



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ  
ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

## **Μηχανική Μάθηση για Διαδικτυακά Δεδομένα**

Μπενοβιας Λεωνιδας

Επιβλέπων: Εμμανουήλ Βαρβαρίγος, Καθηγητής ΕΜΠ

**Αθήνα, Ιούλιος 2023**





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ  
ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

## **Μηχανική Μάθηση για Διαδικτυακά Δεδομένα**

Όνοματεπώνυμο: Λεωνίδας Μπενοβίας

Επιβλέπων: Εμμανουήλ Βαρβαρίγος, Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5η Ιουλίου 2023

Εμμανουήλ Βαρβαρίγος,  
Καθηγητής ΕΜΠ

Θεοδώρα Βαρβαρίγου,  
Καθηγητής ΕΜΠ

Ηρακλής Αβραμόπουλος  
Καθηγητής ΕΜΠ

**Αθήνα, Ιούλιος 2023**

Λεωνίδας Μπενοβίας

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

Copyright © Λεωνίδας Μπενοβίας, 2023

Με επιφύλαξη παντός δικαιώματος

Απαγορεύεται η αντιγραφή, αποθήκευση σε αρχείο πληροφοριών, διανομή, αναπαραγωγή, μετάφραση ή μετάδοση της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό, υπό οποιαδήποτε μορφή και με οποιοδήποτε μέσο επικοινωνίας, ηλεκτρονικό ή μηχανικό, χωρίς την προηγούμενη έγγραφη άδεια του συγγραφέα. Επιτρέπεται η αναπαραγωγή, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν στη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Η έγκριση της διπλωματικής εργασίας από τη Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Ηλεκτρονικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/1932, Άρθρο 202).

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## **Μηχανική Μάθηση για Διαδικτυακά Δεδομένα**

Μπενοβίας Λ. (Επιβλέπων: Βαρβαρίγος Ε.)

### **Περίληψη**

#### **Μηχανική Μάθηση για Δικτυακά Δεδομένα**

Σκοπός αυτής της εργασίας είναι η μελέτη της χρήσης μηχανικής μάθησης στον υπολογισμό Διαδικτυακών Δεδομένων. Γνωστό και ως Network Tomography, είναι ο υπολογισμός εσωτερικών χαρακτηριστικών του δικτύου χρησιμοποιώντας πληροφορία από δεδομένα τελικού σημείου (end-point data).

Ο σκοπός της μελέτης μας είναι η σύγκριση των κλασικών αλγεβρικών μεθόδων για tomography με τις μεθόδους μηχανικής μάθησης (ML). Πιο συγκεκριμένα μελετάμε πως με την χρήση νευρωνικών δικτύων είναι εφικτό γνωρίζοντας ορισμένες μετρικές για ολόκληρα τμήματα του δικτύου να υπολογίσουμε μετρικές όπως το delay για επιμέρους τμήματα του δικτύου, καθώς και πως επηρεάζουν το νευρωνικό δίκτυο αλλά και η γνωστή πληροφορία την δυνατότητα αυτή. Συγκρίνουμε την αποδοτικότητα αυτής της μεθόδου με τις κλασικές αλγεβρικές μεθόδους. Για το σκοπό αυτό παρουσιάζεται αρχικά η σχετική βιβλιογραφία πάνω στο Network Tomography. Η μοντελοποίηση ενός δικτύου υπολογιστών γίνεται με γράφο για τον οποίο χρησιμοποιείται η βιβλιοθήκη Networkx (<https://networkx.org/>), ενώ για το νευρωνικό δίκτυο χρησιμοποιείται Python και το Tensorflow.Keras.

Συμπεραίνεται τελικά ότι η χρήση νευρωνικών δικτύων για τον υπολογισμό επιμέρους μετρικών υπολογιστικών δικτύων είναι εφικτή. Κλείνοντας παρατίθενται εισηγήσεις για περαιτέρω έρευνα.

**Λέξεις κλειδιά:** Network Tomography, νευρωνικά δίκτυα, μηχανική μάθηση, δίκτυο, αλγεβρικές μέθοδοι

## **Μηχανική Μάθηση για Διαδικτυακά Δεδομένα**

Μπενοβιας Λ. (Επιβλέπων: Βαρβαρίγος Ε.)

### **Abstract**

The purpose of this work is to study the use of machine learning in computing Internet Data. Also known as Network Tomography, it is the calculation of internal characteristics of the network using information from end-point data.

The purpose of our study is to compare classical algebraic methods for tomography with machine learning (ML) methods. More specifically, we study how with the use of neural networks it is possible, by knowing certain metrics for entire parts of the network, to calculate metrics such as the delay for individual parts of the network, as well as how the neural network and the known information affect this possibility. We compare the efficiency of this method with classical algebraic methods.

For this purpose, the relevant bibliography on Network Tomography is first presented. A computer network is modeled using a graph using the Networkx library (<https://networkx.org/>), while Python and Tensorflow.Keras are used for the neural network.

It is finally concluded that the use of neural networks to compute individual metric computing networks is feasible. Finally, suggestions for further research are listed.

**Key Words:** Network Tomography, neural networks, machine learning, network, algebraic methods

## **Ευχαριστίες**

Με το πέρας της διπλωματικής μου εργασίας και του συνόλου των σπουδών μου στη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου, θα ήθελα να ευχαριστήσω τους καθηγητές μου για τις πολύτιμες γνώσεις που μου παρείχαν. Ένα μεγάλο ευχαριστώ θέλω να εκφράσω στον επιβλέποντα της εργασίας μου, Βαρβαρίγο Εμμανουήλ για την καθοδήγηση του και τη μεγάλη του βοήθεια σε όλο το διάστημα εκπόνησης της διπλωματικής εργασίας, καθώς και την εξεταστική επιτροπή: κ. Θεοδώρα Βαρβαρίγου, και κ. Ηρακλή Αρβραμόπουλο. Θα ήθελα επίσης να ευχαριστήσω και τον Ιπποκράτη Σαρτζετάκη για την μεγάλη του βοήθεια στην διπλωματική εργασία.

Τέλος, θα ήθελα να ευχαριστήσω θερμά την οικογένεια μου και τους φίλους μου για τη στήριξη τους όλα αυτά τα χρόνια.

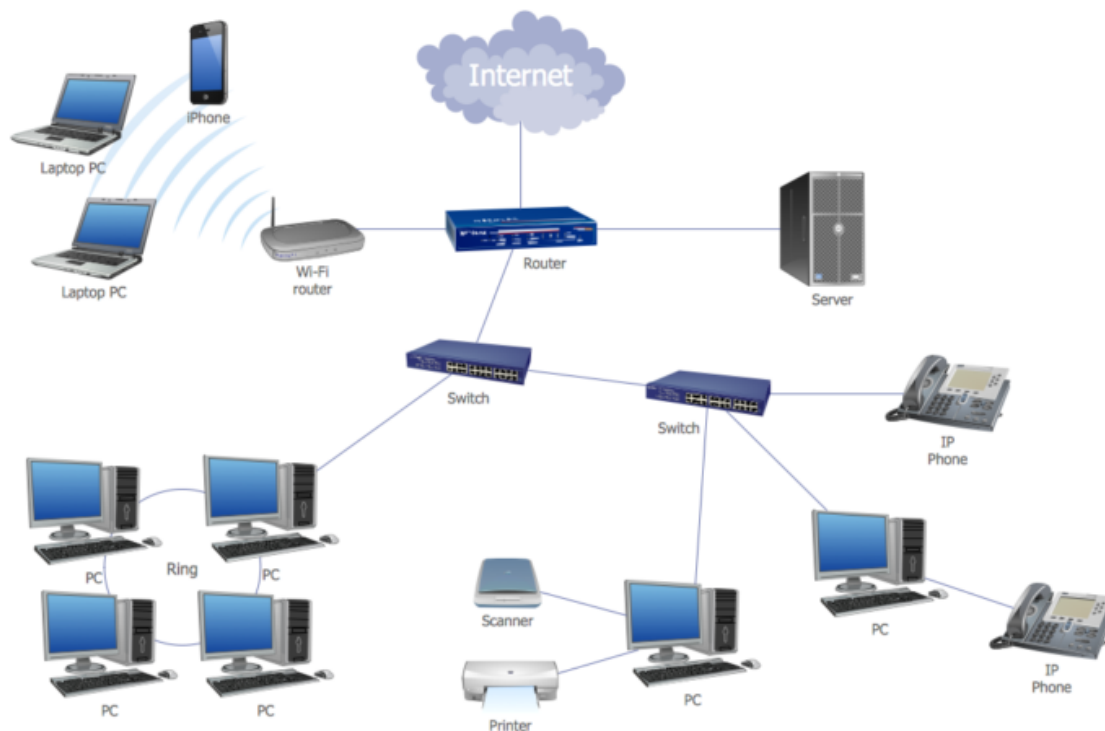
## ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή .....	11
1.1. Δίκτυα Υπολογιστών	
1.2. Μηχανική Μάθηση	
2. Βιβλιογραφική Ανασκόπηση .....	22
2.1 Network Tomography	
2.2 Network Model and Data	
2.3 Network Kriging	
3. Μοντελοποίηση και Μελέτη Προβλήματος .....	26
3.1. Περιγραφή	
3.2. Εκτέλεση	
3.3. Σχέση training και test data στην ακρίβεια του νευρωνικού	
3.4. Πως επηρεάζεται η ακρίβεια του νευρωνικού από την πληροφορία	
3.4.1. Ελλιπής Πληροφορία	
3.4.2. Σχέση της απόδοσης του νευρωνικού με το πλήθος των μονοπατιών	
3.4.3. Σύγκριση πληροφορίας συνδέσμων και κόμβων μονοπατιών	
3.4.4. Απόδοση του νευρωνικού σε σχέση με ανακρίβεια γνωστών δεδομένων	
3.4.5. Σύγκριση με Αλγεβρικές Μεθόδους	
4. Συμπεράσματα .....	51
5. Εισηγήσεις για περαιτέρω έρευνα .....	52
6. Βιβλιογραφία .....	53



# 1. Εισαγωγή

## 1.1 Δίκτυα Υπολογιστών



Ένα δίκτυο υπολογιστών είναι ένα σύνολο υπολογιστών που χρησιμοποιούν ένα κοινό σύστημα πρωτοκόλλων επικοινωνίας πάνω σε ψηφιακές διασυνδέσεις με στόχο την ανταλλαγή πληροφοριών από και προς τους κόμβους του δικτύου. Οι κόμβοι αυτοί του δικτύου μπορεί να είναι προσωπικοί υπολογιστές, servers, networking hardware ή γενικής χρήσης hosts. Αυτοί χαρακτηρίζονται από ξεχωριστά hostnames και network addresses, τα hostnames είναι σαν labels για τους κόμβους και σπανίως αλλάζουν, ενώ τα network addresses συμβάλλουν στον εντοπισμό και προσδιορισμό των κόμβων από τα πρωτόκολλα επικοινωνίας όπως το Internet Protocol.

Τα δίκτυα λοιπόν καθορίζονται από κάποια βασικά χαρακτηριστικά που ορίζουν την Αρχιτεκτονική του δικτύου (Network architecture). Η Αρχιτεκτονική του δικτύου περιλαμβάνει χαρακτηριστικά όπως, η τοπολογία του δικτύου, το μέσο μετάδοσης

του σήματος, τα πρωτόκολλα επικοινωνίας που καθορίζουν και τον τρόπο μετάδοσης της πληροφορίας, το μέγεθος του δικτύου, επιμέρους χαρακτηριστικά όπως το bandwidth και τους μηχανισμούς ελέγχου της μετάδοσης την πληροφορίας.

1. Ο πρώτος τυπικός διαχωρισμός που συναντάται στα δίκτυα υπολογιστών προκύπτει από την γεωγραφική έκταση που καλύπτει το δίκτυο. Οι δυο βασικές κατηγορίες είναι τα τοπικά δίκτυα (Local Area Networks - LAN) και τα δίκτυα ευρείας περιοχής (Wide Area Networks - WAN). Τα τοπικά δίκτυα συνήθως βρίσκονται σε ένα γραφείο, ένα σχολείο ή ένα εργαστήριο και περιορίζονται σχεδόν πάντα στα όρια ενός κτιρίου. Οι ταχύτητες μετάδοσης στα Τοπικά δίκτυα είναι αρκετά υψηλές. Ένα Δίκτυο Ευρείας Περιοχής μπορεί να καλύπτει μία ομάδα κτιρίων (π.χ. βιομηχανίες πανεπιστήμια, κ.λ.π.) ή να καλύπτει την έκταση μίας πόλης ή ακόμη να γεφυρώνει δύο ηπείρους. Συνήθως ένα Δίκτυο Ευρείας Περιοχής χρησιμοποιείται για την διασύνδεση απομακρυσμένων Τοπικών Δικτύων. Το γνωστότερο παράδειγμα δικτύου ευρείας περιοχής είναι το Διαδίκτυο (Internet).

Επίσης υπάρχει μία ενδιαμέση ενδιαφέρουσα κατηγορία δικτύων υψηλής ταχύτητας, τα μητροπολιτικά δίκτυα (Metropolitan Area Networks - MAN). Καλύπτουν γεωγραφικά μία πόλη ή μία μητροπολιτική περιοχή με γειτονικές πόλεις και προάστεια και χρησιμοποιούν σαν μέσο μετάδοσης οπτικές ίνες, σε αυτά τα δίκτυα επίσης υπάρχουν αρκετά υψηλές ταχύτητες μετάδοσης της πληροφορίας.

2. Η τοπολογία λοιπόν του δικτύου είναι ο τρόπος σχεδιαστικά και οργανωτικά που είναι συνδεδεμένοι οι διάφοροι hosts του δικτύου σε αντίθεση με τον πραγματικό φυσικό τους τόπο (Network Topology). Αυτή λοιπόν η διαγραμματική οργάνωση του δικτύου επηρεάζει την αποδοτικότητα, την ασφάλεια και γενικά την μορφή του δικτύου. Υπάρχουν τοπολογίες που πλεονεκτούν σε συγκεκριμένα χαρακτηριστικά και υστερούν σε άλλα. Για παράδειγμα μια τοπολογία "bus network" όπου όλοι οι κόμβοι του δικτύου συνδέονται σε ένα κοινό κεντρικό δίαυλο επικοινωνίας από όπου περνά όλη η πληροφορία. Είναι μια απλή και οικονομική εγκατάσταση και χρησιμοποιείται σε μικρά δίκτυα όπως αυτό ενός σπιτιού, έχει όμως μειονεκτήματα όπως ότι μπορεί μια μόνο αποτυχία στον κεντρικό δίαυλο να οδηγήσει ολόκληρο το

δίκτυο σε πλήρη αποτυχία. Ο γενικός κανόνας ότι οι περισσότερες διασυνδέσεις μεταξύ κόμβων κάνουν το δίκτυο πιο ασφαλές και αξιόπιστο αλλά και ταυτόχρονα είναι πιο ακριβό στην εγκατάσταση.

Κάποιες ενδεικτικές τοπολογίες που συνηθίζονται είναι οι: Bus network, Star network, Ring network, Mesh network, Fully connected network, Tree network.

3. Όταν δύο συστήματα επικοινωνούν με οποιονδήποτε τρόπο (δηλαδή ανταλλάσσουν πληροφορίες), θα πρέπει να ακολουθούν ένα πρωτόκολλο. Το Πρωτόκολλο (Protocol) είναι μία συλλογή κανόνων που διέπουν την επικοινωνία μεταξύ συσκευών, κυκλωμάτων ή ακόμη και τμημάτων λογισμικού. Το σύνολο των κανόνων επικοινωνίας ονομάζεται ειδικότερα Πρωτόκολλο Επικοινωνίας. Οι κανόνες αυτοί καθορίζουν τη μορφή, το χρόνο και τη σειρά μετάδοσης των πληροφοριών στο δίκτυο. Εκτελούν, επίσης, έλεγχο και διόρθωση σφαλμάτων στη διάρκεια μετάδοσης των πληροφοριών. Το πιο γνωστό σύνολο από τέτοια πρωτόκολλα επικοινωνίας είναι το TCP/IP (Transmission Control Protocol/Internet Protocol ) που χρησιμοποιείται για την επικοινωνία των συσκευών που συνδέονται στο διαδίκτυο, αλλά και συχνά ως πρωτόκολλο επικοινωνίας σε εμπορικά δίκτυα. Σύνολα πρωτοκόλλων όπως και το TCP/IP διαχωρίζονται σε επιμέρους στάδια (layers) που επιτελούν διαφορετικές διαδικασίες. Μια βασική λοιπόν διαδικασία είναι και το Routing, δηλαδή η διαδικασία επιλογής μιας διαδρομής (path) για την μετάδοση της πληροφορίας σε ένα δίκτυο ή μεταξύ πολλών δικτύων και συνήθως βρίσκεται στο network access layer.

Συνοπτικά αναφέρουμε ότι το routing πρωτίστως διαχωρίζεται ανάλογα με το πως στέλνουν τα δεδομένα, δηλαδή σε:

- Unicast, όπου η πληροφορία μεταδίδεται από ένα κόμβο σε έναν συγκεκριμένο κόμβο
- Broadcast, η πληροφορία μεταδίδεται σε όλους τους κόμβους του δικτύου
- Multicast, η πληροφορία μεταδίδεται σε ένα σύνολο από κόμβους
- Anycast, η πληροφορία μεταδίδεται σε ένα συγκεκριμένο κόμβο από ένα σύνολο κόμβων

Το αλγοριθμικό κομμάτι του routing όπως αναφέρθηκε αφορά στον τρόπο επιλογής του μονοπατιού. εδώ λοιπόν, οι βασικές κατηγορίες routing μεθόδων διαχωρίζονται σε:

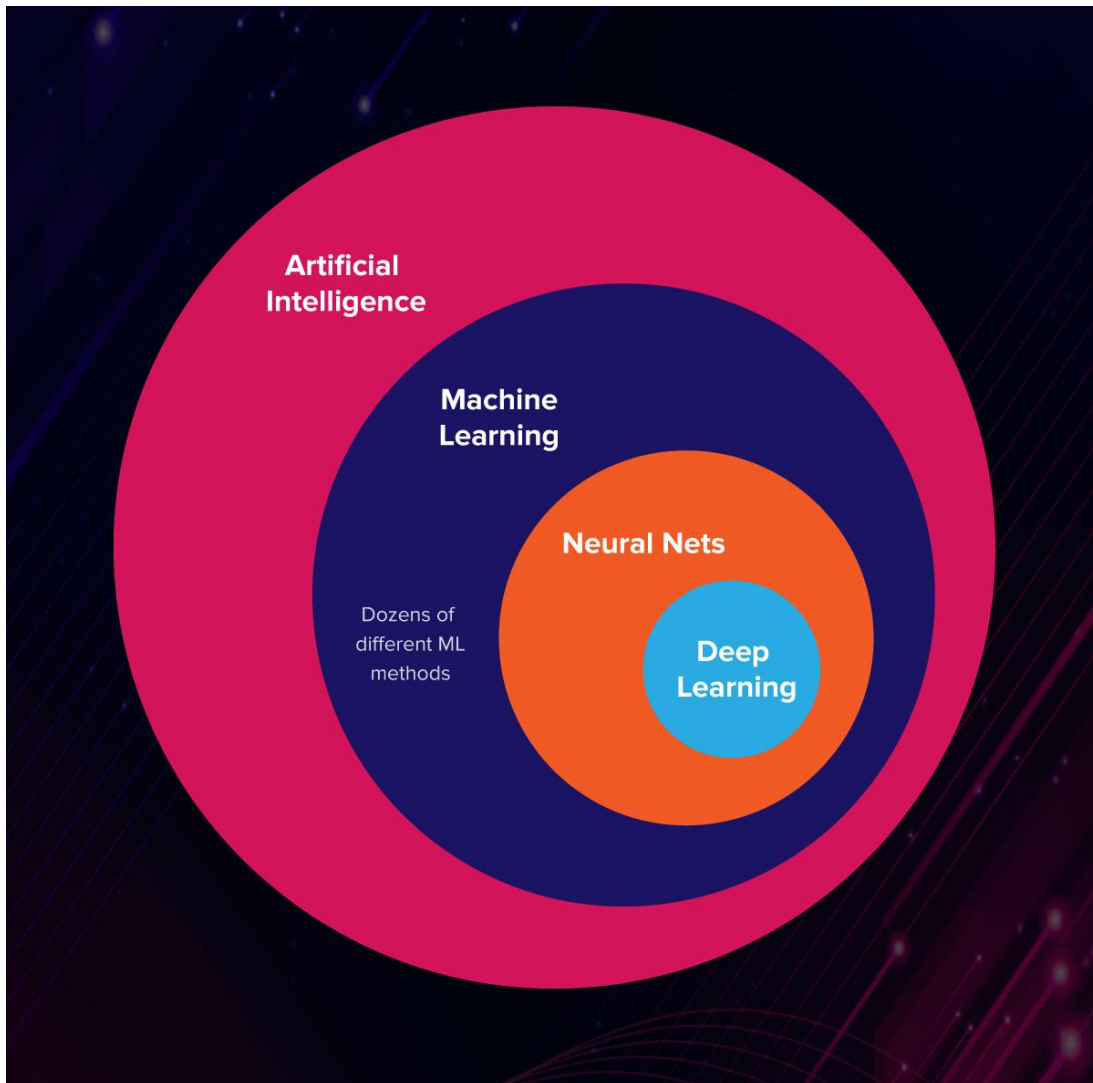
- Prefix-length, όπου προτιμώνται οι κόμβοι με μακρύτερα subnet masks (ανεξάρτητα αν πρόκειται για το ίδιο ή διαφορετικά πρωτόκολλα)
- Metric, όπου συνήθως προτιμούνται οι κόμβοι/σύνδεσμοι με το μικρότερο κόστος για κάποια συγκεκριμένη μετρική (πρόκειται για ένα και μόνο πρωτόκολλο)
- Administrative distance, η μικρότερη απόσταση είναι αυτή που επιλέγεται (εδώ όμως εννοούμε μεταξύ διαφορετικών πρωτοκόλλων).

4. Τέλος βασικά χαρακτηριστικά που καθορίζουν την απόδοση ενός δικτύου, κάποια από τα οποία αναφέρθηκαν και παραπάνω, είναι:

- Quality of Service, με τον όρο αυτό εννοούμε το σύνολο των μηχανισμών και τεχνολογιών που λειτουργούν σε ένα δίκτυο με σκοπό τον έλεγχο της κυκλοφορίας της πληροφορίας ώστε να διασφαλίσουν την αποδοτική λειτουργία των εφαρμογών ιδίως σε περιπτώσεις που η χωρητικότητα του δικτύου είναι περιορισμένη. Συνήθως εννοείται η βελτιστοποίηση της λειτουργίας του δικτύου για την εκτέλεση πολλαπλών εφαρμογών δίνοντας την ανάλογη προτεραιότητα ή καθορίζοντας την ανάθεση των πόρων του δικτύου με τον κατάλληλο τρόπο. Εδώ πρέπει επίσης να αναφερθεί ότι πολλές φορές η απόδοση του δικτύου εξαρτάται από τον σκοπό του. Ένα ενδεικτικό παράδειγμα αυτού είναι τα circuit switched networks για τα οποία συχνά η απόδοση αναφέρεται και ως grade of service και μπορεί για παράδειγμα ο αριθμός των απορριφθέντων κλήσεων να είναι ένα μέτρο σύγκρισης της απόδοσης του δικτύου.
- Bandwidth, μετριέται συνήθως σε bit/s και αναφέρεται στον μέσο ρυθμό με τον οποίο μεταδίδεται η πληροφορία μέσω ενός μονοπατιού ή ακόμη και ενός συνδέσμου. Συχνά αναφερόμαστε σε αυτό και ως throughput. Διαφορετικές τεχνολογίες και αλγόριθμοι επηρεάζουν το τελικό bandwidth που ανατίθεται σε κάθε μέρος ενός δικτύου, και συχνά συναντώνται και δυναμικές μέθοδοι ανάθεσης bandwidth (dynamic bandwidth allocation). Αυτά συνολικά συντελούν στο τελικό throughput του δικτύου.

- Network Delay, όπως είναι προφανές αναφέρεται στον χρόνο που θέλει η πληροφορία για να μεταβεί από την πηγή στον προορισμό και σαφώς αποτελεί σημαντικό μέρος της απόδοσης ενός δικτύου. Ο χρόνος αυτός συχνά αναφέρεται και με τον όρο Network Latency ή lag και μετριέται συνήθως σε milliseconds (ms), ενώ μπορεί να αναφερθεί και ως ping rate. Ο σκοπός είναι πάντα να επιτυγχάνουμε χαμηλό lag, ιδιαίτερα όταν ο χρόνος είναι μικρός αναφερόμαστε σε low-latency network ενώ για μεγάλη καθυστέρηση σε high-latency network. Ενδεικτικά συνηθίζεται κάτω από τα 100ms να θεωρείται ικανοποιητικός ο χρόνος μετάδοσης ενός πακέτου.
- Network congestion, συνήθως προκαλείται όταν το δίκτυο προσπαθεί να εξυπηρετήσει περισσότερη πληροφορία από ότι μπορεί. Έτσι προκύπτει περίπτωση “συνωστισμού” πληροφορίας σε συγκεκριμένο κόμβο ο οποίος δεν καταφέρνει να την εξυπηρετήσει, με αποτέλεσμα περαιτέρω καθυστέρηση (queueing delay), απώλεια πληροφορίας (packet loss) ή ακόμη και μπλοκάρισμα νέων συνδέσμων. Σύγχρονες μέθοδοι αντιμετώπισης τέτοιου είδους προβλημάτων έχουν αναπτυχθεί όπως αλγόριθμοι για congestion control ή traffic control.
- Network resilience, τέλος σημαντική είναι η ικανότητα του δικτύου να συνεχίσει να λειτουργεί σε ένα αποδεκτό επίπεδο και στις περιπτώσεις σφαλμάτων ή άλλων εμποδίων στην ομαλή του λειτουργία. Υπάρχουν περιπτώσεις σφαλμάτων που είναι αναμενόμενα αλλά και άλλων που είναι αναπάντεχα, ο στόχος ωστόσο είναι το δίκτυο να αντιμετωπίζει και τις δυο περιπτώσεις προβλημάτων όσο το δυνατόν γρηγορότερα και αποδοτικότερα.

## 1.2 Μηχανική Μάθηση



Η μηχανική μάθηση (Machine Learning) είναι ένα τμήμα του AI (artificial Intelligence) που ασχολείται με τη δημιουργία μεθόδων μάθησης που επιτρέπουν σε υπολογιστικά συστήματα να “μάθουν” μέσα από ένα σύνολο δεδομένων και επιδιώκουν να μιμηθούν τις λειτουργίες του ανθρώπινου εγκεφάλου. Πιο συγκεκριμένα με την επεξεργασία των δεδομένων και χρήση υπολογιστικών μεθόδων είναι εφικτό να αποκομίσουν γνώση για συγκεκριμένα προβλήματα χωρίς προκαθορισμένα μαθηματικά μοντέλα. Ενδεικτικοί τομείς χρήσης μεθόδων μηχανικής μάθησης είναι η αναγνώριση φωνής (speech recognition), computer vision και autonomous driving vehicles που μελετούνται εκτενώς και συνεχώς εξελίσσονται.

Το machine learning είναι ουσιαστικά η δημιουργία και εκπαίδευση ενός μοντέλου (π.χ νευρωνικό δίκτυο) πάνω σε ένα σύνολο δεδομένων (training data) με σκοπό την κατηγοριοποίηση (classification) ή την πρόβλεψη (prediction) δεδομένων και κατά συνέπεια την λήψη αποφάσεων (decision making) σε ορισμένες εφαρμογές και επιχειρήσεις, χωρίς όμως το μοντέλο να είναι ρητά προγραμματισμένο για μια συγκεκριμένη διαδικασία.

Ένας αλγόριθμος μηχανικής μάθησης μπορεί να διαχωριστεί σε τρία κύρια μέρη.

- Decision making: Γενικά, όπως αναφέρθηκε οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται για να κάνουν μια πρόβλεψη (prediction) ή κατηγοριοποίηση (classification). Σε αυτό το στάδιο με βάση ορισμένα δεδομένα εισόδου, τα οποία μπορούν να είναι labelled ή unlabelled, ο αλγόριθμος παράγει μια εκτίμηση σχετικά με ένα μοτίβο (pattern) στα δεδομένα.
- Error Function: Μια συνάρτηση σφάλματος που αξιολογεί την πρόβλεψη του μοντέλου. Εάν υπάρχουν γνωστά παραδείγματα, μια συνάρτηση σφάλματος μπορεί να κάνει σύγκριση για να αξιολογήσει την ακρίβεια του μοντέλου.
- Model Optimization Process: Στόχος είναι η βελτιστοποίηση του μοντέλου, για αυτό το σκοπό τα βάρη προσαρμόζονται για να μειωθεί η απόκλιση μεταξύ του γνωστού παραδείγματος και της εκτίμησης του μοντέλου. Ο αλγόριθμος θα επαναλάβει αυτή τη διαδικασία «αξιολόγησης και βελτιστοποίησης», ενημερώνοντας τα βάρη αυτόνομα μέχρι να επιτευχθεί ένα όριο ακρίβειας.

Ο διαχωρισμός ανάμεσα σε Machine learning και Deep learning, αφορά κυρίως στον τρόπο με τον οποίο μαθαίνει το σύστημα. Πιο συγκεκριμένα στην πραγματικότητα όπως ήδη αναφέρθηκε το Machine Learning είναι μια υποκατηγορία του AI, ενώ το Deep Learning είναι μια υποκατηγορία των Neural Networks τα οποία αποτελούν υποκατηγορία του Machine Learning. Το κλασικό machine learning συνήθως χρειάζεται την ανθρώπινη συμβολή στον καθορισμό των χαρακτηριστικών των δεδομένων στα οποία εκπαιδεύεται το σύστημα και κατά συνέπεια απαιτεί δομημένα δεδομένα. Το deep learning ωστόσο δεν προϋποθέτει κάποια δομή στα δεδομένα, παίρνει raw data και αποφασίζει μόνο του το σύνολο των χαρακτηριστικών που τα διαφοροποιούν κατά συνέπεια εξαλείφει την ανάγκη για το ανθρώπινο παράγοντα στη διαδικασία της μάθησης.

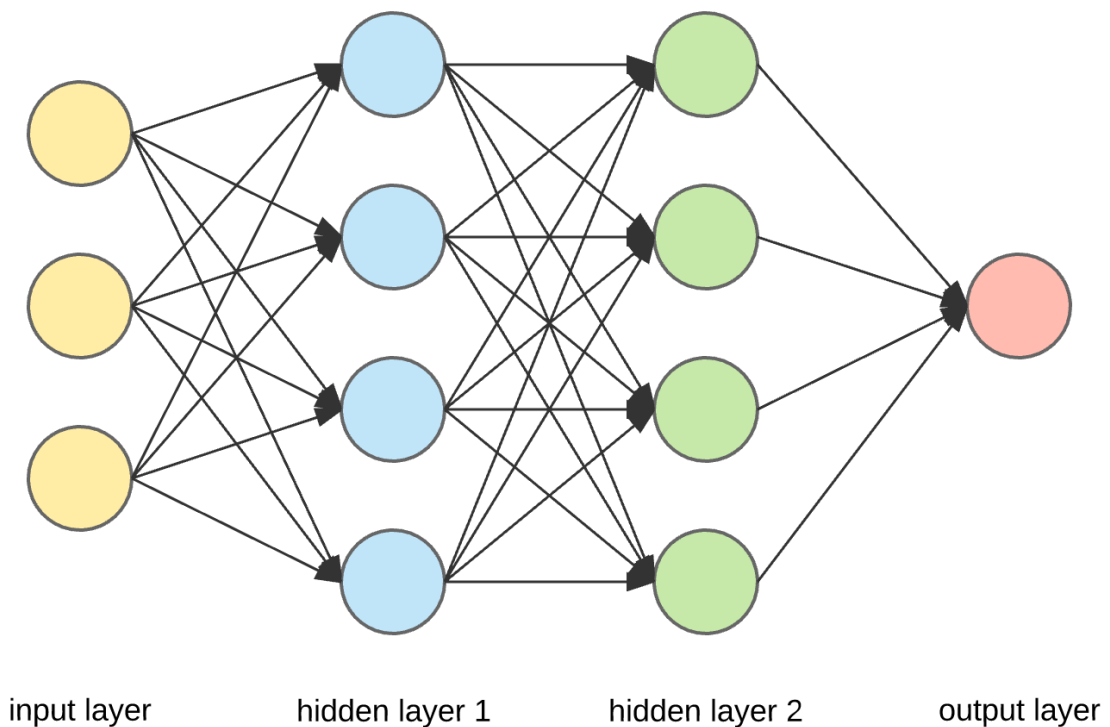
Ένας ακόμη τρόπος να διαχωρίσουμε τις κατηγορίες μηχανικής μάθησης είναι σε “supervised” και “unsupervised” learning.

Στην περίπτωση του Supervised learning, γνωρίζουμε το input και output για συγκεκριμένες περιπτώσεις και επιδιώκουμε την κατασκευή ενός μοντέλου ικανού να κάνει λογικές προβλέψεις με βάση τα παραπάνω για νέα δεδομένα. Οι δύο βασικές κατηγορίες αλγορίθμων είναι classification όπου κατατάσσονται τα δεδομένα σε κατηγορίες, όπως για παράδειγμα στην αναγνώριση φωνής (speech recognition), και regression μέθοδοι που στοχεύουν στην εύρεση μαθηματικών σχέσεων μεταξύ δυο μεταβλητών input και output, μια τέτοια εφαρμογή είναι το algorithmic trading.

Στο Unsupervised Learning το σύστημα προσπαθεί να εκπαιδευτεί στον εντοπισμό κρυφών σχέσεων ή εγγενών δομών στα δεδομένα και στην παραγωγή αποτελεσμάτων χωρίς γνωστά output. Μια γνωστή κατηγορία τέτοιων αλγορίθμων είναι το clustering, όπου στόχος είναι ο διαχωρισμός των δεδομένων σε ομάδες (clusters) αφού εντοπιστούν συγκεκριμένα patterns στα δεδομένα.

Όπως ήδη αναφέρθηκε, ένας σημαντικός κλάδος της μηχανικής μάθησης είναι και αυτός που ασχολείται με την πρόβλεψη αποτελεσμάτων μέσω της χρήσης νευρωνικών δικτύων που μιμούνται τον τρόπο σκέψης του ανθρώπινου εγκεφάλου. Τα νευρωνικά δίκτυα (neural networks) αποτελούνται από ένα σύνολο τεχνητών νευρώνων όπως ο ανθρώπινος εγκέφαλος αποτελείται από βιολογικούς νευρώνες και μπορεί να χρησιμοποιηθούν είτε σε περιπτώσεις supervised ή και unsupervised learning. Το deep learning που αναφέρθηκε παραπάνω αφορά στο πλήθος των διάφορων layer του νευρωνικού δικτύου, και συγκεκριμένα για πάνω από 3 layer θεωρείται deep neural network.





Στην εικόνα βλέπουμε ένα ενδεικτικό μοντέλο νευρωνικού δικτύου. Για να μοντελοποιηθεί λοιπόν η λειτουργία του εγκεφάλου, χρησιμοποιούνται 1 input layer όπου δίνεται η είσοδος (input) στο σύστημα, ακολουθούν 1 ή περισσότερα κρυφά (hidden) layers τα οποία αποτελούνται από νευρώνες (κόμβους) οι οποίοι “συνδέονται” με διαφορετικά βάρη (“weights”) μεταξύ τους που καθορίζουν την επιρροή κάθε νευρώνα στο τελικό αποτέλεσμα. Αλγεβρικά αυτό σημαίνει ότι σε κάθε layer πολλαπλασιάζουμε το input με τα weights του κάθε νευρώνα (έπειτα μπορούμε να προσθέσουμε μια σταθερά “bias”) και στην συνέχεια χρησιμοποιούμε ένα activation function για το τελικό αποτέλεσμα κάθε νευρώνα, δηλαδή:

$f(w * x + b)$ , όπου  $f$  το activation function,  $w$  = weights,  $x$ = input και  $b$ =bias. Το τελευταίο layer είναι αυτό στο οποίο παίρνουμε το τελικό αποτέλεσμα του νευρωνικού δικτύου (output) και συχνά είναι ένα τελικό layer (Dense layer) όπου αθροίζουμε τα αποτελέσματα των νευρώνων για να πάρουμε το τελικό αποτέλεσμα.

Αυτή τη διαδικασία που μόλις περιγράφηκε ονομάζεται “forward propagation” και οδηγεί στον υπολογισμό του αποτελέσματος. Αφού αυτό συγκριθεί με το επιθυμητό αποτέλεσμα μπορούμε να εκτελέσουμε την αντίστροφη διαδικασία “back-propagation” μέσω της οποίας με χρήση συγκεκριμένων αλγορίθμων μπορούμε να επηρεάσουμε τα βάρη (weights) ώστε όταν ξανα-εκτελέσουμε τη διαδικασία πρόβλεψης να έχουμε καλύτερη προσέγγιση του αποτελέσματος.

Η επιλογή του αλγορίθμου για την διαδικασία του back propagation είναι παρά πολύ σημαντική για την ακριβή και ταχεία απόκριση του νευρωνικού δικτύου, ενδεικτικοί αλγόριθμοι back-propagation είναι οι: Stochastic Gradient Descent, Adam, Adagrad, και άλλοι που βελτιστοποιούν αυτήν την διαδικασία για γρηγορότερη ή ακριβέστερη πρόβλεψη.

Η βασική ιδέα του Gradient Descent (GD) είναι η ακόλουθη: ορίζουμε μια συνάρτηση κόστους που εκφράζει την απόσταση από το επιθυμητό αποτέλεσμα, και ο στόχος είναι η ελαχιστοποίηση αυτής της συνάρτησης. Ξεκινώντας λοιπόν από ένα starting point, εάν υπολογίσουμε την κλίση της συνάρτησης, η οποία από τα μαθηματικά δίνεται από την πρώτη παραγωγό, και ακολουθήσουμε αυτή την κλίση ανανεώνοντας τις παραμέτρους, σταδιακά θα μειώσουμε την κλίση μέχρις ότου η παραγωγός τείνει στο 0 όπου και έχουμε σύγκλιση σε τοπικό (ή ολικό) ελάχιστο. Οι δύο βασικές παράμετροι που επηρεάζουν τον αλγόριθμο Gradient Descent (GD), είναι το learning rate και η επιλογή του cost function. Το πρώτο αφορά στον ρυθμό με τον οποίο μεταβάλλουμε τις παραμέτρους σε κάθε επανάληψη και επηρεάζει την ταχύτητα σύγκλισης, ενώ το δεύτερο επηρεάζει την ακρίβεια της σύγκλισης καθώς είναι ο τρόπος που υπολογίζουμε την απόκλιση από το επιθυμητό αποτέλεσμα.

Συνήθως στην επιλογή του κατάλληλου αλγορίθμου υπάρχει ένα trade-off ανάμεσα στην ταχύτητα και την ακρίβεια των προβλέψεων του νευρωνικού.

Τέλος για να κατανοήσουμε πως αξιολογείται ένα νευρωνικό δίκτυο οφείλουμε να αναφέρουμε τις βασικές κατηγορίες αξιολόγησης:

- Convergence, ταχύτητα σύγκλισης του νευρωνικού δικτύου

- Precision, ακρίβεια σύγκλισης του νευρωνικού δικτύου
- Robustness, να δίνει ικανοποιητικά αποτελέσματα σε πολλαπλές και όχι μόνο σε ειδικές περιπτώσεις
- General Complexity, γενικότερη πολυπλοκότητα του δικτύου (όπως η αυξημένη πολυπλοκότητα ενός deep neural net)

Και πρέπει ακόμη να αναφερθούν και δύο βασικά προβλήματα που συχνά παρατηρούνται στα νευρωνικά δίκτυα και αφορούν στο robustness:

- Overfitting, όταν εκπαιδεύσουμε υπερβολικά ένα δίκτυο σε συγκεκριμένα δεδομένα, μπορεί αυτό να αποκτήσει bias σε αυτά και έτσι να δίνει πολύ καλή ακρίβεια σε αυτά τα δεδομένα αλλά να μην μπορεί να επιτύχει ικανοποιητική ακρίβεια σε νέα δεδομένα, δηλαδή έχει μικρό σφάλμα στα training data, αλλά μεγάλο σφάλμα στα testing data
- Underfitting, στην αντίθετη περίπτωση, όπου εκπαιδεύουμε λιγότερο το νευρωνικό δίκτυο, ελλιπή δεδομένα ή όχι αρκετός χρόνος εκπαίδευσης, τότε το νευρωνικό συχνά δεν μπορεί να εντοπίσει τις απαραίτητες σχέσεις μεταξύ των δεδομένων εισόδου και εξόδου με αποτέλεσμα το σφάλμα να είναι μεγάλο και στα training αλλά και στα testing data

## 2. Βιβλιογραφική ανασκόπηση

Μετά από έρευνα στο αντικείμενο βρέθηκε ότι υπάρχουν αρκετές βιβλιογραφικές αναφορές στο θέμα είναι από χώρες όπως οι ΗΠΑ και Ευρωπαϊκές (π.χ Γαλλία) ενώ επίσης άρθρα έχουν δημοσιευθεί από Ινδούς και Κινέζους ακαδημαϊκούς, καθώς επίσης υπάρχει και ελληνική βιβλιογραφία στο αντικείμενο.

Έχουν γίνει ήδη κάποιες μελέτες στο τομέα του Network Tomography.

Σημαντικές μελέτες έχουν γίνει για την αλγεβρική επίλυση του προβλήματος [1],[2]. Μια ιδιαίτερη αναφορά πρέπει να γίνει και στο network Kriging [3]. Οι μελέτες αυτές εμβαθύνουν ακόμη και στην αξιοποίηση τέτοιων μεθόδων στο περιορισμό των δεδομένων που απαιτούνται για ικανοποιητικό προσδιορισμό με γραμμική άλγεβρα καθώς σε μεγάλα δίκτυα το πλήθος των μονοπατιών αυξάνεται εκθετικά σε σχέση με το σύνολο των κόμβων.

Έπειτα πολλά Paper μελετάνε την χρήση νευρωνικών δικτύων για τον υπολογισμό δεδομένων από end-to-end measurements και κάνουν και επεκτάσεις για γενίκευση σε περιπτώσεις χωρίς συγκεκριμένες πληροφορίες για το δίκτυο, καθώς ακόμη προχωρούν και σε προσπάθειες ανακατασκευής του δικτύου [5]. Άλλες μελέτες αφορούν στην χρήση neural networks για υπολογισμό κυρίως προσθετικών μετρικών σε υποθετικά υπολογιστικά δίκτυα (SDN, NFV) καθώς και την σχέση του πλήθους των hidden layer, των νευρώνων και του μεγέθους των δεδομένων με την ικανότητα απόκρισης του νευρωνικού δικτύου [6], [7].

Τέλος υπάρχουν και βιβλιογραφικές μελέτες που προχωρούν στην χρήση νευρωνικών δικτύων για τον εντοπισμό ανωμαλιών και τον ακριβή προσδιορισμό τους για να μπορέσει να εξασφαλιστεί η ομαλή και ασφαλής λειτουργία του δικτύου [8], [9].

Το αντικείμενο αυτό όμως έχει ακόμη πολλά περιθώρια περαιτέρω έρευνας, για αυτό και δίκη μας εργασία προσπαθεί να μελετήσει κάποια παραπάνω

χαρακτηριστικά στην χρήση των νευρωνικών δικτύων για την πρόβλεψη επιμέρους δεδομένων του δικτύου.

## 2.1 Network Tomography

Το Network Tomography είναι ένα ευρέως γνωστό πρόβλημα στον τομέα του network monitoring και ουσιαστικά αφορά στον υπολογισμό εσωτερικών χαρακτηριστικών του δικτύου χρησιμοποιώντας πληροφορία από δεδομένα τελικού σημείου (end-point data).

Ο σκοπός του προβλήματος είναι να αξιοποιήσουμε την πληροφορία που μπορούμε να μετρήσουμε για τελικά σημεία του δικτύου (end point measurements), για να κατανοήσουμε την συνολική κατάσταση του δικτύου και να κατανοήσουμε την πορεία της πληροφορίας σε αυτό. Αυτό μας δίνει την δυνατότητα να διευκολύνουμε την διαδικασία ελέγχου και επίλυσης πιθανόν προβλημάτων που μπορεί να υπάρχουν (π.χ network congestion, faulty nodes). Η σωστή αξιοποίηση την γνωστής πληροφορίας και η κατανόηση της συνολικής κατάστασης του δικτύου είναι πολύ σημαντικές για την αποτελεσματική λειτουργία του δικτύου και άρα την ικανοποιητική ποιότητα επικοινωνίας μεταξύ των χρηστών.

Πιο συγκεκριμένα, επιτυγχάνοντας το υπολογισμό των μετρικών για τα εσωτερικά τμήματα του δικτύου όπως για παράδειγμα η καθυστέρηση (delay) ή ο ρυθμός που χάνει πακέτα (packet loss ratio) ένας κόμβος από τις end-to-end μετρήσεις, το αντικείμενο δηλαδή του Network Tomography, μπορούμε να κατανοήσουμε ποιο τμήμα του δικτύου οφείλεται για την κακή λειτουργία ενός δικτύου και πόσο επηρεάζει το εκάστοτε τμήμα την ομαλή λειτουργία του δικτύου.

Για να επιτευχθεί ο υπολογισμός των επιμέρους χαρακτηριστικών του δικτύου συνήθως γνωρίζοντας την τοπολογία του δικτύου, χρησιμοποιείται το routing των επιμέρους end-to-end μετρήσεων, καθώς επίσης και μετρήσεις για πληροφορίες που διέρχονται από όλους τους κόμβους του δικτύου, καθώς και μετρήσεις που αφορούν τη ροή της πληροφορίας στο δίκτυο όπως το delay και το packet-loss-

ratio που αναφέρθηκαν. Καθότι το routing είναι εφικτό να μεταβάλλεται δυναμικά σε ένα δίκτυο χρησιμοποιώντας network virtualization, μπορούμε αφότου εντοπίσουμε προβληματικά τμήματα του δικτύου να μεταβάλλουμε το routing επιτυγχάνοντας αποτελεσματικότερη λειτουργία στο δίκτυο.

## 2.2 Network Model and Data

Ορίζουμε την πληροφορία που καθορίζει την ποιότητα επικοινωνίας (QoS) του δικτύου ως τα end-to-end measurements που κάνουμε για την καθυστέρηση (delay) του δικτύου και επίσης θεωρούμε γνωστή την τοπολογία του δικτύου.

Το δίκτυο μπορεί να οριστεί ως ένας συνδεδεμένος γράφος (connected graph)  $G = \{V, E\}$ , όπου  $V$  το σύνολο των κόμβων (nodes) και  $E$  το σύνολο των αμφίδρομων συνδέσμων (unidirectional links), και κατα συνέπεια το πλήθος των κόμβων και των συνδέσμων αντίστοιχα είναι  $|V|$  και  $|E|$ . Αν θεωρήσουμε ένα σύνολο  $M$  που αποτελείται από τα εγκατεστημένα μονοπάτια και μας δίνει την κατάσταση του δικτύου, τότε μπορούμε να ορίσουμε τον πίνακα routing

$R_M \in \{0,1\}^{\|M\| \times \|E\|}$ , όπου  $R_M[m, l] = 1$  όταν το μονοπάτι  $m$  περιλαμβάνει τον σύνδεσμο  $l$ , αλλιώς είναι 0. Επίσης εάν ορίσουμε με  $y_M \in R^{\|M\|}$  το διάνυσμα που εκφράζει την τιμή του μονοπατιού  $m$  (πχ συνολικό delay του μονοπατιού  $m$ ), τότε το  $y$  γράφεται ως γραμμικός συνδυασμός του  $R$  και των επιμέρους μετρικών των συνδέσμων  $x \in R^{\|E\|}$ , ως  $y_M = R_M x$ .

Εμείς αναζητούμε τα end-to-end measurements ενός νέου συνόλου  $N$  για τα οποία γνωρίζουμε το routing  $R_N$  και άρα μπορούμε να πούμε ότι είναι ίσο με  $y_N = R_N x$ .

Για το πρόβλημα του Network Tomography θεωρούμε ότι δεν γνωρίζουμε τις επιμέρους μετρικές  $x$ , αλλά γνωρίζουμε τα  $y_M, R_M$  και  $R_N$  και αναζητούμε το  $y_N$ .

Μια από τις μεθόδους επίλυσης του δεδομένου προβλήματος είναι και η αλγεβρική με το Network Kriging.

## 2.3 Network Kriging

Αναφέρθηκε ότι ο προσδιορισμός των ζητούμενων μετρικών του δικτύου είναι εφικτό να γίνει αλγεβρικά. Αυτό γίνεται με χρήση μεθόδων γραμμικής πρόβλεψης (linear prediction), καθότι όπως εξηγήσαμε η τιμή μονοπατιού  $y$  μπορεί να εκφραστεί ως γραμμικός συνδυασμός του routing πίνακα  $R$  και των μετρικών των συνδέσμων  $x$ , από την γραμμική άλγεβρα βρίσκουμε ότι η βέλτιστη γραμμική πρόβλεψη ως προς το μέσο τετραγωνικό σφάλμα (MSE), δίνεται από την ακόλουθη εξίσωση:  $y_N = R_N R_M^T (R_M R_M^T)^+ y_M$ , όπου οι πίνακες ορίστηκαν παραπάνω, και με “+” ορίζεται ο ψευδοαντίστροφος πίνακας, όπως δηλαδή ο Moore-Penrose inverse. Η πολυπλοκότητα της συγκεκριμένης επίλυσης είναι  $O(\|M\|^3)$  όπου  $\|M\|$  το πλήθος των εγκατεστημένων μονοπατιών.

Περαιτέρω έρευνα στον τομέα του Network Kriging όπως αναφέρθηκε αφορά στον περιορισμό του πλήθους των “γνωστών” μονοπατιών (measured paths) που απαιτούνται για τον προσδιορισμό με ικανοποιητική ακρίβεια όλων των μονοπατιών. Αυτό είναι σημαντικό, καθώς το πλήθος των μονοπατιών ενός δικτύου αυξάνεται εκθετικά σε σχέση με το πλήθος των κόμβων. Βασικές ιδέες μέχρι στιγμής επιτυγχάνουν σημαντικές βελτιώσεις στον περιορισμό αυτόν, βασιζόμενες στην εγγενή φύση των δικτύων να χρησιμοποιούν συγκεκριμένους κόμβους σε μεγάλο πλήθος μονοπατιών.

Μια πιο συγκεκριμένη ιδέα είναι να μειωθούν τα μονοπάτια με βάση το πλήθος των virtual nodes (identifiable link subsets) του δικτύου και αποδεικνύεται ότι πράγματι επαρκεί για να ανακτήσουμε με ακρίβεια για το πλήρες σύνολο μονοπατιών τα end-to-end χαρακτηριστικά.

### 3. Μοντελοποίηση και Μελέτη Προβλήματος

#### 3.1 Περιγραφή

Όπως λοιπόν έγινε κατανοητό ο στόχος του Network Tomography είναι ο υπολογισμός χαρακτηριστικών του συνολικού δικτύου από μετρήσεις των χαρακτηριστικών αυτών σε συγκεκριμένα τμήματα του δικτύου. Είναι εδώ σημαντικό να αναφερθεί ότι τα χαρακτηριστικά αυτά, μπορεί να είναι additive metrics ή non-additive metrics. Στην πρώτη περίπτωση αναφερόμαστε σε χαρακτηριστικά όπου το συνολικό αποτέλεσμα είναι το άθροισμα των επιμέρους μετρικών των τμημάτων του δικτύου, όπως είναι το delay ή ακόμη και το packet loss ratio (μπορεί να εκφραστεί ως άθροισμα, εάν εφαρμόσουμε τον λογάριθμο, καθώς  $\log(x*y) = \log(x) + \log(y)$  και  $\log(x/y) = \log(x) - \log(y)$ ), ενώ στην περίπτωση των non-additive, πρόκειται για μετρικές των οποίων το συνολικό αποτέλεσμα δεν εκφράζεται ως γραμμικός συνδυασμός των επιμέρους, ένα τέτοιο παράδειγμα είναι το network congestion, όπου ο βαθμός congestion ενός μονοπατιού ορίζεται από τον “χειρότερο” κόμβο. Η δική μας μελέτη αφορά την υποκατηγορία των additive μετρικών .

Η εργασία αυτή επιχειρεί να εξετάσει τη χρήση νευρωνικών δικτύων στον υπολογισμό additive μετρικών για το συνολικό δίκτυο εφόσον γνωρίζουμε τις αντίστοιχες μετρικές για ένα περιορισμένο πλήθος μονοπατιών.

#### 3.2 Υλοποίηση

Για να πραγματοποιήσουμε την μελέτη, χρειάζεται να μοντελοποιήσουμε τόσο το νευρωνικό δίκτυο όσο και την τοπολογία του δικτύου του οποίου τα δεδομένα θέλουμε να υπολογίσουμε.

Αρχικά, για να μοντελοποιήσουμε την τοπολογία του δικτύου, όπως αναφέρθηκε και παραπάνω χρειάζεται η κατασκευή ενός γράφου που απεικονίζει την νοητή αυτή τοπολογία. Αυτό υλοποιείται ακολούθως:



Χρησιμοποιούμε συνάρτηση που φτιάχνει έναν τυχαίο γράφο μέσω συναρτήσεων του networkx library (<https://networkx.org/>).

“  $G = nx.dense\_gnm\_random\_graph(n,m)$  “ , όπου  $n,m$  το πλήθος των κόμβων και των συνδέσμων αντίστοιχα, και προσθέτουμε τυχαία weights στα edges του γράφου (τους συνδέσμους του γράφου) που αντιστοιχούν στο delay μεταξύ δυο κόμβων του δικτύου,  $G.add\_edge(x, y, weight = 10*random.random())$ .

Έπειτα θέλουμε να “διαβάσουμε” τον γράφο που απεικονίζει την υποθετική αυτή τοπολογία και να υπολογίσουμε τα ελάχιστα μονοπάτια τα οποία και θεωρούμε ότι θα ακολουθούνται στο δίκτυο. Για τον σκοπό χρησιμοποιούμε συνάρτηση που υλοποιεί τον αλγόριθμο του Dijkstra πάνω στον γράφο.

Σημειώνουμε ότι ο αλγόριθμος Dijkstra είναι ο εξής: Ξεκινώντας από έναν κόμβο αρχής, υπολογίζουμε τους γειτονικούς κόμβους και τις αποστάσεις αυτών από τον αρχικό, στην συνέχεια βγάζουμε τον αρχικό κόμβο από το σύνολο των κόμβων και επιλέγουμε το πλησιέστερο κόμβο και επαναλαμβάνουμε την ίδια διαδικασία για αυτόν υπολογίζοντας τον κόμβο με το μικρότερο κόστος (απόσταση) από όλους τους προς εξέταση κόμβους. Αποδεικνύεται ότι κάθε φορά που επιλέγεται ένας κόμβος γνωρίζουμε πλέον για αυτόν τον κόμβο τη πλησιέστερη διαδρομή από τον αρχικό κόμβο, αυτό είναι αποτέλεσμα της επιλογής του πλησιέστερου κόμβου σε κάθε επανάληψη (σημειώνουμε ότι απαραίτητη προϋπόθεση για την ισχύ του αλγορίθμου είναι η μη ύπαρξη αρνητικών συνδέσμων στον γράφο).

Όπως αναφέραμε με τον Dijkstra υπολογίζουμε τα συντομότερα μονοπάτια από έναν συγκεκριμένο κόμβο αρχής. Εμείς όμως εδώ θέλουμε να υπολογίσουμε τα συντομότερα μονοπάτια μεταξύ όλων των πιθανών ζευγαριών κόμβων ( $[v_i, v_j]$ ) του δικτύου (όπως αναφέραμε το δίκτυο θεωρείται undirected και connected άρα πάντα υπάρχει μονοπάτι μεταξύ δυο οποιονδήποτε κόμβων). Για να το επιτύχουμε αυτό τρέχουμε Dijkstra θεωρώντας κάθε κόμβο ως αρχικό.

Κατόπιν κατάλληλης επεξεργασίας φτιάχνουμε τα δεδομένα μας που είναι, 4 αρχεία. Ένα αρχείο που απεικονίζει τον γράφο, ένα αρχείο που εκφράζει ουσιαστικά το routing, απεικονίζει δηλαδή όλα τα path με την μορφή  $a = [0,1,0,\dots,0]$  όπου  $a[j] = 1$  εάν το μονοπάτι  $a$  περιλαμβάνει τον σύνδεσμο  $j$ , ένα

αρχείο που απεικονίζει μόνο τα start και end nodes των μονοπατιών [vi, vj] και τέλος ένα αρχείο που κρατάει τα μήκη των μονοπατιών (δηλαδή το συνολικό delay των μονοπατιών).

Έπειτα κατασκευάζουμε το νευρωνικό δίκτυο. Για αυτό φτιάχνουμε την SequentialModel class, που φτιάχνει ένα keras Sequential model και μέσω μιας συνάρτησης καθορίζονται οι παράμετροι που ορίζουν το νευρωνικό δίκτυο (αριθμός hidden layer, output layer, αριθμός νευρώνων του δικτύου, activation function κ.α).

Ενδεικτικά οι παράμετροι που δίνονται στο μοντέλο φαίνονται στο ακόλουθο τμήμα κώδικα:

```
sm = SequentialModel(input_dim=param_dict['input_dim'],
                    num_layers=param_dict['num_layers'],
                    num_units=param_dict['num_units'],
                    activation=param_dict['activation'],
                    activation_out=param_dict['activation_out'],
                    loss=param_dict['loss'],
                    initializer=param_dict['initializer'],
                    optimizer=param_dict['optimizer'],
                    learning_rate=param_dict['learning_rate'],
                    epochs=param_dict['epochs'],
                    batch_size=param_dict['batch_size'])
```

Μπορούμε επίσης να δούμε ενδεικτικά πως κατασκευάζεται το νευρωνικό δίκτυο, με τις αντίστοιχες παραμέτρους που αφορούν τα χαρακτηριστικά του κάθε layer αλλά και τα optimization functions. Παρατηρούμε ότι στο output layer έχουμε ένα Dense(1) layer χωρίς bias, όπου δηλαδή αθροίζουμε τα αποτελέσματα των νευρώνων για να πάρουμε το τελικό αποτέλεσμα.

```
def build_model(self):
    model = Sequential()
    # Ensure num_units list's length is the same as num_layers
    if self.num_layers != len(self.num_units):
    # Expand the list by repeating number of nodes except for last
    layer
    num_nodes, num_nodes_out = self.num_units[0], self.num_units[-1]
    self.num_units = [i for i in range(self.num_layers-1) for i in
    [num_nodes]]
    self.num_units.append(num_nodes_out)

    # Loop thru all the layers
```

```

for i in range(self.num_layers):
    if i == 0: # input and first hidden layer
        model.add(Dense(units=self.num_units[i],
                        input_dim=self.input_dim,
                        kernel_initializer=self.initializer,
                        activation=self.activation))
    elif i+1 == self.num_layers: # output layer
        model.add(Dense(1, use_bias=False))
    else: # hidden layers
        model.add(Dense(units=self.num_units[i],
                        kernel_initializer=self.initializer,
                        activation=self.activation))

# Compile the model
model.compile(loss=self.loss,
              optimizer=self.optimizer,
              metrics=self.metrics)
return model

```

Τέλος χρησιμοποιείται κώδικας που αφότου διαβάσει τα δεδομένα, δημιουργεί, εκπαιδεύει και αξιολογεί το νευρωνικό. Πιο συγκεκριμένα αρχικοποιούμε το νευρωνικό με παραμέτρους και εκτελούμε ένα k-fold validation ώστε να υπολογίσουμε μια μέση απόδοση του νευρωνικού στα δεδομένα μας και έπειτα δοκιμάζουμε διαφορετικές παραμέτρους ώστε να καταλήξουμε στις βέλτιστες ιδανικά για το πρόβλημα μας. Στην συνέχεια χρησιμοποιείται γραφική απεικόνιση για να γίνει κατανοητό πως επηρεάζουν οι παράμετροι το μέσο τετραγωνικό σφάλμα (MSE, mean squared error). Χρησιμοποιείται k-fold validation κυρίως για το υπολογισμό της ακρίβειας των αποτελεσμάτων, ωστόσο έχουμε υλοποιήσει και μια μέθοδο για να χρησιμοποιούμε μεταβαλλόμενο μέγεθος δεδομένων για train και test sets (δηλαδή κάνουμε split σε test και training data με μεταβαλλόμενο μέγεθος μεταξύ τους), ώστε να δούμε πως αυτό επηρεάζει το μέσο σφάλμα (MSE).

Το K-fold Validation είναι μια πολύ γνωστή μέθοδος αξιολόγησης μοντέλων machine learning και συχνά χρησιμοποιείται και στην αξιολόγηση νευρωνικών δικτύων. Ο βασικός λόγος είναι ότι είναι μια απλή στην χρήση και την κατανόηση μέθοδος που δίνει αποτελέσματα με μικρή εξάρτηση στα δεδομένα που εκπαιδεύουν το νευρωνικό. Η βασική ιδέα είναι ότι (αφού ανακατέψουμε το σύνολο των δεδομένων μας τυχαία), ορίζουμε έναν αριθμό k που είναι τα υποσύνολα στα οποία διασπάτε το συνολικό data, έπειτα για κάθε υποσύνολο το θεωρούμε ως test

data και το υπόλοιπο θεωρείται training data, εκπαιδεύουμε και αξιολογούμε το νευρωνικό με αυτά τα δεδομένα, αποθηκεύουμε το αποτέλεσμα και επαναλαμβάνουμε την διαδικασία για τα υπόλοιπα υποσύνολα σε νέο νευρωνικό, χρησιμοποιούμε όλα τα αποτελέσματα, για παράδειγμα υπολογίζοντας τον μέσο όρο για να καταλήξουμε στην τελική αξιολόγηση του νευρωνικού. Αυτή η ιδέα βοηθά τόσο στην περιορισμό των δεδομένων που χρειάζονται για την εκπαίδευση του μοντέλου καθώς χρησιμοποιούμε το ίδιο σύνολο για να δημιουργήσουμε πολλαπλά διαφορετικά σύνολα training και test data και επίσης μειώνει το κίνδυνο διάσπασης του συνόλου των δεδομένων σε κάποια ειδική περίπτωση που τυγχάνει να έχει υψηλό bias.

Ενδεικτικά τμήμα της υλοποίησης του k-fold validation ακολουθεί:

```
kfold = KFold(n_splits=n_splits,
              shuffle=True,
              random_state=seed))

for train_index, test_index in kfold.split(X, y):

    model = sm.build_model()

    results = model.fit(X[train_index],
                       y[train_index],
                       epochs=sm.epochs,
                       batch_size=sm.batch_size,
                       verbose=0)

    scores = model.evaluate(X[test_index],
                            y[test_index],
                            verbose=0)
    score_lst.append(scores[1])

score_lst = np.array(score_lst)
mean_mse = score_lst.mean()
```

### 3.2 Εκτέλεση:

Input Data: Επιλέγουμε να χρησιμοποιήσουμε ένα γράφο με 30 nodes και 60 links. Θεωρούμε το μέγεθος αυτό ικανοποιητικό για την πειραματική μας μελέτη καθώς είναι αρκετό μεγάλο για να υπάρχει bias στα αποτελέσματα αλλά και ταυτόχρονα όχι τόσο ώστε να απαιτεί χρονοβόρα εκμάθηση του νευρωνικού. Ακόμη, έχει μελετηθεί [5], ότι ο μέσος βαθμός ενός κόμβου (ο οποίος είναι ίσος με  $2\gamma/n$ ) είναι μεταξύ 1 και 5, για αυτό και εμείς επιλέγουμε  $2\gamma/n = 4 \Rightarrow \gamma = 2*n$ , όπου  $\gamma$  links και  $n$  nodes. Τα weights του κάθε link που εκφράζουν την μετρική που μελετάμε (π.χ delay) επιλέγουμε να είναι random αριθμοί στο  $[0,10)$ .

Τα βασικά χαρακτηριστικά που αφορούν στην αξιολόγηση του νευρωνικού όπως αναφέρθηκε, είναι η ταχύτητα σύγκλισης (convergence) και ακρίβεια των προβλέψεων (precision). Δύο ακόμη χαρακτηριστικά που αφορούν στην αξιολόγηση ενός νευρωνικού δικτύου είναι το robustness (να επιτυγχάνουν ικανοποιητικά αποτελέσματα σε όλες τις περιπτώσεις και όχι μόνο σε ειδικές περιπτώσεις) και general complexity (υπολογιστική πολυπλοκότητα).

Μελετώντας λοιπόν την ακρίβεια των προβλέψεων του νευρωνικού ως προς το μέσο τετραγωνικό σφάλμα (MSE) και σε σχέση με τις παραμέτρους του.

Hyperparameter optimization (μέσω k-fold validation):

- layers = Τα layers ενός δικτύου καθορίζουν την πολυπλοκότητα του νευρωνικού δικτύου. Περισσότερα layers κάνουν το δίκτυο πιο χρονοβόρο στην εκμάθηση αλλά είναι εφικτό να εντοπίσουν πιο πολύπλοκες σχέσεις μεταξύ των δεδομένων. Εδώ βρίσκουμε ότι 2 layers είναι το ελάχιστο αλλά είναι και αρκετά ικανό. Θεωρούμε 2 ότι είναι 1 hidden και ένα simple dense(1) output layer (θα μπορούσαμε να υπολογίσουμε μόνο με βάση τα hidden layer και τότε θα λέγαμε ότι 1 είναι αρκετό). Το γεγονός ότι για περισσότερα layer έχουμε μεγαλύτερο σφάλμα οφείλεται σε overfitting.
- Neurons = Ο αριθμός των νευρώνων στο input layer, εκφράζει το πλήθος των μεταβλητών εισόδου που δίνονται στο μοντέλο, ενώ αντίστοιχα στο output layer τις μεταβλητές εξόδου που επιθυμούμε. Για τα hidden layer είναι επίσης

σημαντικό το πλήθος των νευρώνων κυρίως για να αποφευχθούν πιθανά overfitting ή underfitting και να έχουμε ικανοποιητικά αποτελέσματα. Εδώ η βασική μας ιδέα είναι να έχουμε νευρώνες (neurons) όσος ο αριθμός των links. Στο Testing για διαφορετικό αριθμό νευρώνων, βλέπουμε ότι το μοντέλο έχει καλύτερη ακρίβεια όσο αυξάνουμε των αριθμό όπως είναι λογικό (στο διάγραμμα λόγω της κλίμακας δεν είναι πολύ εμφανές, αλλά όντως υπάρχει μια βελτίωση, επίσης αν βάλουμε αρκετά μικρό αριθμό νευρώνων το σφάλμα εκτοξεύεται, για  $\text{num\_units}=4 \rightarrow \text{MSE} > 10$ , αλλά και αυτό θα έκανε την κλίμακα ακόμη πιο μεγάλη και δε θα φαινόταν καλά στο διάγραμμα, το διάγραμμα του αριθμού των νευρώνων είναι το “num\_units”).

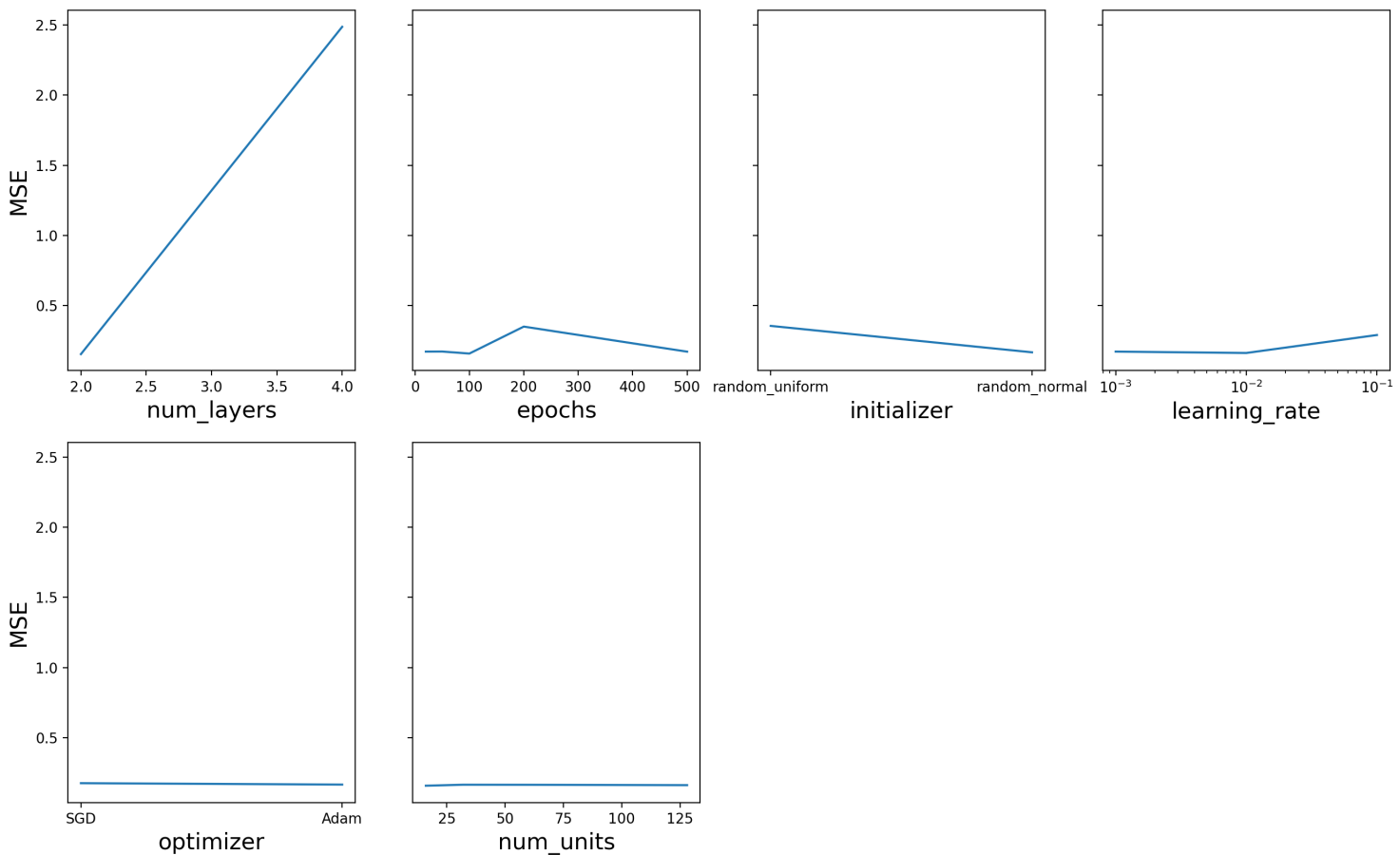
- **Learning\_rate** = όπως αναφέραμε και παραπάνω αφορά στο ρυθμό εκμάθησης του δικτύου, δηλαδή στον ρυθμό με τον οποίο μεταβάλλουμε τις παραμέτρους (τα βάρη) του νευρωνικού δικτύου σε κάθε επανάληψη. βρίσκουμε ότι, 0.001 or 0.01 είναι κάποιες πολύ καλές τιμές. Γενικότερα η επιλογή του learning\_rate εξαρτάται και από τον optimiser (το ίδιο το keras προτείνει default τιμές σε SGD 0.01 ενώ στον Adam 0.001)
- **Optimiser** = Πολύ σημαντικός όπως έχει αναφερθεί είναι και ο αλγόριθμος βελτιστοποίησης που έχει επιλέγει για το νευρωνικό δίκτυο. Εδώ βλέπουμε δυο βασικούς optimiser, ο SGD είναι απλούστερος και έχει ικανοποιητική ακρίβεια, όμως ο Adam το οποίο και επιλέγουμε, είναι βέλτιστη υλοποίηση της ιδέας του SGD για γρηγορότερη απόκριση, καθώς μεταβάλλει το learning rate κατά την διάρκεια εκπαίδευσης του νευρωνικού, ώστε να μπορεί στα αρχικά στάδια να συγκλίνει ταχύτερα, ενώ όταν πλησιάζει στο επιθυμητό αποτέλεσμα μπορεί να ελαττώσει τον ρυθμό ώστε να επιτύχει μεγαλύτερη ακρίβεια .
- **Initialiser**, αφορά στην αρχικοποίηση των τιμών των weights πριν ξεκινήσει το training του νευρωνικού. Γενικά αυτή γίνεται τυχαία, ωστόσο υπάρχουν κάποιες βασικές συναρτήσεις για αυτή την διαδικασία, εδώ δοκιμάζουμε τις random\_normal και random\_uniform. Στη πρώτη η αρχικοποίηση των τιμών

ακολουθεί Gaussian κατανομή (γνωστή και ως bell curve, σχήμα καμπάνας), ενώ στην δεύτερη η η αρχικοποίηση γίνεται ομοιόμορφα σε ένα συγκεκριμένο εύρος τιμών. Βλέπουμε ότι το mse είναι μικρό και στις δυο επιλογές, με το random\_normal να δίνει λίγο καλύτερη ακρίβεια.

- Activation function = Πολύ σημαντική είναι και επιλογή του activation function που μας δίνει την τελική έξοδο κάθε (hidden) layer. Επιλέγουμε να χρησιμοποιήσουμε “relu”, μια άλλη ιδέα θα ήταν η “sigmoid” αλλά θεωρούμε ότι η φύση του “relu”, 0 για  $x < 0$  και γραμμική ( $y = x$ ) για  $x > 0$  ταιριάζει στο πρόβλημα μας.
- Loss = Το loss function είναι αυτό που καθορίζει τον τρόπο με το οποίο εξετάζουμε την σύγκλιση του νευρωνικού. Επιλέγουμε το “MSE” για καλύτερη ακρίβεια, ενώ μια απλούστερη επιλογή θα ήταν το MAE
- Epochs = Epoch, ορίζουμε τον κύκλο που κάνει το νευρωνικό για να εκπαιδευτεί μια φορά σε όλα τα δεδομένα. Είναι πολύ σημαντικό για τον χρόνο εκπαίδευσης του νευρωνικού αλλά και για την ακρίβεια καθώς συχνά οδηγεί σε overfitting ή underfitting. Βρίσκουμε ότι 100, είναι ένας ικανοποιητικός αριθμός για να έχουμε ακριβή και όχι χρονοβόρα εκμάθηση.
- batch\_size = Batch\_size, είναι το πλήθος των δειγμάτων που επεξεργάζεται το νευρωνικό σε κάθε επανάληψη. Η επιλογή του μεγέθους αυτού αφορά στον χρόνο αλλά και την ακρίβεια εκπαίδευσης του νευρωνικού. Μεγάλο batch\_size σημαίνει καλύτερη ακρίβεια αλλά χρονοβόρα εκμάθηση, ενώ μικρότερο μπορεί να οδηγήσει σε ταχύτερη εκπαίδευση και λιγότερη χρήση μνήμης αλλά και χειρότερη ακρίβεια. Εμείς έχουμε ενδεικτικά βάλει 32.

Τα διαγράμματα για τις παραμέτρους του νευρωνικού δικτύου και την σχέση τους με την ακρίβεια του (MSE) ακολουθούν:

## Parameter Tuning Trend



Αφού λοιπόν επιλέξαμε το νευρωνικό μας με βάση τα παραπάνω αποτελέσματα, προχωρούμε σε μια σύγκριση των αποτελεσμάτων του για training με k-fold validation, με split validation καθώς και στη σύγκριση του νευρωνικού με ένα απλό xgbRegressor και επίσης την αλγεβρική μέθοδο network kriging τόσο όσον αφορά τον χρόνο σύγκλισης όσο και την ακρίβεια, έχουμε:

k-fold cross-validation. Time taken: 8.70 seconds  
Mean MSE: 0.16

Split-validation. Time taken: 0.76 seconds  
Mean MSE: 0.0001898284099297598

XGB. Time taken: 0.67 seconds  
Mean MAE: 2.3



```
KRIG. Time taken: 0.03 seconds  
mse_krig = 1.8254768623634543e-28
```

Εδώ να αναφέρουμε ότι ο XGBoost (Extreme Gradient Boosting) είναι μια open-source αποδοτική υλοποίηση του gradient boosting algorithm, και χρησιμοποιείται για regression predictive modelling, όπου θέλουμε δηλαδή αριθμητικές προβλέψεις. Τέλος να αναφέρουμε ότι το αποτέλεσμα όπως φαίνεται δίνεται σε MAE (Mean Average Error) και χρησιμοποιούνται μέθοδοι cross validation για την αξιολόγηση.

Συνοψίζοντας, το k-fold δίνει πολύ υψηλή ακρίβεια σε ικανοποιητικό χρόνο, αν και είναι φυσικά πιο αργό από το απλό split όπως είναι λογικό καθώς ουσιαστικά κάνει k φορές split. Αν και παρατηρούμε μεγαλύτερη ακρίβεια στο split-validation, αναγνωρίζουμε ότι το k-fold είναι πιο αντιπροσωπευτικό καθώς στο split validation ενδέχεται να έχουμε ισχυρό bias στο training data που έτυχε να επιλέξουμε, εάν δηλαδή τυχαία επιλέγαμε άλλο training data μπορεί να βλέπαμε πολύ μεγαλύτερο σφάλμα στο testing data, κάτι που το k-fold validation αποφεύγει. Ο xgbRegressor είναι εξίσου γρήγορος με το training του νευρωνικού με απλό split και λιγότερο ακριβής, επίσης αναμενόμενο καθώς εκτελεί απλό regression. Τέλος η αλγεβρική μέθοδος Network Kriging είναι φυσικά η ταχύτερη (οι πίνακες είναι αρκετά μικροί σε μέγεθος όποτε δεν είναι τόσο σύνθετος ο υπολογισμός) και πιο ακριβής από τα υπόλοιπα.

Εδώ πρέπει λοιπόν να αναφέρουμε συμπερασματικά ότι η σε περίπτωση που είναι γνωστή όλη η πληροφορία (όλα τα links), είναι εφικτό να χρησιμοποιηθούν οι αλγεβρικές μέθοδοι επίλυσης όπως το Network Kriging και να δώσουν το ακριβές αποτέλεσμα. Ωστόσο αν και είναι πιο ακριβής οι αλγεβρικές μέθοδοι, η απόδοση των μεθόδων ML (στην περίπτωση αυτή τα νευρωνικά δίκτυα) είναι σε κάθε περίπτωση ικανοποιητική.

### 3.3 Σχέση training και test data στην ακρίβεια του νευρωνικού.

Με το απλό split σε train και test data,

Split train/test	MSE
25% train / 75% test	>10
50% train / 50% test	<1
75% train / 25% test	<0.1

Χρησιμοποιώντας τον διαχωρισμό σε δυο μόνο σύνολα training και test data, μπορούμε να δούμε πως επηρεάζει το πλήθος των δεδομένων στο οποίο εκπαιδεύουμε το νευρωνικό. Όπως είναι λοιπόν αναμενόμενο όσο μεγαλύτερο είναι το training dataset (αυτό αυτομάτως σημαίνει και μικρότερο test data), τόσο βελτιώνεται και η απόδοση του νευρωνικού.

### 3.4 Πως επηρεάζεται η ακρίβεια του νευρωνικού από την πληροφορία.

Αφού λοιπόν μελετήσαμε πως επηρεάζει το ίδιο το νευρωνικό την ακρίβεια των αποτελεσμάτων, στόχος μας είναι να δούμε πως επηρεάζει και η πληροφορία που έχουμε ως γνωστή για τις παραμέτρους του δικτύου.

#### 3.4.1 Ελλιπής Πληροφορία

Το πρώτο βήμα, είναι να δούμε την ακρίβεια των αποτελεσμάτων σε περίπτωση που για κάποια links δεν έχουμε πληροφορία. Για να επιτευχθεί αυτό, μπορούμε να αντικαταστήσουμε με 0 τις στήλες αυτών των link στο training data (δεν επηρεάζουμε ωστόσο καθόλου το test data). Με βάση και τον ορισμό του routing πίνακα, είναι σαν να θεωρούμε ότι δεν υπάρχουν στα μονοπάτια οι σύνδεσμοι αυτοί. Τρέχοντας με το νευρωνικό στις παραμέτρους που μας έδωσε το k-fold

validation και μεταβάλλοντας το πλήθος των “άγνωστων” συνδέσμων, βλέπουμε ότι:

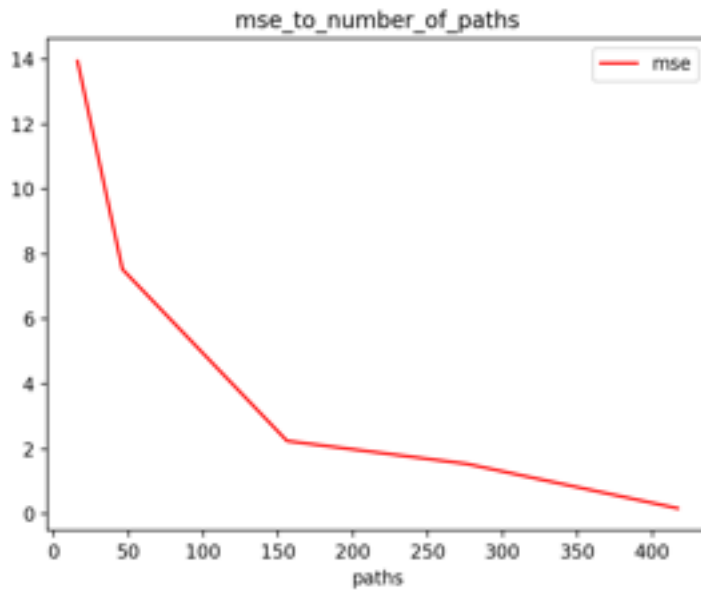
“Unknown links”	MSE
0%	0,16
5%	0,73
10%	1,6
20%	12

Παρατηρείται ότι η ακρίβεια των αποτελεσμάτων παραμένει ικανοποιητική όταν η απώλεια πληροφορίας είναι μικρή (μέχρι και για 10%), αλλά φαίνεται να αυξάνεται εκθετικά για απώλεια πληροφορίας άνω του 10%.

### 3.4.2 Σχέση της απόδοσης του νευρωνικού με το πλήθος των μονοπατιών

Η απόδοση του νευρωνικού αναμένεται να εξαρτάται και από το πλήθος των μονοπατιών που είναι γνωστά (measured paths) και χρησιμοποιούνται στην εκπαίδευση του. Για να δούμε αυτή την συσχέτιση απαιτείται η εξέταση των αποτελεσμάτων του νευρωνικού για εκπαίδευση με μεταβαλλόμενο πλήθος γνωστών μονοπατιών. Για αυτό ακολουθείται η εξής μεθοδολογία: επιλέγονται κάποια links που θεωρούνται μη εγκατεστημένα και αποκλείονται όλα τα path που περιλαμβάνουν αυτά τα links από το input και το output.

Παρατηρώντας το διάγραμμα που παίρνουμε για την σχέση των δύο, βλέπουμε ότι για μικρό αριθμό εγκατεστημένων μονοπατιών έχουμε μεγάλο σφάλμα, το οποίο μικραίνει απότομα καθώς αυξάνουμε τον αριθμό των μονοπατιών, η βελτίωση αυτή φθίνει πάνω από τα 200 path (περίπου 50% εγκατεστημένων μονοπατιών) αλλά συνεχίζουμε να βλέπουμε βελτίωση μέχρι και όταν όλα τα μονοπάτια είναι εγκατεστημένα. Σημειώνεται ότι ο αριθμός των μονοπατιών έχει βέβαια να κάνει με το μέγιστο σύνολο μονοπατιών που είναι 435 (όσο όλα τα πιθανά μονοπάτια).



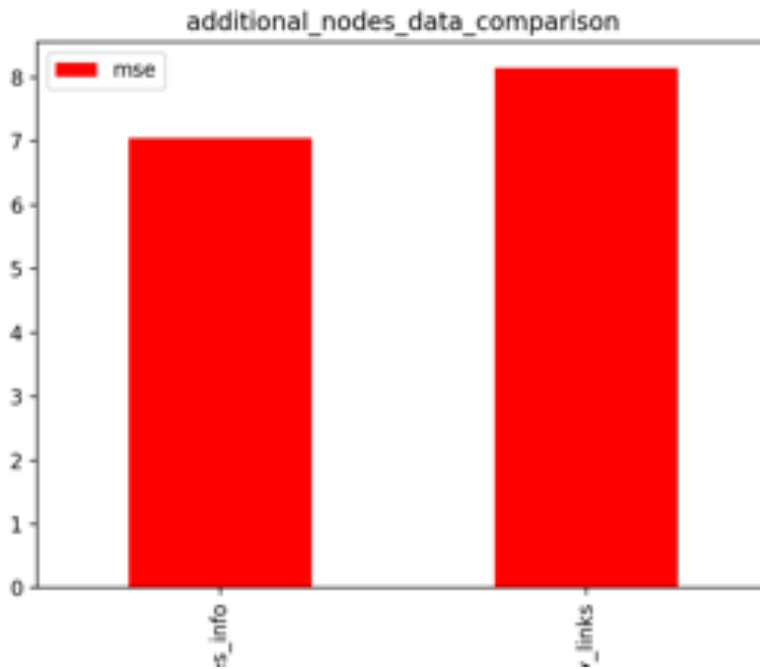
Ακόμη αναφέρουμε ότι εάν χρησιμοποιήσουμε ως test data τα μονοπάτια που είχαμε αποκλείσει ως μη εγκατεστημένα, το σφάλμα σε αυτήν την περίπτωση θα είναι πολύ μεγάλο ανεξάρτητα του αριθμού των μονοπατιών. Αποτέλεσμα αναμενόμενο καθώς σε αυτή την περίπτωση το test data περιλαμβάνει συνδέσμους που δεν υπάρχουν καθόλου στο input.

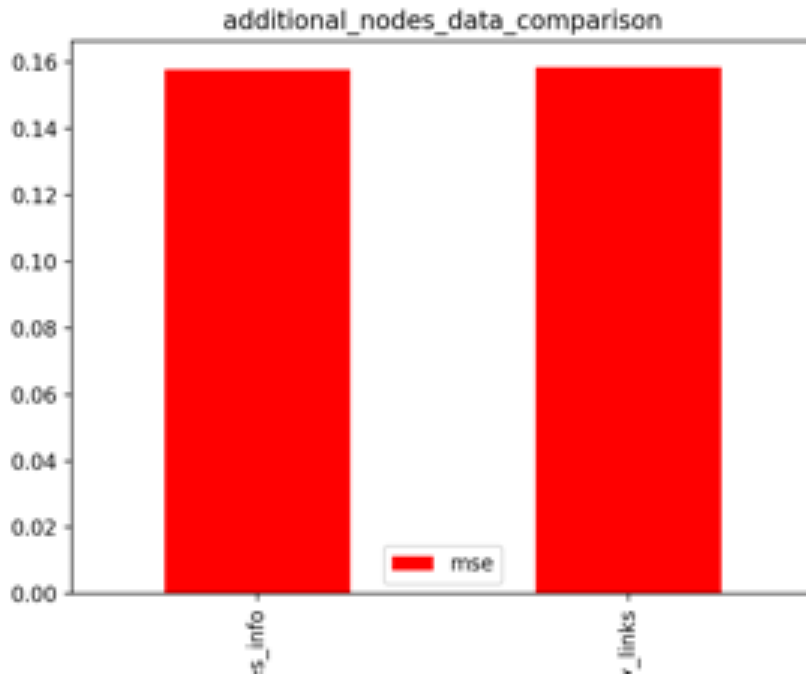
Να αναφέρω ότι ο μέγιστος αριθμός μονοπατιών(435) προκύπτει ως το σύνολο όλων των ζευγών κόμβων και αυτό διότι θεωρείται ότι ο γράφος που απεικονίζει το υποθετικό δίκτυο είναι undirected και connected, άρα μεταξύ δυο οποιονδήποτε κόμβων υπάρχει πάντα μονοπάτι που τους ενώνει.

### 3.4.3 Σύγκριση πληροφορίας συνδέσμων και κόμβων μονοπατιών

Πέραν του πλήθους των μονοπατιών που μελετήσαμε παραπάνω, χρήσιμο θεωρείται να μελετηθεί και η δυνατότητα εκπαίδευσης του νευρωνικού με την επιπλέον πληροφορία των κόμβων αρχής και τέλους των μονοπατιών. Πιο συγκεκριμένα μπορούμε εκπαιδεύσουμε το νευρωνικό με την πληροφορία των συνδέσμων των μονοπατιών και των κόμβων αρχής και τέλους των μονοπατιών και χωρίς αυτήν την πληροφορία. Δηλαδή στην μια περίπτωση το input μας είναι

$[0,1\dots,0,1,\alpha,\beta]$  όπου  $\alpha,\beta$  οι κόμβοι αρχής και τέλους, συγχωνεύουμε δηλαδή τον routing πίνακα με τη πληροφορία αρχικού και τελικού κόμβου, ενώ στην άλλη μόνο ο routing πίνακας, δηλαδή  $[0,1,\dots,0,1]$ .





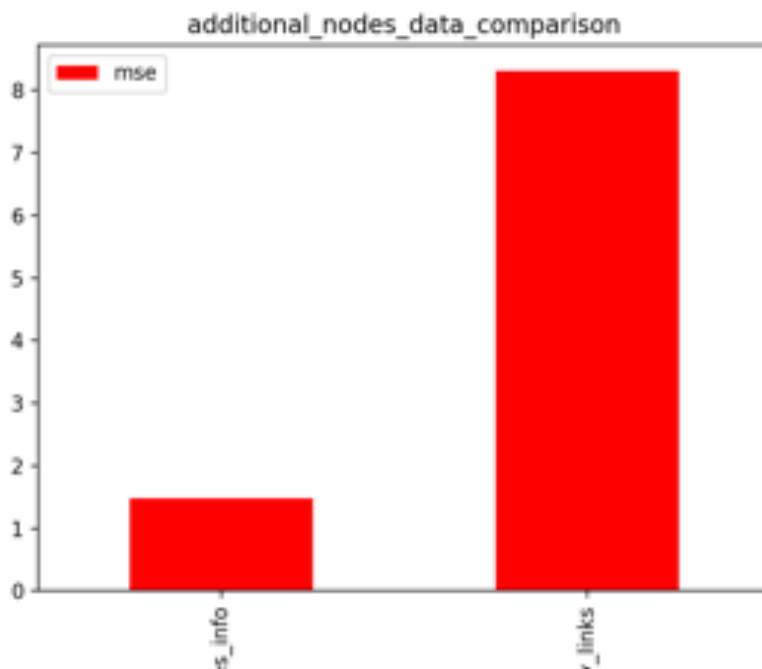
Το αποτέλεσμα είναι τα παραπάνω διαγράμματα που συγκρίνουν το MSE του νευρωνικού με και χωρίς την επιπλέον πληροφορία των κόμβων αρχής και τέλους όπως εξηγήσαμε. Επίσης για να καταλάβουμε τα διαγράμματα, πρέπει να αναφερθεί ότι για να κατανοήσουμε καλύτερα την επιρροή αυτής της επιπλέον πληροφορίας μεταβάλλουμε και εδώ το ποσοστό γνωστής πληροφορίας για τους συνδέσμους. Στο πρώτο διάγραμμα έχουμε 80% γνωστούς συνδέσμους, στην δεύτερη περίπτωση 95% και στην τρίτη γνωρίζουμε όλους τους συνδέσμους. Αυτό μοντελοποιείται επιλέγοντας τυχαία ένα συγκεκριμένο ποσοστό (20% και 5% αντίστοιχα) των συνδέσμων για τους οποίους βάζουμε 0 σαν τιμή σε όλα τα path. Φαίνεται ότι πράγματι η επιπλέον πληροφορία των κόμβων αρχής και τέλους βοηθάει το νευρωνικό σημαντικά στις περίπτωση που έχουμε ελλιπής πληροφορία ως αναφορά τους συνδέσμους, ενώ πολύ λιγότερο όταν έχουμε πλήρη πληροφορία για τους συνδέσμους.

Για υψηλό ποσοστό αγνώστων συνδέσμων το σφάλμα παραμένει ωστόσο υψηλό.

Μια γνωστή μέθοδος προεπεξεργασίας δεδομένων στο machine learning, είναι το one-hot-encoding. Συνήθως χρησιμοποιείται για την μετατροπή categorical data σε

numerical data, το οποία απαιτείται καθώς τα νευρωνικά δίκτυα απαιτούν αριθμητικά δεδομένα ως είσοδο. Η βασική ιδέα είναι η δημιουργία νέων στηλών για κάθε διαφορετική τιμή των δεδομένων με στόχο την μετατροπή της πληροφορίας σε 0 και 1 όπου 0 = False και 1 = True.

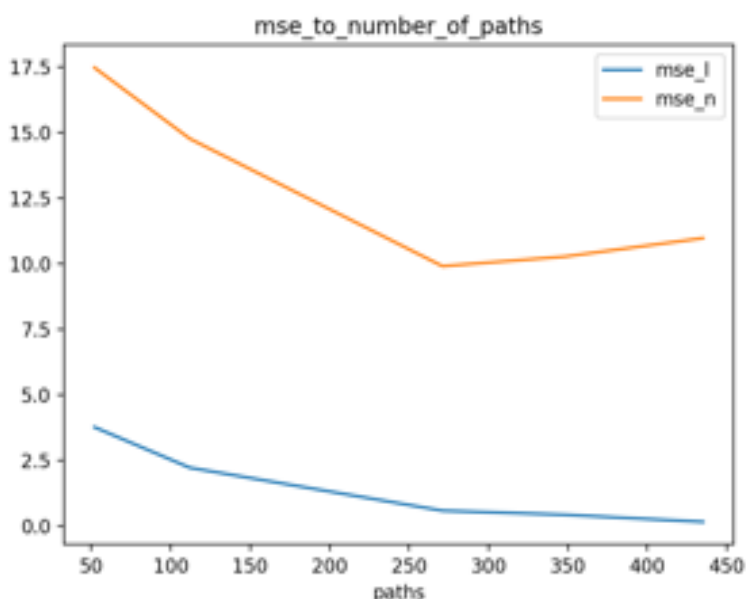
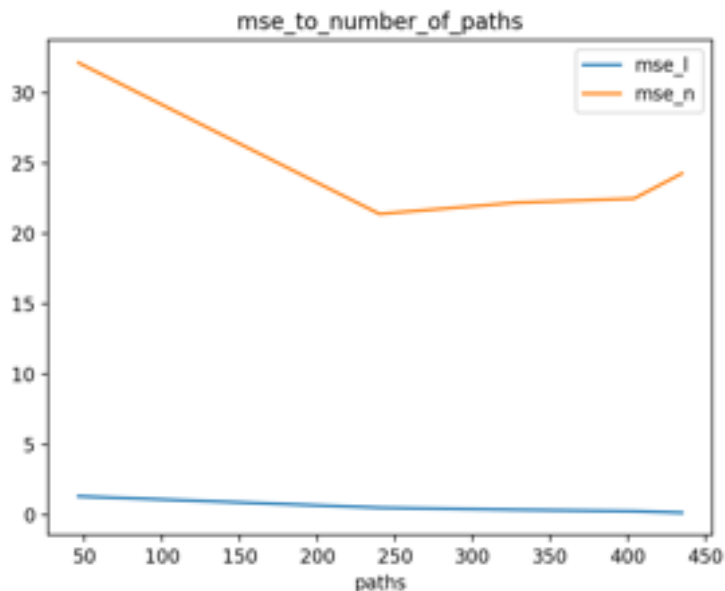
Εμείς εδώ για να βελτιώσουμε αυτό το αποτέλεσμα, κωδικοποιούμε την πληροφορία αρχής και τέλους με one\_hot\_encoding δηλαδή το [α,β] μετατρέπεται σε [0,1,0...,1,0] όπου μόνο τα στοιχεία α, β είναι 1 και έτσι το input είναι πλέον της μορφής [0,1,.....,0,1, 1,0,....,0,1], όπου τα πρώτα m στοιχεία είναι οι σύνδεσμοι και τα τελευταία n είναι οι κόμβοι, τότε το σφάλμα βελτιώνεται σημαντικά. Στο παρακάτω διάγραμμα έχουμε 80% γνωστούς κόμβους και βλέπουμε MSE κοντά στο 1, στο αντίστοιχο διάγραμμα από πάνω είχαμε MSE κοντά το 7.



Σημειώνουμε ότι για τα παραπάνω διαγράμματα: nodes\_info η απόδοση με την επιπλέον πληροφορία των κόμβων(n), ενώ only\_links δίνεται μόνο η γνώση των συνδέσμων(l)

Μέχρι τώρα όμως είδαμε πως βοηθάει η πληροφορία των κόμβων αρχής και τέλους των μονοπατιών ως συμπληρωματική πληροφορία, γεννιέται λοιπόν το ερώτημα εάν επαρκεί και ως η κυρία πληροφορία εκπαίδευσης του νευρωνικού για την σωστή πρόβλεψη των επιμέρους τιμών.

Εκπαιδεύοντας πράγματι το νευρωνικό μόνο με αυτήν πληροφορία, των κόμβων αρχής και τέλους, καταλήγουμε ότι δεν επαρκεί. Όπως φαίνεται από τα διαγράμματα που ακολουθούν δεν μπορούμε να πλησιάσουμε καθόλου σε ακρίβεια τα αποτελέσματα του νευρωνικού με την πληροφορία των συνδέσμων των μονοπατιών.





Στο δεύτερο διάγραμμα επιχειρείται η βελτίωση της απόδοσης δίνοντας ως input τους κόμβους αφού πρώτα τους μετατρέψουμε με one-hot encoding. Δηλαδή το input είναι της μορφής  $[0,1,\dots,1,0]$  όπου 1 έχουν μόνο τα  $i, j$  όπου  $i, j$  οι κόμβοι αρχής και τέλους του μονοπατιού. Αυτό πράγματι βελτιώνει την απόδοση σημαντικά αλλά και πάλι δεν πλησιάζει αυτήν των συνδέσμων όπως είναι λογικό.

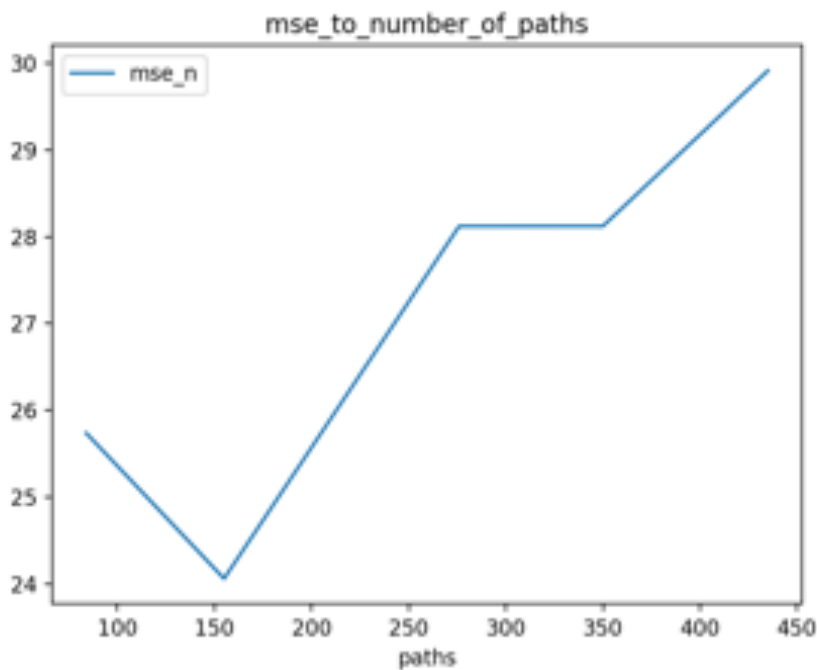
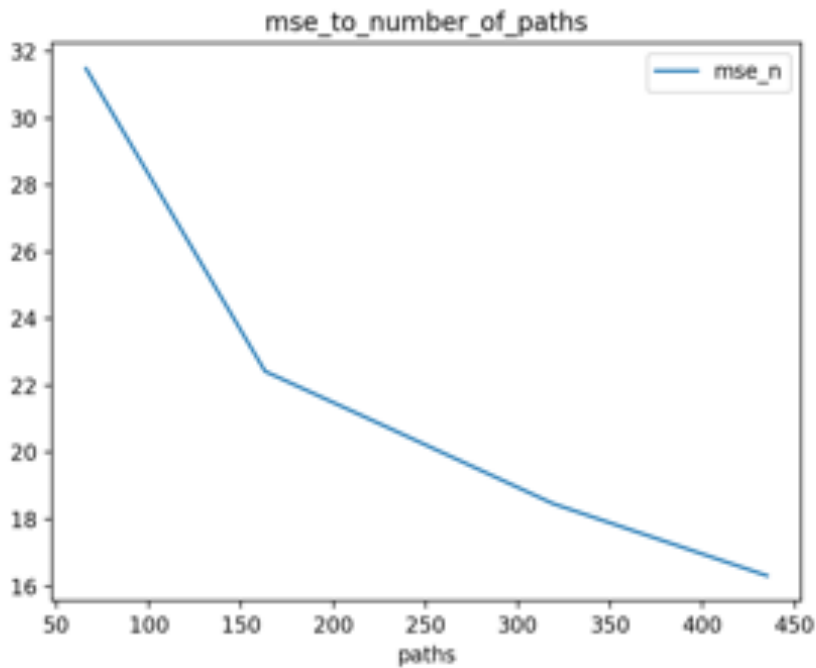
### **3.4.4 Απόδοση του νευρωνικού σε σχέση με ανακρίβεια γνωστών δεδομένων**

Ένα ακόμη αντικείμενο μελέτης είναι η απόδοση του νευρωνικού σε σχέση με την ακρίβεια των δεδομένων που έχουμε ως input. Πιο συγκεκριμένα τίθεται το ερώτημα πως θα επηρεαζόταν η ακρίβεια των αποτελεσμάτων αν δεν έχουμε ακριβή γνώση των μονοπατιών του δικτύου. Ένας τρόπος για να ελεγχθεί αυτό είναι να θεωρήσουμε ότι δεν γνωρίζουμε όλα τα βέλτιστα μονοπάτια, όπως θεωρούσαμε μέχρι στιγμής. Σημειώνουμε εδώ ότι θεωρούμε ότι το δίκτυο επιλέγει να χρησιμοποιεί τα μονοπάτια ελάχιστης καθυστέρησης μεταξύ δυο κόμβων, αν και αυτό είναι μια λειτουργία που καθορίζεται από το πρωτόκολλο επικοινωνίας που χρησιμοποιεί ένα δίκτυο υπολογιστών.

Για να το ερευνήσουμε αυτό στο δικό μας πρόβλημα λοιπόν, θεωρούμε γνωστά τα δεύτερα ή και τρίτα καλύτερα μονοπάτια μεταξύ δυο κόμβων. Σημειώνουμε ότι ένας αλγόριθμος για τον ακριβή προσδιορισμό των δεύτερων και τρίτων βέλτιστων μονοπατιών είναι ο ακόλουθος: Εκτελούμε έναν djikstra, έπειτα βγάζουμε έναν σύνδεσμο από το βέλτιστο μονοπάτι και να ξαναβρίσκουμε το βέλτιστο μονοπάτι (εκτελώντας και πάλι djisktra) όποτε έχουμε το βέλτιστο μονοπάτι χωρίς τον συγκεκριμένο σύνδεσμο. Προσθέτουμε τον σύνδεσμο που αφαιρέσαμε, αφαιρούμε νέο σύνδεσμο και επαναλαμβάνουμε την διαδικασία. Όταν το κάνουμε για όλους τους συνδέσμους έχουμε πλέον όλα τα πιθανά μονοπάτια που έχουν τουλάχιστον ένα διαφορετικό σύνδεσμο από το πραγματικό βέλτιστο. Εάν από αυτά βρούμε το

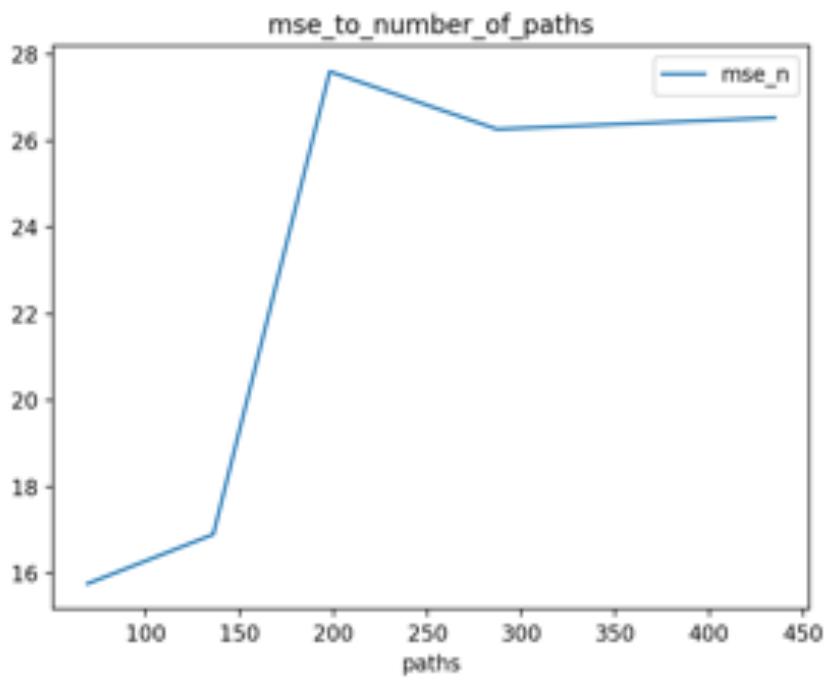
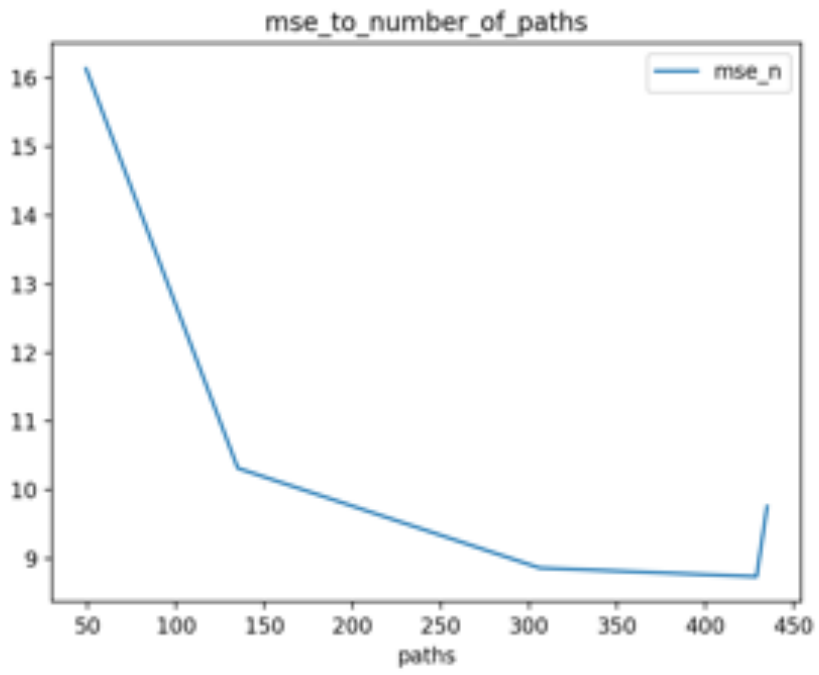
καλύτερο και το δεύτερο καλύτερο από αυτά θα έχουμε βρει το δεύτερο και τρίτο καλύτερο μονοπάτι. Αυτό είναι όμως υπερβολικά χρονοβόρο και θα έπρεπε θεωρητικά να το εκτελούμε για κάθε βέλτιστο μονοπάτι από την αρχή. Εμείς για υπολογίσουμε λοιπόν τα επόμενα καλύτερα μονοπάτια πλην των βέλτιστων, κάνουμε το εξής: Καθώς τρέχει ο Dijkstra, αποθηκεύει και μια λίστα με το δεύτερο καλύτερο μονοπάτι, αυτό το κάνει ως εξής: εφόσον υπολογίσει ένα νέο βέλτιστο μονοπάτι για κάποιο κόμβο, θεωρεί το ως τότε βέλτιστο πλέον το δεύτερο καλύτερο, ενώ αν είναι η πρώτη φορά που βρίσκει μονοπάτι το αποθηκεύει ως καλύτερο αλλά και ως δεύτερο καλύτερο (αυτό ουσιαστικά είναι για την περίπτωση που μόνο ένα μονοπάτι υπάρχει να μπορεί να υπάρχει και στην λίστα με τα “Δευτέρα καλύτερα” ένα μονοπάτι). Αυτή η λύση δημιουργεί ένα dataset που περιλαμβάνει είτε τα καλύτερα είτε τα δεύτερα καλύτερα σε αρκετές περιπτώσεις (για την ακρίβεια δεν είναι απολύτως βέβαια ότι υπολογίζεται πάντα το δεύτερο καλύτερο καθώς αυτό θεωρητικά θα μπορούσε να έχει απορριφθεί εφόσον ελέγχεται αφού έχουμε ήδη βρει το βέλτιστο μονοπάτι, ωστόσο σίγουρα υπολογίζεται ένα ικανοποιητικό δεύτερο μονοπάτι, το οποίο είναι και το ζητούμενο στην συγκεκριμένη μελέτη). Θεωρούμε δηλαδή ότι αυτή η λύση ικανοποιεί αυτό που θέλουμε να ελέγξουμε, δηλαδή πως μια απόκλιση από τα ακριβή δεδομένα στα μονοπάτια επηρεάζει το αποτέλεσμα του νευρωνικού όταν σαν input δίνονται μόνο οι κόμβοι αρχής και τέλους.

Τα αποτελέσματα ακολουθούν στα παρακάτω διαγράμματα:



Τα διαγράμματα αυτά είναι η απόδοση του νευρωνικού σε σχέση με το πλήθος των μονοπατιών με τα οποία εκπαιδεύουμε το νευρωνικό, για την περίπτωση που δίνουμε ανακριβή δεδομένα όπως εξηγήσαμε παραπάνω. Σημειώνουμε ότι για το πρώτο διάγραμμα δίνουμε input αφού εκτελέσουμε πρώτα one-hot encoding και

άρα δίνουμε την πληροφορία των κόμβων αρχής και τέλους ως  $[0,1,0\dots,1,0]$  όπου 1 εάν οι κόμβοι  $i, j$  είναι οι κόμβοι αρχής και τέλους αντίστοιχα, ενώ το δεύτερο το input είναι της μορφής  $[v_i, v_j]$  όπου  $v_i, v_j$  οι κόμβοι αρχής και τέλους των μονοπατιών. Η περίεργη συμπεριφορά στο δεύτερο διάγραμμα θα εξηγηθεί παρακάτω.

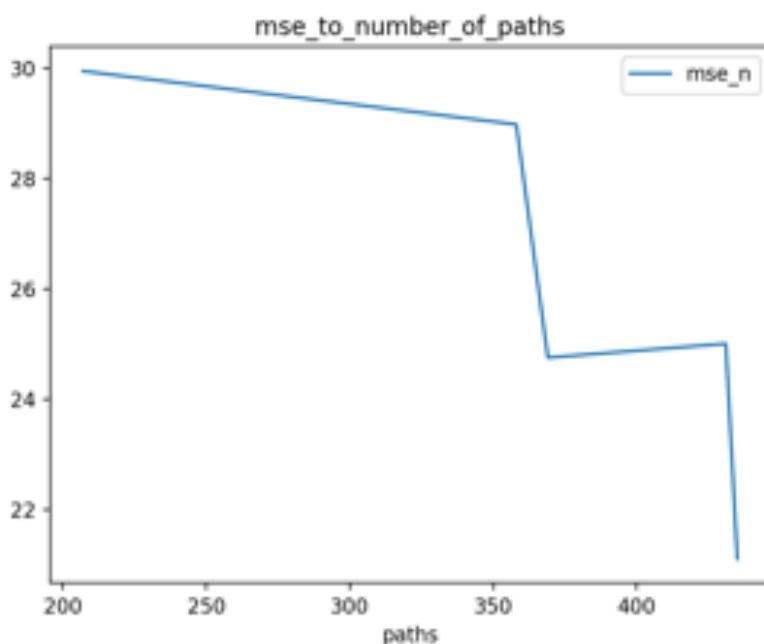
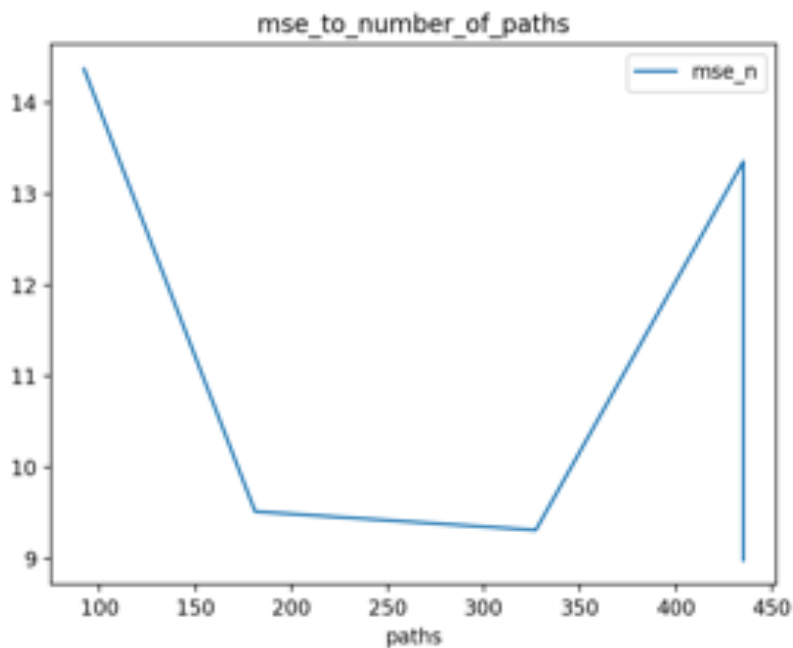


Τα δύο παραπάνω διαγράμματα είναι για την περίπτωση που έχουμε ακριβή πληροφορία, όπως δηλαδή εξηγήσαμε παραπάνω, έχουμε πράγματι τα βέλτιστα μονοπάτια. Ομοίως με παραπάνω το πρώτο μετά από one-hot encoding ενώ το δεύτερο με το απλό input.

Τα αποτελέσματα είναι όπως φαίνεται από το μέσο τετραγωνικό σφάλμα(MSE) στον κάθετο άξονα, ήταν αναμενόμενα χειρότερα στην περίπτωση που δεν έχουμε μόνο τα βέλτιστα μονοπάτια. Ωστόσο η ακρίβεια των προβλέψεων του νευρωνικού και για την περίπτωση που έχουμε τα επόμενα καλύτερα μονοπάτια (όχι τα βέλτιστα) μπορεί μην αποκλίνει υπερβολικά από αυτήν των βέλτιστων μονοπατιών και να βελτιωθεί σημαντικά αν προεπεξεργαστούμε το input με one-hot encoding. Πιο συγκεκριμένα δίνοντας ένα μεγάλο αριθμό μονοπατιών και χρησιμοποιώντας το one-hot encoding μπορεί να ελαττωθεί σημαντικά το μέσο σφάλμα, όπως βλέπουμε:  $MSE = 9$  και  $MSE = 16$  για την περίπτωση που δίνουμε όλο το σύνολο των μονοπατιών στις δύο περιπτώσεις αντίστοιχα.

Όπως αναφέραμε παραπάνω, παρατηρούμε μια περίεργη συμπεριφορά στην απόδοση του νευρωνικού για input μόνο με τους κόμβους αρχής και τέλους, δηλαδή μια αύξουσα καμπύλη σε σχέση με το πλήθος των μονοπατιών το οποίο δεν είναι λογικό. Για να κατανοήσουμε αυτήν την συμπεριφορά, κάναμε αρκετές δοκιμές με το νευρωνικό και τελικά παρατηρήσαμε ότι η επιλογή των ενεργών συνδέσμων επηρεάζει σημαντικά την απόδοση του νευρωνικού σε αυτή την περίπτωση, και καθώς αυτή η επιλογή γίνεται τυχαία, παρατηρούμε διαφορετικά διαγράμματα κάθε φορά. Αυτό ήταν αποτέλεσμα του τρόπου επιλογής μικρότερου αριθμού μονοπατιών για την εκπαίδευση του νευρωνικού. Πιο συγκεκριμένα, είχαμε επιλέξει την εξής μεθοδολογία για την επιλογή μικρότερου πλήθους μονοπατιών: επιλέγω κάποιους συνδέσμους που θεωρώ ανενεργούς και αφαιρώ από το input και το output τα μονοπάτια που χρησιμοποιούν αυτούς τους κόμβους (θεωρώ ότι δεν είναι εγκατεστημένα κατα συνέπεια ούτε τα μονοπάτια). Έτσι κρατάμε ένα μικρότερο σύνολο δεδομένων με ορθή συνοχή. Είναι όμως πιθανό το μικρότερο αυτό dataset να περιλαμβάνει μονοπάτια με ορθά δεδομένα και κατα συνέπεια να μπορεί να δώσει πολύ ικανοποιητικά αποτελέσματα στο μικρότερο

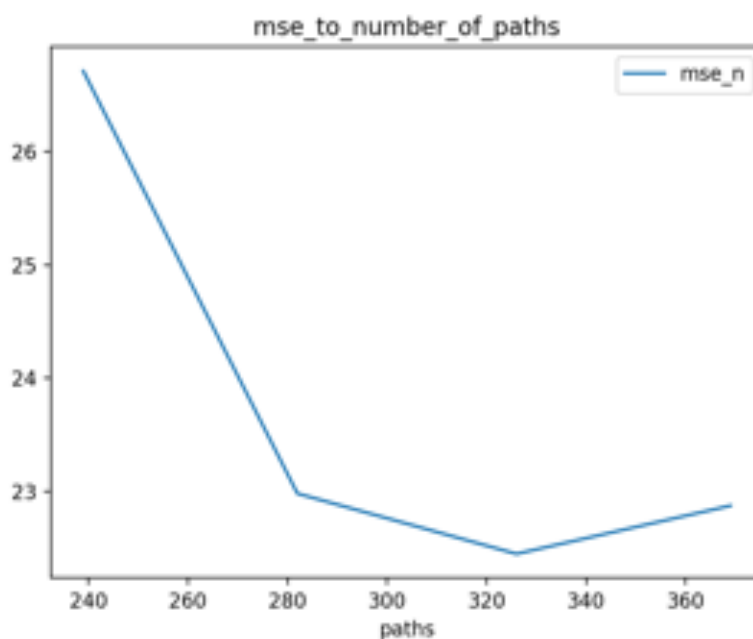
σύνολο δεδομένων. Κάνοντας λοιπόν αρκετές δοκιμές παρατηρήσαμε ότι υπάρχουν περιπτώσεις που ακόμη και όταν δίνουμε τους κόμβους αρχής τέλους με one-hot encoding αλλά και σε σπάνιες περιπτώσεις τους συνδέσμους των μονοπατιών μπορεί να συμβεί να έχουμε μικρότερο σφάλμα σε λιγότερα δεδομένα, όμοια με αυτό που παρατηρείται και εδώ. Ωστόσο είναι πολύ πιο ευαίσθητα στην επιλογή αυτήν των ανενεργών κόμβων τα αποτελέσματα στην περίπτωση που δίνουμε μόνο τους κόμβους αρχής και τέλους χωρίς επεξεργασία.



Ενδεικτικά στα δυο παραπάνω διαγράμματα, το πρώτο διάγραμμα είναι ένα αποτέλεσμα κατόπιν one-hot encoding, όπου τυγχάνει να έχουμε απόδοση συγκριτικά καλά και με πλήθος μονοπατιών μεταξύ 150 και 350 αλλά και ένα spike στην απόδοση κοντά στο πλήρες σύνολο μονοπατιών, ενώ το δεύτερο είναι μια περίπτωση στην οποία η απόδοση φαίνεται να βελτιώνεται με τους κόμβους αρχής και τέλους όπως θα περιμέναμε, σε αντίθεση με αυτό που βλέπαμε στα διαγράμματα προηγουμένως με την αύξηση του σφάλματος για αυξημένο πλήθος μονοπατιών.

Για να αντιμετωπίσουμε αυτό το πρόβλημα, χρησιμοποιήσαμε μια απλούστερη μέθοδο για τη επιλογή διαφορετικού πλήθους μονοπατιών: χρησιμοποιώντας το πλήρες dataset μεταβάλλουμε μόνο το ποσοστό training και test data (δηλαδή στο split των δεδομένων σε training και test data, πιο συγκεκριμένα το ποσοστό των test data μεταξύ 10,20,30,40%. Αυτό σημαίνει πως στο training έχουμε 90,80,70,60% του πλήθους των μονοπατιών αντίστοιχα.

Τα τελικά διαγράμματα ήταν όπως το παρακάτω με μεγαλύτερη σταθερότητα σε όλες τις δοκιμές που έγιναν:



Δηλαδή εδώ παρατηρούμε πιο σταθερά την αναμενόμενη βελτίωση για μεγαλύτερο training data (αν και βλέπουμε και εδώ την βέλτιστη απόδοση τυχαία να μην είναι στην μεγαλύτερη τιμή). Δηλαδή όταν δεν προϋποθέτουμε την ορθή συνοχή ακόμη και για λιγότερα μονοπάτια που μπορεί να οδηγήσει σε υψηλή ακρίβεια για μικρότερο πλήθος μονοπατιών αλλά μεταβάλλουμε το μέγεθος των training και test set έτσι ώστε να εκπαιδεύσουμε με ένα μικρότερο σύνολο δεδομένων (μονοπατιών) και αντίστοιχα έχουμε ένα ευρύτερο σύνολο testing data, το αποτέλεσμα επηρεάζεται από το πλήθος πράγματι των μονοπατιών και όχι τόσο από την επιλογή των εγκατεστημένων και μη μονοπατιών που παρατηρήθηκε παραπάνω.

### **3.4.5 Σύγκριση με Αλγεβρικές Μεθόδους**

Όπως αναφέραμε ένας από τους βασικούς στόχους της εργασίας είναι και η σύγκριση των μεθόδων ML (machine learning), όπως τα νευρωνικά δίκτυα που χειριζόμαστε εδώ, με τις αλγεβρικές μεθόδους επίλυσης του προβλήματος του Network Tomography.

Μελετώντας λοιπόν την χρήση των νευρωνικών δικτύων για διαφορετικά είδη πληροφορίας, παρατηρήσαμε τόσο την δυνατότητα εκπαίδευσης τους και ικανοποιητικής πρόβλεψης σε περιπτώσεις όπου η γνωστή πληροφορία ήταν μόνο αυτή των κόμβων πηγής και προορισμού, καθώς και την δυνατότητα τους να αντισταθμίσουν ελλιπή ή ανακριβή πληροφορία και να πετύχουν ικανοποιητικές προβλέψεις σε περιπτώσεις ελλιπούς και ανακριβούς πληροφορίας.

Αντιθέτως με τις μεθόδους ML, οι αλγεβρικές δεν μπορούνε χρησιμοποιηθούν σε περιπτώσεις που δεν έχουμε καθόλου γνώση για τους συνδέσμους του δικτύου, όπως δηλαδή η περίπτωση που έχουμε γνώση μόνο για τους κόμβους πηγής και προορισμού. Επίσης η χρήση τους σε περιπτώσεις ελλιπούς πληροφορίας θα ήταν ανεπαρκείς καθώς δεν θα μπορούσαν να αντισταθμίσουν την πληροφορία που δεν υπάρχει, ενώ σε περίπτωση ανακριβούς πληροφορίας θα οδηγούσαν σε λανθασμένες προβλέψεις.



#### 4. Συμπεράσματα

Συνοψίζοντας λοιπόν την εργασία, μπορούμε να σταθούμε στα βασικά σημεία της μελέτης.

Η χρήση νευρωνικών δικτύων για την πρόβλεψη επιμέρους τιμών στα δίκτυα υπολογιστών από γνώση τιμών end-to-end μονοπατιών είναι εφικτό να επιφέρει πολύ ικανοποιητική ακρίβεια και να βοηθήσει στον κατανόηση της κατάστασης του δικτύου. καταλήγουμε μάλιστα στο συμπέρασμα ότι δεν απαιτείται η χρήση πολύπλοκων και μεγάλων νευρωνικών δικτύων καθώς και ένα δίκτυο με 1 μόνο hidden layer και νευρώνες ίσους με τον αριθμό των συνδέσμων του δικτύου μπορεί να επιφέρει ικανοποιητική ακρίβεια.

Σημαντικό ρόλο στην επιτυχή χρήση των νευρωνικών δικτύων για πρόβλεψη των επιμέρους τιμών, έχει και η πληροφορία που κατέχουμε ως γνωστή. Πιο συγκεκριμένα όπως είναι αναμενόμενο όσο περισσότερη πληροφορία γνωρίζουμε για το δίκτυο τόσο ακριβέστερες θα είναι και οι προβλέψεις, ωστόσο σε μικρές γνωστικές απώλειες το νευρωνικό κατορθώνει και επιτυγχάνει ικανοποιητικές προβλέψεις. Ακόμη παρατηρείται ότι και σε περιπτώσεις ανακριβούς πληροφορίας, εφόσον φυσικά αυτή δεν αποκλίνει σημαντικά από την πραγματικότητα το νευρωνικό μπορεί και πάλι να πετύχει ικανοποιητική απόδοση. Τέλος μελετώντας το είδος της πληροφορίας που γνωρίζουμε και με το οποίο εκπαιδεύουμε το νευρωνικό, και την σχέση του με την ακρίβεια των προβλέψεων βλέπουμε ότι η γνώση των επιμέρους συνδέσμων των μονοπατιών επαρκεί για την ικανοποιητική απόδοση του νευρωνικού δικτύου. Ωστόσο η γνώση μόνο των κόμβων αρχής και τέλους δεν επαρκεί για ικανοποιητική ακρίβεια του νευρωνικού, αλλά μπορεί να βελτιώνει σημαντικά την ακρίβεια εάν συνδυαστεί με αυτήν των συνδέσμων.

## 5. Εισηγήσεις για πιθανή έρευνα

Η εργασία αυτή αφήνει ανοιχτά ερωτήματα για περαιτέρω έρευνα όπως και το γενικότερο αντικείμενο του Network Tomography. Μια περαιτέρω έρευνα σε πιθανά μοντέλα deep learning για πολυπλοκότερα νευρωνικά δίκτυα θα μπορούσε να είναι ένα αντικείμενο μελέτης. Ακόμη η χρήση των δεδομένων των κόμβων αρχής και τέλους αλλά και των συνδέσμων σε περιπτώσεις που έχουμε γνώση για τμήματα του δικτύου αλλά όχι για end-to-end μονοπάτια. Τέλος πέρα από τα όρια αυτής της εργασίας επέκταση μπορεί να γίνει στην χρήση νευρωνικών δικτύων για πρόβλεψη non-additive μετρικών του δικτύου, ή δικτύων που χρησιμοποιούν διαφορετικά πρωτόκολλα επιλογής μονοπατιών, για παράδειγμα πρωτόκολλα που δεν επιλέγουν το βέλτιστο μονοπάτι ως αυτό με την μικρότερη καθυστέρηση αλλά πιθανώς αυτό με την μικρότερη πιθανότητα congestion και κατα συνέπεια αποτυχίας του δικτύου.

## 6. Βιβλιογραφία

1. Vardi, Y. (1996). "Network Tomography: estimating source-destination traffic intensities from link data". *Journal of the American Statistical Association*. **91** (433): 365–377. [doi:10.2307/2291416](https://doi.org/10.2307/2291416). [JSTOR 2291416](https://www.jstor.org/stable/2291416).
2. Castro, R.; Coates, Mark; Liang, Gang; Nowak, Robert; [Yu, Bin](#) (2004). "Network Tomography: Recent Developments". *Statistical Science*. **19** (3): 499–517. [CiteSeerX 10.1.1.64.8631](https://www.cite-seer.org/10.1.1.64.8631). [doi:10.1214/088342304000000422](https://doi.org/10.1214/088342304000000422).
3. Network Kriging, David B. Chua, *Member, IEEE*, Eric D. Kolaczyk, *Member, IEEE*, and Mark Crovella, *Member, IEEE*
4. Y. Shavitt, X. Sun, A. Wool, and B. Yener, "Computing the unmeasured: An algebraic approach to Internet mapping," in *Proc. IEEE IN- FOCOM*, Apr. 2001, pp. 1646–1654.
5. Neural Network Tomography, Liang Ma, *Member, IEEE*, Ziyao Zhang, *Student Member, IEEE*, and Mudhakar Srivatsa, *Senior Member, IEEE*
6. TOM: a self-trained Tomography solution for Overlay networks Monitoring, Mohamed Rahali, Jean-Michel Sanner, Gerardo Rubino
7. Y. Chen, D. Bindel, and R. H. Katz, "Tomography-based overlay network monitoring," in *Proc. 2003 ACM SIGCOMM Conf. Internet Measurement*, 2003, pp. 216–231.
8. Network Performance Anomaly Detection and Localization ,Paul Barford University of Wisconsin ,Nick Duffield AT&T Labs–Research , Amos Ron University of Wisconsin, Joel Sommers Colgate University
9. Network Tomography using Routing Probability for Undeterministic Routing, Rie TAGYO<sup>†</sup>, Daisuke IKEGAMI<sup>†</sup>, and Ryoichi KAWAHARA<sup>††</sup>, *Members*
10. Quality of Transmission Estimation in WDM and Elastic Optical Networks Accounting for Space–Spectrum Dependencies , I. Sartzetakis, K. Christodoulopoulos, C. P. Tsekrekos, D. Syvridis, and E. Varvarigos

