαιδευτεί. Επειδή όλοι οι νευρώνες ενός στρώματος συνδέονται με όλους τους νευρώνες του επόμενου στρώματος κάθε στρώμα του Multi-layer Perceptron ονομάζεται Dense. Τέτοιου τύπου δίκτυα χρησιμοποιούνται τόσο για προβλήματα κατηγοριοποίησης (classification), παλινδρόμησης (regression), αναγνώριση μοτίβων, φυσική επεξεργασία γλώσσας (Natural Language Processing) καθώς και στην πρόβλεψη εξέλιξης χρονοσειρών.
2. **Convolutional Neural Network:** το οποίο αποτελείται και αυτό από πολλούς νευρώνες και ανάλογα την σχεδιαστική επιλογή πολλά ή λίγα κρυφά στρώματα που διαφέρουν μεταξύ τους. Πιο συγκεκριμένα τα πρώτα στρώματα του Convolutional Neural Network (CNN) είναι μία συστοιχία από Convolutional και Pooling στρώματα. Το Convolutional στρώμα πραγματοποιεί την πράξη της συνέλιξης μεταξύ της εισόδου που δέχεται και του λεγόμενου kernel πίνακα ο οποίος ανανεώνεται μέσω των Backpropagation και Gradient Descent αλγορίθμων για καλύτερα αποτελέσματα. Το αποτέλεσμα της είναι ο λεγόμενος χάρτης (map). Οι χάρτες που παράγονται στην συνέχεια περνούν από το Pooling στρώμα το οποίο τους συσσωματώνει. Τα τελικά στρώματα του Convolutional Neural Network είναι συνήθως Dense στρώματα. Τέτοιου τύπου δίκτυα χρησιμοποιούνται τόσο για προβλήματα κατηγοριοποίησης (classification), αναγνώριση μοτίβων, φυσική επεξεργασία γλώσσας (Natural Language Processing) καθώς και στην πρόβλεψη εξέλιξης χρονοσειρών.

3. Recurrent Neural Network: το οποίο αποτελείται και αυτό από πολλούς νευρώνες και ανάλογα την σχεδιαστική επιλογή πολλά ή λίγα κρυφά στρώματα. Πιο συγκεκριμένα το πρώτο και τα τελευταία στρώμα του Recurrent Neural Network είναι συνήθως Dense στρώματα, ενώ τα κρυφά στρώματα είναι συνήθως RNN στρώματα τα οποία αποτελούνται από νευρώνες οι οποίοι αποθηκεύουν τις προηγούμενες εξόδους του νευρώνα και τις χρησιμοποιούν σε συνδυασμό με τις εισόδους που λαμβάνουν από νευρώνες του προηγούμενου επιπέδου για να παράξουν την καινούρια έξοδο. Ωστόσο, το Recurrent Neural Network έχει ένα κύριο μειονέκτημα που είναι αυτό του Gradient Vanishing, εξαιτίας του οποίου κατά την εκτέλεση του Backpropagation αλγορίθμου η κλίση μειώνεται σημαντικά με αποτέλεσμα να μην μπορεί να συνεισφέρει στην διαδικασία μάθησης του δικτύου. Παρόλα αυτά λύση σε αυτό το πρόβλημα είναι δύο άλλα Τεχνητά Νευρωνικά Δίκτυα που είναι ουσιαστικά υποκατηγορίες των Recurrent Neural Networks:
  - a. Long Short-Term Memory Network: το οποίο αντί για τα RNN στρώματα διαθέτει τα LSTM στρώματα που περιέχουν νευρώνες με λίγο διαφορετικό μηχανισμό σε σχέση με αυτούς των RNN.
  - b. Gated Recurrent Unit Network: το οποίο αντί για τα RNN ή LSTM στρώματα διαθέτει τα GRU στρώματα που περιέχουν νευρώνες με λίγο διαφορετικό μηχανισμό σε σχέση με αυτούς των RNN και των LSTM.

Τέτοιου τύπου δίκτυα χρησιμοποιούνται τόσο για προβλήματα, αναγνώριση μοτίβων, πρόβλεψη εξέλιξης χρονοσειρών καθώς και φυσική επεξεργασία γλώσσας (Natural Language Processing) όπως δημιουργία κειμένου (text generation) και μετάφραση κειμένου [6] [10] [12].

#### 4.4 Μοντέλα Βαθιάς Μάθησης

Τα μοντέλα Βαθιάς Μάθησης όπως τονίστηκε εκπαιδεύουν μοντέλα Τεχνητών Νευρωνικών Δικτύων μέσω μιας επαναληπτικής διαδικασίας βελτιστοποίησης των Βαρών μεταξύ των διαφορετικών στρωμάτων των Τεχνητών Νευρωνικών Δικτύων, έτσι ώστε το δίκτυο να εμφανίζει την επιθυμητή συμπεριφορά. Στα δίκτυα αυτά δεν υπάρχει κάποιος κανόνας που να προσδιορίζει των επιθυμητό αριθμό νευρώνων ή στρωμάτων ωστόσο συνήθως αποτελούνται από πολλά κρυφά στρώματα (hidden layers) και για αυτό τον λόγο ο κλάδος αυτός την Τεχνητής νοημοσύνης έχει λάβει το όνομα «Βαθιά Μάθηση». Επομένως, για να εφαρμοστεί ένα κατάλληλο Τεχνητό Νευρωνικό Δίκτυο, θα πρέπει να αφιερωθεί αρκετός χρόνος ώστε ελεγχθούν διαφορετικοί συνδυασμοί στρωμάτων και νευρώνων. Παρόλα αυτά θα πρέπει να τονιστεί πως όσο αυξάνονται οι νευρώνες και τα στρώματα τόσο θα αυξάνονται και τα Βάρη μεταξύ των νευρών διαφορετικών στρωμάτων που θα πρέπει να βελτιστοποιηθούν με αποτέλεσμα να χρειάζεται μεγαλύτερη υπολογιστική ισχύς για την πραγματοποίηση της διαδικασίας της βελτιστοποίησης των Βαρών αλλά και για την παραγωγή εξόδου από το μοντέλο.

Η μέθοδος που χρησιμοποιείται για την διαδικασία βελτιστοποίησης των Βαρών μεταξύ των διαφορετικών στρωμάτων των Τεχνητών Νευρωνικών Δικτύων απαρτίζεται από συνδυασμό των αλγορίθμων Backpropagation και Gradient Descent. Αρχικά τα δεδομένα που θα χρησιμοποιηθούν για την εκπαίδευση τροφοδοτούνται στο Τεχνητό Νευρωνικό Δίκτυο και δίνουν κάποια έξοδο. Στην συνέχεια μέσω του Backpropagation αλγορίθμου θα υπολογιστεί η κλίση της cost ή loss συνάρτησης (συνάρτηση

που μετρά το σφάλμα μεταξύ της πραγματικής τιμής και της τιμής εξόδου) ως προς τα Βάρη κάθε νευρώνα του στρώματος εξόδου. Έπειτα η διαδικασία επαναλαμβάνεται προς τα πίσω και στα υπόλοιπα στρώματα μέχρι το στρώμα εισόδου, με αποτέλεσμα να αναπαράγονται οι κλίσεις σε όλα τα στρώματα. Τέλος αναλαμβάνει ο Gradient Descent αλγόριθμος ο οποίος τροποποιεί τα βάρη κατά ένα μικρό ποσοστό της τιμής της κλίσης προς την αντίθετη κατεύθυνση των κλίσεων τους. Αυτή η διαδικασία μπορεί να επαναληφθεί πολλές φορές (εποχές) μέχρι το σφάλμα να μειωθεί, αρκεί να πραγματοποιηθεί προσεκτικά ώστε να μην συμβεί υπερπροσαρμογή του μοντέλου.

Επιπροσθέτως τα στρώματα των Τεχνητών Νευρωνικών Δικτύων έχουν κάποιες παραμέτρους, όπως ο αριθμός των νευρώνων (units), η συνάρτηση ενεργοποίησης των νευρώνων (activation function), εποχές (epochs), βελτιστοποιητής (optimizer – υπάρχουν διάφορες υλοποιήσεις του αλγορίθμου Gradient Descent), batch size (ο αριθμός δειγμάτων των δεδομένων που επεξεργασία μαζί σε ένα Forward και Backward πέρασμα κατά την εκπαίδευση) που ονομάζονται Υπερπαραμέτροι (Hyperparameters) και μέσω των διαφορετικών τιμών που μπορούν να λάβουν, διαφοροποιούν την συμπεριφορά του δικτύου καταλήγοντας είτε σε χειρότερα είτε σε καλύτερα αποτελέσματα. Έτσι λοιπόν υπάρχει άλλη μια διαδικασία που ονομάζεται Συντονισμός Παραμέτρων (Hyperparameter Tuning) μέσω της οποίας εκπαιδεύεται το δίκτυο με όλους τους διαφορετικούς συνδυασμούς των παραμέτρων αυτών που ορίζει ο χρήστης και ο συνδυασμός που θα δώσει τα καλύτερα αποτελέσματα χαρακτηρίζεται ως βέλτιστος και επιλέγεται για την υλοποίηση του δικτύου [6] [10].

## 4.5 Python

Όπως προαναφέρθηκε όσο αφορά το πεδίο της μηχανικής και της βαθιάς μάθησης, η γλώσσα προγραμματισμού Python διαθέτει κάποιες εξαιρετικές βιβλιοθήκες οι οποίες θα παρουσιαστούν στα επόμενα κεφάλαια και καθιστούν την προεπεξεργασία των δεδομένων, την υλοποίηση, εφαρμογή και αξιολόγηση των μοντέλων μηχανικής και της βαθιάς μάθησης ιδιαίτερα απλή και γρήγορη, με αποτέλεσμα αυτή την περίοδο να είναι η κυρίαρχη γλώσσα στα πεδία αυτά. [13] [14].

Αναλυτικότερα, η Python είναι μια ανοιχτού κώδικα αντικειμενοστραφής γλώσσα προγραμματισμού υψηλού επιπέδου η οποία διαθέτει τα εξής χαρακτηριστικά:

- ένα πανίσχυρο διερμηνέα
- υψηλού επιπέδου δομές δεδομένων
- απλή σύνταξη
- πολλές βιβλιοθήκες
- δυναμική σημασιολογία
- δυναμική πληκτρολόγηση
- δυναμική δέσμευση

Ως συνέπεια όλων των παραπάνω χαρακτηριστικών είναι η εύκολη μελέτη και υλοποίηση κώδικα με αποτέλεσμα να μειώνεται το κόστος συντήρησης των εφαρμογών αλλά ταυτόχρονα να αυξάνεται και η παραγωγικότητα σε αυτές. Έτσι λοιπόν η python χρησιμοποιείται σήμερα για πληθώρα εφαρμογών όπως:

- Διαδικτυακές εφαρμογές
- Αυτοματισμούς

- Καθημερινές εφαρμογές
- Ανάλυση και απεικόνιση δεδομένων
- Μηχανική και Βαθιά μάθηση

Μια εξαιρετική πλατφόρμα ανοιχτού κώδικα που επιτρέπει την επεξεργασία και εκτέλεση κώδικα στη python είναι αυτή με το όνομα Anaconda. Η πλατφόρμα αυτή προσφέρεται σε περιβάλλοντα Windows, Linux, macOS και στο Cloud. Ακόμη το Anaconda είναι συμβατό με Integrated Development Environments (IDEs) όπως το Jupyter και το Spyder, τα οποία έχει ενσωματώσει [15] [16].

#### 4.5.1 Pandas

Είναι μια βιβλιοθήκη ανοιχτού κώδικα της python που προσφέρει, εύχρηστες δομές δεδομένων, ορισμένα εργαλεία ανάλυσης δεδομένων έτσι ώστε αυτά να μπορούν επεξεργάζονται αμιγώς μέσω της python. Ένα μεγάλο πλεονέκτημα είναι πως έχει μεγάλη απήχηση και κοινότητα με αποτέλεσμα διαρκώς να βελτιώνεται και να συντηρείται [6].

#### 4.5.2 Matplotlib

Είναι μια βιβλιοθήκη ανοιχτού κώδικα της python που προσφέρει την δυνατότητα εύκολης παραγωγής ποικιλίας ποιοτικών διαγραμμάτων τόσο σε έντυπη όσο και σε διαδραστική μορφή. Πιο συγκεκριμένα τα διαγράμματα που διαθέτει είναι τα εξής [6]:

- Απλό γράφημα
- Ιστόγραμμα
- Διάγραμμα φασματικής πυκνότητας ισχύος
- Ραβδόγραμμα
- Διάγραμμα λαθών
- Διάγραμμα διασποράς

#### 4.5.3 Plotly

Είναι μια βιβλιοθήκη ανοιχτού κώδικα της python που προσφέρει πάνω από 40 μοναδικούς τύπους γραφημάτων που καλύπτουν ένα ευρύ φάσμα στατιστικών, οικονομικών, γεωγραφικών, επιστημονικών και τρισδιάστατων περιπτώσεων χρήσης. Έχει δημιουργηθεί αξιοποιώντας την βιβλιοθήκη Plotly της JavaScript γλώσσας προγραμματισμού, η οποία προσφέρει την δυνατότητα δημιουργίας όμορφων διαδραστικών απεικονίσεων [17].

#### 4.5.4 Numpy

Είναι μια βιβλιοθήκη ανοιχτού κώδικα της python που προσφέρει εξαιρετικές λειτουργικότητες για επιστημονικούς υπολογισμούς. Πιο συγκεκριμένα διαθέτει ένα πολυδιάστατο πίνακα με ένα πανίσχυρο σύνολο συναρτήσεων πάνω σε αυτόν οι οποίες πραγματοποιούν μετασχηματισμό Fourier καθώς και πράξεις γραμμικής άλγεβρας. Επίσης είναι εξοπλισμένη με εργαλεία για την ενσωμάτωση κώδικα από πιο χαμηλού επιπέδου γλώσσες [6].

#### 4.5.5 Datetime

Είναι μια βιβλιοθήκη ανοιχτού κώδικα της python που προσφέρει λειτουργικότητες για τον χειρισμό της ώρας και της ημερομηνίας. Πιο συγκεκριμένα διαθέτει αρκετές κλάσεις για την παρουσίαση, επεξεργασία και την μορφοποίηση της ώρας και της ημερομηνίας. Επίσης μια σχεδιαστική λεπτομέρεια αυτής της βιβλιοθήκης είναι πως η ώρα και η ημερομηνία αποθηκεύονται σε διαφορετικά αντικείμενα της python [18].

#### 4.5.6 Scikit-learn ή sklearn

Είναι μια βιβλιοθήκη ανοιχτού κώδικα της python που προσφέρει εργαλεία για την εξόρυξη, προεπεξεργασία, ανάλυση και απεικόνιση δεδομένων. Επίσης διαθέτει συναρτήσεις και κλάσεις για το δημιουργία μοντέλων μηχανικής και βαθιάς μάθησης. Έχει δημιουργηθεί αξιοποιώντας τις βιβλιοθήκες NumPy, SciPy και Matplotlib της python [6].

#### 4.5.6 Keras

Είναι μια βιβλιοθήκη ανοιχτού κώδικα της python που υποστηρίζει την δημιουργία Convolutional Neural Networks, Recurrent Neural Networks καθώς και την εκπαίδευσή τους μέσω μεθόδων βαθιάς μάθησης με εύκολο, γρήγορο και επεκτάσιμο τρόπο. Επίσης οι συναρτήσεις και οι κλάσεις της μπορούν να εκτελεστούν τόσο σε CPU όσο και σε GPU προσφέροντας μεγαλύτερη παραλληλοποίηση με αποτέλεσμα πιο γρήγορα αποτελέσματα και αύξηση της αποδοτικότητας. Έχει δημιουργηθεί αξιοποιώντας τις βιβλιοθήκες TensorFlow, CNTK και Theano της python [6].

#### 4.5.7 Optuna

Είναι μια βιβλιοθήκη ανοιχτού κώδικα της python που μέσω μιας διεπαφής προσφέρει κλάσεις και συναρτήσεις που πραγματοποιούν με εύκολο και γρήγορο τρόπο την βελτιστοποίηση των οριζόμενων από τον χρήστη υπερπαραμέτρων μοντέλων μηχανικής μάθησης. Οι υπερπαραμέτροι και το εύρος των τιμών που θα λάβουν αυτές κατά την διαδικασία της βελτιστοποίησης ορίζονται από τον χρήστη μέσω των συναρτήσεων της διεπαφής του Optuna συντονιστή. [19].

#### 4.5.8 InfluxDB

Είναι μια βιβλιοθήκη ανοιχτού κώδικα της python που προσφέρει μια διεπαφή έτσι ώστε ο χρήστης χρησιμοποιώντας μόνο την python να μπορεί να κάνει ερωτήματα, τροποποιεί, διαγράφει και εισάγει δεδομένα σε μία InfluxDB βάση δεδομένων [20].

#### 4.5.9 Anvil

Το Anvil είναι ένα framework μέσω του οποίου μπορεί να φτιαχτεί μία Γραφική Διεπαφή Χρήστη χρησιμοποιώντας μόνο python η οποία στην συνέχεια υπάρχει η δυνατότητα να δημοσιευθεί στο διαδίκτυο ως ιστοσελίδα. Το λογισμικό του Anvil παρέχεται για τοπική εγκατάσταση αλλά και στο cloud. Πιο αναλυτικά, το Anvil προσφέρει ένα drag-and-drop περιβάλλον για την δημιουργία της γραφικής διεπαφής του χρήστη, το οποίο διαθέτει γραφικά στοιχεία όπως drag-down λίστες, sliders, εικόνες, πεδία για την εισαγωγή κειμένου και διαγράμματα από διάφορες βιβλιοθήκες της python. Επίσης προσφέρεται μια ενσωματωμένη βάση δεδομένων αλλά υποστηρίζει και εξωτερικές βάσεις δεδομένων όπως η MySQL. Αφού πραγματοποιηθεί ο σχεδιασμός της Γραφικής Διεπαφής Χρήστη μπορεί να δημοσιευθεί στο διαδίκτυο ως ιστοσελίδα πολύ εύκολα ενώ ταυτόχρονα δίνεται η δυνατότητα στον χρήστη να περιορίσει επίσης με απλό τρόπο την ορατότητα της σελίδας στο διαδίκτυο. Τέλος το Anvil διαθέτει μια βιβλιοθήκη ανοιχτού κώδικα της python που ονομάζεται *anvil* μέσω της οποίας το Jupyter Notebook του πακέτου Anaconda συνδέεται με το Anvil και έτσι ανάλογα με την επιλογές σε γραφικά στοιχεία όπως drag-down λίστες, sliders θα εκτελούνται κάποιες python συναρτήσεις οι οποίες πιθανότατα επιστρέφουν εικόνες ή διαγράμματα ή κάποιο κείμενο, που μπορεί να τοποθετείται σε κάποιο γραφικό στοιχείο εικόνας ή πεδίου για την εισαγωγή κειμένου ή διαγράμματος αντίστοιχα [21] [22].



## 5 Ανάλυση και Αποτελέσματα Βελτιστοποίησης Μοντέλων Μηχανικής Μάθησης για την Πρόβλεψη Κίνησης Δικτύων 5<sup>ης</sup> Γενεάς

Στο κεφάλαιο αυτό θα παρουσιαστούν τα διάφορα λογισμικά που χρησιμοποιήθηκαν, ο κώδικας που υλοποιήθηκε για να δημιουργηθεί ένα περιβάλλον υλοποίησης και βελτιστοποίησης μοντέλων μηχανικής μάθησης ώστε να προβλεφθεί η κίνηση σε ένα δίκτυο 5ης γενεάς. Έπειτα θα αναλυθεί η ερευνητική διαδικασία βελτιστοποίησης μοντέλων μηχανικής μάθησης που χρησιμοποιήθηκαν ως προς την ακρίβεια πρόβλεψης της εξέλιξής τους στον χρόνο και θα παρατεθούν τα αποτελέσματα.

### 5.1 Περιβάλλον Υλοποίησης και Βελτιστοποίησης Μοντέλων Μηχανικής Μάθησης για την Πρόβλεψη Κίνησης Δικτύων 5<sup>ης</sup> Γενεάς

#### 5.1.1 Προεγκατεστημένα Λογισμικά

Για την δημιουργία του περιβάλλοντος υλοποίησης και βελτιστοποίησης μοντέλων ή ροών μηχανικής μάθησης έτσι ώστε να προβλεφθεί η κίνηση σε δίκτυα 5ης γενεάς πρέπει να χρησιμοποιηθεί ένα πλήρες Δίκτυο 5G (5<sup>ης</sup> Γενεάς). Για την δημιουργία ενός πλήρους Δικτύου 5G αξιοποιήθηκαν κάποια προεγκατεστημένα λογισμικά σε virtual machines ενός Private Cloud. Πιο συγκεκριμένα χρησιμοποιήθηκαν δύο virtual machines, τα οποία θα ονομαστούν στα πλαίσια αυτής της εργασίας ως ubuntu@free5gc-sm και ubuntu@ueransim, με IP διευθύνσεις 192.168.8.133 και 192.168.8.52 αντίστοιχα. Το ubuntu@free5gc-sm virtual machine διαθέτει εγκατεστημένο το λογισμικό ανοιχτού κώδικα free5GC το οποίο αναπαριστά ένα 5G Core Network ενώ το ubuntu@ueransim virtual machine διαθέτει εγκατεστημένο το λογισμικό ανοιχτού κώδικα UERANSIM το οποίο αναπαριστά ένα NG-RAN που έχει και την δυνατότητα δημιουργίας εικονικών συσκευών-συνδρομητών συνδεδεμένων σε αυτό. Τα δύο λογισμικά free5gcore και UERANSIM έχουν ρυθμιστεί κατάλληλα, ώστε να επικοινωνούν μεταξύ τους και να συνιστούν ένα πλήρες 5G Δίκτυο. Επιπλέον από το Private Cloud αξιοποιήθηκε άλλο ένα virtual machine, το οποίο στα πλαίσια αυτής της εργασίας θα ονομαστεί ως jim@jarmusch με IP διεύθυνση 192.168.7.2 και χρησιμοποιείται ως απλός εξυπηρετητής (server) για την αποστολή δικτυακών πακέτων προς τις εικονικές συσκευές-συνδρομητών που είναι συνδεδεμένες στο NG-RAN. Επιπροσθέτως σε ένα άλλο virtual machine του cloud αυτού είναι εγκατεστημένο το λογισμικό ανοιχτού κώδικα Prometheus, το οποίο μέσω των exporters που είναι εγκατεστημένοι στα προαναφερθέντα virtual machines συλλέγει μετρικές (metrics) αριθμητικών χρονοσειρών δικτυακών πόρων από το Δίκτυο 5G, οι οποίες με την σειρά τους διοχετεύονται σε μια βάση δεδομένων InfluxDB για μόνιμη αποθήκευση και έπειτα παροχή στα μοντέλα μηχανικής μάθησης. Να σημειωθεί πως και η βάση δεδομένων InfluxDB βρίσκεται σε ένα διαφορετικό virtual machine του cloud αλλά έχει ρυθμιστεί κατάλληλα ώστε να διασυνδέεται με το Prometheus. Τέλος για την απόκτηση πρόσβασης σε όλα αυτά τα virtual machine που βρίσκονται στο Private Cloud είναι απαραίτητη η σύνδεση στο αντίστοιχο VPN (Virtual Private Network).

##### 5.1.1.1 Δημιουργία ενός Πλήρους Δικτύου 5G (5<sup>ης</sup> Γενεάς) με μια Εικονική Συσκευή-Συνδρομητή

Για την δημιουργία του πλήρους δικτύου κινητής τηλεπικοινωνίας 5G αρχικά πραγματοποιείται σύνδεση στο αντίστοιχο VPN (Virtual Private Network), ώστε να είναι δυνατή η πρόσβαση στο Private Cloud.



Έπειτα πρέπει να ανοιχθεί ένα Ubuntu (Linux) Terminal είτε σε ένα Ubuntu περιβάλλον είτε σε Windows περιβάλλον μέσω Windows Subsystem Linux. Στα πλαίσια της παρούσας εργασίας που πραγματοποιήθηκε σε Windows περιβάλλον χρησιμοποιήθηκε Windows Subsystem Linux για να ανοιχθεί ένα Ubuntu (Linux) Terminal. Στο Ubuntu (Linux) Terminal αρχικά πρέπει γίνει σύνδεση με το ubuntu@free5gc-sm virtual machine μέσω της παρακάτω εντολής και στην συνέχεια να εισαχθεί ο κωδικός που θα ζητηθεί:

- `ssh ubuntu@192.168.8.133`

Έπειτα εκτελούνται οι παρακάτω εντολές για την καθοδήγηση στο directory με το όνομα /free5gc/ και την εκτέλεση του run.sh script, έτσι ώστε να δημιουργηθεί ένα 5G Core Network:

- `cd ~/free5gc`
- `./run.sh`

Επόμενο βήμα στην διαδικασία είναι να ανοιχθεί ένα άλλο Ubuntu (Linux) Terminal και να γίνει σύνδεση με το ubuntu@ueransim virtual machine μέσω της παρακάτω εντολής και στην συνέχεια να εισαχθεί ο κωδικός που θα ζητηθεί:

- `ssh ubuntu@192.168.8.52`

Στην συνέχεια εκτελείται η εντολή καθοδήγησης στο directory με το όνομα /UERANSIM/:

- `cd ~/UERANSIM`

και τέλος εκτελείται η εντολή που ακολουθεί, έτσι ώστε να δημιουργηθεί ένα NG-RAN:

- `build/nr-gnb -c config/free5gc-gnb.yaml`

Έτσι έχει δημιουργηθεί 5G Core Network και ένα NG-RAN τα οποία επικοινωνούν μεταξύ τους λόγω της ρύθμισης παραμέτρων που έχει προηγηθεί, άρα αυτό σημαίνει πώς υπάρχει πλέον διαθέσιμο δίκτυο κινητής τηλεπικοινωνίας 5G.

Επόμενο βήμα στην διαδικασία είναι να ανοιχθεί ένα άλλο Ubuntu (Linux) Terminal και να γίνει σύνδεση με το ubuntu@ueransim virtual machine μέσω της παρακάτω εντολής και στην συνέχεια να εισαχθεί ο κωδικός που θα ζητηθεί:

- `ssh ubuntu@192.168.8.52`

Στην συνέχεια εκτελείται η εντολή καθοδήγησης στο directory με το όνομα /UERANSIM/:

- `cd ~/UERANSIM`

και τέλος εκτελείται η εντολή που ακολουθεί, έτσι ώστε να δημιουργηθεί εικονική συσκευή-συνδρομητής η οποία θα είναι συνδεδεμένη στο NG-RAN και κατά επέκταση στο δίκτυο κινητής τηλεπικοινωνίας 5G:

- `build/nr-ue -c config/free5gc-ue.yaml`

#### 5.1.1.2 Παραγωγή Δικτυακής κίνησης Μεταξύ Εικονική Συσκευή-Συνδρομητή και ενός Εξυπηρετητή

Για τις ανάγκες της εργασίας πρέπει να παραχθεί δικτυακή κίνηση η οποία θα οδηγήσει στην μεταβολή των υπολογιστικών ή δικτυακών πόρων στο virtual machine με το όνομα `ubuntu@free5gc-sm`, όπου είναι ενεργό το 5G Core Network (μέσω του λογισμικού `free5GC`) καθώς και στο virtual machine με το όνομα `ubuntu@ueransim`, όπου είναι ενεργό ένα NG-RAN και μία εικονική συσκευή-συνδρομητής (μέσω του λογισμικού `UERANSIM`). Η δικτυακή αυτή κίνηση για να υλοποιηθεί, θα σταλούν δικτυακά πακέτα από ένα εξυπηρετητή (server) σε μια εικονική συσκευή-συνδρομητής με σταθερό ρυθμό. Αρχικά ωστόσο πρέπει να γίνει να ανοιχθεί ένα άλλο Ubuntu (Linux) Terminal και να γίνει σύνδεση με το `jim@jarmusch` virtual machine του εξυπηρετητή μέσω της παρακάτω εντολής και στην συνέχεια να εισαχθεί ο κωδικός που θα ζητηθεί:

- `ssh jim@192.168.7.2`

Στην συνέχεια εκτελείται η εντολή δημιουργίας ενός iPerf εξυπηρετητή (server) ο οποίος υπακούει στην θυρίδα (port) 5201:

- `iperf3 -s`

Επόμενο βήμα στην διαδικασία είναι να ανοιχθεί ένα άλλο Ubuntu (Linux) Terminal και να γίνει σύνδεση με το `ubuntu@ueransim` virtual machine μέσω της παρακάτω εντολής και στην συνέχεια να εισαχθεί ο κωδικός που θα ζητηθεί:

- `ssh ubuntu@192.168.8.52`

Αμέσως μετά εκτελείται το `iperf3_test.sh` script, το οποίο είναι γραμμένο σε γλώσσα προγραμματισμού bash και πυροδοτεί δικτυακή κίνηση με την αποστολή πακέτων από την εξυπηρετητή (server) προς την εικονική συσκευή-συνδρομητή με σταθερό ρυθμό:

- `./iperf3_test.sh`

Το `iperf3_test.sh` script παράγει δικτυακή κίνηση στέλνοντας δικτυακά πακέτα από τον εξυπηρετητή (server) προς την εικονική συσκευή-συνδρομητή για περίπου δύο ώρες.

Καταληκτικά έχει δημιουργηθεί στα δύο virtual machines ένα δίκτυο κινητής τηλεπικοινωνίας 5G στο οποίο είναι συνδεδεμένες μία εικονική συσκευή-συνδρομητής, ενώ ταυτόχρονα οποιαδήποτε μεταβολή συμβαίνει στην χρήση των υπολογιστικών ή δικτυακών πόρων, εξαιτίας της δικτυακής κίνησης μεταξύ του εξυπηρετητή (server) και της εικονική συσκευή-συνδρομητής, στο virtual machine με το όνομα `ubuntu@free5gc-sm`, όπου είναι ενεργό ένα 5G Core Network (μέσω του λογισμικού `free5GC`) καθώς και στο virtual machine με το όνομα `ubuntu@ueransim`, όπου είναι ενεργό ένα NG-RAN και μία εικονική συσκευή-συνδρομητής (μέσω του λογισμικού `UERANSIM`), επιτηρείται μέσω μετρικών δεδομένων χρονοσειρών που συλλέγονται από το εργαλείο παρακολούθησης Prometheus το οποίο βρίσκεται σε άλλο virtual machine του Private Cloud. Επίσης το Prometheus αποθηκεύει τις μετρικές δεδομένων χρονοσειρών που συλλέγει στην InfluxDB βάση δεδομένων που όπως προαναφέρθηκε και αυτή βρίσκεται σε άλλο virtual machine του Private Cloud.

### 5.1.2 Υλοποίηση Κώδικα

Στην συνέχεια για την ολοκλήρωση του περιβάλλοντος υλοποίησης και βελτιστοποίησης μοντέλων μηχανικής μάθησης για την πρόβλεψη κίνησης σε δίκτυα 5ης γενεάς υλοποιήθηκε κώδικας σε γλώσσα προγραμματισμού python μέσω του οποίου συλλέγονται τα δεδομένα χρονοσειρών δύο μετρικών του 5G Δικτύου από την βάση δεδομένων InfluxDB, γίνεται προεπεξεργασία των δεδομένα χρονοσειρών πριν την εισαγωγή τους σε κάποιο από τα μοντέλα μηχανικής μάθησης, διαχωρισμός των δεδομένων χρονοσειρών σε σετ εκπαίδευσης (train set), επαλήθευσης (validation set), δοκιμής(test set) και τέλος πραγματοποιείται η διαδικασία της βελτιστοποίησης των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης ώστε να παραχθεί το βέλτιστο (μοντέλο) ως προς την ακρίβεια πρόβλεψης της εξέλιξής στον χρόνο των διάφορων μετρικών (metrics) του Δικτύου 5G. Τέλος, παράγονται διαγράμμάτων που περιέχουν τα δεδομένα εκπαίδευσης, επαλήθευσης, δοκιμής και πρόβλεψης του μοντέλου μηχανικής μάθησης με τις βέλτιστες υπερπαραμέτρους. Να τονιστεί πως τα μοντέλα μηχανικής μάθησης που θα χρησιμοποιηθούν στα πλαίσια της παρούσας μεταπτυχιακής εργασίας ανήκουν στη κατηγορία της επιβλεπόμενης μηχανικής μάθησης. Ακόμη οι βιβλιοθήκες της γλώσσας προγραμματισμού python που χρησιμοποιήθηκαν φαίνονται στην εικόνα που ακολουθεί.

```
import datetime
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import plotly.tools as tls
import plotly.graph_objs as go
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense, LSTM, TimeDistributed, Conv1D, MaxPooling1D, Flatten
from sklearn.preprocessing import MinMaxScaler
from influxdb_client import InfluxDBClient
import optuna
import anvil.server
import anvil.media
```

Εικόνα 10: Βιβλιοθήκες της γλώσσας προγραμματισμού Python που χρησιμοποιήθηκαν

#### 5.1.2.1 Συνάρτηση Σύνδεση με την InfluxDB Βάση Δεδομένων και Ανάκτηση δεδομένων

Στην εικόνα που ακολουθεί, παρουσιάζεται ο κώδικας της συνάρτησης *get\_data* σε python που έχει υλοποιηθεί σε Jupyter Notebook και πραγματοποιεί την σύνδεση με την InfluxDB βάση δεδομένων για την ανάκτηση δεδομένων χρονοσειρών διάφορων μετρικών από το Δίκτυο 5G. [20] Πιο συγκεκριμένα η συνάρτηση αυτή έχει ως όρισμα τη μεταβλητή *range\_fetch\_data* μέσω της οποίας καθορίζει μέχρι πόσα λεπτά πίσω στον χρόνο από τωρινή χρονική στιγμή θα συλλεχθούν δεδομένα χρονοσειρών. Επίσης μέσω της συνάρτησης ορίζονται οι μετρικές (metrics) του Δικτύου 5G που είναι επιθυμητές να συλλεχθούν από την βάση δεδομένων. Αυτές οι μετρικές είναι οι *node\_cpu\_seconds\_total* και *node\_network\_receive\_bytes\_total* που αντιστοιχούν στο συνολικό χρόνο που καταναλώνει ο επεξεργαστής (CPU) για interrupts λόγω του Δικτύου 5G και στα συνολικά bytes που λαμβάνονται στο το UPF από τους σταθμούς βάσης (gNB). Να τονιστεί πως οι μετρικές αυτές επιδεικνύουν έμμεσα την κίνηση του δικτύου 5G. Τέλος η συνάρτηση αρχικά αποθηκεύει τα δεδομένα χρονοσειρών από τη βάση δεδομένων σε μια μεταβλητή με DataFrame μορφή ενώ στην έξοδο της, τα επιστρέφει σε μια μεταβλητή με JSON μορφή.

```

@anvil.server.callable
def get_data(range_fetch_data):
    # Connection details
    url = 'http://192.168.7.90:8086'
    token = '7cdJZe84FUVh8bWKZ0uAV6VH1vcw5k4PM8Pvv_5EpsScmhU_A5ZaXSwvIp4vgWb6m222k6qghwT18IOCZDVxFw=='
    org = 'UoA'
    start_time = '-' + range_fetch_data + 'm'
    # Connect to the InfluxDB instance
    client = InfluxDBClient(url=url, token=token, org=org)
    query_api = client.query_api()
    query = 'from(bucket: "telegraf")\
    |> range(start: ' + start_time + ', stop: now())\
    |> filter(fn: (r) => r._measurement == "node_cpu_seconds_total")\
    |> filter(fn: (r) => r._field == "counter")\
    |> filter(fn: (r) => r.url == "http://192.168.8.133:9100/metrics")\
    |> filter(fn: (r) => r.mode == "softirq")\
    |> derivative(unit: 1s, nonNegative: false)\
    |> keep(columns: ["_time", "_measurement", "_value"])\
    |> rename(columns: {value: "load1", time: "Time"})\
    |> pivot(rowKey: ["Time"], columnKey: ["_measurement"], valueColumn: "load1")'
    result = query_api.query_data_frame(org=org, query=query)
    result['Time'] = result['Time'].astype(str)
    result['Time'] = result['Time'].str.split('+').str[0]
    result['Time'] = pd.to_datetime(result['Time'])
    result.index = result['Time']
    result = result.drop(columns = ['result', 'table', 'Time'])
    query_1 = 'from(bucket: "telegraf")\
    |> range(start: ' + start_time + ', stop: now())\
    |> filter(fn: (r) => r._measurement == "node_network_receive_bytes_total")\
    |> filter(fn: (r) => r._field == "counter")\
    |> filter(fn: (r) => r.url == "http://192.168.8.133:9100/metrics")\
    |> filter(fn: (r) => r.device == "upfgrtp")\
    |> derivative(unit: 1s, nonNegative: false)\
    |> keep(columns: ["_time", "_measurement", "_value"])\
    |> rename(columns: {value: "load1", time: "Time"})\
    |> pivot(rowKey: ["Time"], columnKey: ["_measurement"], valueColumn: "load1")'
    result_1 = query_api.query_data_frame(org=org, query=query_1)
    result_1['Time'] = result_1['Time'].astype(str)
    result_1['Time'] = result_1['Time'].str.split('+').str[0]
    result_1['Time'] = pd.to_datetime(result_1['Time'])
    result_1.index = result_1['Time']
    result_1 = result_1.drop(columns = ['result', 'table', 'Time'])
    result['node_network_receive_bytes_total'] = result_1['node_network_receive_bytes_total']
    result = result.to_json(orient='index')
    client.close()
    return result

```

Εικόνα 11: Συνάρτηση ανάκτησης των δύο μετρικών από το 5G δίκτυο

### 5.1.2.2 Συνάρτηση Βελτιστοποίησης των Υπερπαραμέτρων Μοντέλου Μηχανικής Μάθησης

Έπειτα, παρουσιάζεται ο κώδικας της συνάρτησης *objective* σε *rython* που έχει υλοποιηθεί σε *Jupyter Notebook* και πραγματοποιεί την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης που έχει επιλεγεί μέσω του ορίσματος *prediction\_model*, έτσι ώστε να επιτευχθεί όσο το δυνατό μεγαλύτερη ακρίβεια πρόβλεψης. Αναλυτικότερα, η συνάρτηση έχει και κάποια επιπλέον ορίσματα όπως:

- *X\_train, y\_train*: Μέσω των οποίων εισάγονται στο μοντέλο μηχανικής μάθησης το σετ εκπαίδευσης
- *X\_valid, y\_valid*: Μέσω των οποίων εισάγονται στο μοντέλο μηχανικής μάθησης το σετ επαλήθευσης
- *T*: Μέσω του οποίου ορίζεται η διάσταση του διανύσματος δεδομένων εισόδου στο μοντέλο μηχανικής μάθησης
- *HORIZON*: Μέσω του οποίου ορίζεται πόσα μελλοντικά χρονικά βήματα της χρονοσειράς θα προβλέψει το μοντέλο μηχανικής μάθησης δεδομένου ενός διανύσματος εισόδου

- *trail*: Μέσω του οποίου εισάγεται ο συντονιστής Optuna (ο Optuna ορίζεται στην συνάρτηση *time\_series\_prediction\_with\_data\_new* που παρουσιάζεται στην συνέχεια), ο οποίος πραγματοποιεί την βελτιστοποίηση των υπερπαραμέτρων αξιοποιώντας αυτή την συνάρτηση.

Στην εικόνα που ακολουθεί παρουσιάζονται οι υπερπαραμέτροι που έχουν οριστεί και οι τιμές που είναι δυνατό να πάρουν. Το σύνολο των συνδυασμών των υπερπαραμέτρων είναι 108. Ο συντονιστής Optuna κάθε φορά που καλεί αυτή την συνάρτηση ορίζει ένα από αυτούς τους συνδυασμούς. [19]

```
@anvil.server.callable
def objective(trial, prediction_model, X_train, y_train, X_valid, y_valid, T, HORIZON):
    # Define the search space for hyperparameters
    units = trial.suggest_categorical('units', [50, 100, 200])
    activation = trial.suggest_categorical('activation', ['relu', 'tanh'])
    epochs = trial.suggest_categorical('epochs', [100, 200])
    optimizer = trial.suggest_categorical('optimizer', ['adam', 'sgd', 'RMSprop'])
    batch_size = trial.suggest_categorical('batch_size', [32, 64, 128])
```

Εικόνα 12: Η αρχή της *Objective* συνάρτησης που ορίζονται οι υπερπαραμέτροι και οι δυνατές τους τιμές

Στην επόμενη εικόνα μπορεί να παρατηρηθεί ο τρόπος που δημιουργείται, εκπαιδεύεται και επαληθεύεται το επιλεγμένο μοντέλο μέσω της τιμής του ορίσματος *prediction\_model*. Οι τιμές που μπορεί να πάρει το όρισμα *prediction\_model* που ορίζει το μοντέλο μηχανικής μάθησης είναι GRU, LSTM, CNN, MLP και αντιστοιχούν στα τεχνητά νευρωνικά δίκτυα Gated Recurrent Unit Network, Long Short-Term Memory Network, Convolutional Neural Network και Multi-layer Perceptron αντίστοιχα. Επίσης είναι εμφανές πως η συνάρτηση ουσιαστικά επιστρέφει την τιμή της *loss* συνάρτησης, η οποία αξιοποιείται από τον συντονιστή Optuna (ο οποίος ορίζεται στην συνάρτηση *time\_series\_prediction\_with\_data\_new* που παρουσιάζεται στην συνέχεια), για να αξιολογήσει και εν τέλει να επιλέξει στο τέλος τις βέλτιστες υπερπαραμέτρους για το συγκεκριμένο μοντέλο μηχανικής μάθησης.

```
model = Sequential()
if prediction_model == 'GRU':
    model.add(GRU(units = units, input_shape = (T, 1), activation = activation))
elif prediction_model == 'LSTM':
    model.add(LSTM(units = units, input_shape = (T, 1), activation = activation))
elif prediction_model == 'CNN':
    model.add(Conv1D(64, 1, activation=activation, input_shape=(T, 1)))
    model.add(MaxPooling1D())
    model.add(Flatten())
    model.add(Dense(units = units, activation=activation))
elif prediction_model == 'MLP':
    model.add(Dense(units = units, activation=activation, input_dim=T))
model.add(Dense(int(HORIZON)))
model.compile(optimizer = optimizer, loss = 'mse', metrics=['accuracy'])
model.summary()
model_fit = model.fit(X_train, y_train, batch_size = batch_size, epochs = epochs, validation_data = (X_valid, y_valid), verbose=0)
# Evaluate the model accuracy on the validation set.
score = model_fit.history['val_loss'][-1]
return score
```

Εικόνα 13: Η υπόλοιπη της *Objective* συνάρτησης

### 5.1.2.3 Συνάρτηση Δημιουργίας του Βέλτιστου Μοντέλου Μηχανικής Μάθησης

Εν συνεχεία, παρουσιάζεται ο κώδικας της συνάρτησης *model\_optimized* σε *python* που έχει υλοποιηθεί σε *Jupyter Notebook* και δημιουργεί το μοντέλο μηχανικής μάθησης με τις βέλτιστες υπερπαραμέτρους που δέχεται ουσιαστικά από τον συντονιστή Optuna [19] (ο οποίος ορίζεται στην συνάρτηση *time\_series\_prediction\_with\_data\_new* η οποία παρουσιάζεται στην συνέχεια), αφού αυτός ανακαλύπτει τις

βέλτιστες παραμέτρους αξιοποιώντας την συνάρτηση *objective* και έπειτα τις διοχετεύει στην συνάρτηση *model\_optimized*. Αναλυτικότερα, η συνάρτηση έχει ως επιπλέον ορίσματα:

- *prediction\_model*: Μέσω του οποίου ορίζεται το μοντέλο μηχανικής μάθησης που θα δημιουργηθεί
- *units, activation, epochs, optimizer, batch\_size*: Μέσω των οποίων εισάγονται στο μοντέλο μηχανικής μάθησης οι βέλτιστες υπερπαραμέτροι για την δημιουργία του. Οι παράμετροι αυτές είναι: οι μονάδες των Dense στρωμάτων του τεχνητού νευρωνικού δικτύου (*units*), συνάρτηση ενεργοποίησης (*activation function*), εποχές (*epochs*), Βελτιστοποιητής (*optimizer*), *batch\_size* αντίστοιχα.
- *X\_train, y\_train*: Μέσω των οποίων εισάγονται στο μοντέλο μηχανικής μάθησης το σετ εκπαίδευσης
- *X\_valid, y\_valid*: Μέσω των οποίων εισάγονται στο μοντέλο μηχανικής μάθησης το σετ επαλήθευσης
- *T*: Μέσω του οποίου ορίζεται η διάσταση του διανύσματος δεδομένων εισόδου στο μοντέλο μηχανικής μάθησης
- *HORIZON*: Μέσω του οποίου ορίζεται πόσα μελλοντικά χρονικά βήματα της χρονοσειράς θα προβλέψει το μοντέλο μηχανικής μάθησης δεδομένου ενός διανύσματος εισόδου

Στην επόμενη εικόνα μπορεί να παρατηρηθεί ο τρόπος που δημιουργείται, εκπαιδεύεται και επαληθεύεται το επιλεγμένο μοντέλο μέσω της τιμής του ορίσματος *prediction\_model*. Οι τιμές που μπορεί να πάρει το όρισμα *prediction\_model* που ορίζει το μοντέλο μηχανικής μάθησης, είναι GRU, LSTM, CNN, MLP και αντιστοιχούν στα τεχνητά νευρωνικά δίκτυα Gated Recurrent Unit Network, Long Short-Term Memory Network, Convolutional Neural Network και Multi-layer Perceptron αντίστοιχα. Επίσης είναι εμφανές πως η συνάρτηση ουσιαστικά επιστρέφει το δημιουργημένο βέλτιστο μοντέλο σε μια μεταβλητή καθώς και άλλη μια μεταβλητή που περιέχει την ιστορία της εκπαίδευσης του μοντέλου η οποία περιέχει πληροφορίες απόδοσης του μοντέλου όπως η τιμή της *loss* συνάρτησης. [6] [17] [23]

```
@anvil.server.callable
def model_optimized(prediction_model, units, activation, epochs, optimizer, batch_size, X_train, y_train, X_valid, y_valid, T, H):
    model = Sequential()
    if prediction_model == 'GRU':
        model.add(GRU(units = units, input_shape = (T, 1), activation = activation))
    elif prediction_model == 'LSTM':
        model.add(LSTM(units = units, input_shape = (T, 1), activation = activation))
    elif prediction_model == 'CNN':
        model.add(Conv1D(64, 1, activation=activation, input_shape=(T, 1)))
        model.add(MaxPooling1D())
        model.add(Flatten())
        model.add(Dense(units = units, activation=activation))
    elif prediction_model == 'MLP':
        model.add(Dense(units = units, activation=activation, input_dim=T))
        model.add(Dense(int(HORIZON)))
    model.compile(optimizer = optimizer, loss = 'mse', metrics=['accuracy'])
    model.summary()
    model_fit = model.fit(X_train, y_train, batch_size = batch_size, epochs = epochs, validation_data = (X_valid, y_valid), verbose=0)
    return model, model_fit
```

Εικόνα 14: Η συνάρτηση παραγωγής του μοντέλου με τις βέλτιστες υπερπαραμέτρους

#### 5.1.2.4 Συνάρτηση Προεπεξεργασίας και Πρόβλεψης Εξέλιξης Δεδομένων Χρονοσειρών – Παραγωγή Διαγραμμάτων

Άλλη μια συνάρτηση που έχει παραχθεί σε python μέσω Jupyter Notebook ονομάζεται *time\_series\_prediction\_with\_data\_new* και στην οποία αρχικά επιλέγεται το μοντέλο μηχανικής μέσω του ορίσμάτος της *prediction\_model*. Οι τιμές που μπορεί να πάρει το όρισμα *prediction\_model* που ορίζει το μοντέλο μηχανικής μάθησης είναι GRU, LSTM, CNN, MLP και αντιστοιχούν στα τεχνητά νευρωνικά δίκτυα Gated Recurrent Unit Network, Long Short-Term Memory Network, Convolutional Neural Network και Multi-layer Perceptron αντίστοιχα. Εν συνεχεία μετατρέπονται τα δεδομένα χρονοσειρών που εισέρχονται στην συνάρτηση μέσω του ορίσματος *json\_data* από μορφή JSON σε DataFrame. Έπειτα όπως φαίνεται στην παρακάτω εικόνα η συνάρτηση επιλέγει μέσω του ορίσματος *column* την μετρική (metric) του Δικτύου 5G για της οποίας τα δεδομένα χρονοσειράς θα βελτιστοποιηθεί το μοντέλο μηχανικής μάθησης, ως προς την ακρίβεια πρόβλεψης της εξέλιξής της μετρικής στον χρόνο. Οι τιμές που μπορεί να πάρει το όρισμα *column* είναι *node\_cpu\_seconds\_total* και *node\_network\_receive\_bytes\_total* που αντιστοιχούν στο συνολικό χρόνο που καταναλώνουν ο επεξεργαστής (CPU) για interrupts λόγω του Δικτύου 5G και στα συνολικά bytes που λαμβάνονται στο το UPF από τους σταθμούς βάσης (gNB) ενώ ταυτόχρονα επιδεικνύουν έμμεσα την κίνηση του δικτύου 5G.

```
@anvil.server.callable
def time_series_prediction_with_data_new(prediction_model, column, json_data):
    df = pd.read_json(json_data, orient='index')
    if column == 'node_cpu_seconds_total':
        df = df.drop(columns = ['node_network_receive_bytes_total'])
        data_set_selection = 'node_cpu_seconds_total'
    elif column == 'node_network_receive_bytes_total':
        df = df.drop(columns = ['node_cpu_seconds_total'])
        data_set_selection = 'node_network_receive_bytes_total'
```

Εικόνα 15: Συνάρτηση προεπεξεργασίας και πρόβλεψης εξέλιξης δεδομένων χρονοσειρών – παραγωγής διαγραμμάτων (1)

Στην συνέχεια η συνάρτηση προεπεξεργάζεται με την μέθοδο sliding window τα δεδομένα χρονοσειράς της επιλεγμένης μετρικής, δημιουργώντας με διαχωρισμό των δεδομένων χρονοσειράς, το σετ εκπαίδευσης (train set)(70% των δεδομένων), επαλήθευσης (validation set)(10% των δεδομένων), δοκιμής(test set)(20% των δεδομένων) όπως είναι εμφανές στην επόμενη εικόνα. Τα σετ αυτά είναι διαδοχικά χρονικά ξεκινώντας από το σετ εκπαίδευσης, μετά ακολουθεί το σετ επαλήθευσης και τέλος το σετ δοκιμής. Έτσι λοιπόν το κάθε σετ περιέχει ένα μέρος των δεδομένων της χρονοσειράς της εκάστοτε μετρικής που έχει επιλεχθεί.



```

length_df = len(df)
train_rows = int(0.7*length_df)
valid_rows = int(0.1*length_df)
test_rows = length_df - train_rows - valid_rows
train = df.copy()[0:train_rows]
train_rows_new = train_rows + 1
valid = df.copy()[train_rows:train_rows + valid_rows]
test = df.copy()[train_rows + valid_rows:]
scaler = MinMaxScaler()
train[data_set_selection] = scaler.fit_transform(train)
valid[data_set_selection] = scaler.fit_transform(valid)
test[data_set_selection] = scaler.fit_transform(test)
T = 2 # Let the input be a vector of the previous 12 seconds of the values.
HORIZON = 1 # one-step-ahead prediction
train_shifted = train.copy()
train_shifted['y_t+1'] = train_shifted[data_set_selection].shift(-1, freq = '6s')
for t in range(1, T+1):
    train_shifted[str(T-t)] = train_shifted[data_set_selection].shift(T-t, freq = '6s')
y_col = 'y_t+1'
X_cols = []
for i in range(-T+1, 1):
    if i != 0:
        X_cols.append('t{}'.format(i))
    else:
        X_cols.append('t')
train_shifted.columns = ['original'] + [y_col] + X_cols
train_shifted = train_shifted.dropna(how = 'any')
y_train = np.array(train_shifted[y_col])
X_train = np.array(train_shifted[X_cols])
X_train = X_train.reshape(X_train.shape[0], T, 1)
valid_shifted = valid.copy()
test_shifted = test.copy()
valid_shifted['y_t+1'] = valid_shifted[data_set_selection].shift(-1, freq = '6s')
test_shifted['y_t+1'] = test_shifted[data_set_selection].shift(-1, freq = '6s')
for t in range(1, T+1):
    valid_shifted[str(T-t)] = valid_shifted[data_set_selection].shift(T-t, freq = '6s')
for t in range(1, T+1):
    test_shifted[str(T-t)] = test_shifted[data_set_selection].shift(T-t, freq = '6s')
valid_shifted.columns = ['original'] + [y_col] + X_cols
test_shifted.columns = ['original'] + [y_col] + X_cols
valid_shifted = valid_shifted.dropna(how = 'any')
test_shifted = test_shifted.dropna(how = 'any')

y_valid = np.array(valid_shifted[y_col])
y_test = np.array(test_shifted[y_col])
X_valid = np.array(valid_shifted[X_cols])
X_test = np.array(test_shifted[X_cols])
X_valid = X_valid.reshape(X_valid.shape[0], T, 1)
X_test = X_test.reshape(X_test.shape[0], T, 1)

```

Εικόνα 16: Συνάρτηση προεπεξεργασίας και πρόβλεψης εξέλιξης δεδομένων χρονοσειρών – παραγωγής διαγραμμάτων (2)

Συνεχίζοντας ορίζεται ο συντονιστής Optuna ο οποίος ουσιαστικά αξιοποιώντας την *objective* συνάρτηση πραγματοποιεί βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης που έχει επιλεχθεί, με δεδομένα χρονοσειράς της μετρικής του Δικτύου 5G που έχει επιλεχθεί. Στην συνέχεια οι βέλτιστες υπερπαραμέτροι που επιστρέφει ο συντονιστής Optuna [19] διοχετεύονται στην *model\_optimized* συνάρτηση, η οποία επιστρέφει εκπαιδευμένο μέσω του σετ εκπαίδευσης και επαληθευμένο μέσω του σετ επαλήθευσης, το μοντέλο μηχανικής μάθησης που έχει επιλεχθεί με δεδομένα της μετρικής του Δικτύου 5G που έχει επιλεχθεί και τις βελτιστοποιημένες υπερπαραμέτρους. [6] Να τονιστεί πως στο μοντέλο εισάγονται διανύσματα δεδομένων εισόδου δύο διαστάσεων και προβλέπεται το αμέσως επόμενο χρονικά βήμα (χρονοσήμανση) της χρονοσειράς.



```

func = lambda trial: objective(trial, prediction_model, X_train, y_train, X_valid, y_valid, T, HORIZON)
study = optuna.create_study(direction="minimize")
study.optimize(func, n_trials=108, timeout=None)
trial = study.best_trial
units = trial.params['units']
activation = trial.params['activation']
epochs = trial.params['epochs']
optimizer = trial.params['optimizer']
batch_size = trial.params['batch_size']
model, model_fit = model_optimized(prediction_model, units, activation, epochs, optimizer, batch_size, X_train, y_train, X_val

```

Εικόνα 17: Συνάρτηση προεπεξεργασίας και πρόβλεψης εξέλιξης δεδομένων χρονοσειρών – παραγωγής διαγραμμάτων (3)

Αμέσως μετά εισάγοντας στο βέλτιστο αυτό μοντέλο τις χρονοσημάνσεις (timestamps) του σετ δοκιμής, λαμβάνονται οι προβλέψεις για την εξέλιξη της χρονοσειράς στις χρονοσημάνσεις αυτές. Οι πραγματικές τιμές στις χρονοσημάνσεις αυτές είναι γνωστές μέσω του σετ δοκιμής και έτσι λοιπόν η συνάρτηση επιστρέφει διαγράμματα του σετ εκπαίδευσης, του σετ επαλήθευσης, των πραγματικών τιμών του τεστ δοκιμής καθώς και των προβλεπόμενων από το μοντέλο τιμών του τεστ δοκιμής, έτσι ώστε να οπτικοποιηθούν τα δεδομένα χρονοσειράς των σετ εκπαίδευσης και επαλήθευσης και να συγκριθούν αυτά των πραγματικών τιμών του τεστ δοκιμής και των προβλεπόμενων από το μοντέλο τιμών του τεστ δοκιμής, ώστε να αξιολογηθεί η καταλληλότητα του μοντέλου. [17] Επίσης επιστρέφονται πληροφορίες απόδοσης του μοντέλου όπως η τιμή loss συνάρτηση (ποσοστό) αλλά και οι βέλτιστες υπερπαραμέτροι του μοντέλου.

```

predictions = model.predict(X_test)
#valid_loss
val_loss = model_fit.history['val_loss'][-1]
val_loss_ui = 'Validation evaluation--> ' + 'Validation_Loss: ' + str(val_loss*100) + '%'
best_parameters = 'Optimized Parameters--> ' + 'Units: ' + str(units) + ', Activation Function: ' + str(activation) + ',
test_predicted_df = pd.DataFrame(predictions, columns = ['t+' + str(t) for t in range(1, int(HORIZON) + 1)])
test_predicted_df['Time'] = test_shifted.index
test_predicted_df = pd.melt(test_predicted_df, id_vars = 'Time', value_name = 'predicted', var_name = 'h')
test_predicted_df['test'] = np.transpose(y_test).ravel()
test_predicted_df[['predicted', 'test']] = scaler.inverse_transform(test_predicted_df[['predicted', 'test']])
train_df = pd.DataFrame()
train_df['Time'] = train_shifted.index
train_df['train'] = np.transpose(y_train).ravel()
train_df[['train']] = scaler.inverse_transform(train_df[['train']])
valid_df = pd.DataFrame()
valid_df['Time'] = valid_shifted.index
valid_df['valid'] = np.transpose(y_valid).ravel()
valid_df[['valid']] = scaler.inverse_transform(valid_df[['valid']])
fig, ax = plt.subplots()
ax.plot(test_predicted_df['Time'], test_predicted_df['predicted'], '--r-', label = 'predicted')
ax.plot(test_predicted_df['Time'], test_predicted_df['test'], '-b', label = 'test')
ax.plot(train_df['Time'], train_df['train'], '-y', label = 'train')
ax.plot(valid_df['Time'], valid_df['valid'], '-y', label = 'train')
plt.legend()
plt.ylabel(data_set_selection)
plt.xlabel('Time')
plotly_fig = tls.mpl_to_plotly(fig)
data = plotly_fig['data'][0]
data1 = plotly_fig['data'][1]
data2 = plotly_fig['data'][2]
data3 = plotly_fig['data'][3]
plot_predicted = go.Scatter(x=data['x'], y=data['y'], mode='lines', name = 'predicted')
plot_test = go.Scatter(x=data1['x'], y= data1['y'], mode='lines', name = 'test')
plot_train = go.Scatter(x=data2['x'], y= data2['y'], mode='lines', name = 'train')
plot_valid = go.Scatter(x=data3['x'], y= data3['y'], mode='lines', name = 'valid')
return plot_predicted, plot_test, plot_train, plot_valid, val_loss_ui, best_parameters

```

Εικόνα 18: Συνάρτηση προεπεξεργασίας και πρόβλεψης εξέλιξης δεδομένων χρονοσειρών – παραγωγής διαγραμμάτων (4)

#### 5.1.2.5 Κώδικας Σύνδεσης με την Γραφική Διεπαφή Χρήστη του Anvil

Για ευκολία στις διαδικασίες άντλησης δεδομένων χρονοσειρών των επιθυμητών μετρικών του δικτύου 5G, βελτιστοποίησης του μοντέλου ως προς την ακρίβεια πρόβλεψης της εξέλιξης των χρονοσειρών στον

χρόνο αλλά και παρουσίασης των αποτελεσμάτων δημιουργήθηκε μια Γραφική Διεπαφή Χρήστη στο Anvil Framework. Για την διασύνδεση του Jupyter Notebook και του Anvil Framework έτσι ώστε τα γραφικά στοιχεία της Γραφικής Διεπαφής Χρήστη να αξιοποιούν τις συναρτήσεις του Jupyter Notebook γίνεται μέσω της παρακάτω γραμμής κώδικα όπως φαίνεται στην εικόνα. Σε όλες τις προηγούμενες συναρτήσεις μπορεί να παρατηρήσει κανείς επίσης πως πριν τον ορισμό τους υπάρχει το `@anvil.server.callable` όρισμα που είναι απαραίτητο έτσι ώστε να μπορεί να καλεστεί η συνάρτηση από τον εξυπηρετητή (server) του Anvil Framework που έχει στηθεί η Γραφική Διεπαφή Χρήστη. Τέλος το string που βρίσκεται εντός εισαγωγικών πάρθηκε από τον anvil εξυπηρετητή (server) και ουσιαστικά αποτελεί ένα κλειδί για την σύνδεση. [24]

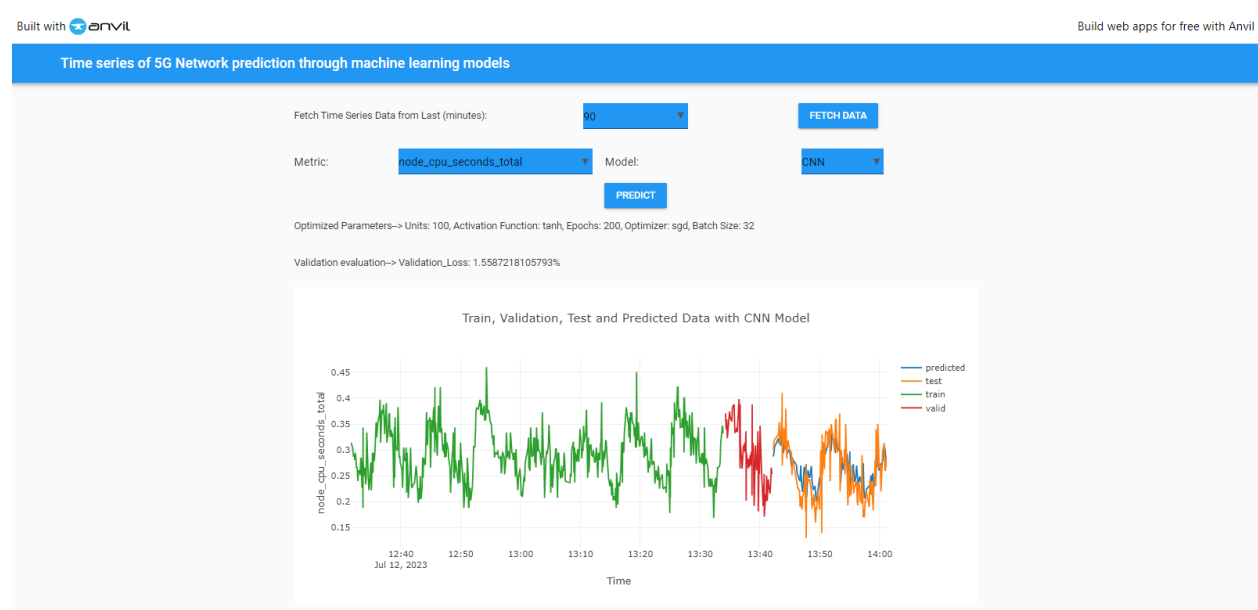
```
anvil.server.connect("server_SITWC4OQAW7TBAMEATUFA633-5BPUVYDAD77UUJVI")
```

### 5.1.3 Υλοποίηση Γραφικής Διεπαφής Χρήστη Μέσω Anvil Framework

Στα πλαίσια της παρούσας εργασίας χρησιμοποιήθηκε η cloud έκδοση του Anvil Framework για την υλοποίηση της Γραφικής Διεπαφής Χρήστη οπότε ο εξυπηρετητή (server) του Anvil βρίσκεται στο cloud. Στα δύο επόμενα υποκεφάλαια θα παρουσιαστεί η Γραφική Διεπαφή Χρήστη που δημιουργήθηκε στο Anvil καθώς και ο κώδικας που υλοποιήθηκε σε γλώσσα προγραμματισμού python για την διασύνδεση των γραφικών στοιχείων με τις συναρτήσεις του Jupyter Notebook.

#### 5.1.3.1 Γραφική Διεπαφή Χρήστη

Αξιοποιώντας το drag-and-drop [24] περιβάλλον του Anvil Framework, όπως φαίνεται στην εικόνα που ακολουθεί, υλοποιήθηκε η Γραφική Διεπαφή Χρήστη που θα διευκολύνει την άντληση δεδομένων χρονοσειρών των επιθυμητών μετρικών του 5G, βελτιστοποίηση του μοντέλου ως προς την ακρίβεια πρόβλεψης της εξέλιξης των χρονοσειρών στον χρόνο αλλά και παρουσίαση των αποτελεσμάτων.



Εικόνα 19: Γραφική Διεπαφή Χρήστη στο Anvil

Πιο συγκεκριμένα η Γραφική Διεπαφή Χρήστη αποτελείται από την **Fetch Time Series Data from Last (minutes)** drop-down λίστα, μέσω της οποίας δίνεται η δυνατότητα επιλογής μέχρι πόσα λεπτά πίσω στον χρόνο από τωρινή χρονική στιγμή θα συλλεχθούν δεδομένα χρονοσειρών από την InfluxDB βάση δεδομένων. Επίσης είναι εμφανές και το κουμπί **FETCH DATA**, το οποίο όταν πατηθεί εκτελεί την `get_data` συνάρτηση με όρισμα την επιλογή από την **Fetch Time Series Data from Last (minutes)** drop-down λίστα και επιστρέφει σε μία μεταβλητή του Anvil εξυπηρετητή (server) με JSON μορφή τα δεδομένα χρονοσειρών των δύο μετρικών `node_cpu_seconds_total` και `node_network_receive_bytes_total` από τη βάση δεδομένων InfluxDB. Επιπλέον στη Γραφική Διεπαφή Χρήστη έχει προστεθεί η **Metric** drop-down λίστα μέσω της οποίας δίνεται η δυνατότητα επιλογής της μετρικής (metric) του Δικτύου 5G για της οποίας τα δεδομένα χρονοσειράς θα βελτιστοποιηθεί ως προς την ακρίβεια πρόβλεψης της εξέλιξής της στον χρόνο, το μοντέλο μηχανικής μάθησης που επιλέγεται από την **Model** drop-down λίστα. Αφού λοιπόν έχουν ανακτηθεί τα δεδομένα χρονοσειρών μέσω του κουμπιού **FETCH DATA**, στην συνέχεια όταν πατηθεί το κουμπί **PREDICT**, που βρίσκεται στην Γραφική Διεπαφή Χρήστη, εκτελείται η `time_series_prediction_with_data_new` συνάρτηση με ορίσματα την επιλογή από τις **Metric, Model** την drop-down λίστα και την μεταβλητή του Anvil εξυπηρετητή (server) σε JSON μορφή που περιέχει δεδομένα χρονοσειρών από τη βάση δεδομένων. Αφού εκτελεστεί η συνάρτηση αυτή επιστρέφονται σε ένα γραφικό στοιχείο κειμένου οι βέλτιστες υπερπαραμέτροι του μοντέλου που επιλέχθηκε από την **Model** drop-down λίστα για την μετρική που επιλέχθηκε από την **Metric** drop-down λίστα καθώς και σε ένα γραφικό στοιχείο κειμένου επιστρέφονται πληροφορίες απόδοσης του εκπαιδευμένου, επαληθευμένου βέλτιστου μοντέλου (εκπαιδευμένο και επαληθευμένο με τις βέλτιστες υπερπαραμέτρους) όπως η `loss` συνάρτηση (ποσοστό). Τέλος η συνάρτηση `time_series_prediction_with_data_new` επιστρέφει σε 4 ξεχωριστά διαγράμματα τα σετ εκπαίδευσης, επαλήθευσης, πραγματικών τιμών του τεστ δοκιμής και των προβλεπόμενων από το μοντέλο τιμών του τεστ δοκιμής, τα οποία τοποθετεί συνδυαστικά σε ένα γραφικό στοιχείο διαγράμματος, έτσι ώστε να οπτικοποιηθούν τα δεδομένα χρονοσειράς της μετρικής που επιλέχθηκε μέσω της **Metric** drop-down λίστας των σετ εκπαίδευσης και επαλήθευσης και να συγκριθούν αυτά των πραγματικών τιμών του τεστ δοκιμής και των προβλεπόμενων από το μοντέλο τιμών του τεστ δοκιμής, ώστε να αξιολογηθεί η καταλληλότητα του μοντέλου.

#### 5.1.3.2 Υλοποίηση Κώδικα για την Σύνδεση των Γραφικών Στοιχείων με τις Συναρτήσεις

Για την διασύνδεση των συναρτήσεων του Jupyter Notebook και των γραφικών στοιχείων της Γραφικής Διεπαφής Χρήστη στο Anvil Framework όπως περιγράφηκε στο προηγούμενο υποκεφάλαιο υλοποιήθηκε κώδικας στον cloud εξυπηρετητή (server) του Anvil και ο οποίος παρατίθεται στην επόμενη εικόνα. Επιπροσθέτως όπως προαναφέρθηκε σε προηγούμενο κεφάλαιο θα πρέπει στο Jupyter Notebook να έχει εκτελεστεί η `anvil.server.connect("server_SITWC4OQAW7TBAMEATUFAG33-5BPUVYDAD77UUVI")` γραμμή κώδικα και όλες οι χρησιμοποιούμενες συναρτήσεις από την Γραφική Διεπαφή Χρήστη του Anvil, να διαθέτουν πριν τον ορισμό τους, το `@anvil.server.callable` όρισμα. [24]

```

from .anvil_designer import FormTemplate
from anvil import *
from anvil import Plot
import anvil.server
import plotly.graph_objects as go
from plotly import *

class Form(FormTemplate):
    def __init__(self, **properties):
        # Set Form properties and Data Bindings.
        self.init_components(**properties)
        data = None
        # Any code you write here will run before the form opens.

    def button_1_click(self, **event_args):
        if self.data != None:
            model_sel = self.model.selected_value
            metric = self.column.selected_value
            data_sel = self.data
            plot_predicted, plot_test, plot_train, plot_valid, loss_accur, best_parameters = anvil.server.call('time_series_prediction_with_data_new', model_sel, metric, data_sel)
            if model_sel == 'GRU':
                layout = go.Layout(title='Train, Validation, Test and Predicted Data with GRU Model', xaxis=dict(title='Time'), yaxis=dict(title='metric'))
            elif model_sel == 'LSTM':
                layout = go.Layout(title='Train, Validation, Test and Predicted Data with LSTM Model', xaxis=dict(title='Time'), yaxis=dict(title='metric'))
            elif model_sel == 'MLP':
                layout = go.Layout(title='Train, Validation, Test and Predicted Data with MLP Model', xaxis=dict(title='Time'), yaxis=dict(title='metric'))
            elif model_sel == 'CNN':
                layout = go.Layout(title='Train, Validation, Test and Predicted Data with CNN Model', xaxis=dict(title='Time'), yaxis=dict(title='metric'))
            self.plot_1.data = [plot_predicted, plot_test, plot_train, plot_valid]
            self.plot_1.layout = layout
            self.accuracy_loss.content = loss_accur
            self.best_params.content = best_parameters

    def fetch_data_click(self, **event_args):
        """This method is called when the button is clicked"""
        range_fetch_data = self.range_to_fetch_data.selected_value
        self.data = anvil.server.call('get_data', range_fetch_data)

```

Εικόνα 20: Κώδικας στον Anvil εξυπηρετητή για την σύνδεση γραφικών στοιχείων με τις συναρτήσεις

## 5.2 Βελτιστοποίηση Μοντέλων Μηχανικής Μάθησης

### 5.2.1 Περιγραφή Ερευνητικής Διαδικασίας Βελτιστοποίησης

Αφού λοιπόν έχει υλοποιηθεί το ενεργό περιβάλλον ενός πλήρους 5G Δικτύου, εκτελείται για τουλάχιστον 90 λεπτά αλλά χωρίς να έχει ολοκληρωθεί το script παραγωγής δικτυακής κίνησης και έχει συνδεθεί η γραφική διεπαφή χρήστη, που έχει δημιουργηθεί στο Anvil, με το Jupyter Notebook στο οποίο έχει υλοποιηθεί ο κώδικας σε python για την ανάκτηση δεδομένων χρονοσειρών των δύο μετρικών (*node\_cpu\_seconds\_total* και *node\_network\_receive\_bytes\_total*) του Δικτύου 5G καθώς και την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης, θα πραγματοποιηθεί ακολούθως μια ερευνητική διαδικασία για την εύρεση του βέλτιστου μοντέλου μηχανικής μάθησης για κάθε μία από τις μετρικές ως προς την ακρίβεια πρόβλεψης της εξέλιξής τους στον χρόνο.

Πιο αναλυτικά αρχικά στην γραφική διεπαφή χρήστη του Anvil θα επιλεγεί η τιμή **90** από την **Fetch Time Series Data from Last (minutes)** drop-down λίστα και στην συνέχεια θα πατηθεί το **FETCH DATA** κουμπί έτσι ώστε να συλλεχθούν από την βάση δεδομένων InfluxDB τα δεδομένα χρονοσειρών των δύο μετρικών του δικτύου 5G στον server του Anvil έως 90 λεπτά πίσω στον χρόνο από τωρινή χρονική στιγμή.

Fetch Time Series Data from Last (minutes):

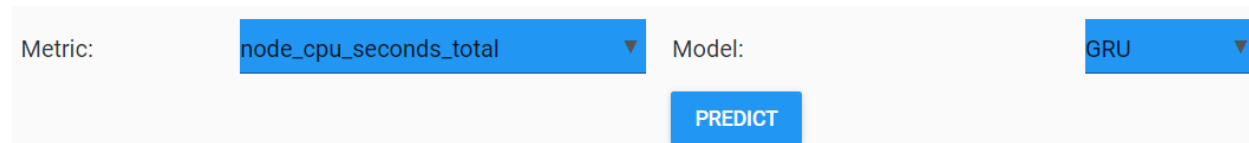
90
▼

FETCH DATA

Εικόνα 21: Επιλογή στην drop-down λίστας της γραφικής διεπαφής χρήστη για την ανάκτηση δεδομένων χρονοσειρών από την βάση δεδομένων των μετρικών *node\_cpu\_seconds\_total* *node\_network\_receive\_bytes\_total* των τελευταίων 90 λεπτών

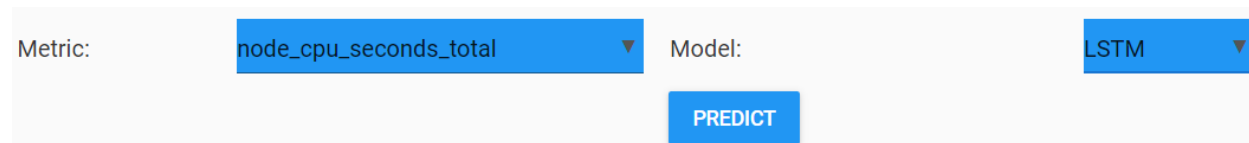
Αφού ανακτήθηκαν τα δεδομένα χρονοσειρών των δύο μετρικών του δικτύου 5G στην συνέχεια επιλέγεται η **node\_cpu\_seconds\_total** μετρική από την **Metric** drop-down λίστα και το **GRU** μοντέλο

μηχανικής μάθησης **Model** drop-down λίστα. Έπειτα πατώντας το **PREDICT** κουμπί, θα επιλεχθούν τα δεδομένα χρονοσειράς της `node_cpu_seconds_total` μετρικής, τα οποία θα χωριστούν σε σετ εκπαίδευσης, επαλήθευσης και θα προεπεξεργαστούν με την μέθοδο sliding window. Έπειτα μέσω του συντονιστή Optuna θα πραγματοποιηθεί η βελτιστοποίηση των τιμών των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης έτσι ώστε να παραχθεί το βέλτιστο μοντέλο με βάση την τιμή της loss συνάρτησης (ποσοστό) που θα οδηγήσει στην βέλτιστη ακρίβεια πρόβλεψης της εξέλιξής μετρικής στον χρόνο. Όσο μικρότερη τιμή έχει η loss συνάρτηση τόσο μεγαλύτερη προβλεπτική ισχύ έχει το μοντέλο. Επομένως οι υπερπαραμέτροι που θα εφαρμοστούν στο μοντέλο και θα καταλήγουν στην μικρότερη τιμή loss συνάρτησης, θεωρούνται οι βέλτιστες παράμετροι του μοντέλου αυτού και θα χρησιμοποιηθούν (οι βέλτιστες παράμετροι) για την εκπαίδευση, την επαλήθευση και την δοκιμή του μοντέλου. Έπειτα λοιπόν αφού το μοντέλο εκπαιδεύεται με το σετ εκπαίδευσης και επαληθεύεται με το σετ επαλήθευσης, δοκιμάζεται με το σετ δοκιμής. Στην συνέχεια στην γραφική διεπαφή χρήστη του Anvil εμφανίζεται η τιμή της loss συνάρτησης και οι βέλτιστες υπερπαραμέτροι του βέλτιστου μοντέλου μηχανικής μάθησης. Επιπλέον στην γραφική διεπαφή χρήστη παρουσιάζεται ένα συνδυαστικό διάγραμμα που περιέχει τα σετ εκπαίδευσης, επαλήθευσης, πραγματικών τιμών του τεστ δοκιμής και των προβλεπόμενων από το βέλτιστο μοντέλο τιμών του τεστ δοκιμής, έτσι ώστε να οπτικοποιηθούν τα δεδομένα χρονοσειράς της μετρικής `node_cpu_seconds_total` που επιλέχθηκε μέσω της **Metric** drop-down λίστας των σετ εκπαίδευσης και επαλήθευσης και να συγκριθούν αυτά των πραγματικών τιμών του τεστ δοκιμής και των προβλεπόμενων από το βέλτιστο μοντέλο τιμών του τεστ δοκιμής.



Εικόνα 22: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης LSTM της μετρικής `node_cpu_seconds_total`

Η διαδικασία βελτιστοποίησης των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης επαναλαμβάνεται με τα ίδια ανακτημένα δεδομένα χρονοσειρών, την ίδια επιλεγμένη μετρική `node_cpu_seconds_total` στην **Metric** drop-down λίστα και επιλέγοντας σειριακά τα υπόλοιπα μοντέλα μηχανικής μάθησης που υπάρχουν στην **Model** drop-down λίστα (LSTM, MLP, CNN) και πατώντας στην συνέχεια το κουμπί **PREDICT**.



Εικόνα 23: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης LSTM της μετρικής `node_cpu_seconds_total`

Metric: node\_cpu\_seconds\_total ▼ Model: MLP ▼

**PREDICT**

Εικόνα 24: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης MLP της μετρικής `node_cpu_seconds_total`

Metric: node\_cpu\_seconds\_total ▼ Model: CNN ▼

**PREDICT**

Εικόνα 25: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης CNN της μετρικής `node_cpu_seconds_total`

Στην συνέχεια αξιοποιώντας τα ίδια ανακτημένα δεδομένα χρονοσειρών αλλά επιλέγοντας την άλλη μετρική **`node_network_receive_bytes_total`** στην **Metric** drop-down λίστα, επαναλαμβάνεται η διαδικασία βελτιστοποίησης των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης επιλέγοντας σειριακά όλα τα μοντέλα μηχανικής μάθησης που υπάρχουν στην **Model** drop-down λίστα (GRU, LSTM, MLP, CNN) πατώντας στην συνέχεια το κουμπί **PREDICT**.

Metric: node\_network\_receive\_bytes\_total ▼ Model: GRU ▼

**PREDICT**

Εικόνα 26: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης GRU της μετρικής `node_network_receive_bytes_total`

Metric: node\_network\_receive\_bytes\_total ▼ Model: LSTM ▼

**PREDICT**

Εικόνα 27: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης LSTM της μετρικής `node_network_receive_bytes_total`

Metric: node\_network\_receive\_bytes\_total ▼ Model: MLP ▼

**PREDICT**

Εικόνα 28: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης MLP της μετρικής `node_network_receive_bytes_total`

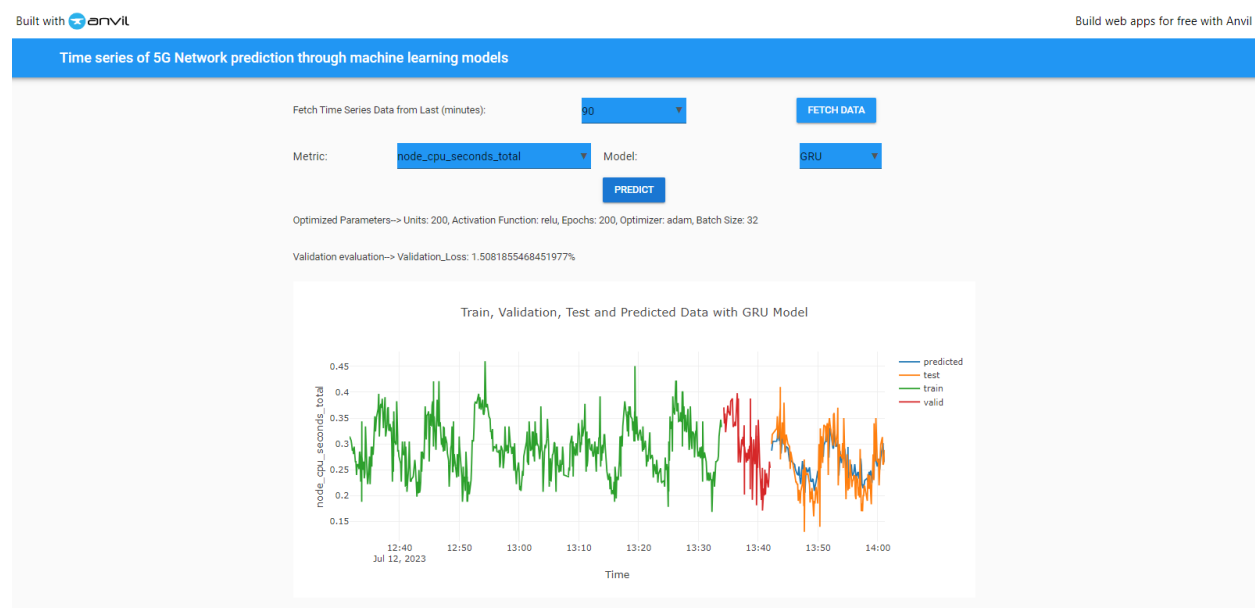
Metric: node\_network\_receive\_bytes\_total Model: CNN

**PREDICT**

Εικόνα 29: Επιλογές στις drop-down λίστες της γραφικής διεπαφής χρήστη για την βελτιστοποίηση των υπερπαραμέτρων του μοντέλου μηχανικής μάθησης CNN της μετρικής `node_network_receive_bytes_total`

### 5.2.2 Αποτελέσματα Ερευνητικής Διαδικασίας Βελτιστοποίησης

Κατά την διαδικασία βελτιστοποίησης συλλέχθηκαν αρχικά μέσω του Prometheus δεδομένα χρονοσειρών των `node_cpu_seconds_total` και `node_network_receive_bytes_total` μετρικών του 5G δικτύου που αντιστοιχούν στο συνολικό χρόνο που καταναλώνουν ο επεξεργαστής (CPU) για interrupts λόγω του Δικτύου 5G και στα συνολικά bytes που λαμβάνονται στο το UPF από τους σταθμούς βάσης (gNB), λόγω της αποστολής δικτυακών πακέτων με σταθερό ρυθμό από τον iPerf εξυπηρετητή (server) στην συνδεδεμένη στο NG-RAN εικονική συσκευή-συνδρομητή. Τα δεδομένα χρονοσειρών των μετρικών αυτών αφού αποθηκεύτηκαν στην βάση δεδομένων InfluxDB αξιοποιήθηκαν στην συνέχεια για την βελτιστοποίηση των υπερπαραμέτρων των τεσσάρων μοντέλων μηχανικής μάθησης (GRU, LSTM, MLP, CNN) που εφαρμόστηκαν στα δεδομένα των δύο μετρικών, ώστε να προβλέπεται με τη μεγαλύτερη ακρίβεια η εξέλιξή τους στον χρόνο. Να τονιστεί πως η βελτιστοποίηση στα μοντέλα έγινε με γνώμονα την ελαχιστοποίηση της τιμής της *loss* συνάρτησης των μοντέλων. Έτσι λοιπόν για κάθε μετρική με την βοήθεια του Optuna συντονιστή βρέθηκαν οι βέλτιστες υπερπαραμέτροι για καθένα από τα τέσσερα μοντέλα που δοκιμάστηκαν καθώς και η τιμή της *loss* συνάρτησης των μοντέλων αυτών με τις βέλτιστες υπερπαραμέτρους. Επίσης για την οπτικοποίηση του αποτελέσματος παράχθηκαν διαγράμματα που περιέχουν το σετ εκπαίδευσης, επαλήθευσης, πραγματικές τιμές του τεστ δοκιμής και τις προβλεπόμενες από τα βέλτιστα μοντέλα τιμών του τεστ δοκιμής. Στις επόμενες εικόνες παρουσιάζονται τα αποτελέσματα της διαδικασίας βελτιστοποίησης οπτικοποιημένα από την γραφική διεπαφή χρήστη στο Anvil.



Εικόνα 30: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης *loss* (ποσοστό) καθώς και το διάγραμμα οπτικοποίηση της ποιότητας του μοντέλου GRU για την μετρική `node_cpu_seconds_total`



## Time series of 5G Network prediction through machine learning models



Εικόνα 31: Η γραφική διεπαφή χρήση στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίηση της ποιότητας του μοντέλου LSTM για την μετρική `node_cpu_seconds_total`

## Time series of 5G Network prediction through machine learning models



Εικόνα 32: Η γραφική διεπαφή χρήση στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίηση της ποιότητας του μοντέλου MLP για την μετρική `node_cpu_seconds_total`



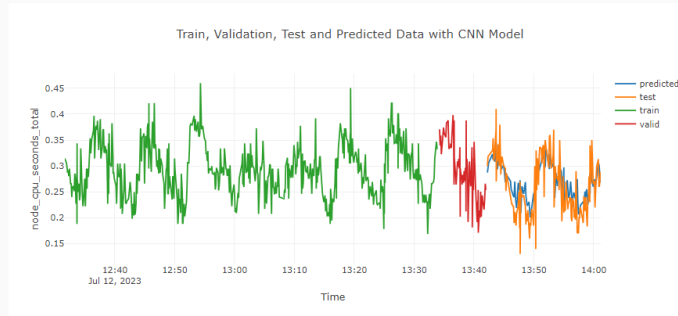
## Time series of 5G Network prediction through machine learning models

Fetch Time Series Data from Last (minutes):

Metric:  Model:

Optimized Parameters--> Units: 100, Activation Function: tanh, Epochs: 200, Optimizer: sgd, Batch Size: 32

Validation evaluation--> Validation\_Loss: 1.5587218105793%



Εικόνα 33: Η γραφική διεπαφή χρήση στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίηση της ποιότητας του μοντέλου CNN για την μετρική `node_cpu_seconds_total`

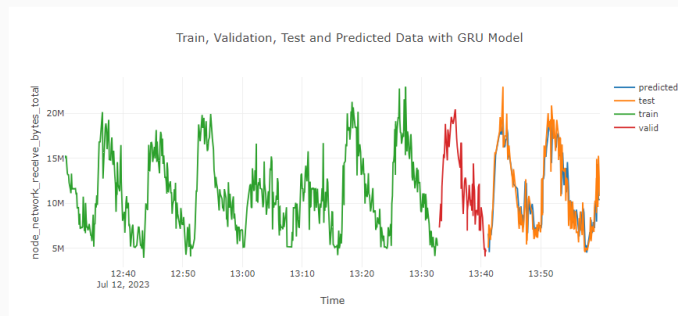
## Time series of 5G Network prediction through machine learning models

Fetch Time Series Data from Last (minutes):

Metric:  Model:

Optimized Parameters--> Units: 50, Activation Function: relu, Epochs: 200, Optimizer: adam, Batch Size: 64

Validation evaluation--> Validation\_Loss: 0.9093128144741058%



Εικόνα 34: Η γραφική διεπαφή χρήση στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίηση της ποιότητας του μοντέλου GRU για την μετρική `node_network_receive_bytes_total`

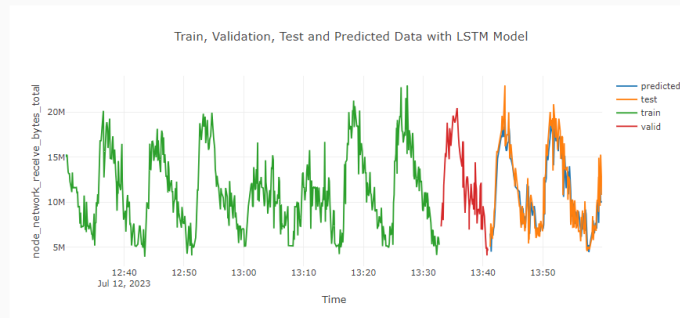
## Time series of 5G Network prediction through machine learning models

Fetch Time Series Data from Last (minutes):

Metric:  Model:

Optimized Parameters--> Units: 50, Activation Function: tanh, Epochs: 200, Optimizer: RMSprop, Batch Size: 32

Validation evaluation--> Validation\_Loss: 0.9385925717651844%



Εικόνα 35: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίηση της ποιότητας του μοντέλου LSTM για την μετρική `node_network_receive_bytes_total`

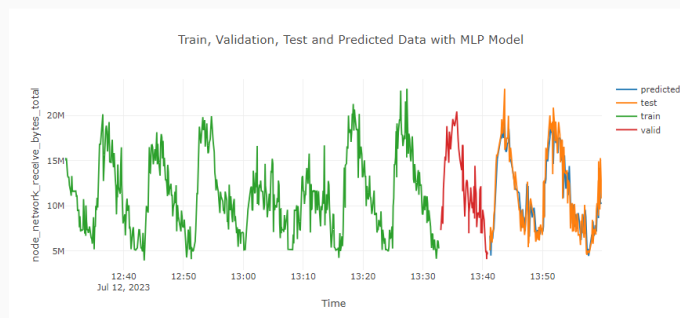
## Time series of 5G Network prediction through machine learning models

Fetch Time Series Data from Last (minutes):

Metric:  Model:

Optimized Parameters--> Units: 50, Activation Function: relu, Epochs: 200, Optimizer: adam, Batch Size: 128

Validation evaluation--> Validation\_Loss: 0.9172100573778152%



Εικόνα 36: Η γραφική διεπαφή χρήστη στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίηση της ποιότητας του μοντέλου MLP για την μετρική `node_network_receive_bytes_total`

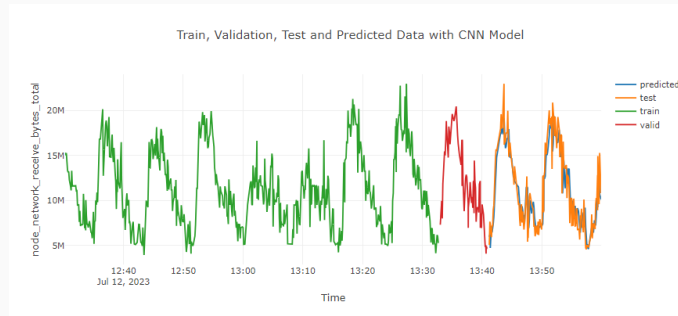
## Time series of 5G Network prediction through machine learning models

Fetch Time Series Data from Last (minutes):

Metric:  Model:

Optimized Parameters--> Units: 200, Activation Function: tanh, Epochs: 200, Optimizer: sgd, Batch Size: 64

Validation evaluation--> Validation\_Loss: 1.046793907880783%



Εικόνα 37: Η γραφική διεπαφή χρήση στο Anvil που περιέχει τις βέλτιστες υπερπαραμέτρους, την τιμή της συνάρτησης loss (ποσοστό) καθώς και το διάγραμμα οπτικοποίηση της ποιότητας του μοντέλου CNN για την μετρική `node_network_receive_bytes_total`

Συνεχίζοντας παρατίθεται από ένας συγκεντρωτικός πίνακες για κάθε μετρική όπου παρουσιάζονται οι βέλτιστες υπερπαραμέτροι καθώς και τιμή της loss συνάρτησης (ποσοστό) για καθένα από τα μοντέλα.

node_cpu_seconds_total						
Models	Optimized Hyperparameters					Loss Function %
	Units	Activation Function	Epochs	Optimizer	Batch Size	
GRU	100	tanh	200	sgd	32	1,55%
LSTM	200	relu	200	adam	32	1,53%
MLP	200	relu	200	RMSprop	128	1,51%
CNN	100	tanh	200	sgd	32	1,55%

Εικόνα 38: Οι βέλτιστες υπερπαραμέτροι καθώς και η τιμή της loss συνάρτησης για κάθε μοντέλο μηχανικής μάθησης που εφαρμόζεται στα δεδομένα της μετρικής `node_cpu_seconds_total`

node_network_receive_bytes_total						
Models	Optimized Hyperparameters					Loss Function %
	Units	Activation Function	Epochs	Optimizer	Batch Size	
GRU	50	relu	200	adam	64	0,90%
LSTM	50	tanh	200	RMSprop	32	0,93%
MLP	50	relu	200	adam	128	0,91%
CNN	200	tanh	200	sgd	64	1,04%

Εικόνα 39: Οι βέλτιστες υπερπαραμέτροι καθώς και η τιμή της loss συνάρτησης για κάθε μοντέλο μηχανικής μάθησης που εφαρμόζεται στα δεδομένα της μετρικής `node_network_receive_bytes_total`

Μέσω αυτών των δύο πινάκων είναι δυνατή η επιλογή του βέλτιστου μοντέλου μηχανικής για την πρόβλεψη της εξέλιξης της κάθε μιας μετρικής στον χρόνο. Οι δύο αυτές μετρικές συσχετίζονται άμεσα με την δικτυακή κίνηση γενικότερα σε ένα δίκτυο 5G, οπότε μπορεί να λεχθεί πως έμμεσα προβλέπεται και η κίνηση του δικτύου 5G στον χρόνο. Έτσι λοιπόν, έπειτα από μελέτη του πίνακα *node\_cpu\_seconds\_total* είναι εύκολο να παρατηρηθεί πως την καλύτερη δυνατή πρόβλεψη εξέλιξης της μετρικής *node\_cpu\_seconds\_total* στον χρόνο είναι σε θέση να προσφέρει το MLP (Multi Layer Perceptron) μοντέλο μηχανικής μάθησης διότι έχει την πιο μικρή τιμή loss συνάρτησης 1,51% και βέλτιστες υπερπαραμέτρους: 200 Units στα Dense στρώματα του τεχνητού νευρωνικού δικτύου, την relu ως Activation Function, 200 Εποχές (Epochs) εκπαίδευσης, τον RMSprop βελτιστοποιητή (optimizer) και 64 Batch Size. Ωστόσο και τα υπόλοιπα μοντέλα μηχανικής μάθησης (GRU, LSTM, CNN) του πίνακα της μετρικής *node\_cpu\_seconds\_total* παράγουν αρκετά καλές προβλέψεις αφού και σε αυτά η τιμή της loss συνάρτησης είναι αρκετά χαμηλή και παραπλήσια με αυτή του MLP μοντέλου, αλλά σε αντίθεση με το MLP μοντέλο καταναλώνουν αρκετά περισσότερους υπολογιστικούς πόρους γεγονός που καθιστά βέλτιστη λύση το MLP μοντέλο. Αντίστοιχα μελετώντας τον πίνακα *node\_network\_receive\_bytes\_total* μπορεί να εξαχθεί το συμπέρασμα πως η καλύτερη δυνατή πρόβλεψη εξέλιξης της μετρικής *node\_network\_receive\_bytes\_total* στον χρόνο παράγεται από το GRU (Gated Recurrent Unit) μοντέλο μηχανικής μάθησης το οποίο διαθέτει την μικρότερη τιμή loss συνάρτησης 0.9% και βέλτιστες υπερπαραμέτρους: 50 Units στα Dense στρώματα του τεχνητού νευρωνικού δικτύου, την relu ως Activation Function, 200 Εποχές (Epochs) εκπαίδευσης, τον adam βελτιστοποιητή (optimizer) και 64 Batch Size. Καλή συμπεριφορά πρόβλεψης εξέλιξης της μετρικής *node\_network\_receive\_bytes\_total* στον χρόνο επιδεικνύουν και τα μοντέλα LSTM και MLP διότι έχουν παραπλήσια και χαμηλή τιμή loss συνάρτησης, με αποτέλεσμα ιδιαίτερα το MLP μοντέλο που καταναλώνει λιγότερους υπολογιστικούς πόρους από το GRU να αποτελεί μια καλή εναλλακτική σε περίπτωση περιορισμού των υπολογιστικών πόρων στο σύστημα. Τέλος στην περίπτωση της μετρικής *node\_network\_receive\_bytes\_total*, όπως μαρτυρά ο αντίστοιχος πίνακας το μοντέλο CNN έχει αρκετά μεγαλύτερη τιμή loss συνάρτησης από τα υπόλοιπα μοντέλα και σε συνδυασμό με την υψηλή κατανάλωση υπολογιστικών πόρων καθίσταται μια όχι τόσο καλή επιλογή για την πρόβλεψη εξέλιξης της μετρικής *node\_network\_receive\_bytes\_total* στον χρόνο.

## 6 Επίλογος

Η τεχνολογία δικτύων κινητής επικοινωνίας 5G (5ης γενεάς) έχει ήδη εφαρμοστεί σχεδόν πλήρως, αντικαθιστώντας τις προηγούμενες τεχνολογίες δικτύων κινητής επικοινωνίας αφού ικανοποιεί απαιτήσεις της βιομηχανίας και της κοινωνίας που οι προηγούμενες γενιές δικτύων δεν μπορούσαν, όπως τις υψηλότερες ταχύτητες μεταφοράς δεδομένων, μικρούς χρόνους απόκρισης, μεταφορά μεγαλύτερου όγκου δεδομένων, αλλά και δυνατότητες για την ανάπτυξη του εγχειρήματος της βιομηχανίας 4ης γενεάς. Όπως περιγράφηκε σε προηγούμενο κεφάλαιο, κάθε νέα γενεά δικτύου προσφέρει νέες δυνατότητες. Συγκεκριμένα η 5η γενεά προσφέρει την αυτοματοποίηση διαφόρων λειτουργιών του δικτύου καθώς και την πρόβλεψη νέων. Επίσης η τεχνολογία δικτύων 5G (5ης γενεάς) επιτρέπει την machine to machine επικοινωνία και την επικοινωνία εφαρμογών που ανήκουν στον τομέα του Internet of Things ενώ ταυτόχρονα προσφέρεται υψηλή αξιοπιστία. Ακόμη, όπως όλες οι προηγούμενες τεχνολογίες δικτύων κινητής επικοινωνίας, έτσι και αυτή της 5ης γενεάς αποτελείται από δύο μέρη όσον αφορά την αρχιτεκτονική της: 1) το Δίκτυο Ραδιοπρόσβασης NG-RAN (New Generation Radio Access Network ή NG-RAN) και 2) το Δίκτυο Πυρήνα 5ης γενεάς (5G Core Network ή 5GC). Το Δίκτυο Πυρήνα 5ης γενεάς (5G Core Network) είναι η ‘καρδιά’ του τηλεπικοινωνιακού δικτύου 5G (5ης γενεάς) και μέσω αυτού δημιουργείται αξιόπιστη και ασφαλής σύνδεση στο δίκτυο για τις διάφορες συσκευές που είναι συνδεδεμένες σε αυτό, ενώ ταυτόχρονα παρέχει πρόσβαση και στις υπηρεσίες του δικτύου. Παρόλο που το Δίκτυο Πυρήνα 5ης γενεάς μοιάζει αρκετά με το Δίκτυο Πυρήνα των προηγούμενων γενεών διαφέρει αρκετά στη διαδικασία του signaling και στον τρόπο διασύνδεσης των λειτουργιών που γίνεται μέσω μιας Service-Based διεπαφής ενώ, όπως είναι φυσικό υπάρχουν και νέες λειτουργίες. Από την άλλη πλευρά το Δίκτυο Ραδιοπρόσβασης NG-RAN αποτελείται από πολλαπλούς ραδιοσταθμούς βάσης καθένας από τους οποίους εξυπηρετεί την λήψη και την μετάδοση δεδομένων από συνήθως τρία ‘κελιά’ (μεγάλα γεωγραφικά τμήματα) και οι οποίοι είναι συνδεδεμένοι μεταξύ τους, με τις συσκευές, αλλά και με το 5G Core Network. Το Δίκτυο Ραδιοπρόσβασης NG-RAN αξιοποιεί την τεχνολογία διαμόρφωσης OFDM που έχει πολλά πλεονεκτήματα στην μετάδοση ενός σήματος σε μια συσκευή, όπως εξηγήθηκε σε προηγούμενη ενότητα, προσφέροντας βελτιωμένη ποιότητα σήματος και καλύτερη αξιοποίηση του φάσματος. Τέλος, η πραγματοποίηση διαδικασιών signaling και μεταφοράς δεδομένων μεταξύ των ραδιοσταθμών βάσης και μεταξύ των ραδιοσταθμών βάσης και του 5G Core Network καθώς και η επικοινωνία μεταξύ συσκευών και ραδιοσταθμών βάσης γίνεται μέσω διαφόρων διεπαφών και των αντίστοιχων πρωτοκόλλων που ακολουθούνται σε κάθε διεπαφή σύμφωνα με την βιβλιογραφία του 5G Core Network.

Για την επιτήρηση και πρόβλεψη της κίνησης των δεδομένων μέσω της κατανάλωσης υπολογιστικών και δικτυακών πόρων ενός δικτύου 5G (5ης γενεάς) που είναι ενεργό σε ένα υπολογιστικό σύστημα, αξιοποιούνται συστήματα (λογισμικά) παρακολούθησης και μοντέλα μηχανικής μάθησης αντίστοιχα. Πιο συγκεκριμένα μέσω του συστήματος παρακολούθησης συλλέγονται μετρικές δεδομένων χρονοσειρών του δικτύου κινητής τηλεπικοινωνίας 5G (5ης γενεάς) που είναι ενεργό στην υπολογιστική υποδομή, διαμέσου της επεξεργασίας των οποίων είναι δυνατή η βελτιστοποίηση της ροής των δεδομένων του υπολογιστικού συστήματος και της κατανάλωσης υπολογιστικών και δικτυακών πόρων του, εξαιτίας διαφόρων λειτουργιών του δικτύου ενώ μέσω των μοντέλων μηχανικής μάθησης τα δεδομένα χρονοσειρών διαφόρων μετρικών του δικτύου που συλλέγονται αξιοποιούνται ώστε να

εκπαιδευτούν κατάλληλα τα μοντέλα για να μπορούν στην συνέχεια να προβλέπουν την εξέλιξη στον χρόνο της ροής των δεδομένων του υπολογιστικού συστήματος καθώς και της κατανάλωσης υπολογιστικών και δικτυακών πόρων του, αποφεύγοντας έτσι σφάλματα που μπορεί να οδηγήσουν σε κατάρρευση του.

Όσο αφορά τα δεδομένα χρονοσειρών, σύμφωνα με τον ορισμό τους, είναι μια ακολουθία δεδομένων μιας μετρήσιμης ποσότητας ενός συστήματος, που έχουν ληφθεί σε συγκεκριμένες χρονικές στιγμές, οι οποίες συνήθως ισαπέχουν μεταξύ τους και με την βοήθειά τους είναι εφικτή η επίλυση και η πρόβλεψη προβλημάτων σε ένα σύστημα, γι' αυτό και συλλέγονται τέτοιου είδους δεδομένα από κάθε σύστημα παρακολούθησης. Ωστόσο όπως τονίστηκε σε προηγούμενο κεφάλαιο, είναι πολύ απαιτητική η αποθήκευση και η γρήγορη ανάκλησή των χρονοσειρών, αφού πρόκειται για μεγάλο όγκο δεδομένων που συσσωρεύεται συνεχώς. Γι' αυτόν τον λόγο δημιουργήθηκαν οι ειδικές βάσεις δεδομένων χρονοσειρών που ικανοποιούν την αποθήκευση και την γρήγορη ανάκληση δεδομένων χρονοσειρών, αλλά και προσφέρουν εργαλεία για την διευκόλυνση της ανάλυσης και διοχέτευσης των δεδομένων αυτών. Κάθε σύγχρονο σύστημα παρακολούθησης ενός υπολογιστικού συστήματος διαθέτει μια βάση δεδομένων χρονοσειρών, την οποία μπορούν να αξιοποιούν τα μοντέλα μηχανικής μάθησης για την εκπαίδευσή τους. Αξιοσημείωτο παραδείγματα ενός συστήματος παρακολούθησης και μιας βάσης δεδομένων χρονοσειρών είναι το Prometheus και η InfluxDB αντίστοιχα, που αναλύθηκαν διεξοδικά.

Συνεχίζοντας, επειδή στο πλαίσιο της παρούσας μεταπτυχιακής εργασίας σκοπός ήταν να δημιουργηθεί ένα περιβάλλον υλοποίησης και βελτιστοποίησης ροών μηχανικής μάθησης για την πρόβλεψη κίνησης σε ένα δίκτυο κινητής επικοινωνίας 5G (5ης γενιάς) που είναι ενεργό σε ένα υπολογιστικό σύστημα, προαναφέρθηκαν και αναλύθηκαν εκτενώς σε προηγούμενο κεφάλαιο οι έννοιες της τεχνητής νοημοσύνης, της μηχανικής μάθησης, των τεχνητών νευρωνικών δικτύων και της βαθιάς μάθησης. Αναλυτικότερα, έγινε μια σύντομη ιστορική αναδρομή στο πεδίο της τεχνητής νοημοσύνης και περιγράφηκαν οι κατηγορίες μοντέλων μηχανικής μάθησης καθώς και υποκατηγορίες τους. Επιπροσθέτως, παρουσιάστηκε η αρχιτεκτονική των τεχνητών δικτύων, τα οποία αποτελούνται από τους λεγόμενους νευρώνες και τα λεγόμενα βάρη και μπορούν να ενταχθούν σε διάφορες κατηγορίες ανάλογα την αρχιτεκτονική των νευρώνων που αξιοποιούν και της συνδεσιμότητας μεταξύ τους μέσω των βαρών. Επεξηγήθηκε ακόμη το πεδίο της βαθιάς μάθησης, το οποίο ουσιαστικά είναι ένας κλάδος της μηχανικής μάθησης, που ασχολείται με την ανάπτυξη μεθόδων για την εκπαίδευση τεχνητών νευρωνικών δικτύων. Έπειτα μελετήθηκαν διάφορες βιβλιοθήκες της γλώσσας προγραμματισμού Python, οι οποίες διευκολύνουν την υλοποίηση και εφαρμογή μοντέλων μηχανικής μάθησης, την ανάκτηση δεδομένων από βάσεις δεδομένων, την επεξεργασία των δεδομένων αυτών πριν την εισαγωγή τους στα μοντέλα, την οπτικοποίηση των αποτελεσμάτων πρόβλεψης των μοντέλων αυτών καθώς και της υλοποίησης μιας εύχρηστης γραφικής διεπαφής χρήστη (Anvil framework).

Η θεωρητική περιγραφή της μηχανικής μάθησης και της γλώσσας προγραμματισμού Python μέσω της οποίας υλοποιούνται όλα τα μοντέλα μηχανικής μάθησης, συνοδεύτηκε από ένα σύντομο οδηγό πραγμάτωσης του περιβάλλοντος υλοποίησης και βελτιστοποίησης ροών ή μοντέλων μηχανικής μάθησης για την πρόβλεψη κίνησης σε ένα δίκτυο 5G (5ης γενιάς) που είναι ενεργό σε private cloud. Σύμφωνα με αυτόν, αρχικά πραγματοποιήθηκε η υλοποίηση ενός πλήρους δικτύου κινητής επικοινωνίας 5G (5ης Γενιάς) σε ένα private cloud αξιοποιώντας το virtual machine με το όνομα ubuntu@free5gc-sm

που περιέχει λογισμικό προσομοίωσης ενός 5G Core Network *free5GC* και το virtual machine με το όνομα *ubuntu@ueransim* που περιέχει λογισμικό προσομοίωσης ενός NG-RAN *Ueransim* το οποίο έχει συνδεσμένη μια εικονική συσκευή σ' αυτό. Στα δύο αυτά virtual machines έχουν ρυθμιστεί κατάλληλα οι παράμετροί τους ώστε να επικοινωνούν μεταξύ τους πραγματώνοντας το 5G δίκτυο ενώ ταυτόχρονα να παρακολουθούνται από το σύστημα παρακολούθησης Prometheus που βρίσκεται σε ένα άλλο virtual machine του private cloud, έτσι ώστε να συλλέγονται δεδομένα χρονοσειρών διαφόρων μετρικών του δικτύου κινητής επικοινωνίας 5G (5ης Γενεάς) τα οποία θα αποθηκεύονται σε μια InfluxDB βάση δεδομένων που βρίσκεται σε άλλο virtual machine. Έπειτα εκτελέστηκε ένα script το οποίο αποστέλλει δικτυακά πακέτα από ένα iPerf εξυπηρετητή (server) που βρίσκεται σε ένα άλλο virtual machine του private cloud με το όνομα *jim@jarmusch* στην εικονική συσκευή-συνδρομητή με σταθερό ρυθμό.

Επίσης υλοποιήθηκε κώδικας, του οποίου το ένα μέρος ουσιαστικά κατά την διάρκεια αποστολής πακέτων από τον iPerf εξυπηρετητή στην εικονική συσκευή-συνδρομητή ανακτά δεδομένα χρονοσειρών των τελευταίων 90 λεπτών δύο μετρικών από το δίκτυο κινητής επικοινωνίας 5G (5ης Γενεάς) *node\_cpu\_seconds\_total* και *node\_network\_receive\_bytes\_total*, που αντιστοιχούν στο συνολικό χρόνο που καταναλώνει ο επεξεργαστής (CPU) για interrupts λόγω του Δικτύου 5G και στα συνολικά bytes που λαμβάνονται στο το UPF από τους σταθμούς βάσης (gNB) επιδεικνύοντας έμμεσα την κίνηση του δικτύου. Το υπόλοιπο μέρος του κώδικα βελτιστοποιεί τις υπερπαραμέτρους του GRU, LSTM, MLP ή CNN μοντέλου μηχανικής μάθησης που εφαρμόζεται στα δεδομένα χρονοσειράς κάθε μετρικής έτσι ώστε για κάθε ζευγάρι μοντέλου-μετρικής να προβλέπεται με τη μεγαλύτερη δυνατή ακρίβεια η εξέλιξη της μετρικής αυτής στον χρόνο. Επίσης κατασκευάστηκε μέσω του Anvil framework μια εύχρηστη γραφική διεπαφή χρήστη για την διευκόλυνση της διαδικασίας βελτιστοποίησης των υπερπαραμέτρων των μοντέλων και την παρουσίαση, οπτικοποίηση των αποτελεσμάτων με εύληπτο τρόπο.

Καταληκτικά, αφού στήθηκε το περιβάλλον υλοποίησης και βελτιστοποίησης ροών ή μοντέλων μηχανικής μάθησης για την πρόβλεψη κίνησης σε ένα δίκτυο 5G (5ης γενεάς) το οποίο είναι ενεργό σε ένα private cloud, πραγματοποιήθηκε η ερευνητική διαδικασία της μεταπτυχιακής εργασίας, η οποία έχει στόχο την εύρεση του βέλτιστου μοντέλου μηχανικής μάθησης για κάθε μία από τις μετρικές που επιδεικνύουν έμμεσα την κίνηση του δικτύου, ως προς την ακρίβεια πρόβλεψης της εξέλιξής τους στον χρόνο. Μέσω της γραφικής διεπαφής χρήστη ανακτήθηκαν αρχικά τα δεδομένα χρονοσειράς των δύο μετρικών και έπειτα αυτά διοχετεύθηκαν στο κομμάτι κώδικα που πραγματοποιεί βελτιστοποίηση των υπερπαραμέτρων του GRU, LSTM, MLP ή CNN μοντέλου μηχανικής μάθησης στα δεδομένα χρονοσειράς κάθε μιας από τις μετρικές, έτσι ώστε σειριακά για κάθε ζευγάρι μετρικής-μοντέλου να ευρίσκονται οι βέλτιστες υπερπαραμέτροι, η loss συνάρτηση και να παράγεται ένα διάγραμμα που παρουσιάζεται το σετ εκπαίδευσης, επαλήθευσης του μοντέλου καθώς και οι πραγματικές και προβλεπόμενες από το μοντέλο τιμές του τεστ δοκιμής.

Παρατηρώντας λοιπόν τα αριθμητικά αποτελέσματα των πινάκων και τα διαγράμματα του κεφαλαίου 5.2.2 που προέκυψαν μέσω της ερευνητικής διαδικασίας, ακριβέστερη και αποδοτικότερη πρόβλεψη εξέλιξής της μετρικής *node\_cpu\_seconds\_total* στον χρόνο όπως προαναφέρθηκε αποδίδει το MLP (Multi Layer Perceptron) μοντέλο μηχανικής μάθησης με βέλτιστες υπερπαραμέτρους: 200 Units στα Dense στρώματα του τεχνητού νευρωνικού δικτύου, την relu ως Activation Function, 200 Εποχές (Epochs) εκπαίδευσης, τον RMSprop βελτιστοποιητή (optimizer) και 64 Batch Size, αφού διαθέτει την ελάχιστη

τιμή loss συνάρτησης σε σύγκριση με τα υπόλοιπα μοντέλα (GRU, LSTM, CNN) και απαιτεί τους λιγότερους υπολογιστικούς πόρους. Αντίστοιχα πολύ ποιοτική πρόβλεψη εξέλιξης της μετρικής *node\_network\_receive\_bytes\_total* στον χρόνο προσφέρει το GRU (Gated Recurrent Unit) μοντέλο μηχανικής μάθησης το οποίο συγκρινόμενο με τα υπόλοιπα μοντέλα διαθέτει την μικρότερη τιμή loss συνάρτησης και βέλτιστες υπερπαραμέτρους: 50 Units στα Dense στρώματα του τεχνητού νευρωνικού δικτύου, την relu ως Activation Function, 200 Εποχές (Epochs) εκπαίδευσης, τον adam βελτιστοποιητή (optimizer) και 64 Batch Size. Ωστόσο πολύ καλή και αποδοτική πρόβλεψη εξέλιξης της μετρικής *node\_network\_receive\_bytes\_total* στον χρόνο προσφέρει και το MLP (Multi Layer Perceptron) μοντέλο μηχανικής μάθησης, με βέλτιστες υπερπαραμέτρους: 50 Units στα Dense στρώματα του τεχνητού νευρωνικού δικτύου, την relu ως Activation Function, 200 Εποχές (Epochs) εκπαίδευσης, τον adam βελτιστοποιητή (optimizer) και 128 Batch Size καθώς διαθέτει την αμέσως μεγαλύτερη και πολύ παραπλήσια τιμή loss συνάρτησης σε σύγκριση με το GRU ενώ απαιτεί πολύ λιγότερους υπολογιστικούς πόρους από το GRU.



## Βιβλιογραφία

- [1] X. C. J. W. a. J. W. H. Yin, «Deep Learning Based Intelligent Resource Management in Fog Radio Access Networks,» IEEE Transactions on Vehicular Technology, 2019.
- [2] P. H. M. O. L. F. S. S. C. M. STEFAN ROMMER, 5G CORE NETWORKS Powering Digitalization, Cambridge, Massachusetts: Academic Press, 2020.
- [3] P. A. JOEL BASTOS, Hands-On Ifrastructure Monitoring with Prometheus, Birmingham: Packt Publishing Ltd, 2019.
- [4] «ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΧΡΟΝΟΣΕΙΡΩΝ,» 25 8 2018. [Ηλεκτρονικό]. Available: <https://eclass.aegean.gr/modules/document/file.php/SAS142/KEF1.pdf>.
- [5] Δ. Κουγιουμτζής, «Ανάλυση Δεδομένων,» 11 3 2011. [Ηλεκτρονικό]. Available: <https://users.auth.gr/dkugiu/Teach/DataAnalysis/NotesDataAnalysis.pdf>.
- [6] F. Lazzeri, Machine Learning for Time Series Forecasting with Python, 2020.
- [7] «Stackhero for InfluxDB,» [Ηλεκτρονικό]. Available: <https://www.stackhero.io/en/services/InfluxDB/documentations/Introduction>.
- [8] «Digital Guide ΙΟΝΟΣ,» 12 5 2023. [Ηλεκτρονικό]. Available: <https://www.ionos.com/digitalguide/hosting/technical-matters/what-is-influxdb/>.
- [9] «we are community,» 12 5 2021. [Ηλεκτρονικό]. Available: <https://wearecommunity.io/communities/india-java-user-group/articles/891>.
- [10] C. Huang και A. Petukhina, Applied Time Series Analysis and Forecasting with Python, 2022.
- [11] «javatpoint,» 2021. [Ηλεκτρονικό]. Available: <https://www.javatpoint.com/types-of-machine-learning>.
- [12] P. Vadapalli, «upGrad,» 29 12 2020. [Ηλεκτρονικό]. Available: <https://www.upgrad.com/blog/types-of-neural-networks/>.
- [13] Coursera, «Coursera,» 15 6 2023. [Ηλεκτρονικό]. Available: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>.
- [14] «Python,» 2023. [Ηλεκτρονικό]. Available: <https://www.python.org/doc/essays/blurb/>.
- [15] «Domino,» 2023. [Ηλεκτρονικό]. Available: <https://www.dominodatalab.com/data-science-dictionary/anaconda>.
- [16] D. Ellis, «HubSpot,» 20 1 2023. [Ηλεκτρονικό]. Available: <https://blog.hubspot.com/website/anaconda-python>.

- [17] «plotly | Graphing Libraries,» 2023. [Ηλεκτρονικό]. Available: <https://plotly.com/python/getting-started/>.
- [18] S. Starkman, «towardsdatascience,» 23 6 2023. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/a-simple-guide-to-pythons-datetime-library-55eb6ce845b5>.
- [19] «OPTUNA,» 2018. [Ηλεκτρονικό]. Available: <https://optuna.readthedocs.io/en/stable/index.html>.
- [20] «influxdata,» [Ηλεκτρονικό]. Available: <https://docs.influxdata.com/influxdb/cloud/api-guide/client-libraries/python/>.
- [21] R. Ritz, «towardsdatascience,» 11 8 2023. [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/an-introduction-to-anvil-full-stack-web-apps-with-nothing-by-python-cbab06392d13>.
- [22] meredydd, «github,» 28 4 2023. [Ηλεκτρονικό]. Available: <https://github.com/anvil-works/anvil-runtime>.
- [23] D. KARAR, «kaggle,» [Ηλεκτρονικό]. Available: <https://www.kaggle.com/code/dkdevmallya/deep-learning-cnn-lstm-time-series-forecasting>.
- [24] «Anvil,» 2023. [Ηλεκτρονικό]. Available: <https://anvil.works/>.
- [25] harkiran78, «geeksforgeeks,» [Ηλεκτρονικό]. Available: <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>.
- [26] «aws,» 2023. [Ηλεκτρονικό]. Available: <https://aws.amazon.com/what-is/neural-network/>.
- [27] J. G. a. K. S. A. Saxena, «Predictive Maintenance in Industry 4.0: A Comprehensive Review,» IEEE Transactions on Reliability, 2018.
- [28] ITU, «AI for Good,» 2023. [Ηλεκτρονικό]. Available: <https://aiforgood.itu.int/new-itu-standard-to-introduce-machine-learning-into-5g-networks/>.
- [29] M. Z. S. A. I. a. J. T. A. Imran, «Machine Learning for 5G Networks: Applications, Challenges, and Opportunities,» IEEE Communications Surveys & Tutorials, 2019.