ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ ΤΗΛΕΜΑΤΙΚΗΣ

Μελέτη και Υλοποίηση Μεθόδων Βαθιάς Μηχανικής Μάθησης για την Εκτίμηση της Δικτυακής Κίνησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΕΤΡΟΣ Δ. ΜΑΡΑΤΟΣ

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π

Αθήνα, Σεπτέμβριος 2024

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ ΤΗΛΕΜΑΤΙΚΗΣ

Μελέτη και Υλοποίηση Μεθόδων Βαθιάς Μηχανικής Μάθησης για την Εκτίμηση της Δικτυακής Κίνησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΕΤΡΟΣ Δ. ΜΑΡΑΤΟΣ

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 16η Σεπτεμβρίου 2024.

.....	.....	.....
Συμεών Παπαβασιλείου Καθηγητής Ε.Μ.Π	Ελένη Στάη Επ. Καθηγήτρια Ε.Μ.Π	Ιωάννα Ρουσσάκη Αν. Καθηγήτρια Ε.Μ.π

Αθήνα, Σεπτέμβριος 2024

.....
ΠΕΤΡΟΣ Δ. ΜΑΡΑΤΟΣ
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Ο πίνακας κίνησης είναι μια αφηρημένη δομή που περιγράφει την δικτυακή κίνηση (σε πακέτα ή bits ανά δευτερόλεπτο) που μεταφέρεται μεταξύ όλων των ζευγών ακραίων κόμβων μιας δικτυακής τοπολογίας. Η δομή αυτή καθίσταται ιδιαίτερα χρήσιμη στους διαχειριστές δικτύων μεγάλης κλίμακας για τον σχεδιασμό, την παρακολούθηση και την επίλυση προβλημάτων. Δυστυχώς, η απευθείας μέτρηση του πίνακα κίνησης είναι ακριβή υπολογιστικά και οδηγεί στην παραγωγή σημαντικού όγκου διαχειριστικών μηνυμάτων. Μια εναλλακτική λύση είναι η χρήση των ομαδοποιημένων φορτίων κίνησης που διέρχονται από τις ζεύξεις της τοπολογίας (και που μπορούν να αποκτηθούν εύκολα μέσω SNMP) ως δεδομένα για την έμμεση εξαγωγή του πίνακα κίνησης. Το πρόβλημα αυτό είναι γνωστό ως Εκτίμηση Πίνακα Κίνησης και ανήκει στην γενικότερη κατηγορία προβλημάτων της Τομογραφίας Δικτύου. Με δεδομένο ότι το πλήθος των ακραίων κόμβων είναι μεγαλύτερο από τον αριθμό των ζεύξεων του δικτύου, η Εκτίμηση Πίνακα Κίνησης μοντελοποιείται ως ένα γραμμικό αντίστροφο πρόβλημα που είναι υπό-ορισμένο (ill-posed), δηλαδή δεν επιδέχεται μοναδική λύση για είσοδο ενός συγκεκριμένου διανύσματος φορτίων ζεύξεων. Μια κατηγορία μοντέλων Βαθιάς Μηχανικής Μάθησης ειδικά σχεδιασμένη για την επίλυση αυτού του τύπου προβλημάτων είναι τα Αντιστρέψιμα Νευρωνικά Δίκτυα. Τα μοντέλα αυτά είναι εκ κατασκευής αντιστρέψιμα και κατόπιν εκπαίδευσης πάνω σε ένα γνωστό μετασχηματισμό δύνανται να αναπαραστήσουν και την αντίστροφη διαδικασία. Στην παρούσα εργασία κατασκευάζουμε μια αρχιτεκτονική Αντιστρέψιμου Νευρωνικού Δικτύου η οποία χρησιμοποιείται ως κύριος υπολογιστικός πυρήνας σε τρεις διαφορετικούς τρόπους εκπαίδευσης και λειτουργίας, με τον τελευταίο να μπορεί να παράγει και ρεαλιστικούς συνθετικούς πίνακες κίνησης. Επιπλέον τα τρία αυτά μοντέλα συνοδεύονται από ένα στάδιο προ-επεξεργασίας για την μείωση της διάστασης της εισόδου (τα Αντιστρέψιμα Νευρωνικά Δίκτυα επιβάλλουν οι διαστάσεις της εισόδου και της εξόδου να ταυτίζονται) το οποίο μοντελοποιείται μέσω ενός Αυτοκωδικοποιητή. Οι προτεινόμενες μέθοδοι αξιολογούνται πειραματικά πάνω σε ένα δημόσια διαθέσιμο δίκτυο κορμού, και συγκρίνονται τόσο μεταξύ τους όσο και με καθιερωμένες μεθόδους της βιβλιογραφίας με τα Αντιστρέψιμα Νευρωνικά Δίκτυα να εμφανίζουν πολύ ανώτερη απόδοση.

Λέξεις Κλειδιά:
τομογραφία δικτύου, παρακολούθηση δικτύου, πίνακας κίνησης, εκτίμηση πίνακα κίνησης, σύνθεση πίνακα κίνησης, μηχανική μάθηση, βαθιά μάθηση, νευρωνικά δίκτυα, γεννητικά μοντέλα, αυτοκωδικοποιητής, αντιστρέψιμα νευρωνικά δίκτυα

Abstract

The traffic matrix is an abstract structure that describes network traffic in terms of packets or data bytes per second transferred between all pairs of end nodes in a network topology. This structure is particularly useful for large-scale network administrators for planning, monitoring, and troubleshooting. Unfortunately, directly measuring the traffic matrix is computationally expensive and leads to the generation of a significant volume of management messages. An alternative solution is to use the aggregated traffic loads passing through the links of the topology (which can be easily obtained via SNMP) as data for the indirect estimation of the traffic matrix. This problem is known as Traffic Matrix Estimation and belongs to the broader category of Network Tomography problems. Given that the number of end nodes is greater than the number of network links, Traffic Matrix Estimation is modeled as an underdetermined linear inverse problem, meaning that it does not admit a unique solution for a given input vector of link loads. A category of Deep Learning models specifically designed to solve this type of problem is Invertible Neural Networks. These models are inherently invertible and, once trained on a known transformation, can represent the inverse process as well. In this work, we construct an Invertible Neural Network architecture, which is used as the main computational core in three different training and operation modes, the last of which can also produce realistic synthetic traffic matrices. Additionally, these three models are accompanied by a preprocessing stage for reducing the input dimension (Invertible Neural Networks require that the input and output dimensions match), which is modeled through an Autoencoder. The proposed methods are experimentally evaluated on a publicly available backbone network and are compared both among themselves and with established methods from the literature, with Invertible Neural Networks demonstrating significantly superior performance.

Keywords:
network tomography, network monitoring, traffic matrix, traffic matrix estimation, traffic matrix synthesis, machine learning, deep learning, neural networks, generative models, autoencoder, invertible neural networks

Ευχαριστίες

Με τη συγγραφή της παρούσας διπλωματικής εργασίας ολοκληρώνεται το ταξίδι των προπτυχιακών σπουδών μου, και για αυτό θα ήθελα να ευχαριστήσω τα άτομα που με καθοδήγησαν, βοήθησαν αλλά και ενέπνευσαν σε όλη την πορεία του.

Αρχικά, θα ήθελα να ευχαριστήσω τον Καθηγητή κ. Συμεών Παπαβασιλείου για την πολύτιμη βοήθειά του στην επιλογή και τη διαμόρφωση του θέματος αυτής της εργασίας. Η συμβολή του ήταν καθοριστική για τον προσδιορισμό των επιστημονικών ενδιαφερόντων μου αλλά και των μελλοντικών ακαδημαϊκών σχεδίων μου.

Επιπλέον, θα ήθελα να ευχαριστήσω τον αναπληρωτή καθηγητή κ. Βασίλειο Καρυώτη και τον υποψήφιο διδάκτορα κ. Γρηγόρη Κακκάβα. Η συμβολή τους υπήρξε καθοριστικής σημασίας για την επιτυχή ολοκλήρωση της διπλωματικής εργασίας χάρη στις εύστοχες παρατηρήσεις και οδηγίες τους.

Στο ταξίδι των προπτυχιακών σπουδών μου δεν βάδιζα μόνος, αλλά μαζί με τους φίλους μου, τους οποίους ευγνωμονώ για την στήριξη, το γέλιο αλλά και την αγάπη που προσέφεραν. Θέλω όμως να κάνω ειδική αναφορά στους Γεράσιμο Μουντάκη, Μάρκο Δεληγιάννη και Αθανάσιο Πυλιώτη οι οποίοι υπήρξαν πραγματικοί συνοδοιπόροι αλλά και πηγή έμπνευσης και ενθάρρυνσης σε αυτό το ταξίδι. Τους ευχαριστώ βαθιά μέσα από την καρδιά μου.

Τέλος, το ταξίδι αυτό δεν θα μπορούσε να πραγματοποιηθεί χωρίς την στήριξη, τη φροντίδα και την αγάπη των γονιών μου, οι οποίοι είναι πάντα δίπλα μου και με ωθούν στο να πετυχαίνω τα όνειρά μου. Δεν υπάρχουν λόγια για να εκφράσω την αμέριστη αγάπη αλλά και την ευγνωμοσύνη προς το πρόσωπό τους οπότε περιορίζομαι στην υπόσχεση ότι θα είμαι πάντα κοντά τους. Θα ήθελα λοιπόν να αφιερώσω την εργασία αυτή στους γονείς μου.

Πέτρος Μαράτος
Σεπτέμβριος 2024

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Στη σύγχρονη εποχή, η ραγδαία εξέλιξη και διάδοση του Διαδικτύου [1] με την ποικιλία των εφαρμογών που προσφέρει, το έχουν καταστήσει ως τον κύριο τρόπο επικοινωνίας παγκοσμίως. Η ανάπτυξη εφαρμογών όπως η ροή δεδομένων βίντεο, τα διαδικτυακά παιχνίδια και τα προγράμματα άμεσης ανταλλαγής μηνυμάτων καθώς και η εκρηκτική αύξηση των μεταδιδόμενων δεδομένων από κινητές συσκευές έχουν οδηγήσει στον σχηματισμό πολύπλοκων ροών κίνησης δεδομένων μεταξύ των οντοτήτων που συμμετέχουν στο παγκόσμιο αυτό δίκτυο.

Με δεδομένη την ολοένα και αυξανόμενη σημαντικότητα του Διαδικτύου και τις απαιτήσεις για ποιοτικότερη παροχή υπηρεσιών, η ακριβής γνώση της κίνησής του καθίσταται πλέον απαραίτητη. Το Διαδίκτυο στην πράξη αποτελείται από μία συλλογή κατανεμημένων δικτύων, με τον κάθε πάροχο να είναι υπεύθυνος για την παρακολούθηση και ρύθμιση της κίνησης που μεταδίδεται εντός του δικτύου του. Για την εξασφάλιση της ποιότητας υπηρεσίας αλλά και την πραγματοποίηση εργασιών διαχείρισης και συντήρησης του δικτύου, ο διαχειριστής χρειάζεται να γνωρίζει με ακρίβεια την κίνηση που ανταλλάσσεται μεταξύ των δικτυακών συσκευών, με την εξαγωγή αυτών των πληροφοριών σε μεγάλα δίκτυα κορμού να είναι υπολογιστικά ακριβή ή να εμπίπτει σε κανονισμούς προστασίας προσωπικών δεδομένων. Η γνώση της κίνησης που μεταδίδεται μεταξύ των δικτυακών οντοτήτων ενός δικτύου μπορεί να οργανωθεί σε μια αφηρημένη δομή που ονομάζεται πίνακας κίνησης.

Ο πίνακας κίνησης [2] είναι μια δομή όπου κάθε στοιχείο της μοντελοποιεί την ροή κίνησης, δηλαδή την ροή των πακέτων ή δεδομένων, που πηγάζουν από έναν κόμβο πηγή και αποστέλλονται μέσω του δικτύου σε έναν κόμβο προορισμού. Ανάλογα με το δίκτυο, οι ακραίοι κόμβοι μπορεί να είναι δρομολογητές ή και τερματικές συσκευές, δηλαδή συσκευές τελικών χρηστών. Σε κάθε περίπτωση, η ροή κίνησης ενός ζεύγους ακραίων κόμβων πιθανώς διέρχεται μέσω πολλαπλών ενδιάμεσων δικτυακών συσκευών όπως δρομολογητές ή μεταγωγείς, οι οποίες συνδέονται απευθείας μέσω ζεύξεων. Αυτό σημαίνει πως κάθε ζεύξη του δικτύου επιτρέπει την πολυπλεξία πολλαπλών ροών κίνησης σε ένα

ομαδοποιημένο φορτίο ζεύξης, το οποίο είναι εύκολα μετρήσιμο.

Λόγω των προκλήσεων της απευθείας μέτρησης των πινάκων κίνησης από τους ακραίους κόμβους αναπτύχθηκαν πολλαπλές τεχνικές για την έμμεση εξαγωγή των πινάκων κίνησης, ένα πρόβλημα που είναι γνωστό ως Εκτίμηση Πίνακα Κίνησης [3]. Το πρόβλημα αυτό εντάσσεται στο γενικότερο πλαίσιο έμμεσης εξαγωγής χαρακτηριστικών του δικτύου από εύκολα μετρήσιμες οντότητες που ονομάζεται Τομογραφία Δικτύου [4]. Συγχρόνως, η εκρηκτική ανάπτυξη των μοντέλων Μηχανικής Μάθησης και ειδικότερα των Νευρωνικών Δικτύων [5], τα οποία είναι κατάλληλα για την μοντελοποίηση μη γραμμικών αλλά και χώρο-χρονικών σχέσεων της δικτυακής κίνησης, έδωσε την απαραίτητη ώθηση για την εξέλιξη των τεχνικών εκτίμησης, καθιστώντας αυτά τα εργαλεία απαραίτητα για την αξιόπιστη διαχείριση των δικτύων.

1.1 Κίνητρο Εργασίας

Η ολοένα αυξανόμενη πολυπλοκότητα στην συμπεριφορά της Διαδικτυακής κίνησης αλλά και οι απαιτήσεις για εγγύηση στην ποιότητα παροχής υπηρεσιών (Quality of Service) οδηγούν τους διαχειριστές των δικτύων κορμού (τα δίκτυα των παρόχων που είναι υπεύθυνα για την διασύνδεση μικρότερης κλίμακας οικιακών και εταιρικών δικτύων με το υπόλοιπο διαδίκτυο) στην ανάγκη για ακριβή προσδιορισμό της κίνησης που ανταλλάσσεται μεταξύ των κόμβων του δικτύου. Με άλλα λόγια, στόχος των διαχειριστών είναι ο καθορισμός του πίνακα κίνησης του δικτύου ανά τακτά χρονικά διαστήματα μέτρησης. Η γνώση του πίνακα κίνησης μπορεί να ενσωματωθεί σε πληθώρα εφαρμογών διαχείρισης και σχεδιασμού των δικτύων [6] και άρα να εξασφαλίσει ότι το δίκτυο συμμορφώνεται με τις απαραίτητες προδιαγραφές για την αξιόπιστη παροχή υπηρεσιών.

Μια βασική χρήση του πίνακα κίνησης είναι η παροχή πληροφοριών για τη βελτιστοποίηση του δικτύου. Πιο συγκεκριμένα, ο πίνακας κίνησης δύναται να χρησιμοποιηθεί για τον σχεδιασμό των χωρητικοτήτων των ζεύξεων (Capacity Planning) [7], δηλαδή τη διασφάλιση επαρκούς εύρους ζώνης για την υποστήριξη των τωρινών αλλά και των μελλοντικών χρηστών του δικτύου, με ελάχιστο κόστος. Επιπλέον, η γνώση της κίνησης εντός του δικτύου διαμορφώνει τις πολιτικές δρομολόγησης, δηλαδή τον καθορισμό των συντομότερων μονοπατιών (το ελάχιστο πλήθος ενδιάμεσων κόμβων που απαιτούνται για την επικοινωνία ενός ζεύγους ακραίων κόμβων) αλλά και εναλλακτικών διαδρομών ούτως ώστε να επιτευχθεί καταμερισμός κίνησης στις ζεύξεις με στόχο την αποφυγή συμφόρησης.

Γίνεται εύκολα αντιληπτό πως ο πίνακας κίνησης ενός δικτύου αποτελεί ένα χρονικό στιγμιότυπο της συμπεριφοράς της κίνησης στις συνδέσεις των ακραίων κόμβων και παρουσιάζει σημαντικές μεταβολές με την πάροδο του χρόνου. Αυτό σημαίνει πως, με την εξασφάλιση άμεσης και ακριβούς μέτρησης των ροών κίνησης σε τακτά χρονικά διαστήματα, ο διαχειριστής δύναται να παρακολουθεί σε σχεδόν πραγματικό χρόνο τη συμπεριφορά του δικτύου και να ειδοποιείται για τυχόν αποκλίσεις ή προβλήματα στη λειτουργία του. Η διαδικασία αυτή είναι γνωστή ως Ανίχνευση Ανωμαλιών (Anomaly Detection) [8]

και αφορά την ανίχνευση, τον καθορισμό και την διαδικασία ανάκαμψης από προβλήματα που οφείλονται σε απρόσμενες και σημαντικές αλλαγές στην κίνηση του δικτύου. Τέτοια συμβάντα σχετίζονται με τεχνικά προβλήματα του δικτύου (δυσλειτουργία ζεύξης ή δικτυακής συσκευής) αλλά και με επιθέσεις κακόβουλου λογισμικού όπως Κατανεμημένες Επιθέσεις Άρνησης Υπηρεσιών (Distributed Denial of Service (DDoS))[9].

Έχοντας αναλύσει την σημασία της γνώσης του πίνακα κίνησης από τους διαχειριστές του δικτύου, προκύπτει το ερώτημα του πώς μπορούν να μετρηθούν αποδοτικά και με ακρίβεια οι ροές κίνησης μεταξύ όλων των ζευγών των ακραίων κόμβων ενός δικτύου, ειδικά όταν το δίκτυο είναι δίκτυο κορμού με χιλιάδες ακραίους κόμβους. Μια πρώτη προσέγγιση είναι η εγκατάσταση ειδικού λογισμικού όπως το NetFlow [10] σε όλους τους ακραίους κόμβους. Το λογισμικό αυτό είναι υπεύθυνο για την ανάλυση όλων των πακέτων που στέλνονται και λαμβάνονται από τον κόμβο με στόχο τον καθορισμό του ζεύγους πηγής-προορισμού και άρα την αύξηση του αντίστοιχου μετρητή που μοντελοποιεί την ροή κίνησης. Οι πληροφορίες αυτές θα πρέπει να στέλνονται σε έναν κεντρικό κόμβο διαχείρισης του δικτύου ανά τακτά χρονικά διαστήματα ώστε να συνδυαστούν και να προκύψει ο συνολικός πίνακας κίνησης. Η ανταλλαγή αυτών των πληροφοριών οδηγεί σε σημαντική αύξηση της κίνησης εντός του δικτύου, περιορίζοντας το διαθέσιμο εύρος ζώνης για την εξυπηρέτηση των τελικών χρηστών. Επιπλέον η ανάλυση των πακέτων μέσω λογισμικού στους ακραίους κόμβους, επιβάλλει μη αμελητέο κόστος και χρόνο επεξεργασίας των πακέτων γεγονός που μειώνει τη συνολική επίδοση του δικτύου. Τέλος, ακόμα και σε δίκτυα όπου η απευθείας μέτρηση των ροών κίνησης δεν επιβαρύνει την απόδοση του δικτύου, η αποστολή πληροφοριών για την κίνηση στους ακραίους κόμβους δύναται να παραβιάζει νομοθεσίες που αφορούν ευαίσθητα προσωπικά δεδομένα αλλά και να συνιστά κίνδυνο για την ασφάλεια του δικτύου (υποκλοπή πληροφοριών για την κίνηση και διάρθρωση του δικτύου).

Παρόλο που η απευθείας μέτρηση των ροών κίνησης στους ακραίους κόμβους του δικτύου είναι μη αποδοτική ή δεν επιτρέπεται, η μέτρηση της ομαδοποιημένης κίνησης των ροών στις ζεύξεις των ενδιάμεσων κόμβων είναι εύκολη και γρήγορη. Αυτό σημαίνει πως οι διαχειριστές του δικτύου μπορούν άμεσα να μάθουν πληροφορίες για την κατάσταση της κίνησης στις ζεύξεις και με δεδομένο ότι το πλήθος των ζεύξεων είναι συνήθως πολύ μικρότερο από το πλήθος των ζευγών ακραίων κόμβων, τα μηνύματα που ανταλλάσσονται για την συλλογή αυτών των μετρήσεων είναι αισθητά λιγότερα. Με τις εύκολα διαθέσιμες μετρήσεις της κίνησης των φορτίων ζεύξεων προκύπτει η ανάγκη για κατασκευή τεχνικών οι οποίες να εξάγουν έμμεσα τον πλήρη πίνακα κίνησης εξασφαλίζοντας τόσο υψηλή ακρίβεια όσο και ταχύτητα στην εκτίμηση ούτως ώστε οι πίνακες να λαμβάνονται σχεδόν σε πραγματικό χρόνο και άρα να χρησιμοποιηθούν ικανοποιητικά για την ανίχνευση ανωμαλιών. Το πρόβλημα εξαγωγής των ροών κίνησης από τις πολυπλεγμένες μετρήσεις των φορτίων ζεύξεων ονομάζεται Εκτίμηση Πίνακα Κίνησης και ανήκει στην ευρύτερη κατηγορία προβλημάτων έμμεσης εξαγωγής χαρακτηριστικών του δικτύου από διαθέσιμα δεδομένα που ονομάζεται Τομογραφία Δικτύου. Το πρόβλημα αυτό δύναται να μοντελοποιηθεί ως ένα υπό-ορισμένο αντίστροφο πρόβλημα και άρα για δεδομένες μετρήσεις των φορτίων των ζεύξεων δεν επιδέχεται μοναδική λύση, γεγονός που αυξάνει σημαντικά

την πολυπλοκότητά του. Λαμβάνοντας όμως υπόψιν τη σημασία του πίνακα κίνησης στη διαμόρφωση και διαχείριση των δικτύων, αλλά και τις δυσκολίες στον απευθείας προσδιορισμό του, η ανάγκη για εύρεση αποδοτικής και χωρίς σφάλματα μεθόδου για την επίλυση της Εκτίμησης Πίνακα Κίνησης καθίσταται ολοένα και περισσότερο απαραίτητη.

1.2 Στόχοι Διπλωματικής

Για την επίλυση του προβλήματος της Εκτίμησης Πίνακα Κίνησης έχουν προταθεί ποικίλες μέθοδοι που αποσκοπούν στην αναλυτική ή στατιστική προσέγγισή του μέσω υποθέσεων για τις χώρο-χρονικές συσχετίσεις των ροών κίνησης. Η εκρηκτική ανάπτυξη των μοντέλων Μηχανικής Μάθησης, τα οποία δύνανται να μοντελοποιήσουν μη γραμμικούς και πολύπλοκους μετασχηματισμούς των δεδομένων χωρίς να βασίζονται σε σύνθετη μαθηματική επεξεργασία, έδωσε νέα πνοή στην κατασκευή μεθόδων για την Εκτίμηση Πίνακα Κίνησης καθώς τα μοντέλα αυτά παρουσιάζουν σημαντικά μειωμένα σφάλματα εκτίμησης και ελαχιστοποιούν τον χρόνο που απαιτείται για την παραγωγή του πίνακα κίνησης. Ο πρώτος λοιπόν στόχος αυτής της διπλωματικής εργασίας είναι η μελέτη και η παρουσίαση της πλούσιας βιβλιογραφίας σχετικά με την πληθώρα τεχνικών για την Εκτίμηση Πίνακα Κίνησης δίνοντας ιδιαίτερη έμφαση στις αρχιτεκτονικές μοντέλων Βαθιάς Μηχανικής Μάθησης.

Όπως αναφέρθηκε και στο Κεφάλαιο 1.1, η Εκτίμηση Πίνακα Κίνησης μπορεί να εκφραστεί ως ένα αντίστροφο πρόβλημα, με τον μετασχηματισμό του πίνακα κίνησης στα φορτία των ζεύξεων να είναι γνωστός (η γνώση παρέχεται μέσω πληροφοριών για την δρομολόγηση). Στο πλαίσιο αυτό, η επίλυση του προβλήματος ανάγεται στην αναζήτηση τρόπου για την αποδοτική αναπαράσταση της αντίστροφης άγνωστης διαδικασίας. Μια πρόσφατα αναπτυσσόμενη κατηγορία μοντέλων Βαθιάς Μηχανικής Μάθησης η οποία ειδικεύεται στην αποδοτική μοντελοποίηση αντίστροφων συστημάτων ονομάζεται Αντιστρέψιμα Νευρωνικά Δίκτυα [11]. Τα μοντέλα αυτά εκπαιδεύονται πάνω στον γνωστό μετασχηματισμό του προβλήματος, μαθαίνοντας έμμεσα και την αντίστροφη κατεύθυνση εκμεταλλευόμενα την εκ κατασκευής αντιστρεψιμότητά τους. Βασικός λοιπόν στόχος αυτής της διπλωματικής εργασίας είναι η ανάπτυξη μεθόδων που ενσωματώνουν τα Αντιστρέψιμα Νευρωνικά Δίκτυα ως υπολογιστικό πυρήνα με στόχο την αποδοτική Εκτίμηση Πίνακα Κίνησης. Επιπλέον μία από τις μεθόδους δύναται να χρησιμοποιηθεί και σε εργασίες παραγωγής ρεαλιστικών συνθετικών πινάκων κίνησης. Οι προτεινόμενες μέθοδοι αξιολογούνται πειραματικά πάνω σε ένα δημόσια διαθέσιμο σύνολο δεδομένων που έχει συλλεχθεί από πραγματικό δίκτυο κορμού.

1.3 Συνεισφορές

Σε αυτήν την ενότητα παρουσιάζονται συνοπτικά οι κύριες συνεισφορές της εργασίας:

- Μελέτη Βιβλιογραφίας και παρουσίαση μεθόδων για την επίλυση του προβλήματος της Εκτίμησης Πίνακα Κίνησης. Ιδιαίτερη έμφαση θα δοθεί στις τεχνικές που αναπτύσσουν αρχιτεκτονικές Βαθιάς Μηχανικής Μάθησης.
- Αναλυτική περιγραφή και παρουσίαση των κύριων αρχιτεκτονικών των Αντιστρέψιμων Νευρωνικών Δικτύων.
- Κατασκευή μοντέλου που βασίζεται στην αρχιτεκτονική των Αυτοκωδικοποιητών (Autoencoders) για την πραγματοποίηση Ελάττωσης Διάστασης.
- Κατασκευή μεθόδου αναφοράς χρησιμοποιώντας Αντιστρέψιμα Νευρωνικά Δίκτυα για μια πρώτη εκτίμηση της απόδοσης αυτής της κατηγορίας μοντέλων στην Εκτίμηση Πίνακα Κίνησης.
- Ανάπτυξη της βασικής μεθόδου που εκμεταλλεύεται πλήρως την αντιστρεψιμότητα των Αντιστρέψιμων Νευρωνικών Δικτύων.
- Επέκταση της βασικής μεθόδου ούτως ώστε το μοντέλο να λειτουργεί ως μέσο παραγωγής ρεαλιστικών συνθετικών πινάκων κίνησης. Το μοντέλο αυτό επιλύει την Εκτίμηση Πίνακα Κίνησης μετασχηματίζοντάς την σε πρόβλημα βελτιστοποίησης.
- Πειραματική αξιολόγηση όλων των προτεινόμενων μεθόδων πάνω στο δημόσιο σύνολο δεδομένων του δικτύου κορμού Abilene [12].
- Για την ανάδειξη της υπεροχής των μεθόδων που ενσωματώνουν Αντιστρέψιμα Νευρωνικά Δίκτυα, κατασκευάζονται και αξιολογούνται στο ίδιο σύνολο δεδομένων δύο βασικές αρχιτεκτονικές καθιερωμένων μοντέλων Βαθιάς Μηχανικής Μάθησης από την βιβλιογραφία.

1.4 Δομή Εργασίας

Η παρούσα διπλωματική εργασία οργανώνεται ως εξής. Στο Κεφάλαιο 1 παρέχεται μια σύντομη εισαγωγή στο πρόβλημα της Εκτίμησης Πίνακα Κίνησης και αναλύονται οι στόχοι και οι συνεισφορές της εργασίας. Στο Κεφάλαιο 2. ορίζεται επίσημα η Εκτίμηση Πίνακα Κίνησης ως υποπρόβλημα της Τομογραφίας Δικτύου και παρουσιάζεται εκτενώς η σχετιζόμενη βιβλιογραφία με κύρια έμφαση τις μεθόδους που ενσωματώνουν μοντέλα Βαθιάς Μηχανικής Μάθησης. Στο Κεφάλαιο 3. περιγράφεται η κεντρική ιδέα μαζί με τις βασικές αρχιτεκτονικές των Αντιστρέψιμων Νευρωνικών Δικτύων καθώς και μια εισαγωγή σε εναλλακτικές αντιστρέψιμες προσεγγίσεις. Στο Κεφάλαιο 4. παρουσιάζονται αναλυτικά όλες οι προτεινόμενες μεθοδολογίες. Πιο συγκεκριμένα παρέχεται μια βασική

εποπτεία της κύριας ροής εργασίας, προσδιορίζεται η γενική αρχιτεκτονική των νευρωνικών δικτύων που θα χρησιμοποιηθούν και αναλύονται οι τρεις τρόποι λειτουργίας των Αντιστρέψιμων Νευρωνικών Δικτύων. Στο Κεφάλαιο 5. πραγματοποιείται η πειραματική αξιολόγηση των προτεινόμενων μεθόδων και η σύγκρισή τους με τις βασικές αρχιτεκτονικές Μηχανικής Μάθησης που έχουν χρησιμοποιηθεί στη βιβλιογραφία. Το Κεφάλαιο 6. συνιστά τον επίλογο της εργασίας όπου ερευνώνται οι μελλοντικές επεκτάσεις της χρήσης των Αντιστρέψιμων Νευρωνικών Δικτύων στην Εκτίμηση Πίνακα Κίνησης. Τέλος στο Παράρτημα Α παρέχονται λεπτομέρειες σχετικά με το περιβάλλον εργασίας στο οποίο εκτελέστηκε η πειραματική διαδικασία, ενώ στο Παράρτημα Β παρουσιάζεται ένα ενδεικτικό τμήμα του κώδικα των πειραμάτων.

Κεφάλαιο 2

Εκτίμηση Πίνακα Κίνησης

2.1 Τομογραφία Δικτύου

Η Τομογραφία Δικτύου (Network Tomography) [13, 4] είναι μια τεχνική παρακολούθησης δικτύων όπου περιλαμβάνει την εκτίμηση της επίδοσης αλλά και των χαρακτηριστικών του δικτύου μέσα από μετρήσεις της κίνησης σε ένα περιορισμένο υποσύνολο δικτυακών κόμβων. Περιλαμβάνει προβλήματα που έχουν ως στόχο την έμμεση εξαγωγή μεγεθών, που είναι δύσκολο να μετρηθούν απευθείας, χρησιμοποιώντας ομαδοποιημένες και εύκολα προσβάσιμες τιμές τους.

Με βάση την προσέγγιση στο πρόβλημα και το επίπεδο ανάλυσης, η Τομογραφία Δικτύου ταξινομείται σε τρεις κατηγορίες (όπως περιγράφονται στο [4]):

- Εκτίμηση Παραμέτρων σε Επίπεδο Ζεύξης
- Εκτίμηση Έντασης Κίνησης σε Επίπεδο Μονοπατιού
- Εξαγωγή Τοπολογίας

Η πρώτη κατηγορία αποτελείται από μετρήσεις κίνησης από άκρο σε άκρο με στόχο την εκτίμηση προσθετικών ή πολλαπλασιαστικών χαρακτηριστικών στις ζεύξεις του δικτύου. Για παράδειγμα, μετρώντας χρονικές καθυστερήσεις στην ανταλλαγή πακέτων μεταξύ ακραίων κόμβων και με την υπόθεση ότι η συνολική καθυστέρηση στην επικοινωνία υπολογίζεται ως το άθροισμα των επιμέρους καθυστερήσεων σε κάθε ζεύξη του μονοπατιού που ενώνει τους δύο κόμβους, τίθεται ως στόχος η εύρεση της καθυστέρησης που υφίσταται σε κάθε ζεύξη. Μια μετρική που διαθέτει πολλαπλασιαστική ιδιότητα είναι η εύρεση του ρυθμού απωλειών σε κάθε ζεύξη και ανάγεται σε προσθετική με την χρήση λογαριθμικής συνάρτησης. Στην Εκτίμηση Παραμέτρων σε Επίπεδο Ζεύξης το πρόβλημα ανάγεται στην αποδοτική επιλογή ζευγών ακραίων κόμβων για την πραγματοποίηση της μέτρησης. Εφαρμογές αυτής της κατηγορίας μπορούν να βρεθούν στα [14, 15, 16, 17].

Η Εκτίμηση Έντασης Κίνησης σε Επίπεδο Μονοπατιού αφορά την εκτίμηση της κίνησης μεταξύ όλων των ζευγών ακραίων κόμβων του δικτύου, έχοντας γρήγορη και χωρίς επιβάρυνση πρόσβαση στην ομαδοποιημένη/πολυπλεγμένη κίνηση σε κάθε ζεύξη του δικτύου. Η Εκτίμηση Πίνακα Κίνησης εμπίπτει σε αυτήν την κατηγορία Τομογραφίας Δικτύου.

Τέλος, η Εξαγωγή Τοπολογίας έχει ως στόχο να συμπεράνει πληροφορίες για την δομή του δικτύου, που συνήθως εκφράζονται από τον πίνακα δρομολόγησης, όταν η δομή αυτή είναι άγνωστη. Πιο συγκεκριμένα, επιστρατεύονται μετρήσεις από άκρο σε άκρο χωρίς την συνεργασία των ενδιάμεσων κόμβων του δικτύου για να ανιχνευθεί η ομοιότητα στην κίνηση των ζευγών κόμβων η οποία είναι επιθυμητό να είναι αύξουσα μονοτονική συνάρτηση του πλήθους ζεύξεων στο μονοπάτι που ενώνει τους κόμβους. Με αυτόν τον τρόπο δύναται να αποκαλυφθεί η λογική τοπολογία του δικτύου με μεθόδους όπως ιεραρχική ομαδοποίηση [18, 19], μέγιστη πιθανοφάνεια [20] και Μπεϋζιανή εξαγωγή [21].

2.2 Ορισμός Προβλήματος

Στο πλαίσιο της Εκτίμησης Έντασης Κίνησης σε Επίπεδο Μονοπατιού στην Τομογραφία Δικτύου ορίζεται το πρόβλημα της Εκτίμησης Πίνακα Κίνησης.

Ορισμός 2.2.1: Η Εκτίμηση Πίνακα Κίνησης (ΕΠΚ) αποτελεί την διαδικασία εξαγωγής της δικτυακής κίνησης μεταξύ όλων των ζευγών κόμβων πηγής – προορισμού μέσα από την πολυπλεγμένη κίνηση στις ζεύξεις του δικτύου.

Κάθε ζεύγος πηγής – προορισμού χαρακτηρίζεται από μία ροή κίνησης πακέτων ή bits δεδομένων ανά δευτερόλεπτο (Origin-Destination Flow) και όλες οι ροές μπορούν να συγκεντρωθούν σε μία δομή που ονομάζεται Πίνακας Κίνησης Δικτύου. Για την επικοινωνία μεταξύ δύο κόμβων παρεμβάλλονται πολλές ενδιάμεσες ζεύξεις που συνδέουν δρομολογητές μεταξύ τους αν μελετάμε IP δίκτυα κορμού όπως τα δίκτυα Παρόχων Υπηρεσιών Διαδικτύου. Αυτό σημαίνει πως, ανάλογα με την πολιτική δρομολόγησης, είναι δυνατό διαφορετικές ροές κίνησης να έχουν κοινό σημαντικό μέρος του μονοπατιού ενδιάμεσων κόμβων μέχρι να φτάσουν στον προορισμό. Συνεπώς, μέσα από κάθε ζεύξη διέρχονται πολλές ροές κίνησης οι οποίες πολυπλέκονται σε μια ομαδοποιημένη κίνηση που ονομάζεται φορτίο ζεύξης (link load).

Ας θεωρήσουμε ένα δίκτυο με n κόμβους και m ζεύξεις. Η Εκτίμηση Πίνακα Κίνησης μπορεί να περιγραφεί από ένα σύστημα γραμμικών εξισώσεων ως εξής:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon \tag{2.1}$$

όπου:

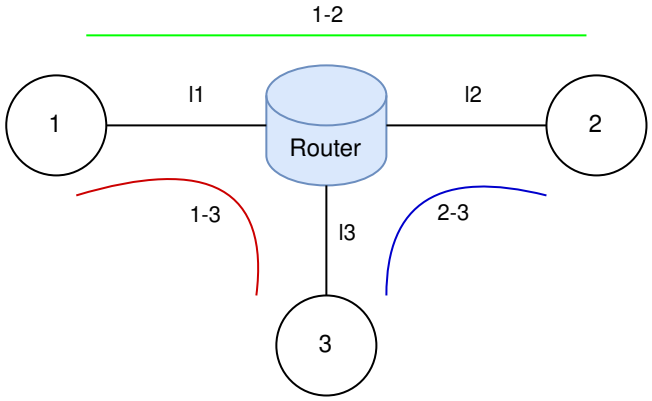
- **x** είναι ο πίνακας κίνησης του δικτύου διάστασης $n \times n$ ο οποίος υποδεικνύει τον όγκο της κίνησης για όλα τα ζεύγη πηγής προορισμού του δικτύου. Συνήθως αναπαρίσταται ως ένα στοιβαγμένο διάνυσμα διάστασης $n^2 \times 1$.

- **A** είναι ο πίνακας δρομολόγησης του δικτύου διάστασης $m \times n^2$ ο οποίος έχει στη θέση (i, j) την τιμή 1 αν η ροή του j-στου ζεύγους πηγής-προορισμού διέρχεται από την i-στη ζεύξη του δικτύου, αλλιώς έχει την τιμή 0.
- **y** είναι το διάνυσμα διάστασης $m \times 1$ το οποίο εκφράζει το φορτίο κίνησης που διέρχεται από την κάθε ζεύξη.
- ϵ είναι το διάνυσμα διάστασης $m \times 1$ το οποίο εκφράζει προσθετικό θόρυβο αυθαίρετα πολύπλοκης μορφής, με στόχο τον έλεγχο της ανθεκτικότητας της διαδικασίας εκτίμησης.

Σε αντιπαραβολή με την Τομογραφία Δικτύου, ο πίνακας x συνιστά τις κρυφές παραμέτρους του δικτύου ενώ το διάνυσμα y είναι οι μετρήσεις που αποκτώνται εύκολα και οδηγούν στην έμμεση εκτίμηση των ζητούμενων μεγεθών. Στις περισσότερες εφαρμογές εκτίμησης, ο πίνακας δρομολόγησης θεωρείται σταθερός για όλες τις χρονικές στιγμές (στατική δρομολόγηση) ενώ αν είχε επιλεχθεί δυναμική δρομολόγηση ο A θα ήταν συνάρτηση του χρόνου $A(t)$. Επιπλέον, το διάνυσμα θορύβου μπορεί να θεωρηθεί αμελητέο και άρα να αγνοηθεί, οδηγώντας στην τελική έκφραση του συστήματος:

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{2.2}$$

Για την καλύτερη κατανόηση του προβλήματος, παρατίθεται ένα παράδειγμα τοπολογίας δικτύου 3 κόμβων (Σχήμα 2.1) σύμφωνα με την οποία ορίζονται οι οντότητες y, A, x και αντικαθίστανται στην σχέση (2.2).



Σχήμα 2.1: Τοπολογία δικτύου με 3 κόμβους και 1 δρομολογητή

Η σχέση (2.2) γράφεται:

$$\begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{OD}_{1,1} \\ \mathrm{OD}_{1,2} \\ \mathrm{OD}_{1,3} \\ \mathrm{OD}_{2,1} \\ \mathrm{OD}_{2,2} \\ \mathrm{OD}_{2,3} \\ \mathrm{OD}_{3,1} \\ \mathrm{OD}_{3,2} \\ \mathrm{OD}_{3,3} \end{bmatrix} \tag{2.3}$$

Το πρόβλημα της Εκτίμησης Πίνακα Κίνησης ανάγεται στην επίλυση του αντίστροφου γραμμικού συστήματος που περιγράφεται από την εξίσωση (2.3), για την εξαγωγή του πίνακα x από τις μετρήσεις ζεύξεων y για γνωστό και σταθερό πίνακα δρομολόγησης A. Δεδομένου ότι, στην γενική περίπτωση, το πλήθος των ζεύξεων του δικτύου είναι πολύ μικρότερο από το τετράγωνο των ακραίων κόμβων, δηλαδή $m < n^2$, ο πίνακας δρομολόγησης δεν προκύπτει πλήρους βαθμού. Το γεγονός αυτό καθιστά το σύστημα υπό-ορισμένο και άρα επιδέχεται πολλαπλές λύσεις που ικανοποιούν τις μετρήσιμες τιμές του φορτίου στις ζεύξεις. Οι λύσεις αυτές φυσικά διαφέρουν από την πραγματική τιμή του πίνακα κίνησης που οδήγησε στην παραγωγή του διανύσματος y. Αν ο πίνακας A ήταν πλήρους βαθμού, τότε το πρόβλημα θα αναγόταν στην εύρεση του αντίστροφου A^{-1} με τον οποίο θα πολλαπλασιαζόταν το διάνυσμα y για τον υπολογισμό μοναδικής λύσης για τον πίνακα x.

2.3 Μετρήσεις Φορτίου Ζεύξεων

Όπως έχουμε αναφέρει, η απευθείας παρακολούθηση των ροών κίνησης σε κάθε κόμβο του δικτύου καθώς και η αποπολύπλεξη της κίνησης στις ζεύξεις επιβαρύνουν σημαντικά την επίδοση των δικτυακών συσκευών. Μια γρήγορη και εύκολα υλοποιήσιμη εναλλακτική αποτελεί η μέτρηση του συνολικού φορτίου κίνησης σε κάθε ζεύξη του δικτύου. Το φορτίο ζεύξης αποτελεί τον συνολικό αριθμό bytes ή πακέτων που διέρχονται μέσα από την ζεύξη και είναι εύκολα ανακτήσιμο μέσω του ευρέως διαδεδομένου πρωτοκόλλου διαχείρισης δικτύων Simple Network Management Protocol (SNMP) [22].

Τα δεδομένα του SNMP αποθηκεύονται για κάθε διεπαφή σε μια δενδρική δομή δεδομένων που ορίζει διαχειριζόμενα αντικείμενα με τυποποιημένο τρόπο και που ονομάζεται Management Information Base (MIB). Τα αντικείμενα της MIB, δηλαδή ο αριθμός των εισερχόμενων ή εξερχόμενων bytes/πακέτων για κάθε ζεύξη της δικτυακής συσκευής μπορούν να ανακτηθούν με μηνύματα snmpget μέσω του πρωτοκόλλου διαχείρισης SNMP και να σταλούν κεντρικά στο σύστημα διαχείρισης δικτύου. Το σύστημα διαχείρισης στέλνει περιοδικά αιτήματα SNMP για την μέτρηση του φορτίου (τυπικά κάθε 5 λεπτά) μέσω πρωτοκόλλου μεταφοράς UDP στη θύρα 161 [2].

Το SNMP, αν και είναι ένας εύκολος και οικονομικός τρόπος για την ανάκτηση του φορτίου ζεύξης, παρουσιάζει ορισμένα ελαττώματα. Αρχικά, λόγω του ότι χρησιμοποιεί ως υποδομή μεταφοράς το πρωτόκολλο UDP, το SNMP καθίσταται επιρρεπές σε σφάλματα κατά την επικοινωνία των SNMP πρακτόρων (οι δικτυακές συσκευές) με το κεντρικό σύστημα διαχείρισης καθώς το UDP δεν υλοποιεί έλεγχο απωλειών και ορθότητας στα πακέτα. Επιπλέον, δύναται να υπάρξουν διακυμάνσεις στο χρονικό διάστημα ανάμεσα σε διαδοχικές μετρήσεις λόγω ανεπαρκώς σχεδιασμένων υλοποιήσεων των SNMP πρακτόρων και του συστήματος διαχείρισης δικτύου. Τέλος, το SNMP δεν μπορεί να παρέχει στον διαχειριστή περισσότερες πληροφορίες για την κίνηση στις ζεύξεις καθώς δεν μπορεί να διαχωρίσει τις ροές κίνησης και να εξάγει σύνθετα στατιστικά.

Παρόλα τα ελαττώματά του, το SNMP παραμένει το εργαλείο αιχμής για την ανάκτηση του φορτίου των ζεύξεων και οδηγεί στην εφαρμογή τεχνικών εκτίμησης δικτυακής κίνησης για την αξιόπιστη εξαγωγή του πίνακα κίνησης.

2.4 Σχετιζόμενη Βιβλιογραφία

Η εκτίμηση πινάκων κίνησης έχει μελετηθεί εκτενώς τις τελευταίες δεκαετίες εξερευνώντας ένα μεγάλο φάσμα μεθοδολογιών όπως στατιστικές τεχνικές, που λειτουργούν ως αρχικές κατανομές για την μοντελοποίηση των OD Flows, αλλά και επιστράτευση μοντέλων βαθιάς μηχανικής μάθησης για την επίλυση του αντίστροφου προβλήματος.

Μια πρώτη κατηγορία τεχνικών για την εκτίμηση πινάκων κίνησης προτάθηκε από τους Vardi et al. [23] και Cao et al. [24] όπου κάθε OD Flow μοντελοποιείται ως μια ανεξάρτητη τυχαία μεταβλητή που ακολουθεί την κατανομή Poisson και Gaussian αντίστοιχα. Με αυτόν τον τρόπο επιβάλλονται επιπλέον περιορισμοί στο υπό-ορισμένο σύστημα εξισώσεων του προβλήματος και άρα επιτυγχάνεται η εκτίμηση του πίνακα που συμμορφώνεται με τις προσδιοριζόμενες σχέσεις. Τα μοντέλα αυτά όμως δεν εκμεταλλεύονται την χωρική και χρονική συσχέτιση μεταξύ των OD Flows και η επίδοσή τους βασίζεται σημαντικά στην αρχική κατανομή με την οποία μοντελοποιούνται. Επιπλέον η κίνηση σε πραγματικά δίκτυα είναι πολύ πιο πολύπλοκη και άρα η υπόθεση ότι τα OD Flows είναι ανεξάρτητα και ακολουθούν συγκεκριμένη δομή καθιστά τις μεθόδους αυτές μη ρεαλιστικές για την εκτίμηση κίνησης γεγονός που μειώνει αισθητά την απόδοσή τους. Μια δεύτερη γενιά μεθόδων χρησιμοποιεί επιπλέον πληροφορία που προέρχεται από μετρήσεις δευτερευόντων πηγών του συστήματος όπως η εξαγωγή των φορτίων ξεύξης με τη βοήθεια του πρωτοκόλλου SNMP. Η βασικότερη από αυτές τις μεθόδους η οποία έχει χρησιμοποιηθεί εκτενώς σε εμπορικές εφαρμογές προτάθηκε από τους Zhang et al. [25] και ονομάζεται Tomogravity. Θεωρώντας έναν κόμβο πηγής i και έναν κόμβο προορισμού j και υποθέτοντας ανεξαρτησία ανάμεσα στην πηγή και τον προορισμό ορίζουμε ως x_i^{IN} όλη την εισερχόμενη κίνηση στο δίκτυο μέσω του i και ως x_j^{OUT} όλη την εξερχόμενη κίνηση του δικτύου μέσω του j. Η κίνηση του δικτύου από τον i στον j αναπαρίσταται

με βάση το απλό μοντέλο βαρύτητας ως εξής:

$$x_{i,j} = x_i^{\text{IN}} \left(\frac{x_j^{\text{OUT}}}{\sum_j x_j^{\text{OUT}}} \right) \tag{2.4}$$

Αυτό σημαίνει ότι η κίνηση που προωθεί ο i στον j είναι ανάλογη της κίνησης που εξέρχεται από το δίκτυο μέσω του j ως προς την συνολική εξερχόμενη κίνηση. Το μοντέλο υποθέτει ανεξαρτησία πηγής-προορισμού και όχι ανεξαρτησία των OD Flows όπως υπέθεταν τα μοντέλα της προηγούμενης κατηγορίας. Για την μοντελοποίηση δικτύων όπου η πολιτική δρομολόγησης δεν οδηγεί σε ανεξαρτησία πηγής προορισμού γίνεται χρήση γενικευμένων σχέσεων βαρύτητας. Τα μοντέλα βαρύτητας λειτουργούν ως αρχικές κατανομές για τα OD Flows αξιοποιώντας τις χωρικές τους συσχετίσεις και η μέθοδος ανάγεται σε πρόβλημα ελαχιστοποίησης του μέσου τετραγωνικού σφάλματος της σχέσης:

$$\min \left(\|y - Ax\|_2^2 + \lambda^2 K(x \| x') \right) \tag{2.5}$$

Ο πρώτος παράγοντας της σχέσης ελαχιστοποιείται όταν ο πίνακας-διάνυσμα x παράγει φορτία ζεύξης Ax που ταυτίζονται με αυτά που μετρήθηκαν μέσω SNMP. Ο δεύτερος παράγοντας εξασφαλίζει πως ο πίνακας που θα επιλεχθεί συμμορφώνεται με μια αρχική εκτίμηση x' που παράγεται από το μοντέλο βαρύτητας (χρήση Kullback-Leibler και παραμέτρου για έλεγχο συμμόρφωσης και κανονικοποίηση).

Μία άλλη μέθοδος που εντάσσεται σε αυτήν την κατηγορία ονομάζεται Αλλαγή Δρομολόγησης (Route Change) και αναπτύχθηκε από τους Soule et al. [26]. Με την μέθοδο αυτή οι συγγραφείς αποσκοπούν στην αύξηση του βαθμού του πίνακα A τροποποιώντας τα βάρη των ζεύξεων του δικτύου και εξάγοντας νέες SNMP μετρήσεις με βάση το νέο σχήμα δρομολόγησης. Αυτό έχει ως αποτέλεσμα την απόκτηση επιπρόσθετων γραμμικώς ανεξάρτητων σχέσεων του συστήματος και επαναλαμβάνεται μέχρις ότου ο τελικός πίνακας δρομολόγησης A (που τώρα περιέχει πολλά διαφορετικά στιγμιότυπα δρομολόγησης) να είναι πλήρους βαθμού. Η τεχνική αυτή δεν εξασφαλίζει την λύση του προβλήματος καθώς τα καινούρια φορτία ζεύξεων καταγράφονται σε επόμενες χρονικά στιγμές διότι απαιτείται η πάροδος ενός χρονικού διαστήματος για την αλλαγή της πολιτικής δρομολόγησης. Στην Αλλαγή Δρομολόγησης τα OD Flows είναι ανεξάρτητα και λαμβάνονται υπόψη μόνο οι χρονικές συσχετίσεις ανά ροή.

Στις μεθόδους που αναφέρθηκαν κατασκευάζεται ένα μοντέλο αναπαράστασης των πινάκων κίνησης και αποκτώνται μετρήσεις από δευτερεύουσες πηγές του συστήματος χωρίς όμως να γίνεται χρήση πραγματικών πινάκων κίνησης που έχουν εξαχθεί σε προηγούμενες χρονικές στιγμές από το δίκτυο. Με την ανάπτυξη και συνεχή βελτίωση των μεθόδων παρακολούθησης δικτύου όπως το NetFlow καθίσταται δυνατή η μέτρηση των OD Flows στους κόμβους του δικτύου και η εξαγωγή μερικών ή ολικών πινάκων κίνησης. Η διαδικασία αυτή παραμένει πολύ ακριβή για να χρησιμοποιείται συνεχώς, δύναται όμως να ενισχύσει τις μεθόδους εκτίμησης του πίνακα κίνησης με περιορισμένη αλλά στοχευμένη αξιοποίηση.

Μια τεχνική που ενσωματώνει πραγματικούς πίνακες κίνησης για την επίλυση του προβλήματος προτείνεται από τους Papagiannaki et al. [27] και ονομάζεται μέθοδος

Fanout. Θεωρώντας το μέγεθος $x(i, j, t)$, το οποίο συμβολίζει την κίνηση που εισέρχεται στο δίκτυο μέσω του κόμβου i και που στη συνέχεια εξέρχεται μέσω του κόμβου j τη χρονική στιγμή t , υπολογίζονται οι fanout ποσότητες για κάθε κόμβο ως:

$$f(i, j, t) = \frac{x(i, j, t)}{\sum_j x(i, j, t)} \tag{2.6}$$

Οι ποσότητες αυτές περιγράφουν το ποσοστό της κίνησης που εισέρχεται στον κόμβο i και εξέρχεται από τον j ως προς την συνολική εισερχόμενη κίνηση. Έχει παρατηρηθεί πως τα fanout διανύσματα παρουσιάζουν ημερήσια περιοδικότητα και άρα η μέθοδος Fanout μετρά τα πραγματικά OD Flows σε κάθε κόμβο για 24 ώρες ανά τακτά χρονικά διαστήματα (για παράδειγμα ανά 10 λεπτά). Στη συνέχεια, για κάθε διάστημα μέτρησης, υπολογίζονται τα fanout διανύσματα. Για την εκτίμηση του πίνακα $x(i, j, t)$ για κάποια επόμενη χρονική στιγμή t επιλέγεται το κατάλληλο fanout διάνυσμα (αυτό που αντιστοιχεί στο διάστημα μέτρησης που περιέχει την t) και πολλαπλασιάζεται με τα φορτία ζεύξεων που προσπίπτουν στον κόμβο i για την χρονική στιγμή t. Η μέθοδος λοιπόν δεν προσπαθεί να επιλύσει το αντίστροφο σύστημα, αλλά αξιοποιεί τόσο τις χωρικές όσο και τις χρονικές συσχετίσεις των OD Flows μέσω των fanout και της περιοδικότητας του συστήματος. Σε ένα δυναμικό σύστημα η κίνηση στου κόμβους μπορεί να μεταβληθεί αισθητά και άρα η μέθοδος απαιτεί μια αποδοτική επανάληψη της μέτρησης των πραγματικών πινάκων κίνησης (συνήθως πραγματοποιείται μόνο στους κόμβους που εμφανίζουν την μεγαλύτερη μεταβολή) για την ενημέρωση των fanout διανυσμάτων.

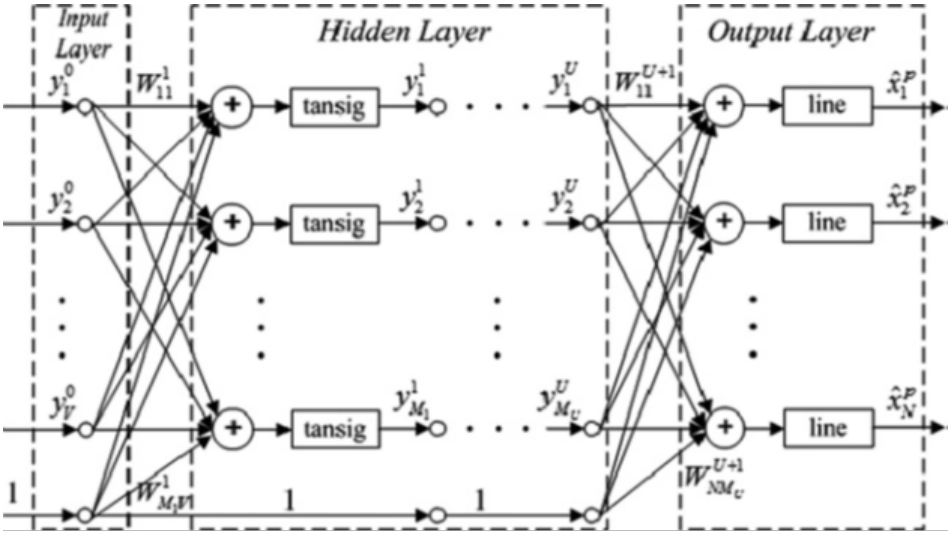
Στην εργασία των Soule et al. [28] περιγράφονται δύο επιπλέον αναλυτικές μέθοδοι που αξιοποιούν μετρήσεις πινάκων κίνησης σε περίοδο μιας ημέρας. Η πρώτη μέθοδος αποσκοπεί στην ελάττωση της διάστασης του προβλήματος καθιστώντας το καλά ορισμένο και επιτρέποντας την ακριβή επίλυση του συστήματος. Πιο συγκεκριμένα, επιστρατεύεται η τεχνική της Ανάλυσης Κύριων Συνιστωσών - Principal Components Analysis (PCA) για τον εντοπισμό των k πιο σημαντικών (κύριων) συνιστωσών που δύναται να περιγράψουν με ελάχιστο σφάλμα όλους τους πίνακες κίνησης που λαμβάνονται κατά το διάστημα μέτρησης. Η διάσταση k είναι πολύ μικρότερη από την αρχική διάσταση του πίνακα x και άρα το σύστημα του προβλήματος καθίσταται καλά ορισμένο. Για την εκτίμηση πίνακα κίνησης από τα φορτία ζεύξης y επιλύεται το σύστημα ούτως ώστε να βρεθεί το διάνυσμα του χώρου που ορίζουν οι κύριες συνιστώσες και το οποίο ανακατασκευάζει τον x με βάση την PCA αποσύνθεσή του. Η δεύτερη μέθοδος, ονομάζεται Kalman Filtering και χρησιμοποιεί χωρικά μοντέλα κατάστασης από την θεωρία δυναμικών γραμμικών συστημάτων για την παρακολούθηση της εξέλιξης του συστήματος.

Όλες οι μέθοδοι που αναφέρθηκαν, βασίζονται σε στατιστικές και αναλυτικές μεθόδους για την επίλυση του προβλήματος της εκτίμησης πίνακα κίνησης. Στα σημερινά δίκτυα όμως, τόσο η εσωτερική δομή τους όσο και η σχέσεις μεταξύ των ροών κίνησης που διέρχονται από αυτά εμφανίζουν σημαντική πολυπλοκότητα. Το γεγονός αυτό οδηγεί τα OD Flows στο να μην είναι ανεξάρτητα, να μην ακολουθούν συγκεκριμένη «κανονική» δομή και το μέγεθος του πίνακα κίνησης να καθυστερεί σημαντικά τον υπολογισμό της εκτίμησής του καθώς παράγεται μέσα από υπολογιστικά ακριβούς μαθηματικούς με-

τασχηματισμούς. Καθίσταται λοιπόν απαραίτητη η ανακάλυψη νέων μεθόδων που δύνανται να μοντελοποιήσουν επιτυχώς τις πολύπλοκες στατιστικές σχέσεις μεταξύ των ροών κίνησης αλλά και να παράγουν γρήγορα την εκτίμηση του πίνακα κίνησης.

Η πρόσφατη εκρηκτική ανάπτυξη των νευρωνικών δικτύων [29] οδήγησε στην κατασκευή μιας οδηγούμενης από δεδομένα νέας οικογένειας μοντέλων Μηχανικής Μάθησης, η οποία για είσοδο παλαιότερα χρονικά στιγμιότυπα των πινάκων κίνησης ανιχνεύει τις χωρικές και χρονικές συσχετίσεις των ροών και προβαίνει αποδοτικά σε εκτίμηση του πίνακα κίνησης με βάση τα φορτία ζεύξεων y.

Στην εργασία των Jiang et al. [30] προτείνεται για πρώτη φορά μοντέλο που ενσωματώνει νευρωνικά δίκτυα για την εκτίμηση του πίνακα κίνησης. Η μέθοδος βασίζεται σε ένα εμπροσθοδιάδοτο (feed-forward) δίκτυο με είσοδο τα φορτία ζεύξεων y για την εκτίμηση του στοιβαγμένου πίνακα σε διάνυσμα X. Για την εκμετάλλευση της χωρικής-χρονικής συσχέτισης των πινάκων κίνησης χρησιμοποιείται ένα στάδιο προ επεξεργασίας της εισόδου όπου λαμβάνονται οι k προηγούμενες τιμές του y για μια χρονική στιγμή t, συνδυάζονται και παράγεται η τελική είσοδος $\hat{y(t)}$. Η είσοδος διαδίδεται στο feed-forward δίκτυο (Σχήμα 2.2), παράγεται το διάνυσμα $\hat{X}$ και υπολογίζεται το σφάλμα μέσω της σχέσης $y = AX$, όπου A ο πίνακας δρομολόγησης. Το σφάλμα χρησιμοποιείται για την ενημέρωση των παραμέτρων του μοντέλου με τον αλγόριθμο Backpropagation. Η εκτίμηση του πίνακα $\hat{X}$ σε περιβάλλον ελέγχου δεν ικανοποιεί απαραίτητα την σχέση $y = A\hat{X}$ και την θετικότητα των στοιχείων του $\hat{X}$. Η συμμόρφωση με τους περιορισμούς αυτούς παρέχεται ενσωματώνοντας ως μετα-επεξεργαστικό στάδιο τον αλγόριθμο Iterative Proportional Fitting (IPFP). Το μοντέλο με όνομα BPTME συγκρίνεται πάνω στο Abilene dataset με το εμπορικό μοντέλο Tomogravity, το οποίο υπερκερνά σε όλες τις προσδιοριζόμενες μετρικές. Οι συγγραφείς καταλήγουν πως το BPTME είναι πολύ πιο γρήγορο στην εκτίμηση του πίνακα κίνησης λόγω της απλοποιημένης μαθηματικής μοντελοποίησής του, της ανθεκτικότητάς του στον θόρυβο και της μεγαλύτερης ακρίβειας στις προβλέψεις του.



Σχήμα 2.2: Αρχιτεκτονική Feed Forward Νευρωνικού Δικτύου. Πηγή [30]

Οι Omidvar et al. [31] πρότειναν την ενσωμάτωση γενετικών αλγορίθμων με στόχο την αποδοτική ενημέρωση των παραμέτρων ενός νευρωνικού δικτύου που εκτιμά τον πίνακα κίνησης. Πιο συγκεκριμένα, χρησιμοποιείται ένα μη γραμμικό autoregressive feed-forward νευρωνικό δίκτυο, το οποίο μοντελοποιεί τον πίνακα κίνησης ως χρονοσειρά όπου κάθε πίνακας X εξαρτάται από ένα παράθυρο προηγούμενων χρονικά πινάκων και μια εξωγενή είσοδο, δηλαδή τις μετρήσεις ζεύξης y. Για την εκπαίδευση του μοντέλου χρησιμοποιείται η μέθοδος Levenberg-Marquardt ως συνάρτηση κόστους. Για την ενημέρωση των βαρών και των μεροληψιών του δικτύου χρησιμοποιείται ένας γενετικός αλγόριθμος ο οποίος αρχίζοντας από ένα τυχαίο σύνολο πιθανών λύσεων και εφαρμόζοντας επαναληπτικά τις ενέργειες «επιλογή», «διασταύρωση» και «μετάλλαξη» επιλέγει ως βέλτιστη αυτή που ελαχιστοποιεί την συνάρτηση κόστους που έχει οριστεί. Στο στάδιο της «επιλογής» προσδιορίζεται ένας αριθμός ικανοποιητικών λύσεων από τον συνολικό χώρο παραμέτρων ως προς την ελαχιστοποίηση της συνάρτησης κόστους που επρόκειτο να συνδυαστούν στο στάδιο της «διασταύρωσης». Με την «μετάλλαξη» ενισχύεται η εξερεύνηση του χώρου λύσεων καθώς κάθε παράμετρος έχει μικρή πιθανότητα να μεταβληθεί σε μια τυχαία τιμή.

Μια εναλλακτική μέθοδος που χρησιμοποιεί ένα Συνάρτηση Ακτινικής Βάσης - Radial Basis Function (RBF) νευρωνικό δίκτυο για την εκτίμηση του πίνακα κίνησης αναπτύσσεται από τους Jiang et al. [32]. Το δίκτυο αυτό αποτελείται από τρία επίπεδα, το στρώμα εισόδου , το κρυφό στρώμα και το στρώμα εξόδου και δέχεται ως είσοδο τα φορτία ζεύξεων y παράγοντας την εκτίμηση για τον πίνακα κίνησης X. Αρχικά εφαρμόζεται ένα προ-επεξεργαστικό στάδιο στα διανύσματα εισόδου για κανονικοποίηση και στη συνέχεια η είσοδος τροφοδοτείται στους νευρώνες του κρυμμένου επιπέδου. Κάθε νευρώνας αποτελεί μια συνάρτηση ακτινικής βάσης η τιμή της οποίας εξαρτάται από την απόσταση της εισόδου του νευρώνα από ένα κέντρο/βάρος. Στη συγκεκριμένη υλοποίη-

ση η συνάρτηση που επιλέγεται είναι η Gaussian και η έξοδος του νευρώνα οδηγείται στο στρώμα εξόδου, όπου συνδυάζεται με την έξοδο των υπόλοιπων νευρώνων του κρυφού στρώματος για την παραγωγή της εκτίμησης του πίνακα X. Η εκτίμηση αυτή μπορεί να μην συμμορφώνεται με την συνθήκη $y = AX$ και το ότι οι τιμές του X πρέπει να είναι θετικές. Για τον καθορισμό του τελικού αποτελέσματος επιστρατεύεται ο αλγόριθμος Iterative Proportional Fitting (IPFP).

Στο [33] οι συγγραφείς κατασκεύασαν μέθοδο που βασίζεται στην ανάλυση συχνότητας και σε νευρωνικό δίκτυο με στόχο την ανακατασκευή των πινάκων κίνησης. Πιο συγκεκριμένα, κάθε OD flow του πίνακα κίνησης ερμηνεύεται ως χρονοσειρά, κανονικοποιείται για κάθε χρονική στιγμή με βάση ένα παράθυρο q επόμενων τιμών, και υφίσταται ανάλυση συχνότητας με την χρήση wavelet μετασχηματισμών. Στο πεδίο τις συχνότητας διαχωρίζονται οι χαμηλές από τις υψηλές συχνότητες και ανακατασκευάζονται τα αντίστοιχα χρονικά σήματα $z_{low}(t)$, $z_{high}(t)$ με τον αντίστροφο wavelet μετασχηματισμό. Στη συνέχεια χρησιμοποιείται ένα autoregressive μοντέλο για την αναλυτική μοντελοποίηση του $z_{low}(t)$ το οποίο εκφράζει την αλλαγή της τάσης στην αντίστοιχη ροή. Το $z_{high}(t)$ (που εκφράζει τη διακύμανση) χρησιμοποιείται για την εκπαίδευση ενός feed-forward νευρωνικού δικτύου με Backpropagation όπου δέχεται ως είσοδο όλα τα υψηλής συχνότητας OD flows για τις χρονικές στιγμές $t-1$ και $t-2$ καθώς και τα φορτία ζεύξης $y(t)$ για την κατασκευή του $z_{high}(t)$. Το τελικό αποτέλεσμα παράγεται από τον συνδυασμό των ανακατασκευασμένων $z_{low}(t)$, $z_{high}(t)$ κατόπιν αφαίρεσης της αρχικής κανονικοποίησης και συμμόρφωσης με τους προσδιοριζόμενους περιορισμούς. Για την πειραματική αξιολόγηση του μοντέλου χρησιμοποιείται το Abilene σύνολο δεδομένων και δοκιμάζονται τέσσερις wavelet βασικές συναρτήσεις, με την Haar να παρουσιάζει τις καλύτερες επιδόσεις.

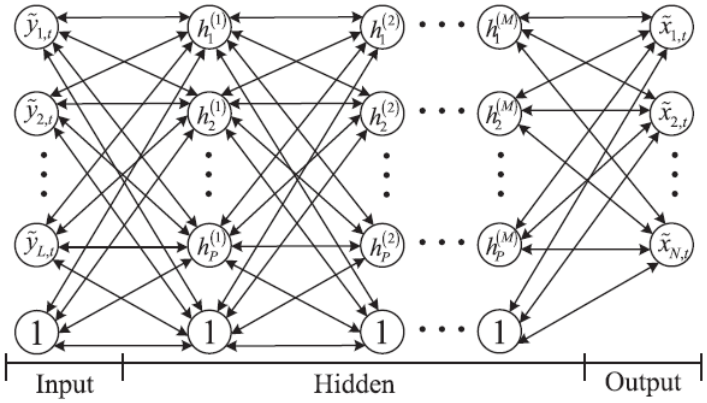
Στην προσπάθεια αναζήτησης περισσότερο αποτελεσματικών τρόπων για την ενημέρωση των παραμέτρων του νευρωνικού δικτύου οι Hussain et al. [34] προτείνουν τη χρήση ενός feed-forward νευρωνικού δικτύου του οποίου οι παράμετροι ενημερώνονται με την μέθοδο Levenberg – Marquardt με στόχο την εκτίμηση του πίνακα κίνησης. Ο αλγόριθμος αυτός συνδυάζει τις μεθόδους Gradient Descent και Gauss Newton για την ενημέρωση των βαρών του νευρωνικού, εναλλάσσοντας μεταξύ των δύο μεθόδων κατά την εκπαίδευση. Η Gradient Descent ελαχιστοποιεί τη συνάρτηση κόστους μετακινούμενη στην αντίθετη κατεύθυνση από την παράγωγο αυτής, γεγονός που μπορεί να οδηγήσει σε πιο «απότομες» ενημερώσεις. Η Gauss Newton ελαχιστοποιεί το άθροισμα των τετραγωνικών σφαλμάτων γραμμικοποιώντας το πρόβλημα γύρω από την τρέχουσα εκτίμηση αλλά απαιτεί ικανοποιητική αρχική πρόβλεψη. Η μέθοδος Levenberg – Marquardt ενημερώνει τις παραμέτρους με Gradient Descent αν το σφάλμα είναι υψηλό και αλλάζει δυναμικά σε Gauss Newton όταν το σφάλμα μειώνεται για εξασφάλιση πιο ομαλής σύγκλισης. Η μέθοδος Levenberg – Marquardt χρησιμοποιείται και στο [31] αλλά ως συνάρτηση κόστους και όχι για την απευθείας ενημέρωση των βαρών (η ενημέρωση γίνεται με γενετικό αλγόριθμο) όπως γίνεται στο [34].

Στη συνέχεια τίθεται το ερώτημα σχετικά με το αν δύναται να κωδικοποιηθεί στο μοντέλο μάθησης πληροφορία που αφορά τη δομή του δικτύου ούτως ώστε να μην είναι

απαραίτητη η έμμεση εξαγωγή της από τα δεδομένα, διευκολύνοντας έτσι την ανίχνευση των σχέσεων μεταξύ των OD Flows. Στο πλαίσιο αυτό, οι Zhou et al. [35] επεκτείνουν την είσοδο του νευρωνικού δικτύου με πληροφορίες σχετικές με την δρομολόγηση, ενσωματώνοντας παράλληλα και τον αλγόριθμο Expectation Maximization (EM). Ειδικότερα υπολογίζεται ο Moore-Penrose αντίστροφος A^+ του πίνακα δρομολόγησης A με, την μεταξύ άλλων, Singular Value Decomposition μέθοδο. Ο πίνακας αυτός πολλαπλασιάζεται με τις μετρήσεις ζεύξεων y συμπεριλαμβάνοντας έτσι πληροφορία για την δρομολόγηση στην επεκτεταμένη είσοδο. Τα ζεύγη A^+y, X δίνονται ως δείγματα εκπαίδευσης σε feed forward νευρωνικό δίκτυο το οποίο καλείται να εκτιμήσει τον πίνακα κίνησης χωρίς να χρειάζεται να μοντελοποιήσει και τον τρόπο δρομολόγησης, γεγονός που βελτιώνει αισθητά την ακρίβεια στην εκτίμηση. Η έξοδος του μοντέλου δεν συμμορφώνεται με τις σχέσεις $y = AX, X \geq 0$ και άρα υποβάλλεται σε ένα μετα-επεξεργαστικό στάδιο όπου οι αρνητικές τιμές αντικαθίστανται από την ελάχιστη θετική τιμή της αντίστοιχης ροής και στη συνέχεια εφαρμόζεται ο EM αλγόριθμος (επαναληπτικός αλγόριθμος για επίλυση ασταθών εξισώσεων) για τον καθορισμό της τελικής εκτίμησης.

Όλες οι αρχιτεκτονικές νευρωνικών δικτύων που έχουν αναφερθεί μέχρι στιγμής απαρτίζονται από ρηχά δίκτυα με λίγα επίπεδα και νευρώνες γεγονός που περιορίζει αισθητά την αναπαραστατική ικανότητα των μοντέλων μάθησης. Η κατασκευή βαθύτερων αρχιτεκτονικών απαιτεί σημαντικά αυξημένη υπολογιστική ισχύ, την οποία δεν μπορεί να προσφέρει μια «κλασσική» μονάδα κεντρικής επεξεργασίας. Η εμφάνιση και η διαρκής εξέλιξη των επιταχυντών υλικού όπως η Μονάδα Επεξεργασίας Γραφικών (Graphical Processing Unit - GPU) επιτρέπουν την δημιουργία νευρωνικών δικτύων με δεκάδες επίπεδα και εκατομμύρια παραμέτρους, με την κατηγορία των μοντέλων αυτών να ονομάζεται βαθιά μηχανική μάθηση.

Από τους πρώτους που επιχείρησαν τη χρήση μοντέλου βαθιάς μηχανικής μάθησης στον πεδίο της εκτίμησης πίνακα κίνησης ήταν οι Nie et al. [36]. Στην εργασία τους προτείνεται μια μέθοδος που ενσωματώνει Δίκτυα Βαθιάς Πίστης (Deep Belief Networks - DBN) (Σχήμα 2.3) για την πρόβλεψη και την εκτίμηση του πίνακα κίνησης. Ένα DBN αποτελείται από πολλά διαδοχικά Restricted Boltzmann Machines (RBM) επίπεδα. Το RBM είναι ένας μη κατευθυνόμενος γράφος δύο επιπέδων, του ορατού και του κρυμμένου, όπου κάθε νευρώνας του ορατού συνδέεται με όλους τους κρυμμένους και κάθε κρυμμένος νευρώνας με όλους τους ορατούς. Κάθε νευρώνας μοντελοποιεί μια στοχαστική διαδικασία και ακολουθεί την Gaussian ή την Bernoulli κατανομή. Η από κοινού κατανομή των δύο επιπέδων βασίζεται στη συνάρτηση ενέργειάς τους και μέσω αυτής μπορούν να βρεθούν σχέσεις για τις πιθανότητες που ακολουθεί κάθε νευρώνας ως συνάρτηση βαρών και μεροληψιών. Για την πρόβλεψη του πίνακα κίνησης, το σύνολο δεδομένων χωρίζεται σε δείγματα μέσω ενός κυλιόμενου παραθύρου που αποτελεί την είσοδο του DBN με τον επόμενο χρονικά πίνακα να είναι η ζητούμενη έξοδος. Για την εκτίμηση δίνονται ως είσοδοι τα φορτία ζεύξης y και λαμβάνεται ως έξοδος ο ανακατασκευασμένος πίνακας X κατόπιν εφαρμογής του IPFP αλγόριθμου στην έξοδο του DBN. Το μοντέλο αξιολογείται στα Abilene και Geant σύνολα δεδομένων, και παρουσιάζει χαμηλότερη μεροληψία και μικρότερη διακύμανση απότι η PCA μέθοδος [28] με την οποία συγκρίνεται.



Σχήμα 2.3: Αρχιτεκτονική Δικτύου Βαθιάς Πίστης. Πηγή [36]

Σε επόμενη εργασία οι Nie et al. [37] εξερευνούν την αποτελεσματικότητα των Δικτύων Βαθιάς Πίστης σε δίκτυα εκτός των IP Backbone που έχουν μελετηθεί εκτενώς στην βιβλιογραφία. Ειδικότερα εφαρμόζουν DBN για την πρόβλεψη και την εκτίμηση πινάκων κίνησης σε δίκτυο που αφορά κέντρα δεδομένων. Η κίνηση στα κέντρα δεδομένων παρουσιάζει περισσότερες διακυμάνσεις σε σχέση με την κίνηση που παρατηρείται σε ISP δίκτυα. Για την πρόβλεψη, τα δείγματα κατασκευάζονται από ένα κυλιόμενο παράθυρο με την αμέσως επόμενη χρονική στιγμή να αποτελεί την ζητούμενη έξοδο. Το DBN χρησιμοποιείται για την εκμάθηση των στατιστικών ιδιοτήτων μιας ροής κίνησης, με την έξοδό του να τροφοδοτεί ένα μοντέλο λογιστικής παλινδρόμησης για την παραγωγή της τελικής πρόβλεψης. Η εκπαίδευση γίνεται με την προσέγγιση της κλίσης της αρνητικής λογαριθμικής πιθανοφάνειας που έχει οριστεί ως στόχος εκπαίδευσης. Για την εκτίμηση, δίνονται ως είσοδος στο DBN τα φορτία ζεύξης y και παράγεται η τελική εκτίμηση X ύστερα από εφαρμογή του IPFP αλγόριθμου. Το μοντέλο αξιολογείται σε πραγματικά δίκτυα κέντρων δεδομένων και εμφανίζει σημαντική βελτίωση σε σχέση με τις PCA, TomoGravity και SRMF μεθόδους με τις οποίες συγκρίνεται. Πιο συγκεκριμένα παρουσιάζει χαμηλότερη μεροληψία με μικρή διακύμανση αλλά και ικανοποιητική προβλεπτική ικανότητα για βραχυπρόθεσμες προβλέψεις.

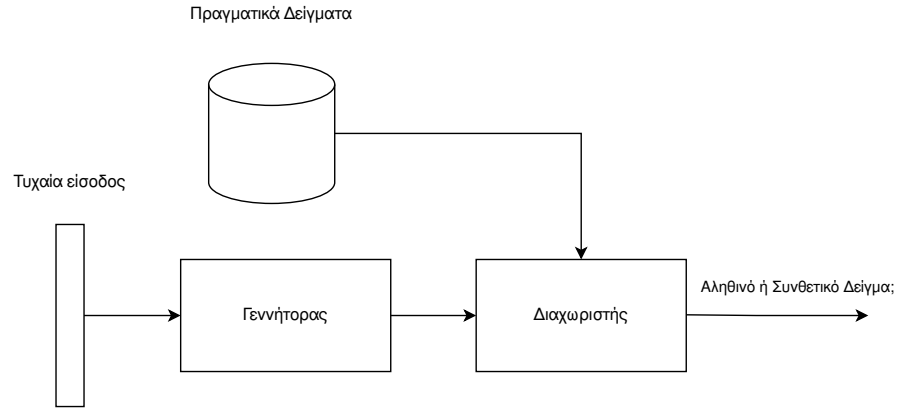
Η ραγδαία ανάπτυξη των τεχνικών για την ανάλυση εικόνων με μεθόδους βαθιάς μηχανικής μάθησης οδήγησε στη μοντελοποίηση του πίνακα κίνησης ως μια δισδιάστατη οντότητα, κατά αναλογία με τις εικόνες, ενσωματώνοντας εργαλεία όπως τα συνελικτικά επίπεδα [38]. Οι Emami et al. [39] κατασκευάζουν για πρώτη φορά συνελικτικό νευρωνικό δίκτυο και το συνδυάζουν με ένα Graph Embedding στάδιο για την ενσωμάτωση της τοπολογίας του δικτύου στην είσοδο του μοντέλου. Πιο συγκεκριμένα, λαμβάνονται υπόψιν όλοι οι κόμβοι της τοπολογίας και κατασκευάζεται ο adjacency πίνακας ο οποίος έχει στη θέση $[i, j]$ τιμή 0 αν οι κόμβοι i, j δεν συνδέονται απευθείας, ενώ έχει το αντίστοιχο φορτίο ζεύξης αν οι κόμβοι είναι γείτονες. Με αυτόν τον τρόπο κωδικοποιείται πέρα από την είσοδο y και πληροφορία σχετική με την τοπολογία του δικτύου και άρα το νευρωνικό δεν χρειάζεται να «μαθαίνει» και την πληροφορία δρομολόγησης για την

εκτίμηση. Ο πίνακας αυτός παράγεται για κάθε στιγμιότυπο της εισόδου y και τροφοδοτείται στο συνελικτικό νευρωνικό δίκτυο για την παραγωγή της εκτίμησης του πίνακα X. Το νευρωνικό δίκτυο αποτελείται από το συνελικτικό μέρος που είναι μια αλληλουχία από Conv2D, συνάρτηση ενεργοποίησης και max-pool επιπέδων για την εκμάθηση των χωρικών σχέσεων της εισόδου και από το πλήρως συνδεδεμένο σκέλος για την παραγωγή του τελικού στοιβαγμένου διανύσματος X.

Μια σημαντική κατηγορία μοντέλων μηχανικής μάθησης που αναπτύσσεται θεαματικά και έχει προσφέρει επιδόσεις αιχμής στην παραγωγή συνθετικών αντικειμένων ονομάζεται Generative Learning [40]. Σε αυτό το περιβάλλον μάθησης, το μοντέλο ανιχνεύει ιδιότητες του χώρου δεδομένων με τα οποία εκπαιδεύεται και μπορεί να παράξει, για τυχαία θορυβώδη είσοδο, συνθετικά δείγματα, ιδανικά μη διαχωρίσιμα από τα πραγματικά δεδομένα. Τα μοντέλα αυτά συνιστούν την σύγχρονη τάση στην εκτίμηση του πίνακα κίνησης των δικτύων με αποτελέσματα που υπερκερνούν σχεδόν όλες τις προηγούμενες τεχνικές.

Η πρώτη χρήση Generative Learning μοντέλου μάθησης εντοπίζεται στην εργασία των Xu et al. [41] όπου προτείνονται δύο μέθοδοι για την εκτίμηση του πίνακα κίνησης, η Proj-D και η GAN-D. Η διαπίστωση ότι δεν υπάρχουν πολλά διαθέσιμα σύνολα δεδομένων με πλήρεις μετρήσεις παλαιότερων πινάκων ενός δικτύου οδηγεί στην κατασκευή μοντέλων που εκτιμούν τον πίνακα κίνησης με βάση τα φορτία ζεύξης και τον περιορισμό που εισάγει η κατανομή που ακολουθούν οι ροές κίνησης (η οποία μπορεί εύκολα να μετρηθεί). Η Proj-D είναι ένας επαναληπτικός κυκλικός αλγόριθμος προβολής των ροών κίνησης πάνω στις σχέσεις $y_i = a_i x$ για κάθε στοιχείο του διανύσματος y. Ανά τακτά χρονικά διαστήματα, το σημείο x (το οποίο ενημερώνεται σε κάθε επανάληψη για να συμμορφώνεται καλύτερα με όλες τις ζητούμενες σχέσεις) αναπροσαρμόζεται έτσι ώστε να προκύπτει από την Αθροιστική Συνάρτηση Κατανομής (Cumulative Distribution Function) που ακολουθούν οι ροές κίνησης του δικτύου.

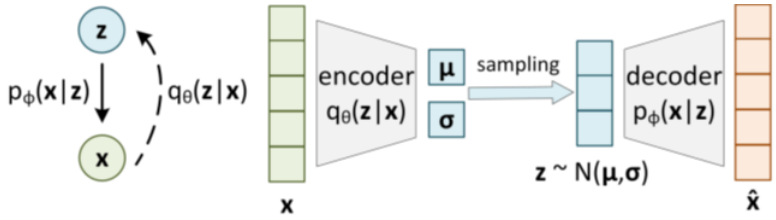
Σε περίπτωση που υπάρχουν παλαιότερες πλήρεις μετρήσεις των πινάκων κίνησης, αναπτύσσεται το μοντέλο GAN-D (Σχήμα 2.4) το οποίο εκπαιδεύεται πάνω σε αυτές και δύναται να πραγματοποιήσει εκτιμήσεις. Πιο συγκεκριμένα το GAN-D αποτελείται από δύο νευρωνικά δίκτυα, τον γεννήτορα και τον διαχωριστή, και μοντελοποιείται ως παίγνιο ανάμεσα στις δύο οντότητες. Στόχος του γεννήτορα (που δέχεται ως είσοδο ένα λανθάνον διάνυσμα) είναι να παράγει συνθετικούς πίνακες κίνησης που μοιάζουν έντονα με τους πραγματικούς έτσι ώστε ο διαχωριστής να μην μπορεί να διαπιστώσει αν ένας πίνακας είναι αληθινός ή συνθετικός. Για την εκτίμηση του πίνακα ο εκπαιδευμένος γεννήτορας, αρχίζοντας από ένα αρχικό διάνυσμα εισόδου, παράγει ένα συνθετικό πίνακα X με στόχο να ελαχιστοποιήσει το μέσο τετραγωνικό σφάλμα των y, AX με το διάνυσμα εισόδου να ενημερώνεται επαναληπτικά προς αυτήν την κατεύθυνση. Οι δύο μέθοδοι αξιολογούνται πειραματικά στα Abilene και GEANT σύνολα δεδομένων, είτε μόνο με γνωστή κατανομή των ροών κίνησης είτε με διαθέσιμους παλαιότερους πίνακες, με το Proj-D να αποδίδει καλύτερα στην πρώτη περίπτωση ενώ το GAN-D παρουσιάζει αυξημένη ικανότητα εκτίμησης στο δεύτερο περιβάλλον ελέγχου.



Σχήμα 2.4: Βασική Αρχιτεκτονική GAN.

Στην εργασία τους οι Kakkavas et al. [3] αναπτύσσουν έναν Παραλλαγμένο Αυτοκωδικοποιητή (Variational Autoencoder (VAE)) για την επίλυση της σύνθεσης και εκτίμησης πινάκων κίνησης. Το VAE αποτελείται από δύο νευρωνικά δίκτυα, τον κωδικοποιητή και τον αποκωδικοποιητή, με τον κωδικοποιητή να παράγει μια κατανομή στον λανθάνοντα χώρο και όχι μεμονωμένα δείγματα. Το VAE (Σχήμα 2.5) εκπαιδεύεται πάνω σε παλαιότερους χρονικά πίνακες τους οποίους κωδικοποιεί αντιστοιχίζοντάς τους σε μια Κανονική κατανομή του λανθάνοντος χώρου από την οποία επιλέγεται, για κάθε πίνακα, ένα τυχαίο δείγμα για ανακατασκευή μέσω του αποκωδικοποιητή. Στόχος της εκπαίδευσης είναι να μειωθεί το σφάλμα ανακατασκευής αλλά και η λανθάνουσα κατανομή να μοιάζει με την Κανονική. Στη συνέχεια διατηρείται μόνο ο αποκωδικοποιητής ο οποίος συνθέτει πίνακες κίνησης για τυχαία δείγματα του λανθάνοντος χώρου αλλά και εκτιμά πίνακες βασισμένους στις μετρήσεις ζεύξεων y ελαχιστοποιώντας την αντικειμενική συνάρτηση $MSE(y, AD(z))$ (όπου $D(z)$ η έξοδος του αποκωδικοποιητή) μέσω επαναληπτικού βελτιστοποιητή για το λανθάνον διάνυσμα z. Το μοντέλο αξιολογείται στο Abilene σύνολο δεδομένων εξετάζοντας δύο παραλλαγές του στόχου ελαχιστοποίησης και μια ενιαία διαδικασία εκπαίδευσης που εκτιμά πίνακες κατά την διάρκεια εκπαίδευσης του VAE. Τα πειράματα δείχνουν ότι ο ενοποιημένος τρόπος εκπαίδευσης παρουσιάζει το χαμηλότερο σφάλμα εκτίμησης για συνάρτηση ελαχιστοποίησης:

$$\arg\min_{z} \left[\|y - A \cdot D(z)\|_2^2 + c \cdot \|z\|_2^2 \right] \tag{2.7}$$



Σχήμα 2.5: Αρχιτεκτονική Variational Autoencoder. Πηγή [3]

Σε επόμενή τους εργασία, οι Kakkavas et al. [42] επεκτείνουν την αρχιτεκτονική που αναπτύσσεται στο [3] σχετικά με την χρήση VAE για την εκτίμηση πίνακα κίνησης, ενσωματώνοντας επίπεδα όπως το attention, self-attention [43] και ConvLSTM [44]. Αρχικά προτείνεται η εισαγωγή attention μηχανισμών ανάμεσα στα συνελικτικά επίπεδα του VAE έτσι ώστε να αναδεικνύονται τα σημαντικά χαρακτηριστικά της εισόδου (προσδίδοντάς τους ένα μεγάλο συντελεστή βάρους) και να περιορίζονται τα επαναλαμβανόμενα και μη σχετιζόμενα στοιχεία. Το attention εφαρμόζεται τόσο στον άξονα των καναλιών όσο και στις χωρικές διαστάσεις της εισόδου. Μια διαφορετική προσέγγιση στο ΕΠΚ είναι η ερμηνεία των πινάκων κίνησης ως χρονική ακολουθία όπου δίνεται ως είσοδος ένα παράθυρο k προηγούμενων χρονικά πινάκων από τον $X(t)$ καθώς και τα φορτία ζεύξεων $y(t)$ για να μπορέσει να εκτιμηθεί ο πίνακας. Στην μοντελοποίηση αυτή ταιριάζει η χρήση Recurrent Neural Networks (RNN) που ειδικεύονται στην εκμάθηση χαρακτηριστικών μιας ακολουθίας. Μια αποδοτική κατηγορία των RNN είναι τα Long Short-Term Memory (LSTM) τα οποία μαθαίνουν μεγάλης διάρκειας χαρακτηριστικά μέσω ενός μηχανισμού πυλών και κυττάρων μνήμης. Στην εργασία αντικαθίσταται ο πολλαπλασιασμός πινάκων εντός του LSTM κυττάρου με συνέλιξη γεγονός που επιτρέπει την αποτύπωση χωρικών πέρα από χρονικών σχέσεων. Τέλος μια ακόμα προσθήκη είναι η ενσωμάτωση self-attention μηχανισμών στην ConvLSTM εκδοχή, που εξερευνά την εξάρτηση που έχει κάθε στοιχείο της ακολουθίας με όλα τα υπόλοιπα αναδεικνύοντας τα σημαντικά χαρακτηριστικά. Τα μοντέλα αξιολογούνται στο Abilene και στο Geant σύνολο και εμφανίζουν σημαντική βελτίωση σε όλες τις μετρικές σε σχέση με το VAE, με το Self-Attention ConvLSTM να εμφανίζει την χαμηλότερη SRE μετρική μαθαίνοντας ικανοποιητικά τις χωρικές-χρονικές συσχετίσεις των TM.

Σε προβλήματα που σχετίζονται με την εκτίμηση του πίνακα κίνησης όπως η πρόβλεψη του επόμενου πίνακα σε μια χρονική ακολουθία πινάκων αλλά και η συμπλήρωση OD Flows που λείπουν από έναν πίνακα κίνησης, το Generative Learning παρέχει ικανοποιητικές λύσεις που μπορούν να εφαρμοστούν και στην εκτίμηση του πίνακα. Οι Sacco et al. [45] παρουσιάζουν μέθοδο που συνδυάζει ένα Κρυφό Μαρκοβιανό Μοντέλο (Hidden Markov Model (HMM)) με έναν Adversarial Autoencoder (AAE) για την συμπλήρωση και την πρόβλεψη πινάκων κίνησης. Το HMM μοντελοποιεί την χρονική ακολουθία πινάκων ως μια αλυσίδα Markov της οποίας η κατάσταση (δηλαδή ο πλήρης πίνακας) εξαρτάται μόνο από την προηγούμενη και από παλαιότερες εξωτερικές παρατηρήσεις. Οι παρατηρήσεις αυτές είναι η ίδια χρονοσειρά πινάκων, με τους πίνακες όμως να έχουν χω-

ρίς τιμή ορισμένες ροές κίνησης. Στόχος λοιπόν του HMM είναι με δεδομένη την σειρά ανολοκλήρωτων πινάκων και τον προηγούμενο ολοκληρωμένο πίνακα X_{t-1} να μπορεί να συμπληρώσει τον πίνακα X_t με βάση τον πίνακα με τις πιθανότητες μετάβασης και να προβλέψει τον επόμενο X_{t+1}. Για την συμπλήρωση του πίνακα εκπαιδεύεται το AAE δίκτυο το οποίο αποτελείται από έναν κωδικοποιητή, έναν αποκωδικοποιητή, έναν γεννήτορα και έναν διαχωριστή και στοχεύει τόσο στην μείωση του σφάλματος ανακατασκευής των δειγμάτων, όσο και στην αντιστοίχιση των δειγμάτων του λανθάνοντα χώρου με μια αρχική κατανομή, με τον διαχωριστή να αδυνατεί να αποφασίσει αν ένα δείγμα ακολουθεί την αρχική κατανομή ή όχι. Πιο συγκεκριμένα το AAE δέχεται ως είσοδο τους ημι-ολοκληρωμένους πίνακες και εξάγει τις συμπληρωμένες εκδοχές τους. Το μοντέλο αξιολογείται στα Abilene, Geant και Mawi σύνολα και παρουσιάζει ικανοποιητικά αποτελέσματα τόσο στην συμπλήρωση όσο και στην πρόβλεψη.

Τέλος, οι Yuan et al. [46] επιστρατεύουν ένα μοντέλο αιχμής για την παραγωγή συνθετικών δειγμάτων, το Denoising Diffusion Probabilistic Model (DDPM), με στόχο την εκτίμηση του πίνακα κίνησης. Το DDPM για κάθε δείγμα του συνόλου δεδομένων διαχέει επαναληπτικά μια μικρή ποσότητα θορύβου μέχρις ότου το δείγμα να γίνει πλήρως θόρυβος και να αντιστοιχιστεί με γνωστή κατανομή (forward pass). Στη συνέχεια αν ληφθεί δείγμα από αυτήν την κατανομή, το μοντέλο μαθαίνει να αναιρεί την επίδραση του θορύβου, παράγοντας έτσι ένα στιγμιότυπο του χώρου δεδομένων (backward pass). Στην αρχιτεκτονική που προτείνεται, εφαρμόζεται ένα προεπεξεργαστικό στάδιο για την μείωση της διάστασης των δεδομένων εισόδου και της κανονικοποίησής τους σε έναν λανθάνοντα χώρο, πραγματοποιείται το forward και backward pass του DDPM και λαμβάνεται ένας πίνακας στον λανθάνοντα χώρο. Ο πίνακας αυτός τροφοδοτεί ένα ανακατασκευαστικό δίκτυο όπου αναιρείται η μείωση διαστάσεων και αφού πολλαπλασιαστεί με τον πίνακα δρομολόγησης A προκύπτει η εκτίμηση $\hat{y}$ η οποία συγκρίνεται με τα πραγματικά φορτία ζεύξεων y για τον υπολογισμό του σφάλματος. Το πρόβλημα λοιπόν μοντελοποιείται ως ελαχιστοποίηση με βελτιστοποιητή κλίσης της $\left\| y - A\left(\hat{X}(z)\right)\right\|_2^2$ όπου z ένα διάνυσμα θορύβου που διαδίδεται με backward pass στο DDPM. Το μοντέλο αξιολογείται στα Abilene και GEANT και παρουσιάζει αισθητά καλύτερες επιδόσεις σε σχέση με τα VAE και GAN.

Κεφάλαιο 3

Αντιστρέψιμα Νευρωνικά Δίκτυα

3.1 Βασική Ιδέα

Τα αντίστροφα προβλήματα καλύπτουν ένα σημαντικό εύρος φυσικών αλλά και τεχνολογικών εφαρμογών. Τα προβλήματα αυτά σχετίζονται με τον υπολογισμό των παραμέτρων του συστήματος που ευθύνονται για την παραγωγή ενός συνόλου παρατηρήσιμων τιμών. Ένα τυπικό αντίστροφο πρόβλημα εκφράζεται μέσα από την σχέση:

$$y = f(x) + \epsilon \tag{3.1}$$

Στη σχέση αυτή, η παράμετρος x αποτελεί τα κρυφά χαρακτηριστικά του συστήματος τα οποία μέσω της ντετερμινιστικής συνάρτησης f μετασχηματίζονται (forward μετασχηματισμός) στα παρατηρήσιμα δεδομένα y. Η ποσότητα ϵ κωδικοποιεί τον θόρυβο του συστήματος και μπορεί να αγνοηθεί με ασφάλεια στις περισσότερες εφαρμογές. Στόχος λοιπόν του προβλήματος είναι η εύρεση του αντίστροφου μετασχηματισμού f^{-1} (backward μετασχηματισμός) και άρα η εξαγωγή της μεταβλητής $x = f^{-1}(y)$. Αν το σύστημα είναι καλά ορισμένο (στην περίπτωση του γραμμικού συστήματος εξισώσεων, ο πίνακας μετασχηματισμού f είναι πλήρους βαθμού) τότε το πρόβλημα επιδέχεται μοναδική λύση. Στην πλειονότητα όμως των εφαρμογών, το σύστημα είναι υπό-ορισμένο (ill-posed) γεγονός που σημαίνει ότι πολλαπλά στιγμιότυπα του μεγέθους x ικανοποιούν την σχέση (3.1).

Ποικίλες τεχνικές Μηχανικής Μάθησης έχουν επιστρατευθεί για την επίλυση των αντίστροφων προβλημάτων και μπορούν να ομαδοποιηθούν σε δύο κύριες κατηγορίες. Η πρώτη οικογένεια λύσεων βασίζεται στην κατασκευή ενός νευρωνικού δικτύου πολλαπλών επιπέδων το οποίο εκπαιδεύεται πάνω στα διαθέσιμα δεδομένα y και παράγει ως έξοδο τις ζητούμενες παραμέτρους x. Στη συνέχεια υπολογίζεται το σφάλμα της εξόδου του δικτύου με τις πραγματικές τιμές του x και το μοντέλο βελτιώνεται επαναληπτικά. Η προσέγγιση αυτή προσπαθεί να μάθει απευθείας τον αντίστροφο μετασχηματισμό του συστήματος και βασίζεται στην ύπαρξη διαθέσιμου συνόλου δεδομένων από πραγματικά

ζεύγη (x, y).

Η δεύτερη κατηγορία τεχνικών λειτουργεί σε Generative Learning [40] περιβάλλον εκπαίδευσης και χρησιμοποιεί μοντέλα όπως Variational Autoencoders (VAE) [47] και Generative Adversarial Networks (GAN) [48]. Στόχος των μοντέλων είναι η εκμάθηση των ιδιοτήτων του χώρου του προβλήματος με έμφαση στην παραγωγή πειστικών συνθετικών δειγμάτων για είσοδο ένα διάνυσμα θορύβου στον λανθάνοντα χώρο. Τα μοντέλα αυτά χρησιμοποιούνται ως αρχικές κατανομές παραγωγής δειγμάτων και μπορούν να ενσωματωθούν στην επίλυση ενός προβλήματος βελτιστοποίησης για την επιλογή του δείγματος x που ικανοποιεί καλύτερα την σχέση (3.1).

Και οι δύο οικογένειες λύσεων εμφανίζουν σημαντικούς περιορισμούς. Η πρώτη κατηγορία χαρακτηρίζεται από ένα εγγενές σφάλμα στον μετασχηματισμό των δεδομένων εισόδου y στην έξοδο x λόγω του ότι το x συνήθως είναι πολύ μεγαλύτερης διάστασης και άρα υπάρχει απώλεια πληροφορίας. Η δεύτερη κατηγορία δεν μπορεί να ανακατασκευάσει χωρίς σημαντικό σφάλμα τα δεδομένα με τα οποία εκπαιδεύεται (μαθαίνει τις ιδιότητες του χώρου και όχι απευθείας μετασχηματισμούς συγκεκριμένων δειγμάτων) και είναι ευαίσθητη σε σφάλματα αν το ζητούμενο x διαθέτει διαφορετικά στατιστικά χαρακτηριστικά από αυτά της κατανομής εκπαίδευσης.

Μια πρόσφατα αναπτυσσόμενη κατηγορία μοντέλων Μηχανικής Μάθησης ειδικά σχεδιασμένη για την επίλυση αντίστροφων προβλημάτων είναι τα Αντιστρέψιμα Νευρωνικά Δίκτυα (Invertible Neural Networks – INN) [49, 50, 51, 11]. Η κύρια χρήση των INN είναι η εκτίμηση πυκνότητας της κατανομής που ακολουθούν τα δείγματα ενός συνόλου δεδομένων.

Ο χώρος δεδομένων περιγράφεται από πολύπλοκες κατανομές άγνωστης δομής, και ένας τρόπος να μοντελοποιηθεί είναι μέσω της αντιστοίχισης των δειγμάτων του σε δείγματα μιας γνωστής, εύκολης στην μαθηματική αναπαράσταση κατανομής στον λανθάνοντα χώρο. Στόχος του INN είναι να βρεθεί ένας μετασχηματισμός $z = f(x)$ από τον χώρο δεδομένων X στον λανθάνοντα χώρο Z όπου η γνωστή κατανομή που ακολουθούν τα z δείγματα παραγοντοποιείται, δηλαδή οι συνιστώσες της είναι ανεξάρτητες. Αν ο μετασχηματισμός f είναι αντιστρέψιμος και η διάσταση του X είναι ίση με την διάσταση του Z τότε από τον τύπο της αλλαγής μεταβλητής ισχύει:

$$p_X(x) = p_Z(f(x)) \cdot \left| \det \left(\frac{\partial f(x)}{\partial x} \right) \right| \tag{3.2}$$

Ο πίνακας $\frac{\partial f(x)}{\partial x}$ ονομάζεται Ιακωβιανός (Jacobian) του μετασχηματισμού f και εφόσον το f είναι αντιστρέψιμο καθίσταται δυνατή η δειγματοληψία στο Z και η «σωστή» ανάκτηση του δείγματος στον χώρο X. Η κεντρική ιδέα των INN είναι χρήση μετασχηματισμών f που αντιστρέφονται εύκολα και συγχρόνως παρουσιάζουν παρακολουθήσιμη (tractable) ορίζουσα του Ιακωβιανού πίνακα. Ένας τύπος μετασχηματισμών που ικανοποιούν αυτές τις απαιτήσεις είναι συναρτήσεις f των οποίων ο Ιακωβιανός πίνακας είναι τριγωνικός και άρα η ορίζουσά του υπολογίζεται ως το γινόμενο των στοιχείων της διαγώνιου.

Όλα τα παραπάνω μας δείχνουν ότι, για γνωστή αρχική κατανομή p_Z (συνήθως Γκαουσιανή) του λανθάνοντος χώρου και γνωστό forward αντιστρέψιμο μετασχηματισμό f, το μοντέλο μάθησης INN κωδικοποιεί την αντιστοίχιση μέσω του f των δειγμάτων X σε δείγματα Z μαθαίνοντας παράλληλα την αντίστροφη διαδικασία εκμεταλλευόμενο την εκ κατασκευής αντιστρεψιμότητά του. Αξίζει επίσης να σημειωθεί πως δεδομένου ότι ο μετασχηματισμός f είναι ντετερμινιστικός και άρα και η αντίστροφη διαδικασία είναι ντετερμινιστική, το μοντέλο, κατόπιν σύγκλισης, μπορεί να ανακατασκευάσει πλήρως τα δείγματα εκπαίδευσης X και συγχρόνως να παράξει αληθοφανή συνθετικά δείγματα καθώς έχει μοντελοποιήσει σωστά την κατανομή του χώρου δεδομένων.

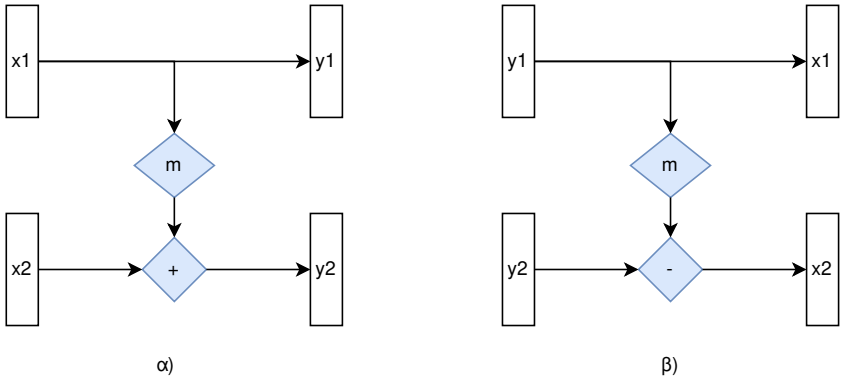
3.2 Κύριες Αρχιτεκτονικές INN

Η εργασία των Dinh et al. [49] παρουσιάζει για πρώτη φορά μοντέλο INN για την εκτίμηση πυκνότητας της κατανομής του χώρου δεδομένων. Βασιζόμενοι στον τύπο αλλαγής μεταβλητής (3.2), οι συγγραφείς επιλέγουν ως μετασχηματισμό f και βασικό λειτουργικό πυρήνα του μοντέλου μάθησης, μια αμφίδρομη συνάρτηση που ονομάζεται Συζευκτικό Στρώμα (Coupling Layer). Το Coupling Layer (Σχήμα 3.1) είναι προσθετικής μορφής όπου η είσοδος x χωρίζεται σε δύο τμήματα x_1, x_2 με διαστάσεις $d, D-d$ αντίστοιχα (υποθέτουμε ότι ο χώρος X έχει διάσταση R^D). Η έξοδος y παράγεται από την συνένωση των y_1, y_2 διαστάσεων $d, D-d$ που υπολογίζονται ως εξής:

$$y_1 = x_1 \tag{3.3}$$
$$y_2 = x_2 + m(x_1) \tag{3.4}$$

Η συνάρτηση m είναι μια αυθαίρετα πολύπλοκη (όχι απαραίτητα αντιστρέψιμη) συνάρτηση $d \to D-d$ που μοντελοποιείται από ένα νευρωνικό δίκτυο με ReLU [52] ενεργοποίηση. Εύκολα προκύπτει η είσοδος x από την έξοδο y με τον αντίστροφο μετασχηματισμό να είναι:

$$x_1 = y_1 \tag{3.5}$$
$$x_2 = y_2 - m(y_1) \tag{3.6}$$



Σχήμα 3.1: Συζευκτικό Στρώμα α) και το αντίστροφό β) του για το μοντέλο NICE

Ο Ιακωβιανός πίνακας του μετασχηματισμού ισούται με:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} I_d & 0 \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} \tag{3.7}$$

Η Ιακωβιανή ορίζουσα του μετασχηματισμού ισούται με $\det(\frac{\partial y_2}{\partial x_2}) = 1$ και η συνολική αρχιτεκτονική του δικτύου προκύπτει από την χρήση πολλών διαδοχικών Coupling layer, με τα τμήματα της εισόδου x_1, x_2 να εναλλάσσονται έτσι ώστε όλα τα στοιχεία της εισόδου να επηρεάζουν την τελική έξοδο. Η χρήση πολλών Coupling Layers συνιστά σύνθεση συναρτήσεων και άρα η τελική ορίζουσα του Ιακωβιανού πίνακα του δικτύου προκύπτει ως το γινόμενο των επιμέρους οριζουσών σε κάθε επίπεδο παραμένοντας tractable και στην περίπτωση αυτή ισούται με 1 (Διατηρεί τον Όγκο - Volume Preserving).

Γενικά είναι επιθυμητό να αυξηθεί το εύρος των χαρακτηριστικών που δύναται να μοντελοποίησει το δίκτυο και άρα εισάγεται ως τελικό στρώμα ένα επίπεδο επανακλιμάκωσης (rescaling) όπου κάθε στοιχείο της εξόδου πολλαπλασιάζεται με ένα παράγοντα S_{ii} (στην ουσία η έξοδος πολλαπλασιάζεται με ένα διαγώνιο πίνακα S διάστασης $D \times D$. Τέλος επιλέγεται μια εύκολα υπολογίσιμη αρχική κατανομή του λανθάνοντος χώρου p_Z διάστασης D (συνήθως Gaussian) της οποίας τα στοιχεία είναι ανεξάρτητα μεταξύ τους. Η εκπαίδευση του μοντέλου βασίζεται στον τύπο αλλαγής μεταβλητής (εφαρμόζοντας λογάριθμο και στα δύο μέλη) και εκφράζεται ως εκτιμητής μέγιστης πιθανοφάνειας με στόχο εκπαίδευσης:

$$\log(p_X(x)) = \log(p_Z(f(x))) + \log\left(\left|\det\left(\frac{\partial f(x)}{\partial x}\right)\right|\right) \tag{3.8}$$

Και αφού τα στοιχεία της p_Z είναι ανεξάρτητα μεταξύ τους και εφαρμόζεται και το rescaling επίπεδο προκύπτει ο τελικός στόχος εκπαίδευσης:

$$\log(p_X(x)) = \sum_{i=1}^{D} \left[\log(p_{Z_i}(f_i(x))) + \log\left(|S_{ii}|\right)\right] \tag{3.9}$$

Το κριτήριο αυτό ονομάζεται Μη Γραμμική Ανεξάρτητη Εκτίμηση Συνιστωσών (Nonlinear Independent Components Estimation (NICE)).

Στο [50] επεκτείνεται η αρχιτεκτονική που παρουσιάστηκε στο [49] εισάγοντας μια νέα INN εκδοχή με όνομα Real-valued Non-Volume Preserving (RealNVP) δίκτυο. Ακολουθώντας την συλλογιστική του NICE, το RealNVP αποτελείται από αμφίδρομα Coupling Layers των οποίων όμως η ορίζουσα του Ιακωβιανού δεν είναι απαραίτητα μοναδιαία και άρα το μοντέλο δεν διατηρεί τον όγκο. Η αλλαγή αυτή επιτρέπει την ικανοποιητική μοντελοποίηση πολύπλοκων δομών στον χώρο δεδομένων με το επίπεδο να παραμένει εύκολα αντιστρέψιμο και η Ιακωβιανή ορίζουσα tractable. Πιο συγκεκριμένα χρησιμοποιείται το εξής Coupling Layer (Σχήμα 3.2) για είσοδο $x = (x_1, x_2)$ και έξοδο $y = (y_1, y_2)$ με διαστάσεις $d, D - d$ αντίστοιχα:

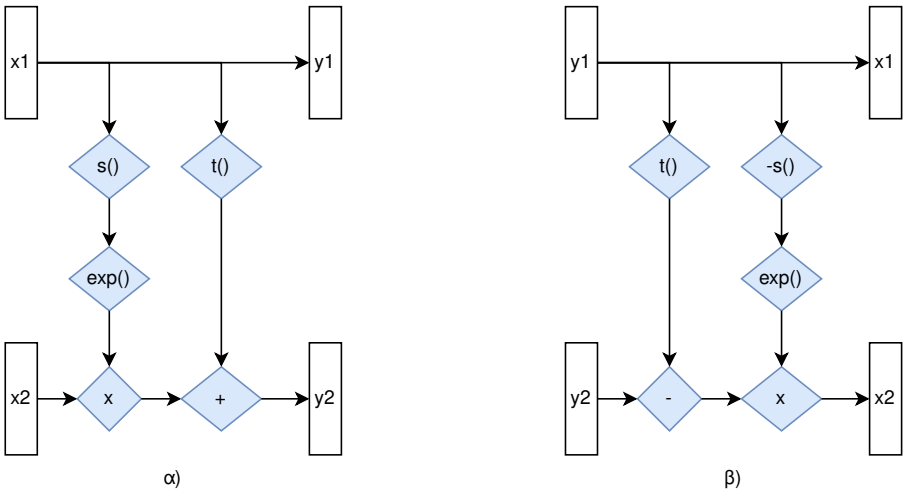
$$y_1 = x_1 \tag{3.10}$$

$$y_2 = x_2 \odot exp(s(x_1)) + t(x_1) \tag{3.11}$$

Τα s, t είναι αυθαίρετα πολύπλοκες, όχι απαραίτητα αντιστρέψιμες συναρτήσεις $d \rightarrow D - d$ και μπορούν να εκφραστούν από βαθιά νευρωνικά δίκτυα. Ο τελεστής $\odot$ είναι το γινόμενο Hadamard ή αλλιώς ο πολλαπλασιασμός στοιχείου με στοιχείο των δύο διανυσμάτων. Εύκολα προκύπτει και ο αντίστροφος μετασχηματισμός:

$$x_1 = y_1 \tag{3.12}$$

$$x_2 = (y_2 - t(y_1)) \odot exp\,(-s(y_1)) \tag{3.13}$$



Σχήμα 3.2: Συζευκτικό Στρώμα α) και το αντίστροφό β) του για το μοντέλο RealNVP

Ο Ιακωβιανός πίνακας του επιπέδου είναι:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} I_d & 0 \\ \frac{\partial y_2}{\partial x_1} & \text{diag}[\exp(s(x_1))] \end{bmatrix} \tag{3.14}$$

Η ορίζουσα του Ιακωβιανού πίνακα ισούται με $exp(\sum_j s(x_1)_j)$.

Ο χωρισμός της εισόδου σε x_1, x_2 μπορεί να γίνει ντετερμινιστικά με μάσκα συνέλιξης ή με μάσκα σκακιέρας στον άξονα των καναλιών και των χωρικών διαστάσεων αντίστοιχα (σε κάθε επίπεδο αλλάζουν οι μάσκες έτσι ώστε όλα τα στοιχεία της αρχικής εισόδου να επηρεάζουν την τελική έξοδο). Επιπλέον εισάγονται τεχνικές Ομαδοποιημένης Κανονικοποίησης (Batch Normalization) για να διευκολύνουν την διάδοση της εισόδου στην εκπαίδευση, επιτρέποντας αυξημένο αριθμό στοιβαγμένων Coupling layers και άρα βαθύτερη αρχιτεκτονική. Τέλος, για λόγους απόδοσης, ένας αριθμός στοιχείων της εξόδου λαμβάνει την τελική του τιμή σε κάποιο ενδιάμεσο επίπεδο και παύει να χρησιμοποιείται στον υπολογισμό του υπόλοιπου αποτελέσματος, μειώνοντας τον αριθμό των παραμέτρων και τη συνολική υπολογιστική πολυπλοκότητα.

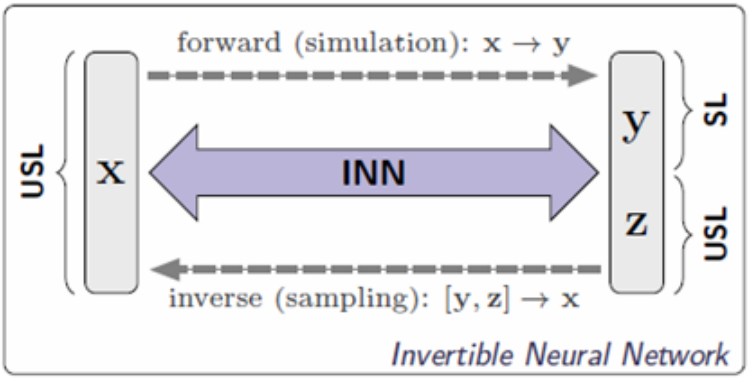
Δεδομένου ότι το RealNVP διαθέτει λιγότερο περιοριστικής μορφής Coupling Layer και υποστηρίζει βαθύτερες αρχιτεκτονικές σε σχέση με το NICE, εμφανίζει πολύ μεγαλύτερη αναπαραστατική ικανότητα και καθίσταται το πιο επιδραστικό μοντέλο INN στη βιβλιογραφία.

Το επόμενο βήμα στην εξέλιξη των INN μοντέλων πραγματοποιούν οι Kingma et al. στο [51]. Στην εργασία αυτή αναπτύσσεται το μοντέλο Glow το οποίο επεκτείνει τις INN αρχιτεκτονικές που παρουσιάστηκαν. Το Glow αποτελείται από μια σειρά από ροές (flows) με κάθε ροή να έχει ένα Actnorm επίπεδο, ένα επίπεδο αντιστρέψιμης 1×1 συνέλιξης και ένα αφινικό (affine) Coupling Layer. Το Actnorm layer αντικαθιστά το batch normalization που εφαρμόζεται στο RealNVP καθώς έχει αποδειχθεί πως ο θόρυβος που εισάγει είναι αντιστρόφως ανάλογος με το μέγεθος του batch και άρα για μεγάλες εικόνες όπου η μνήμη είναι περιορισμένη και το μέγεθος του batch μικρό, η υλοποίηση αυτή δεν είναι ικανοποιητική. Το Actnorm πραγματοποιεί έναν αφινικό μετασχηματισμό στην έξοδο των ενεργοποιήσεων χρησιμοποιώντας παραμέτρους για κλιμάκωση και μεροληψία ανά κανάλι της εισόδου. Η 1×1 συνέλιξη αντικαθιστά τις σταθερές εκ των προτέρων αντιμεταθέσεις της εισόδου που εισάγει το RealNVP με μια προς εκμάθηση αντιστρέψιμη πράξη. Χρησιμοποιώντας LU αποσύνθεση, ο πίνακας W που χρησιμοποιείται για την συνέλιξη μπορεί να υπολογιστεί σε $O(c)$ χρόνο, όπου c ο αριθμός των καναλιών (η είσοδος είναι μια εικόνα με χωρικές διαστάσεις $h \times w$ και c κανάλια). Τέλος ως Coupling Layer χρησιμοποιούνται παραλλαγές των αμφίδρομων συναρτήσεων του RealNVP. Το μοντέλο συγκρίνεται με τα NICE και RealNVP σε σύνολα εικόνων και παρουσιάζει καλύτερη απόδοση.

Όλα τα μοντέλα που αναφέρθηκαν μέχρι τώρα είναι γνωστά στην βιβλιογραφία ως Μοντέλα Κανονικοποιημένων Ροών (Normalizing Flow Models) και όχι ως INN παρόλου που εντάσσονται σε αυτήν την κατηγορία από την κατασκευή τους. Οι Ardizzone et al. [11] εισάγουν για πρώτη φορά τον όρο INN και επιστρατεύουν το μοντέλο αυτό για την επίλυση αντίστροφων προβλημάτων.

Όπως αναλύθηκε στο Κεφάλαιο 3.1, σε αντίθεση με τις κλασσικές τεχνικές νευρωνικών δικτύων που προσπαθούν να εκτιμήσουν το x απευθείας από το y, τα INN αποσκοπούν στην εκμάθηση του forward και γνωστού μετασχηματισμού $y = f(x)$ και εκμεταλ-

λευόμενα την αντιστρέψιμη φύση τους να μπορούν να προσομοιώσουν και τον αντίστροφο μετασχηματισμό $x = f^{-1}(y)$. Στη γενική περίπτωση ένα δείγμα του χώρου δεδομένων έχει διάσταση D ενώ οι μετρήσεις y έχουν διάσταση m όπου $m \leq D$. Όπως έχει αναφερθεί, τα INN για την λειτουργία τους απαιτούν οι διαστάσεις του χώρου δεδομένων και του λανθάνοντος χώρου να ταυτίζονται και άρα για να εξασφαλιστεί η συνθήκη και να αποφευχθεί η απώλεια πληροφορίας οι συγγραφείς προτείνουν την εξής ενέργεια (Σχήμα 3.3). Ο πραγματικός forward μετασχηματισμός έστω $y = s(x)$ προσομοιώνεται από το INN από την forward αντιστοίχιση $[y, z] = f(x)$ όπου z είναι μια λανθάνουσα μεταβλητή που ακολουθεί την Gaussian κατανομή $p_Z(z)$ διάστασης $D - m$ και y οι διαθέσιμες μετρήσεις του προβλήματος. Το $[y, z]$ συμβολίζει την συνένωση των δύο διανυσμάτων σε ένα κοινό διάνυσμα διάστασης D. Με αυτόν τον τρόπο καθίσταται δυνατή η μοντελοποίηση και της αντίστροφης διαδικασίας με είσοδο ένα τυχαίο δείγμα της κατανομής $p_Z(z)$ και την μέτρηση y. Με άλλα λόγια η αρχική κατανομή p_Z ωθείται μέσα από το νευρωνικό δίκτυο στον χώρο X με συνθήκη την μέτρηση y. Η τροποποίηση αυτή εξακολουθεί να διαθέτει tractable Ιακωβιανή ορίζουσα και να ικανοποιεί όλες τις INN προϋποθέσεις. Η προσέγγιση



Σχήμα 3.3: INN Αρχιτεκτονική. Πηγή [11]

Η αρχιτεκτονική που προτείνεται βασίζεται στο RealNVP με τα Coupling Layers να έχουν την εξής μορφή για είσοδο $x = [x_1, x_2]$ και έξοδο $y = [y_1, y_2]$:

$$y_1 = x_1 \odot exp\left(s_2(x_2)\right) + t_2(x_2) \tag{3.15}$$

$$y_2 = x_2 \odot exp\left(s_1(x_1)\right) + t_1(x_1) \tag{3.16}$$

Και ο αντίστροφος μετασχηματισμός:

$$x_2 = (y_2 - t_1(y_1)) \odot exp\left(-s_1(y_1)\right) \tag{3.17}$$

$$x_1 = (y_1 - t_2(y_2)) \odot exp\left(-s_2(y_2)\right) \tag{3.18}$$

Τα s_i, t_i είναι αυθαίρετα πολύπλοκα νευρωνικά δίκτυα με leaky ReLU [53] ενεργοποιήσεις.

Για την εκπαίδευση του μοντέλου χρησιμοποιείται μια Επιβλεπόμενη Συνάρτηση Απωλειών (Supervised Loss Function) L_y για να εξασφαλιστεί ότι το $\hat{y}$ που παράγει το INN προσεγγίζει την πραγματική τιμή y καθώς και μία συνάρτηση απωλειών L_z για τη συμμόρφωση των λανθανόντων $\hat{z}$ με την αρχική Gaussian κατανομή. Επιπλέον η L_z διασφαλίζει ότι η πληροφορία που κωδικοποιείται με την λανθάνουσα μεταβλητή δεν κωδικοποιείται και στην μέτρηση y, με άλλα λόγια οι δύο οντότητες δεν προσομοιώνουν κοινή πληροφορία. Τέλος για την επιτάχυνση της σύγκλισης του μοντέλου επιστρατεύεται και η μη-Επιβλεπόμενη backward απώλεια L_x όπου μειώνει το σφάλμα μεταξύ του εκτιμώμενου $\hat{x}$ με το x κατά την αντίστροφη χρήση του μοντέλου. Για τις L_z, L_x χρησιμοποιείται μια συνάρτηση πυρήνα ως συνάρτηση σφάλματος, η Maximum Mean Discrepancy [54]. Αποδεικνύεται ότι αν οι συναρτήσεις σφάλματος τείνουν στο 0 τότε το μοντέλο μπορεί να προσομοιώσει απόλυτα την πραγματική posterior κατανομή $p(x|y)$ του χώρου δεδομένων με συνθήκη τις μετρήσεις y.

Το μοντέλο αξιολογείται σε τεχνητά αλλά και πραγματικά σύνολα δεδομένων από προβλήματα κινηματικής και αστροφυσικής, με ικανοποιητική επίδοση.

3.3 Εναλλακτικές Προσεγγίσεις

Έχοντας αναλύσει τις βασικές αρχιτεκτονικές των INN μοντέλων, τίθεται το ερώτημα σχετικά με το πώς δύνανται να χρησιμοποιηθούν στην πράξη για την επίλυση των αντίστροφων προβλημάτων. Ένας τρόπος είναι να θεωρηθεί ως διάσταση D των INN η διάσταση του διανύσματος y (έστω m) από την σχέση (3.1) και αφού η είσοδος και η έξοδος του INN πρέπει να έχουν την ίδια διάσταση, αναγκαστικά και το x θα έχει διάσταση m. Αυτό σημαίνει πως θα πρέπει να ενσωματωθεί και ένα στάδιο μείωσης διάστασης για την μετατροπή των δειγμάτων x διάστασης $n > m$ σε δείγματα διάστασης m. Το γεγονός αυτό καθιστά περιοριστική την αρχιτεκτονική INN, ειδικά αν απόκλιση των n, m είναι μεγάλη, καθώς το στάδιο μείωσης διαστάσεων επιφέρει σημαντική απώλεια πληροφορίας.

Όπως αναφέραμε στο Κεφάλαιο 3.1, τα INN κατόπιν σύγκλισης, εκτός από την χωρίς σφάλματα ανακατασκευή των δειγμάτων εκπαίδευσης, μαθαίνουν και την υποκείμενη κατανομή του χώρου δεδομένων. Αυτό σημαίνει πως μπορούν να χρησιμοποιηθούν αποδοτικά ως γεννήτορες συνθετικών δειγμάτων και να ενσωματωθούν σε πρόβλημα βελτιστοποίησης για την επίλυση του αντίστροφου προβλήματος. Η προσέγγιση αυτή δεν επηρεάζεται από τη διάσταση των μετρήσεων y αλλά αφορά αποκλειστικά τον χώρο δεδομένων X, διευρύνοντας την εκφραστική ικανότητα του μοντέλου.

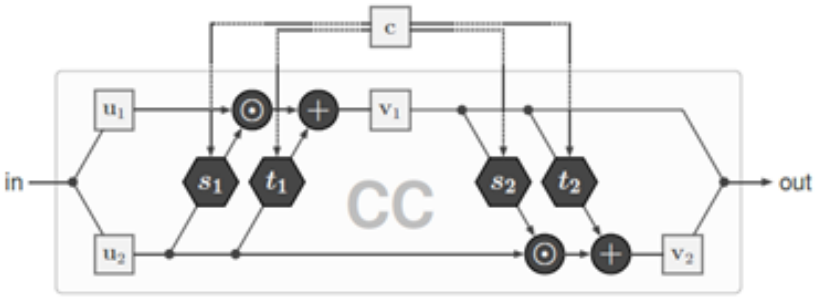
Την διαδικασία αυτή προτείνουν οι Asim et al. [55] όπου γίνεται χρήση INN μοντέλου ως προ-εκπαιδευμένος γεννήτορας για εφαρμογές εικόνων όπως Αφαίρεση Θορύβου από εικόνες (Image Denoising), Συμπιεστική Δειγματοληψία (Compressive Sensing) και Συμπλήρωση Εικόνας (Inpainting). Η κύρια αρχιτεκτονική για εφαρμογές που απαιτούν την χρήση γεννητόρων είναι τα GAN τα οποία εκ κατασκευής εμφανίζουν υψηλό σφάλμα ανακατασκευής των δειγμάτων εκπαίδευσης. Επίσης αδυνατούν να συνθέσουν

δείγματα εκτός της κατανομής του χώρου εκπαίδευσης (ο διαχωριστής χαρακτηρίζει άμεσα μια εικόνα ως ψευδή αν δεν ακολουθεί την κατανομή των δεδομένων εκπαίδευσης συνεπώς δεν ενθαρρύνεται η δημιουργία της). Οι συγγραφείς εκπαιδεύουν ένα INN μοντέλο (και συγκεκριμένα το Glow) με μη επιβλεπόμενο τρόπο πάνω σε εικόνες έτσι ώστε να αντιστοιχιστεί η πολύπλοκη κατανομή του χώρου δεδομένων στην αρχική Gaussian κατανομή του λανθάνοντος χώρου. Το INN μπορεί στη συνέχεια να χρησιμοποιηθεί ως προ-εκπαιδευμένη αρχική κατανομή του χώρου δεδομένων και άρα να εφαρμοσθεί σε πολλαπλές αντίστροφες εφαρμογές. Για το Image Denoising, στόχος είναι η ανάκτηση της εικόνας x από μετρήσεις της μορφής $y = Ax + \eta$ όπου η θόρυβος. Το πρόβλημα μπορεί να λυθεί ως ελαχιστοποίηση με κατάβαση κλίσης στον λανθάνοντα χώρο αντικαθιστώντας όπου x την αντίστροφη μέθοδο του INN έστω $G(z)$. Προκύπτει η ακόλουθη σχέση:

$$\min_{z \in \mathbb{R}^n} \|AG(z) - y\|^2 + \gamma \|z\|^2 \tag{3.19}$$

Για Compressing Sensing δίνονται υποδειγματοληπτημένες μετρήσεις y της εικόνας x και ζητείται η ανάκτησή της. Ως στόχος ελαχιστοποίησης χρησιμοποιείται η άνω σχέση με $\gamma = 0$. Τέλος το Inpainting δέχεται ως είσοδο μια εικόνα x με μάσκα (λείπουν στοιχεία της εισόδου) και στόχος του μοντέλου είναι να τη συμπληρώσει. Και στις τρεις εφαρμογές το INN αποδίδει καλύτερα από τους GAN ανταγωνιστές του και άρα συνιστά μία ικανοποιητική αρχική κατανομή του χώρου δεδομένων.

Μια εναλλακτική προσέγγιση για την επίλυση αντίστροφων προβλημάτων προτείνεται στην εργασία [56] όπου αναπτύσσεται για πρώτη φορά η αρχιτεκτονική των υπόσυνθήκη INN (conditional INN (cINN). Το μοντέλο αποτελεί μια επέκταση της κλασσικής INN δομής η οποία χρησιμοποιείται σε εφαρμογές σύνθεσης, ενσωματώνοντας ένα διάνυσμα συνθήκης c που ονομάζεται πλαίσιο (context) για την παραγωγή/σύνθεση αποτελεσμάτων σχετικών με το context c. Οι συγγραφείς κατασκευάζουν ένα βασικό Coupling INN και βασιζόμενοι στην παρατήρηση ότι τα δίκτυα s, t δεν αντιστρέφονται, συνενώνουν στην είσοδο των δικτύων αυτών και το διάνυσμα συνθήκης c. Όμως το c δεν είναι αποδοτικό να συμπεριληφθεί ολόκληρο στην είσοδο των s, t αλλά μόνο τα σημαντικά χαρακτηριστικά του. Για αυτό κατασκευάζεται και ένα feed-forward δίκτυο που εκπαιδεύεται παράλληλα με το INN (μπορεί να χρησιμοποιηθεί και ως προ εκπαιδευμένο) για να μετασχηματίσει το $c \to \hat{c}$ μειώνοντας την διάστασή του. Το μοντέλο εκπαιδεύεται με μέγιστη λογαριθμική πιθανοφάνεια. Η δομή του Coupling Layer του cINN παρουσιάζεται στο Σχήμα 3.4.



Σχήμα 3.4: Δομή υπό συνθήκης Coupling Layer (Conditional Coupling (CC)). Πηγή [56]

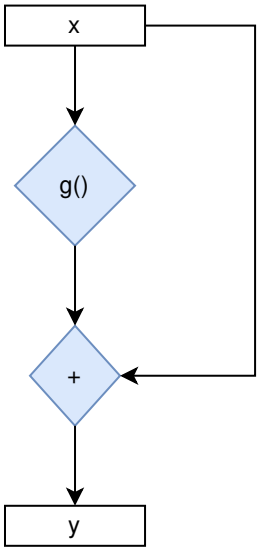
Η προσέγγιση των cINN απελευθερώνει την διάσταση του λανθάνοντος χώρου, η οποία πλέον ταυτίζεται με την διάσταση του χώρου δεδομένων x, με την αντιστοίχιση των δύο χώρων να πραγματοποιείται χωρίς σημαντικές απώλειες. Επιπλέον, το διάνυσμα μετρήσεων y δεν είναι απαραίτητο να χρησιμοποιηθεί μόνο στο πρόβλημα βελτιστοποίησης αλλά συνιστά το context του δικτύου καθοδηγώντας τα συνθετικά δείγματα που παράγει ο γεννήτορας cINN.

Πέρα όμως από τις «κλασσικές» αρχιτεκτονικές INN, η επιστημονική κοινότητα αναζητά τρόπους για την επίτευξη αντιστρεψιμότητας και σε άλλες κατηγορίες μοντέλων Μηχανικής Μάθησης.

Μια πολύ διαδεδομένη αρχιτεκτονική δικτύων Βαθιάς Μηχανικής Μάθησης με εντυπωσιακά αποτελέσματα στο πεδίο της Επεξεργασίας Εικόνων αποτελούν τα Υπολειμματικά Νευρωνικά Δίκτυα (Residual Neural Networks (ResNets)) [57]. Ο βασικός πυρήνας των ResNets είναι το Υπολειπόμενο Μπλοκ (Residual Block - Σχήμα 3.5) όπου για είσοδο x και έξοδο y ισχύει η σχέση:

$$y = x + g(x) \tag{3.20}$$

όπου g ένα αυθαίρετα πολύπλοκο νευρωνικό δίκτυο. Ένα ResNet υλοποιείται ως μια ακολουθία πολλών διαδοχικών Residual Blocks.



Σχήμα 3.5: Δομή Residual Block

Στην εργασία τους, οι Behrmann et al. [58] κατασκευάζουν ένα αντίστροφο νευρωνικό δίκτυο βασιζόμενο στην ResNet υλοποίηση με όνομα Αντιστρέψιμο Υπολειμματικό Δίκτυο (Invertible Residual Network). Όλες οι INN αρχιτεκτονικές που έχουν αναφερθεί μέχρι στιγμής έχουν σχεδιαστεί για εκτίμηση πυκνότητας και αποδίδουν σε εφαρμογές σύνθεσης, αλλά όχι τόσο καλά σε εφαρμογές διαχωρισμού και ταξινόμησης. Επιπλέον επιβάλλουν αντιστρεψιμότητα μέσω της αρχιτεκτονικής τους έτσι ώστε να υπάρχει και tractable Ιακωβιανή ορίζουσα, γεγονός που τα περιορίζει σχεδιαστικά και ενδεχομένως μειώνει την ικανότητα μάθησης πολύπλοκων χαρακτηριστικών. Τα i-ResNets εξασφαλίζουν αντιστρεψιμότητα τόσο στην forward όσο και στην backward λειτουργία τους αν κάθε υπολειπόμενο τμήμα της μορφής $I + g_{\theta_t}$ διαθέτει Lipschitz σταθερά μικρότερη από 1.

$$\text{Lip}(g_{\theta_t}) < 1, \text{ for all } t = 1, \ldots, T \tag{3.21}$$

Η παρατήρηση αυτή προκύπτει αν παρομοιαστεί η σχέση που περιγράφει το υπολειπόμενο επίπεδο με την μέθοδο του Euler για επίλυση Συνήθων Διαφορικών Εξισώσεων (Ordinary Differential Equations). Το backward πέρασμα πραγματοποιείται με έναν επαναληπτικό αλγόριθμο με αρχική τιμή είτε την έξοδο y είτε τυχαίο διάνυσμα.

Για να ικανοποιηθεί η συνθήκη και να εξασφαλιστεί αντιστρεψιμότητα θα πρέπει να χρησιμοποιούνται συσταλτικές (contractive) συναρτήσεις ενεργοποίησης όπως ReLU και Tanh στο υπολειμματικό τμήμα και να κανονικοποιούνται τα βάρη των γραμμικών επιπέδων έτσι ώστε $||W_i||_2 < 1$.

Παρόλο που η i-ResNet αρχιτεκτονική είναι αντιστρέψιμη, δεν διαθέτει tractable Ιακωβιανή ορίζουσα. Με δεδομένο ότι ισχύει η Lipschitz συνθήκη, δύναται να κατασκευαστεί αλγόριθμος που προσεγγίζει το $log\ (det\ (Jac(g_{\theta_t})))$, εκμεταλλευόμενος το ίχνος του πίνακα, και εκφράζοντας την ζητούμενη ποσότητα ως μια πεπερασμένη σειρά. Με άλλα

Algorithm 1 Αλγόριθμος Υπολογισμού i-ResNet. Πηγή [58]

Require: output from residual layer y, contractive residual block g, number of fixed-point iterations n

1: $x^0 \leftarrow y$
2: **for** $i = 0, \ldots, n$ **do**
3: $\quad x^{i+1} \leftarrow y - g(x^i)$
4: **end for**

λόγια, για ένα μικρό σφάλμα, καθίσταται δυνατός ο υπολογισμός της Ιακωβιανής ορίζουσας χωρίς να είναι απαραίτητο να έχει ο πίνακας άνω τριγωνική δομή και επιτρέποντας ισχυρότερη εκφραστική δύναμη στο μοντέλο. Η αρχιτεκτονική αξιολογείται σε σύνολα εικόνων τόσο πειράματα διαχωρισμού όσο και σύνθεσης και συγκρίνεται με άλλες παραδοσιακές INN τεχνικές και με τα «απλά» ResNets. Τα πειράματα δείχνουν πως τα i-ResNets εμφανίζουν πολύ καλύτερη απόδοση στα πειράματα διαχωρισμού και είναι ανταγωνιστικά σε Generative Learning περιβάλλον εκτέλεσης.

Γίνεται λοιπόν αντιληπτό πως για την ύπαρξη tractable Ιακωβιανής ορίζουσας η αρχιτεκτονική του forward μετασχηματισμού f πρέπει να είναι συγκεκριμένης μορφής, περιορίζοντας την εκφραστικότητα του μοντέλου. Τα i-ResNets χαλαρώνουν αυτούς τους περιορισμούς και βασίζονται στην προσέγγιση της Ιακωβιανής ορίζουσας. Στο ίδιο πλαίσιο, οι συγγραφείς του [59] προτείνουν μια μέθοδο για την υπολογιστικά αποδοτική προσέγγιση της Ιακωβιανής Ορίζουσας ενός INN το οποίο δεν διαθέτει περιορισμένη ή συγκεκριμένης δομής αρχιτεκτονική. Πιο συγκεκριμένα, η μέθοδος βασίζεται στην ιδέα του Chen et al. [60] όπου μοντελοποιεί τη backward διάδοση, δηλαδή την παραγωγή ενός δείγματος X με δεδομένο ένα λανθάνον διάνυσμα z_0, ως λύση ODE και το μοντέλο εκπαιδεύεται με βάση την log-density:

$$\log p(\mathbf{z}(t_1)) = \log p(\mathbf{z}(t_0)) - \int_{t_0}^{t_1} \mathrm{Tr}\left(\frac{\partial f}{\partial \mathbf{z}(t)}\right) dt. \tag{3.22}$$

Οι συγγραφείς προτείνουν έναν αποδοτικό τρόπο υπολογισμού του ίχνους του Ιακωβιανού πίνακα σε γραμμικό χρόνο (ως προς την διάσταση εισόδου/εξόδου του INN) με την μέθοδο Hutchinson's Εκτίμηση Ίχνους [61] όπου το ίχνος ενός πίνακα A μπορεί να προσεγγισθεί από: $\mathrm{Tr}(A) = \mathbb{E}_{p(\epsilon)}[\epsilon A \epsilon^T]$ όπου ϵ ένα διάνυσμα θορύβου με μέση τιμή 0 και διασπορά 1. Στη συνέχεια, οι συγγραφείς επιστρατεύουν τη γενικευμένη αυτή μέθοδο προσέγγισης της ορίζουσας του Ιακωβιανού πίνακα έτσι ώστε να συνδυαστεί με ένα INN μη-περιορισμένης αρχιτεκτονικής με όνομα FFJORD το οποίο δύναται να χρησιμοποιηθεί ως «κλασσικό» INN σε εφαρμογές αντίστροφων προβλημάτων.

Κλείνοντας, αξίζει να αναφερθεί η μελέτη των Kruse et al. [62] στην οποία αναπτύσσονται και συγκρίνονται αρχιτεκτονικές INN (κλασσικές και μη) πάνω σε αντίστροφα προβλήματα φυσικής. Αρχικά εισάγονται οι ιδιότητες του αυστηρής αντιστρεψιμότητας (hard invertibility) και χαλαρής αντιστρεψιμότητας (soft invertibility) με την πρώτη να δηλώνει πως η αρχιτεκτονική του μοντέλου είναι υπεύθυνη για την πλήρη αντιστρεψιμότητα των forward και backward μετασχηματισμών ενώ η δεύτερη διασφαλίζει αντιστρεψιμότητα κατόπιν σύγκλισης. Στη συνέχεια κατασκευάζονται 10 μοντέλα μερικά από τα οποία είναι: το INN που προτάθηκε στο [11] βασισμένο στο RealNVP, ένα cINN [56], ένα autoregressive flow μοντέλο [63], ένα invertible ResNet [58], ένας αντιστρέψιμος και ένας απλός αυτοκωδικοποιητής [64] και ένας υπό – συνθήκη κωδικοποιητής [65]. Από τα πειράματα προκύπτουν πως οι βασισμένες σε Coupling Layer αρχιτεκτονικές (RealNVP και cINN) αλλά και ο απλός αυτοκωδικοποητής (ο αποκωδικοποιητής είναι η αντίστροφη διαδικασία του κωδικοποιητή κατόπιν σύγκλισης) αποδίδουν καλύτερα στην επίλυση αντίστροφων προβλημάτων.

Κεφάλαιο 4

Μεθοδολογία

4.1 Επισκόπηση Μεθόδου

Όπως αναφέρθηκε στο Κεφάλαιο 2, η Εκτίμηση Πίνακα Κίνησης (ΕΠΚ) αφορά την εξαγωγή του πίνακα κίνησης x (τα στοιχεία του οποίου ονομάζονται ροές κίνησης πηγής-προορισμού (OD Flows)) από τις εύκολα μετρήσιμες τιμές y της ομαδοποιημένης κίνησης στις ζεύξεις, για γνωστό πίνακα δρομολόγησης A. Το πρόβλημα λοιπόν εκφράζεται ως:

$$y = Ax \tag{4.1}$$

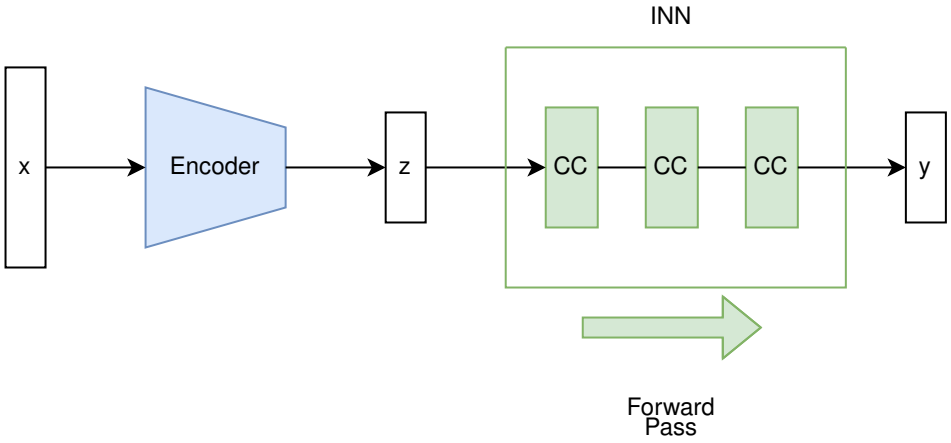
και εντάσσεται στην κατηγορία των αντίστροφων γραμμικών προβλημάτων. Μία τεχνική Βαθιάς Μηχανικής Μάθησης, ειδικά σχεδιασμένη για την επίλυση αντίστροφων προβλημάτων είναι τα Αντιστρέψιμα Νευρωνικά Δίκτυα (Invertible Neural Networks (INN)) τα οποία αναλύονται εκτενώς στο Κεφάλαιο 3.

Μία από τις βασικές ιδιότητες των INN μοντέλων είναι πως οι διαστάσεις της εισόδου και της εξόδου τους πρέπει να ταυτίζονται. Στην *ΕΠΚ* όμως, στη γενική περίπτωση, ο πίνακας x (αντιμετωπίζεται ως ένα στοιβαγμένο διάνυσμα) έχει πολύ μεγαλύτερη διάσταση από τις μετρήσεις y. Για αυτόν τον λόγο, ως στάδιο προ-επεξεργασίας των δεδομένων, θα εφαρμόσουμε μια διαδικασία Ελάττωσης Διαστάσεων (Dimensionality Reduction) [66] ούτως ώστε να μετασχηματίσουμε τους πίνακες-διανύσματα x, που κωδικοποιούν OD Flows, σε διανύσματα μικρότερης κατάλληλης διάστασης που επιτρέπει την ενσωμάτωση των INN στην διαδικασία επίλυσης του προβλήματος. Ειδικότερα, ως στάδιο ελάττωσης της διάστασης του χώρου δεδομένων, θα κατασκευαστεί ένας Αυτοκωδικοποιητής (Autoencoder) [67] και συγκεκριμένα ένας συνελικτικός (convolutional) Autoencoder [68] για να πραγματοποιήσει μη γραμμική απεικόνιση [69] των δειγμάτων του χώρου δεδομένων X στον μειωμένης διάστασης λανθάνοντα χώρο Z.

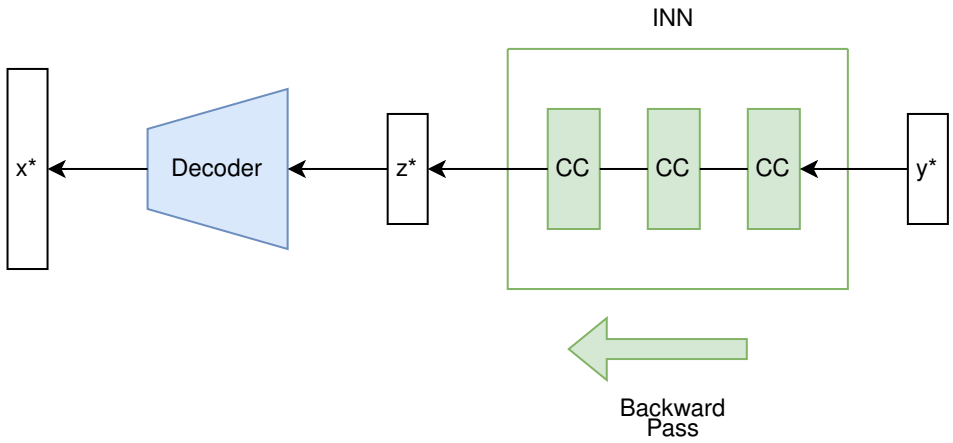
Η συνολική ροή εργασίας της προτεινόμενης μεθόδου για την ΕΠΚ μπορεί να συνοψιστεί ως εξής. Αρχίζοντας από παρατηρήσεις προηγούμενων χρονικών στιγμιότυπων του

πίνακα κίνησης $X = \{x_t\}_{t=1}^{T}$ και πολλαπλασιάζοντας κάθε δείγμα με τον πίνακα δρομολόγησης A, μπορούμε να αποκτήσουμε τις μετρήσεις των φορτίων των ζεύξεων και άρα να δημιουργήσουμε ένα σύνολο δεδομένων από ζεύγη (X, y). Στη συνέχεια, χρησιμοποιούμε τους πίνακες x του συνόλου για την εκπαίδευση του Autoencoder και κρατώντας μόνο τον εκπαιδευμένο κωδικοποιητή (encoder) μετασχηματίζουμε το σύνολο X στο σύνολο του λανθάνοντος χώρου $Z = \{z_t\}_{t=1}^{T}$. Αυτές οι απεικονίσεις τροφοδοτούν ένα INN δίκτυο το οποίο κατά την εκπαίδευση μαθαίνει τον γνωστό forward μετασχηματισμό $f : Z \to Y$ (έμμεσα γνωστό γιατί ο γνωστός μετασχηματισμός αφορά τον χώρο δεδομένων X) ενώ συγχρόνως μοντελοποιεί, μέσω της αντιστρεψιμότητας της αρχιτεκτονικής του, και τον backward μετασχηματισμό $f^{-1} : Y \to Z$. Στη φάση της εκτίμησης, ένα διάνυσμα μετρήσεων y^* δίνεται ως είσοδος στην αντίστροφη μέθοδο του INN και άρα παράγεται μια εκτίμηση z^* στον λανθάνοντα χώρο Z η οποία στην συνέχεια μεταφράζεται μέσω του εκπαιδευμένου αποκωδικοποιητή (decoder) στην τελική εκτίμηση του πίνακα κίνησης του δικτύου x^*. Μια γενική εποπτεία για την προτεινόμενη ροή εργασίας παρέχεται από τα Σχήματα 4.1 και 4.2.

Στη συνέχεια του κεφαλαίου θα αναλυθεί η αρχιτεκτονική του Convolutional Autoencoder και οι διαφορετικές εκδοχές του τρόπου λειτουργίας και εκπαίδευσης του INN μοντέλου.



Σχήμα 4.1: Forward Ροή Εργασίας. CC = Coupling Layer



Σχήμα 4.2: Backward Ροή Εργασίας. CC = Coupling Layer

4.2 Αρχιτεκτονική Αυτοκωδικοποιητή

Ο Αυτοκωδικοποιητής (Autoencoder) είναι ένα νευρωνικό δίκτυο που εμπίπτει στην κατηγορία της Μη Επιβλεπόμενης Μάθησης [70] και χρησιμοποιείται για την εκμάθηση κωδικοποιήσεων για δεδομένα που δεν διαθέτουν ετικέτες. Αποτελείται από δύο οντότητες (Σχήμα 4.3), τον Κωδικοποιητή (Encoder) και τον Αποκωδικοποιητή (Decoder) οι οποίες υλοποιούνται, τυπικά, ως feed-forward νευρωνικά δίκτυα.

Ο Κωδικοποιητής λαμβάνει ως είσοδο τα δείγματα του χώρου δεδομένων και τα μετασχηματίζει σε μια συμπιεσμένη, χαμηλότερης διάστασης αναπαράσταση στον λανθάνοντα χώρο που είναι γνωστός ως Κώδικας (Code). Ο Αποκωδικοποιητής λαμβάνει ως είσοδο μια αναπαράσταση του λανθάνοντος χώρου και προσπαθεί να ανακατασκευάσει το δείγμα του χώρου δεδομένων από το οποίο προέρχεται η αναπαράσταση. Μαθηματικά, για διάνυσμα εισόδου x έχουμε:

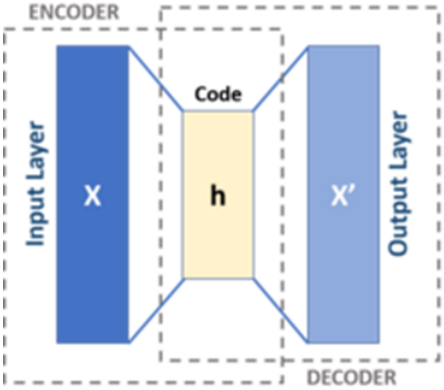
$$z = Enc(x) \tag{4.2}$$

$$\hat{x} = Dec(z) \tag{4.3}$$

όπου $Enc()$ και $Dec()$ αναπαριστούν τα νευρωνικά δίκτυα του Κωδικοποιητή και του Αποκωδικοποιητή Αντίστοιχα.

Για την εκπαίδευση του Αυτοκωδικοποιητή, τα δείγματα εκπαίδευσης κωδικοποιούνται μέσω του Κωδικοποιητή και στη συνέχεια ανακατασκευάζονται από τον Αποκωδικοποιητή με ζητούμενο να είναι ο προσδιορισμός των παραμέτρων των δύο νευρωνικών δικτύων ούτως ώστε να ελαχιστοποιηθεί το σφάλμα ανακατασκευής. Με άλλα λόγια, ο Αυτοκωδικοποιητής για είσοδο ένα διάνυσμα x έχει ως έξοδο ένα $\hat{x}$ (ιδανικά $\hat{x} = x$) ενώ συγχρόνως, αν απομονωθεί ο Κωδικοποιητής, το μοντέλο δύναται να παραγάγει αποδοτικά και με ελάχιστα σφάλματα κωδικοποιήσεις μικρότερης διάστασης για τα δείγματα

του χώρου δεδομένων. Το σφάλμα ανακατασκευής για την εκπαίδευση του δικτύου εκφράζεται συνήθως ως από το Μέσο Τετραγωνικό Σφάλμα (Mean Square Error) ή από το Cross-Entropy.



Σχήμα 4.3: Αρχιτεκτονική Autoencoder. Πηγή Wikipedia

Στην εφαρμογή μας, ο Αυτοκωδικοποιητής, χρησιμοποιείται για τη μείωση της διάστασης του χώρου δεδομένων, σε διάσταση που καθίσταται «βολική» για την ενσωμάτωση του INN στην ροή εργασίας για την επίλυση του προβλήματος. Για τη μείωση διάστασης, ο Αυτοκωδικοποιητής εκπαιδεύεται πάνω στα ιστορικά δεδομένα (προηγούμενα στιγμιότυπα του πίνακα κίνησης x) και άρα μαθαίνει μια αποδοτική κωδικοποίηση των πινάκων στον λανθάνοντα χώρο Z ελαχιστοποιώντας την συνάρτηση απωλειών Huber Loss [71] μεταξύ της εισόδου x και της ανακατασκευασμένης $\hat{x}$:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{if } |a| \leq \delta, \\ \delta(|a| - \frac{1}{2}\delta) & \text{if } |a| > \delta \end{cases} \tag{4.4}$$

όπου $a = x - \hat{x}$. Η συνάρτηση απωλειών Huber συνδυάζει τα χαρακτηριστικά της τετραγωνικής απώλειας και της απώλειας απόλυτου. Για μικρές τιμές του a, η απώλεια είναι τετραγωνική (quadratic) και για μεγαλύτερες τιμές είναι γραμμική (linear). Αυτό κάνει τη συνάρτηση Huber πιο ανθεκτική σε εκτός ορίων τιμές (outliers) σε σχέση με την απλή τετραγωνική απώλεια. Μετά την εκπαίδευση του Αυτοκωδικοποιητή, απομονώνουμε τον Κωδικοποιητή και μετασχηματίζουμε όλα τα δεδομένα εκπαίδευσης σε αναπαραστάσεις τους στον λανθάνοντα χώρο. Ο Αποκωδικοποιητής θα χρησιμοποιηθεί μόνο στην φάση εκτίμησης για την επαναφορά του παραγόμενου από το INN δείγματος του χώρου Z στον αρχικό χώρο δεδομένων Z.

Στο Κεφάλαιο 4.1, αναφέρθηκε πως θα γίνει χρήση ενός Συνελικτκού Αυτοκωδικοποιητή [68]. Ο Αυτοκωδικοποιητής αυτός διατηρεί τα ίδια χαρακτηριστικά με τον απλό Αυτοκωδικοποιητή, με τη διαφορά ότι τα επίπεδα των νευρωνικών δικτύων που απαρτίζουν τις δομικές του συνιστώσες δεν είναι μόνο γραμμικά επίπεδα (που αποτελούν πυρήνα

των παραδοσιακών νευρωνικών δικτύων πολλαπλών επιπέδων) αλλά συμπεριλαμβάνουν και Συνελικτικά Επίπεδα (Convolutional Layers).

Τα συνελικτικά επίπεδα [38] μοντελοποιούν την είσοδό τους ως πλέγμα στοιχείων διατεταγμένων σε χωρικές διαστάσεις. Πιο συγκεκριμένα, κάθε δείγμα εισόδου χαρακτηρίζεται από τις χωρικές διαστάσεις ύψους (h) και του πλάτους (w) αλλά και από μια διάσταση βάθους/καναλιών (c). Σε αυτήν την δομή, κατά την διάδοση μέσω του συνελικτικού επιπέδου, εφαρμόζεται ένας πυρήνας (kernel) που αποτελεί τις παραμέτρους του επιπέδου διαστάσεων $h' \times w' \times c$ (η διάσταση του βάθους ταυτίζεται πάντα με τον αριθμό καναλιών της εισόδου) και πραγματοποιείται η πράξη της συνέλιξης. Η διαδικασία της συνέλιξης, τοποθετεί το φίλτρο σε κάθε πιθανή θέση στην εικόνα έτσι ώστε το φίλτρο να επικαλύπτει πλήρως την εικόνα και να πραγματοποιεί εσωτερικό γινόμενο μεταξύ των παραμέτρων του πυρήνα και του αντίστοιχου πλέγματος εισόδου. Σε κάθε επίπεδο, μπορούν να εφαρμοστούν πολλαπλοί πυρήνες με την έξοδο της συνέλιξης κάθε πυρήνα και της εισόδου να ονομάζεται χάρτης χαρακτηριστικών (feature map). Στόχος του Συνελικτικού επιπέδου είναι η ανίχνευση χωρικών ιδιοτήτων και χαρακτηριστικών της εισόδου, τα οποία συνδυάζονται και οδηγούν στην αναγνώριση ολοένα και πιο πολύπλοκων δομών όσο βαθαίνει η αρχιτεκτονική του δικτύου. Για παράδειγμα, στον τομέα των εικόνων τα επίπεδα που βρίσκονται πιο κοντά στην είσοδο αναγνωρίζουν απλές δομές της εικόνας όπως γραμμές, ενώ τα ανώτερα επίπεδα δύνανται να αναγνωρίζουν ακόμα και πρόσωπα.

Για είσοδο διάστασης $h \times w \times c$ και d πυρήνες διάστασης $h' \times w' \times c$ προκύπτει έξοδος διάστασης $(h - h' + 1) \times (w - w' + 1) \times d$.

Η διαδικασία της συνέλιξης μειώνει τις χωρικές διαστάσεις της εξόδου σε σχέση με την είσοδο γεγονός που οδηγεί σε απώλεια πληροφορίας κατά μήκος των ορίων της εισόδου. Για την αποφυγή του φαινομένου αυτού, επιστρατεύεται η διαδικασία της πλήρωσης (padding) δηλαδή της προσθήκης P μηδενικών στοιχείων σε κάθε διάσταση της εισόδου με στόχο τη διατήρηση του χωρικού αποτυπώματος. Επίσης, η κλασσική συνέλιξη πραγματοποιεί εσωτερικό γινόμενο σε κάθε δυνατή θέση που μπορεί να τοποθετηθεί ο πυρήνας πάνω στο πλέγμα εισόδου. Για την μείωση της κοκκιότητας της συνέλιξης, χρησιμοποιείται η έννοια του βήματος όπου ο πυρήνας τοποθετείται σε διαδοχικές θέσεις 1, S + 1, 2S + 1 και ούτω καθεξής, με το S να ονομάζεται βήμα (stride). Συνεπώς για padding P και stride S, η έξοδος του συνελικτικού επιπέδου θα έχει διάσταση:

$$\left(\frac{h - h' + 2P}{S} + 1, \frac{w - w' + 2P}{S} + 1, d\right) \tag{4.5}$$

Στην έξοδο του Συνελικτικού επιπέδου, εφαρμόζεται μια συνάρτηση ενεργοποίησης και οι παράμετροι των πυρήνων (καθώς και οι μεροληψίες τους) εκπαιδεύονται με τον αλγόριθμο του Backpropagation [72].

Ερμηνεύοντας τους πίνακες κίνησης ως εικόνες διάστασης $n \times n \times 1$, ο Συνελικτικός Αυτοκωδικοποιητής μετασχηματίζει (μέσω του Κωδικοποιητή) το δείγμα εισόδου x σε ένα διάνυσμα του λανθάνοντος χώρου διάστασης d.

4.3 Αρχιτεκτονική INN

Με τη βοήθεια του Αυτοκωδικοποιητή το σύνολο δεδομένων X, δηλαδή των ιστορικών πινάκων κίνησης του δικτύου, μετασχηματίζεται σε δείγματα του λανθάνοντος χώρου Z διάστασης d και τροφοδοτούν το INN δίκτυο της ροής εργασίας. Υποθέτοντας μια αμφίδρομη συνάρτηση $f : \mathbb{R}^d \to \mathbb{R}^d$ η οποία αντιστοιχίζει δείγματα $z \in \mathbb{R}^d$ του λανθάνοντος χώρου σε μετρήσεις φορτίων ζεύξεων $y \in \mathbb{R}^d$ και θεωρώντας μια απλή αρχική κατανομή p_Y προκύπτει μέσω του τύπου αλλαγής μεταβλητής (3.2) η σχέση:

$$p_Z(z) = p_Y(f(z)) \cdot \left| \det \left(\frac{\partial f(z)}{\partial z} \right) \right| \tag{4.6}$$

Εφαρμόζοντας τη λογαριθμική συνάρτηση και στα δύο μέλη της σχέσης προκύπτει:

$$\log(p_Z(z)) = \log(p_Y(f(z))) + \log \left(\left| \det \left(\frac{\partial f(z)}{\partial z} \right) \right| \right) \tag{4.7}$$

Όπου $\frac{\partial f(z)}{\partial z}$ είναι ο Ιακωβιανός πίνακας του μετασχηματισμού f. Επιδιώκοντας η αμφίδρομη f να διαθέτει tractable ορίζουσα του Ιακωβιανού πίνακα, επιλέγεται η χρήση διαδοχικών αφινικών Coupling Layer επιπέδων όπως αυτά περιγράφονται στο [50] και στο Κεφάλαιο 3.2. Πιο συγκεκριμένα, η είσοδος σε κάθε επίπεδο χωρίζεται σε δύο τμήματα ίσης διάστασης $\mathbb{R}^{d/2}$, $z = [z_1, z_2]$ και μετασχηματίζεται ως εξής:

$$y_1 = z_1 \tag{4.8}$$

$$y_2 = z_2 \odot exp\left(s(z_1)\right) + t(z_1) \tag{4.9}$$

όπου, ο τελεστής $\odot$ είναι ο πολλαπλασιασμός στοιχείου με στοιχείο των δύο διανυσμάτων και τα s, t είναι αυθαίρετα πολύπλοκα συναρτήσεις $\mathbb{R}^{d/2} \to \mathbb{R}^{d/2}$.

Η έξοδος y του επιπέδου μπορεί να υπολογιστεί ως η συνένωση των y_1, y_2 ενώ η ορίζουσα του Ιακωβιανού πίνακα ως exp $\left(\sum_j s(x_1)_j\right)$ εκμεταλλευόμενη την τριγωνική του δομή. Δεδομένου ότι το INN αποτελείται από πολλά διαδοχικά Coupling Layers συνιστώντας σύνθεση των συναρτήσεων f, η ορίζουσα του Ιακωβιανού υπολογίζεται εύκολα ως το γινόμενο (άθροισμα σε λογαριθμική μορφή) των επιμέρους οριζουσών γεγονός που επιτρέπει την tractable ιδιότητά της. Ο μετασχηματισμός αυτός ονομάζεται forward διάδοση στο INN δίκτυο και μπορεί εύκολα να αντιστραφεί. Ο αντίστροφος μετασχηματισμός (backward διάδοση) για είσοδο $y = [y_1, y_2]$ περιγράφεται από την σχέση:

$$z_1 = y_1 \tag{4.10}$$

$$z_2 = (y_2 - t(y_1)) \odot exp\left(-s(y_1)\right) \tag{4.11}$$

Με στόχο την αύξηση της ευελιξίας και της εκφραστικής ικανότητας των forward και backward διαδόσεων, τα s, t υλοποιούνται ως αλληλουχία διαδοχικών συνελικτικών 1D επιπέδων ακολουθούμενα από ένα γραμμικό, δηλαδή πλήρως συνδεδεμένο, επίπεδο.

Παρατηρώντας τις σχέσεις των μετασχηματισμών, διαπιστώνουμε εύκολα πως μόνο τα μισά στοιχεία της εισόδου επηρεάζονται από τις δομές του επιπέδου, επιτρέποντας στο άλλο μισό της εισόδου να διαδίδεται στην έξοδο χωρίς αλλαγές μειώνοντας έτσι την αναπαραστατική ικανότητα του επιπέδου. Για την αποφυγή αυτού του φαινομένου, πριν από κάθε Coupling Layer η είσοδος χωρίζεται στα δύο τμήματα και αντιμετατίθεται έτσι ώστε κάθε στοιχείο της τελικής εξόδου της ακολουθίας των Coupling Layers να εξαρτάται από όλα τα στοιχεία της εισόδου.

4.4 Τρόποι Λειτουργίας INN

Σε αυτό το σημείο θα αναλύσουμε τους εναλλακτικούς τρόπους με τους οποίος λειτουργεί, εκπαιδεύεται και παράγει τις εκτιμήσεις των πινάκων κίνησης το INN μοντέλο.

Ο πρώτος τρόπος λειτουργίας ονομάζεται **INN Baseline** (μοντέλο βάσης) και χρησιμοποιείται ως μια πρώτη εκτίμηση της εκφραστικής ικανότητας της INN αρχιτεκτονικής. Πιο συγκεκριμένα, σε αυτό το περιβάλλον λειτουργίας, οι πίνακες κίνησης x $(n \times n)$ μετασχηματίζονται σε διανύσματα z ίδιας διάστασης m με την διάσταση των φορτίων ζεύξεων y. Ο μετασχηματισμός πραγματοποιείται από τον Κωδικοποιητή του Αυτοκωδικοποιητή, ο οποίος έχει προ-εκπαιδευτεί για την αξιόπιστη αυτή μείωση διάστασης. Στη συνέχεια κατασκευάζεται το INN μοντέλο που περιγράφηκε στο Κεφάλαιο 4.3, το οποίο όμως εξετάζουμε και χρησιμοποιούμε ως ένα τυπικό feed-forward νευρωνικό δίκτυο αγνοώντας τόσο την αντίστροφη μέθοδο που περιέχει όσο και τον μετασχηματισμό 4.6 που το χαρακτηρίζει. Στόχος αυτής της ενέργειας είναι ο προσδιορισμός του κατά πόσο η αλληλουχία των Coupling Layers μπορεί να μοντελοποιήσει ικανοποιητικά τον αντίστροφο μετασχηματισμό $z = f^{-1}(y)$ και να αποτελέσει βάση στην επίδοση των επόμενων τρόπων λειτουργίας.

Για την εκπαίδευσή του, το INN δέχεται ως είσοδο ένα φορτίο ζεύξης y και κατόπιν διάδοσής του μέσω forward μετασχηματισμού (4.8), (4.9) στα Coupling Layers, παράγει ως έξοδο ένα $\hat{z}$ το οποίο συγκρίνεται με την πραγματική μετασχηματισμένη μορφή z του αρχικού πίνακα κίνησης x. Στη συνέχεια υπολογίζεται το σφάλμα μεταξύ $\hat{z}$ και z επιστρατεύοντας ως συνάρτηση απωλειών την Huber Loss. Το σφάλμα διαδίδεται στις παραμέτρους του δικτύου με Backpropagation. Για την *ΕΠΚ* και την αξιολόγηση του μοντέλου, δίνεται ως είσοδος η τιμή του διανύσματος y, υπολογίζεται το $\hat{z}$ και στη συνέχεια μέσω του Αποκωδικοποιητή εκτιμάται το $\hat{x}$.

Αξίζει να σημειωθεί πως για λόγους απόδοσης, οι τιμές των z και y κανονικοποιούνται με κλιμάκωση ελαχίστου μεγίστου (min-max scaling) ως $z' = \frac{z - min_z}{max_z - min_z}$ και $y' = \frac{y - min_y}{max_y - min_y}$ αντίστοιχα, με την εκπαίδευση να πραγματοποιείται πάνω στα κανονικοποιημένα μεγέθη. Την ίδια κανονικοποίηση για τους πίνακες x επιβάλουμε και στον Αυτοκωδικοποιητή κατά την δική του εκπαίδευση. Η ενέργεια αυτή περιορίζει το εύρος των δυνατών τιμών που μπορεί να λάβει η είσοδος/έξοδος των μοντέλων γεγονός που επιτρέπει την αποδοτικότερη "εστίαση" της μάθησης στο εύρος [0,1].

Ο επόμενος τρόπος λειτουργίας, **INN Original**, αξιοποιεί πλήρως τα χαρακτηριστικά της INN αρχιτεκτονικής με την κύρια ιδέα που το χαρακτηρίζει να είναι η ταυτόχρονη βελτιστοποίηση των απωλειών τόσο στο πεδίο των μετρήσεων y όσο και στον λανθάνοντα χώρο Z. Και σε αυτόν τον τρόπο λειτουργίας ο Αυτοκωδικοποιητής εκπαιδεύεται κατάλληλα για τον μετασχηματισμό των πινάκων κίνησης x σε διανύσματα z ίδιας διάστασης με τα y. Οι αλλαγές όμως εντοπίζονται στον βρόχο εκπαίδευσης. Ειδικότερα, το INN Original μοντέλο δέχεται ως είσοδο τα δείγματα z τα οποία διαδίδονται με τον forward μετασχηματισμό (4.8), (4.9) στα Coupling Layers για την παραγωγή ενός $\hat{y}$ αλλά και τις ακριβούς τιμής της ορίζουσας του Ιακωβιανού πίνακα, με τον τρόπο που περιγράφηκε στο Κεφάλαιο 4.3. Στη συνέχεια υπολογίζεται η συνάρτηση απωλειών του forward μετασχηματισμού ως εξής:

$$L_y = \text{MSE}(\hat{y}, y) - c \cdot \log\det\left(\frac{\partial f(z)}{\partial z}\right) \tag{4.12}$$

όπου:

$$\text{MSE}(y, \hat{y}) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{4.13}$$

για διανύσματα y διάστασης m και c μια παράμετρος βάρους. Ο πρώτος όρος της σχέσης (4.12) αποσκοπεί στην ελαχιστοποίηση της απόκλισης μεταξύ της εκτιμώμενης και της πραγματικής τιμής των φορτίων ζεύξεων και άρα πραγματοποιεί έμμεσα την μεγιστοποίηση πιθανοφάνειας του όρου $p_Y(f(z))$ της σχέσης (4.6). Ο δεύτερος όρος στοχεύει στην ελαχιστοποίηση της αρνητικής λογαριθμικής πιθανοφάνειας του αντίστοιχου όρου στην σχέση (4.7). Με άλλα λόγια, στόχος της απώλειας L_y είναι η forward διάδοση να προσεγγίσει (και ιδανικά να ταυτιστεί) τον πραγματικό forward μετασχηματισμό $f : Z \to Y$.

Αφού υπολογιστεί και διαδοθεί το σφάλμα στο δίκτυο για την ενημέρωση των παραμέτρων, πραγματοποιείται η αντίστροφη διάδοση στο επίπεδα του INN. Αυτό σημαίνει πως επιστρατεύεται η αντίστροφη μέθοδος των Coupling Layers (4.10), (4.11) ούτως ώστε για είσοδο το πραγματικό δείγμα y (το ίδιο με το οποίο συγκρίθηκε το $\hat{y}$ προηγουμένως) να υπολογιστεί η εκτίμηση $\hat{z}$. Το $\hat{z}$ συγκρίνεται με το πραγματικό z και υπολογίζεται η συνάρτηση απωλειών του backward μετασχηματισμού ως:

$$L_z = \text{MSE}(\hat{z}, z) \tag{4.14}$$

και κατόπιν διαδίδεται με Backpropagation για την ενημέρωση των παραμέτρων του δικτύου. Η ελαχιστοποιήση αυτής της συνάρτησης, οδηγεί στην αξιόπιστη προσέγγιση του αντίστροφου μετασχηματισμού $f^{-1} : Y \to Z$ από την backward μέθοδο του INN. Το γεγονός ότι συνδυάζονται οι L_y και L_z επιτρέπει στο INN να συγκλίνει ταχύτερα αλλά και να επιτύχει βελτιωμένες αποδόσεις αφού η προσέγγιση και των δύο μετασχηματισμών υποβοηθάται από την ελαχιστοποίηση των δύο συναρτήσεων αλλά και από την ενσωμάτωση της Ιακωβιανής ορίζουσας ως στόχο εκπαίδευσης.

Για την αξιολόγηση του μοντέλου, το εκπαιδευμένο πλέον INN δέχεται ως είσοδο το διάνυσμα του φορτίου ζεύξης y, παράγει μέσω της αντίστροφης μεθόδου την εκτίμηση

$\hat{z}$ και μέσω του Αποκωδικοποιητή επαναφέρει το δείγμα στον χώρο δεδομένων όπου και προκύπτει η τελική εκτίμηση $\hat{x}$.

Η ροή εργασίας που περιγράφηκε ενσωματώνει, για την αύξηση της αναπαραστατικής ικανότητας του μοντέλου, min-max scaling στα X, Y, Z σύνολα.

Ο τελευταίος τρόπος λειτουργίας ονομάζεται **INN Extended** και στηρίζεται στην επέκταση του διανύσματος των φορτίων ζεύξεων y μέσω ενός διανύσματος w του οποίου οι τιμές λαμβάνονται από Γκαουσιανή κατανομή. Ο πίνακας κίνησης x με διάσταση $n \times n$ μετασχηματίζεται μέσω του Κωδικοποιητή σε ένα λανθάνον διάνυσμα z διάστασης d η οποία είναι μεγαλύτερη από τη διάσταση των φορτίων των ζεύξεων έστω m, δηλαδή $d > m$.Όμως τα INN για να λειτουργήσουν απαιτούν η διάσταση της εισόδου να ταυτίζεται με την διάσταση της εξόδου. Αυτό σημαίνει πως ο forward μετασχηματισμός αντιστοιχεί τα διανύσματα του χώρου Z διάστασης d σε διανύσματα διάστασης επίσης d της επεκτεταμένης μορφής $[y, w]$ όπου $w \in \mathbb{R}^{d-m}$. Ο στόχος της ενσωμάτωσης του w είναι η κωδικοποίηση επιπρόσθετης πληροφορίας των αμφίδρομων μετασχηματισμών, η οποία αλλιώς θα χανόταν λόγω της μείωσης διάστασης.

Για την εκπαίδευση του μοντέλου ακολουθείται παρόμοια λογική με τον Original INN τρόπο λειτουργίας. Αρχικά, ένα δείγμα x μετασχηματίζεται στο z και μέσω της forward διάδοσης παράγεται ένα διάνυσμα $\hat{y}_e = [\hat{y}, \hat{w}]$ διάστασης d. Το $\hat{y}_e$ συγκρίνεται με την συνένωση του y με το w. Κάθε στοιχείο του w προκύπτει τυχαία από την Γκαουσιανή κατανομή με μέση τιμή και τυπική απόκλιση τα αντίστοιχα μεγέθη που εκφράζουν όλα τα διανύσματα y που ανήκουν στο σύνολο εκπαίδευσης:

$$f(w_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(w_i - \mu)^2}{2\sigma^2}\right) \text{για i = 1,2,...,d-m} \tag{4.15}$$

Ύστερα, υπολογίζεται η προς ελαχιστοποίηση συνάρτηση απωλειών του forward μετασχηματισμού:

$$L_y = w_1 * \text{MSE}(\hat{y}, y) + w_2 * \text{MSE}(\hat{w}, w) - c \cdot \log\det\left(\frac{\partial f(z)}{\partial z}\right) \tag{4.16}$$

όπου w_1, w_2 παράμετροι βάρους που βασίζονται στον λόγο των διαστάσεων των y, w. Ο πρώτος όρος επιδιώκει την ταύτιση του παραγόμενου $\hat{y}$ με την πραγματική μέτρηση φορτίου ζεύξης, ενώ ο δεύτερος όρος την κωδικοποίηση επιπρόσθετης πληροφορίας αλλά και την αντιστοίχηση των επεκτεταμένων στοιχείων του z σε στοιχεία που ακολουθούν την προσδιοριζομένη Γκαουσιανή κατανομή.

Στη συνέχεια, το πραγματικό y συνενώνεται με το ίδιο τυχαίο w και διαδίδεται μέσω της αντίστροφης μεθόδου του INN για την παραγωγή του $\hat{z}$ το οποίο επιθυμούμε να ελαχιστοποιεί την backward συνάρτηση απωλειών $L_z = \text{MSE}(\hat{z}, z)$. Για να διασφαλιστεί ότι το μοντέλο μαθαίνει την υποκείμενη Γκαουσιανή μεταβλητή που ακολουθεί η επέκταση w, διαφορετικά τυχαία δείγματα w επιλέγονται για κάθε εποχή εκπαίδευσης όπου χρησιμοποιείται το δείγμα x. Η συνολική διαδικασία επαναλαμβάνεται για όλα τα δείγματα του συνόλου εκπαίδευσης.

Για την εκτίμηση πίνακα κίνησης, η διαδικασία ακολουθεί τη λογική των μοντέλων παραγωγής συνθετικών δειγμάτων. Με δεδομένο ένα διάνυσμα φορτίων ζεύξεων y^*, στόχος είναι η εύρεση βέλτιστου w^* το οποίο κατόπιν συνένωσης με το y^* να παράγει το z^* και άρα το x^* που ελαχιστοποιεί την σχέση (2.3). Με άλλα λόγια η εκτίμηση εκφράζεται ως το πρόβλημα ελαχιστοποίησης:

$$\arg\min_{w} \left\| y^* - A \cdot d\left(f^{-1}([y^*, w])\right)\right\|_2^2 \tag{4.17}$$

όπου $d()$ αντιπροσωπεύει τον εκπαιδευμένο Αποκωδικοποιητή και f^{-1} την αντίστροφη μέθοδο του INN. Η διαδικασία αυτή μπορεί να επιταχυνθεί με την εύρεση ενός "καλού" σημείου w_0 ως το σημείο που ελαχιστοποιεί την συνάρτηση στόχο μεταξύ N τυχαία επιλεγμένων σημείων. Με άλλα λόγια το $w_0 = w_k$ ικανοποιεί την σχέση:

$$\left\| y^* - A \cdot d\left(f^{-1}([y^*, w_k])\right)\right\|_2^2 \leq \left\| y - A \cdot d\left(f^{-1}([y^*, w_i])\right)\right\|_2^2 \tag{4.18}$$

Για i από 1 μέχρι N.

Μετά την επιλογή του "καλού" αρχικού σημείου, εφαρμόζεται στοχαστική διαδικασία ελαχιστοποίησης χρησιμοποιώντας έναν βελτιστοποιητή όπως ο Adam [73] με στόχο τον καθορισμό του βέλτιστου $\hat{x}^* = d(f^{-1}([y^*, w^*]))$.

Και σε αυτόν τον τρόπο λειτουργίας, εφαρμόζεται min-max scaling κανονικοποίηση.

4.5 Συνάρτηση Ενεργοποίησης

Σε αυτή την ενότητα θα αναλύσουμε τις βασικές Συναρτήσεις Ενεργοποίησης δίνοντας έμφαση στα πλεονεκτήματα αλλά και τα μειονεκτήματα που παρουσιάζουν, με στόχο την αιτιολόγηση της επιλογής της καταλληλότερης συνάρτησης για την προτεινόμενη μεθοδολογία.

Η Συνάρτηση Ενεργοποίησης (Activation Function) αποτελεί μια καθοριστική οντότητα στην εκπαίδευση και στη λειτουργία των δικτύων βαθιάς Μηχανικής Μάθησης καθώς βελτιώνει την αναπαραστατική ικανότητα των μοντέλων. Κάθε νευρώνας του δικτύου υπολογίζει την τιμή που επρόκειτο να διαδώσει στους νευρώνες των ανώτερων στρωμάτων ως άθροισμα των γινομένων των παραμέτρων-βαρών με τις τιμές που λαμβάνει ως είσοδο. Το γεγονός αυτό μοντελοποιεί την σχέση της εισόδου με την έξοδο ως γραμμική και άρα περιορίζει το μοντέλο στην αποδοτική εκμάθηση αποκλειστικά γραμμικών συσχετίσεων των δεδομένων μεταξύ τους. Όμως στην πράξη τα δεδομένα εισόδου-εξόδου συνήθως χαρακτηρίζονται από μη γραμμικούς μετασχηματισμούς και άρα το μοντέλο οφείλει να ενσωματώσει μη γραμμικά συστατικά για την προσέγγιση και κωδικοποίηση αυτών των μετασχηματισμών. Η Συνάρτηση Ενεργοποίησης εφαρμόζεται ως επιπρόσθετο στάδιο στον υπολογισμό της τιμής για κάθε νευρώνα του δικτύου επιβάλλοντας μη γραμμικά χαρακτηριστικά στην τελική έξοδο που θα προωθηθεί στο δίκτυο.

Οι πρώτες μη γραμμικές Συναρτήσεις Ενεργοποίησης που αξιοποιήθηκαν σε νευρωνικά δίκτυα είναι η Σιγμοειδής (Sigmoid) [74] και η Υπερεκθετική (Tanh) [75] οι οποίες περιορίζουν την έξοδο σε εύρος τιμών [0,1] και [-1,1] αντίστοιχα. Το περιορισμένο εύρος των συναρτήσεων αυτών αλλά και το γεγονός ότι είναι παραγωγίσιμες καθιστά τις τεχνικές βελτιστοποίησης που βασίζονται σε κατάβαση κλίσης περισσότερο σταθερές και καλώς ορισμένες. Παρόλα αυτά για μεγάλο εύρος τιμών στο πεδίο της εισόδου, η έξοδος αυτών των συναρτήσεων οδηγείται σε κορεσμό και άρα η κλίση σε αυτά τα σημεία εμφανίζει πολύ χαμηλές τιμές (τείνει στο 0) επιβραδύνοντας σημαντικά την εκπαίδευση (σε κάθε επανάληψη η βελτίωση των βαρών είναι αμελητέα). Σε ένα δίκτυο πολλών επιπέδων η εμφάνιση σχεδόν μηδενικής κλίσης σε κάποιο από αυτά είναι μια ενέργεια χωρίς δυνατότητα ανάκαμψης, με την κλίση να μηδενίζεται ως την τελική έξοδο του δικτύου, με το φαινόμενο αυτό να είναι γνωστό ως Εξαφάνιση Κλίσης [76].

Για την αντιμετώπιση της Εξαφάνισης Κλίσης εφαρμόστηκαν Συναρτήσεις Ενεργοποίησης που δεν περιορίζουν το εύρος του συνόλου τιμών, θυσιάζοντας όμως την παραγωγισιμότητά τους. Η κύρια συνάρτηση που εμπίπτει σε αυτήν την κατηγορία και που πειραματικά εμφανίζει θεαματική βελτίωση σε σχέση με τις προαναφερθέντες συναρτήσεις είναι η ReLU [52]:

$$\text{ReLU}(x) = \max(0, x) \tag{4.19}$$

για είσοδο x. Η συνάρτηση αυτή λειτουργεί ως ταυτότητα για θετική είσοδο επιτρέποντας την πλήρη έκφραση του θετικού πεδίου ορισμού, μηδενίζεται όμως για αρνητική είσοδο γεγονός που της παρέχει την επιθυμητή μη γραμμικότητα. Επιπλέον είναι πολύ πιο εύκολη και άρα γρηγορότερη στον υπολογισμό σε σχέση με την Σιγμοειδή και την Υπερεκθετική. Το κύριο μειονέκτημα της ReLU όμως είναι ένα φαινόμενο που ονομάζεται Νέκρωση Νευρώνων [77]. Πιο συγκεκριμένα, εάν κατά την εκπαίδευση, όλες οι τιμές προ-ενεργοποίησης της ReLU μεταβούν σε ένα εύρος όπου η κλήση είναι 0 ανεξάρτητα της εισόδου (αυτό μπορεί να συμβεί άμα όλα τα βάρη του νευρώνα είναι αρνητικά ενώ η είσοδός του πάντα μη αρνητική ή αν ο ρυθμός μάθησης είναι υψηλός) η ReLU δεν θα πυροδοτηθεί ποτέ και άρα η κλίση του σφάλματος ως προς τα βάρη πριν τη ReLU θα είναι 0. Αυτό σημαίνει πως τα βάρη του νευρώνα δεν θα ενημερωθούν ποτέ ξανά και άρα ο νευρώνας "νεκρώνει".

Μια λύση που προτάθηκε για την αντιμετώπιση του φαινομένου είναι η ενσωμάτωση "διαρροής" για το αρνητικό τμήμα του πεδίου ορισμού. Η αναβάθμιση αυτή ονομάζεται leakyReLU [53] και εκφράζεται από την σχέση:

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0 \end{cases} \tag{4.20}$$

όπου $\alpha \in (0, 1)$ και ρυθμίζει την κλίση των τιμών που διαρρέονται. Με την προσθήκη της διαρροής, ο νευρώνας δεν μπορεί να παράξει μηδενική έξοδο ανεξαρτήτως εισόδου και άρα η κλίση του σφάλματος θα είναι διάφορη του μηδενός, επιτρέποντας έτσι την συνεχή ενημέρωση των παραμέτρων.

Μια συνάρτηση ενεργοποίησης που συνδυάζει τα πλεονεκτήματα όσων συναρτήσεων

παρουσιάστηκαν προηγουμένως και που θα χρησιμοποιηθεί στην εφαρμογή μας είναι η Συνάρτηση Ενεργοποίησης Swish [78]:

$$\text{Swish}(x) = x \cdot \sigma(\beta \cdot x) \tag{4.21}$$

όπου $\sigma()$ είναι η Σιγμοειδής συνάρτηση και η β είναι παράμετρος προς εκμάθηση:

$$\sigma(\beta \cdot x) = \frac{1}{1 + e^{-\beta \cdot x}} \tag{4.22}$$

Η Swish είναι μια ομαλή και παραγωγίσιμη συνάρτηση η οποία απελευθερώνει την κλασσική Σιγμοειδή από τον κορεσμό της εξόδου, πολλαπλασιάζοντάς την με την είσοδο x γεγονός που προσδίδει ευελιξία και επιτρέπει την αποδοτική κωδικοποίηση μεγάλου εύρους τιμών εισόδου. Με άλλα λόγια αν η είσοδος είναι εντοπισμένη σε χαμηλές απόλυτες τιμές, η Σιγμοειδής υπερισχύει ενώ αν οι τιμές είναι πολύ υψηλές, η Swish λειτουργεί όπως η ReLU (πρακτικά η Σιγμοειδής εκφυλίζεται στην ακραία τιμή της που είναι το 1 για θετικές τιμές και 0 για αρνητικές). Στην πράξη έχει επιβεβαιωθεί πειραματικά πως η Swish εμφανίζει πολύ καλύτερες επιδόσεις σε σχέση με τις leakyReLU και ReLU, μειώνοντας αισθητά το πρόβλημα των νεκρών νευρώνων αλλά και της Εξαφάνισης Κλίσης. Η γενικεύμενη Swish ενσωματώνει και την παράμετρο β η οποία μαθαίνεται ως παράμετρος βάρους από το νευρωνικό δίκτυο και άρα εμφανίζει διαφορετική τιμή ανά νευρώνα, κλιμακώνοντας κατάλληλα την Σιγμοειδή έτσι ώστε να μοντελοποιεί με τον βέλτιστο τρόπο την σχέση εισόδου-εξόδου.

Δεδομένου ότι οι ροές κίνησης του πίνακα κίνησης x επιβάλλεται να έχουν θετικές τιμές, στην εφαρμογή μας θα επιστρατεύσουμε την απόλυτη τιμή της Swish ως τελική Συνάρτηση Ενεργοποίησης γεγονός που εξασφαλίζει θετικότητα τόσο στην έξοδο όσο και στα ενδιάμεσα αποτελέσματα των επιπέδων της αρχιτεκτονικής.

4.6 Θέματα Αρχικοποίησης και Βελτιστοποίησης

Σε αυτήν την ενότητα παρουσιάζονται οι τεχνικές αρχικοποίησης των βαρών των νευρωνικών δικτύων αλλά και η οντότητα βελτιστοποίησης που θα χρησιμοποιηθούν στην ροή εργασίας για την εκπαίδευση των μοντέλων (INN και Αυτοκωδικοποιητής).

Η αρχικοποίηση των παραμέτρων/βαρών του νευρωνικού δικτύου συνιστά μία υψίστης σημασίας σχεδιαστική απόφαση, καθώς επηρεάζει την αποδοτικότητα την εκπαίδευσης των μοντέλων Βαθιάς Μηχανικής Μάθησης.

Μια πρώτη ιδέα για την αρχικοποίηση των βαρών είναι να οριστεί η ίδια αρχική τιμή για όλα τα βάρη του δικτύου. Αυτό σημαίνει πως όλοι οι νευρώνες του πρώτου κρυφού επιπέδου θα έχουν κοινή είσοδο και βάρη, παράγοντας έτσι ακριβώς την ίδια έξοδο και κατ' επέκταση το ίδιο σφάλμα το οποίο θα ενημερώσει όλες τις παραμέτρους με ομοιόμορφο τρόπο. Η συμπεριφορά αυτή θα διαδοθεί στα ανώτερα στρώματα του δικτύου και άρα το

μοντέλο εκφυλίζεται αποκτώντας πλήρη συμμετρία. Με άλλα λόγια η κοινή αρχικοποίηση οδηγεί όλους τους νευρώνες να έχουν την ίδια συμπεριφορά και άρα να κωδικοποιούν την ίδια πληροφορία, γεγονός που ελαττώνει την ικανότητα μάθησης του μοντέλου.

Μια καλύτερη προσέγγιση είναι η τυχαία αρχικοποίηση των βαρών, η οποία καταπολεμά την συμμετρία του δικτύου και διευρύνει την εκφραστικότητα του μοντέλου. Και αυτή η μέθοδος όμως απαιτεί προσοχή καθώς δύναται να προκύψει αρχικοποίηση που να οδηγεί τόσο σε Εξαφάνιση [76] όσο και σε Έκρηξη [79] (οι κλίσεις μεγεθύνονται συνεχώς κατά την διάδοση από επίπεδο σε επίπεδο μέχρις ότου να είναι μη διαχειρίσιμες από μια βαθιά αρχιτεκτονική) των κλίσεων του νευρωνικού δικτύου, εμποδίζοντας την εκπαίδευση.

Στην εργασία τους, οι Glorot και Bengio [80] προτείνουν ότι η διακύμανση της εξόδου κάθε επιπέδου πρέπει να ισούται με την διακύμανση της εισόδου έτσι ώστε να αποφευχθεί τόσο ο κορεσμός όσο και η εξαφάνιση του σήματος. Το ίδιο θα πρέπει να ισχύει και για τη διακύμανση των κλίσεων πριν και μετά την διάδοσή τους από το επίπεδο. Για την εξασφάλιση αυτής της συνθήκης ορίζονται τα μεγέθη fan_{in}, fan_{out} ως ο αριθμός εισόδων και εξόδων αντίστοιχα για κάθε επίπεδο του νευρωνικού δικτύου και υπολογίζεται η ποσότητα $fan_{avg} = (fan_{in} + fan_{out})/2$. Το μέγεθος αυτό χρησιμοποιείται για την τυχαία αρχικοποίηση των βαρών του δικτύου μέσω της Γκαουσιανής κατανομής με μέση τιμή 0 και διακύμανση $\sigma^2 = \frac{1}{fan_{avg}}$, με την μέθοδο αυτή να είναι γνωστή ως Αρχικοποίηση Xavier. Η συγκεκριμένη τεχνική αρχικοποίησης αποδίδει καλά αν ως συνάρτηση ενεργοποίησης στα επίπεδα των νευρωνικών δικτύων χρησιμοποιείται η Σιγμοειδής.

Δεδομένου ότι, όπως αναφέρθηκε στην προηγούμενη ενότητα, για την ροή εργασίας έχει επιλεχθεί το απόλυτο της Swish ενεργοποίησης, η μέθοδος των αρχικοποίησης των βαρών πρέπει να προσαρμοστεί. Για αυτόν τον λόγο, θα επιστρατευθεί η Αρχικοποίηση He [81] η οποία έχει σχεδιαστεί ειδικά για την ReLU αλλά και τις παραλλαγές-βελτιώσεις της όπως η Swish, γεγονός που επαληθεύεται εμπειρικά σε πολλές μελέτες. Η αρχικοποίηση He επιλέγει τυχαία την αρχική τιμή των παραμέτρων σε κάθε επίπεδο μέσω της Γκαουσιανής κατανομής με μέση τιμή 0 και διακύμανση $\sigma^2 = \frac{2}{fan_{in}}$.

Για την εκπαίδευση ενός μοντέλου Νευρωνικών Δικτύων, εκτός από την ”έξυπνη” επιλογή της αρχικής τιμής των παραμέτρων, εξίσου σημαντικός παράγοντας που καθορίζει την αποτελεσματικότητα της διαδικασίας είναι η κατάλληλη επιλογή του βελτιστοποιητή. Ο βελτιστοποιητής, είναι υπεύθυνος για την ενημέρωση των παραμέτρων του δικτύου προς την κατεύθυνση που ελαχιστοποιεί την συνάρτηση απωλειών, δηλαδή τον στόχο εκπαίδευσης. Οι βελτιστοποιητές χρησιμοποιούν τις κλίσεις που υπολογίζονται μέσω του Backpropagation ούτως ώστε να ενημερώσουν επαναληπτικά τα βάρη των νευρώνων για την ελαχιστοποίηση του συνολικού σφάλματος.

Η βασικότερη τεχνική βελτιστοποίησης ονομάζεται Κατάβαση Κλίσης (Gradient Descent) [82] και είναι ένας επαναληπτικός αλγόριθμος πρώτης τάξης για την ελαχιστοποίηση μια παραγωγίσιμης συνάρτησης στόχου, δηλαδή της συνάρτησης απωλειών. Σε κάθε επανάληψη, υπολογίζεται η πρώτης τάξης παράγωγος της συνάρτησης απωλειών σε σχέση με τις παραμέτρους του νευρωνικού, οι οποίες ενημερώνονται προς την αντίθετη κατεύ-

θυνση από αυτήν προς την οποία αυξάνεται η κλίση. Η Κατάβαση Κλίσης δίνεται από την σχέση:

$$\theta_t = \theta_{t-1} - \eta \nabla_\theta J(\theta) \tag{4.23}$$

όπου θ_{t-1} οι προηγούμενες τιμές των παραμέτρων του δικτύου, η μια παράμετρος που καθορίζει την κλιμάκωση της ενημέρωσης των παραμέτρων σε κάθε επανάληψη, γνωστή ως ρυθμός μάθησης και $\nabla_\theta J(\theta)$ η πρώτης τάξης παράγωγος της συνάρτησης απωλειών. Ο ρυθμός μάθησης αν έχει χαμηλή τιμή οδηγεί σε χαμηλής κατ' απόλυτη τιμή ενημερώσεις στις παραμέτρους του δικτύου, απαιτώντας περισσότερες επαναλήψεις για την εύρεση του ελαχίστου. Αν όμως έχει μεγαλύτερη τιμή, τότε η ενημερώσεις είναι πιο επιδραστικές και η σύγκληση είναι πιο γρήγορη, με κίνδυνο όμως την εμφάνιση ταλαντώσεων. Η εφαρμογή της ενημέρωσης των βαρών μπορεί να πραγματοποιηθεί αφού υπολογιστεί η έξοδος του δικτύου για όλα τα δείγματα εκπαίδευσης (Batch Gradient Descent), μετά από κάθε δείγμα του συνόλου εκπαίδευσης (Stochastic Gradient Descent) αλλά και μετά από μικρές ομάδες δεδομένων εκπαίδευσης (Mini-batch Gradient Descent).

Η Κατάβαση Κλίσης, ενημερώνει τις παραμέτρους του δικτύου με βάση την υπολογιζόμενη παράγωγο της συνάρτησης απωλειών για της εκάστοτε επανάληψη, αγνοώντας την συμπεριφορά των παραγώγων των προηγούμενων επαναλήψεων. Πάνω σε αυτήν την αδυναμία, στηρίζεται η ανάπτυξη του βελτιστοποιητή ορμής (Momentum) [83] ο οποίος σε κάθε επανάληψη ενημερώνει τις παραμέτρους του μοντέλου, προσθέτοντας ένα διάνυσμα ορμής. Το διάνυσμα ορμής ενημερώνεται με βάση την πρώτης τάξης παράγωγο της συνάρτησης απωλειών. Ο αλγόριθμος ενημέρωσης συνοψίζεται στην ακόλουθη σχέση:

$$\mathbf{m} \leftarrow \beta \mathbf{m} - \eta \nabla_\theta J(\theta) \tag{4.24}$$

$$\theta \leftarrow \theta + \mathbf{m} \tag{4.25}$$

Στην προτεινόμενη μέθοδο θα χρησιμοποιηθεί ο Adam (Adaptive Moment Estimation) [73] για την ενημέρωση των παραμέτρων των μοντέλων. Το Adam είναι μια μέθοδος βελτιστοποίησης που υπολογίζει μεμονωμένους προσαρμοστικούς ρυθμούς μάθησης για διαφορετικές παραμέτρους μέσω εκτιμήσεων των ορμών των κλίσεων πρώτης και δεύτερης τάξης. Για την επίτευξη αυτού του στόχου, το Adam διατηρεί εκθετικά αποσβεννύμενους μέσους όρους παλαιότερων κλίσεων αλλά και τετραγώνων των κλίσεων. Ενσωματώνει επίσης εκτιμήσεις πρώτων και δεύτερων ορμών με διόρθωση μέσω μεροληψίας. Κατόπιν πειραματικής αξιολόγησης, το Adam παρουσιάζει πολύ καλύτερες επιδόσεις σε σχέση με την Κατάβαση Κλίσης αναφορικά με την ταχύτητα σύγκλισης αλλά και την ανθεκτικότητα.

Κεφάλαιο 5

Υλοποίηση Μεθόδων & Πειραματική Αξιολόγηση

5.1 Σύνολο Δεδομένων

Για την πειραματική αξιολόγηση της προτεινόμενης μεθόδου, θα χρησιμοποιηθεί το δημοσίως διαθέσιμο σύνολο δεδομένων από το δίκτυο Abilene [12], το οποίο είναι ένα δίκτυο κορμού των Ηνωμένων Πολιτειών της Αμερικής και αφορά την σύνδεση μεγάλων πανεπιστημιακών ιδρυμάτων (Σχήμα 5.1).

Σχήμα 5.1: Τοπολογία Abilene Δικτύου. Πηγή [12]

Το δίκτυο αυτό αποτελείται από 12 κύριους κόμβους (με δύο εξ αυτών να βρίσκονται

στην Atlanta) οι οποίοι συνδέονται με 30 εσωτερικές ζεύξεις. Κάθε κόμβος διαθέτει επίσης δύο επιπρόσθετες ζεύξεις, μία για την εισερχόμενη και μια για την εξερχόμενη κίνηση οι οποίες τον συνδέουν με εξωτερικούς κόμβους (εκτός του Abilene). Κάθε εσωτερική ζεύξη έχει χωρητικότητα 9920000 kbps, εκτός από την ζεύξη Atlanta-Indianopolis που έχει χωρητικότητα 2480000 kbps. Το σύνολο δεδομένων περιέχει μέσους όρους μετρήσεων των πινάκων κίνησης σε διαστήματα των 5 λεπτών για 24 εβδομάδες ξεκινώντας από 1η Μαρτίου έως τη 10η Σεπτεμβρίου του 2004. Οι πίνακες κίνησης δεν είναι συμμετρικοί (στη θέση i, j καταγράφονται τα δεδομένα που με πηγή τον κόμβο i έχουν προορισμό τον j, κίνηση που δεν είναι απαραίτητο να είναι αμφίδρομη) και η διαγώνιος εν γένει χαρακτηρίζεται από μη μηδενικά στοιχεία καθώς μοντελοποιεί την κίνηση από το εξωτερικό δίκτυο που εισέρχεται και εξέρχεται από τον ίδιο κόμβο του Abilene.

Για την επεξεργασία του συνόλου δεδομένων από την προτεινόμενη μέθοδο, εξάγονται οι πίνακες κίνησης x και ο πίνακας δρομολόγησης A. Πιο συγκεκριμένα, το σύνολο δεδομένων περιέχει, για κάθε διάστημα μέτρησης, πέντε διαφορετικές εκδοχές του πίνακα κίνησης εκ των οποίων η πρώτη αφορά τις πραγματικές τιμές ενώ οι υπόλοιπες αφορούν τα αποτελέσματα μοντέλων εκτίμησης. Για την πειραματική αξιολόγηση θα χρησιμοποιηθούν αποκλειστικά οι 144 τιμές (12×12 ζεύγη), οι οποίες ελέγχονται για σφάλματα. Ειδικότερα αν μια ροή κίνησης έχει τιμή μεγαλύτερη από την χωρητικότητα των ζεύξεων που ενώνουν τους δύο κόμβους, η τιμή της αντικαθίσταται από την μικρότερη χωρητικότητα του μονοπατιού. Επιπλέον οι τιμές όλων των ροών κίνησης μετατρέπονται σε Mbps. Ο πίνακας δρομολόγησης A περιλαμβάνεται σε ξεχωριστό αρχείο και αποτελεί έναν πίνακα με τιμές 0 ή 1 και διαστάσεις 30×144. Ο πίνακας έχει τιμή 1 στη θέση (i, j) αν η κίνηση της ροής του j-στου ζεύγους διέρχεται από την ζεύξη i, αλλιώς έχει τιμή 0. Τα διανύσματα των φορτίων ζεύξης y διάστασης 30×1 δεν περιέχονται στο σύνολο δεδομένων, μπορούν όμως να ανακτηθούν με πολλαπλασιασμό του πίνακα δρομολόγησης με το διάνυσμα διάστασης 144×1 που εκφράζει τον πίνακα δρομολόγησης.

Για την πειραματική αξιολόγηση, θα χρησιμοποιηθούν οι πίνακες κίνησης 14 συνεχόμενων εβδομάδων και συγκεκριμένα από την 6η έως την 19η εβδομάδα. Οι πίνακες αυτοί διαιρούνται σε δύο σύνολα, το σύνολο εκπαίδευσης και το σύνολο ελέγχου. Το σύνολο εκπαίδευσης περιέχει το πρώτο 80% των δειγμάτων (22579) ενώ το σύνολο ελέγχου περιέχει το επόμενο χρονικά 20% (5645) των πινάκων κίνησης.

5.2 Υλοποίηση

Σε αυτήν την ενότητα παρουσιάζονται αναλυτικά οι αρχιτεκτονικές των νευρωνικών δικτύων που θα χρησιμοποιηθούν για την πειραματική αξιολόγηση της μεθόδου.

Συνελικτικός Αυτοκωδικοποιητής

Η αρχιτεκτονική του Συνελικτικού Αυτοκωδικοποιητή παρουσιάζεται στους πίνακες 5.1 και 5.2. Το μοντέλο αποτελείται από δύο νευρωνικά δίκτυα, τον Κωδικοποιητή και τον Αποκωδικοποιητή.

Ο Κωδικοποιητής λαμβάνει ως είσοδο έναν πίνακα κίνησης x με διαστάσεις 12×12 δημιουργώντας μια επιπλέον διάσταση καναλιών. Η νέα είσοδος είναι της μορφής $12 \times 12 \times 1$. Στη συνέχεια η είσοδος τροφοδοτεί 3 συνελικτικά νευρωνικά επίπεδα τα οποία μειώνουν τις χωρικές διαστάσεις του πίνακα και συγχρόνως διευρύνουν το πλήθος των feature maps. Η έξοδος της συνελικτικής φάσης μετατρέπεται σε διάνυσμα μίας διάστασης (Flatten) και μέσω ενός γραμμικού επιπέδου παράγεται η τελική κωδικοποίηση. Για τους **INN Baseline** και **INN Original** η τελική έξοδος θα έχει διάσταση $d = 30$ ενώ στο πλαίσιο του **INN Extended** επιλέγεται η διάσταση $d = 40$.

Ο Αποκωδικοποιητής πραγματοποιεί την αντίστροφη διαδικασία, λαμβάνοντας ως είσοδο ένα διάνυσμα διάστασης 30 ή 40 αντίστοιχα και το επαναφέρει στην αρχική διάσταση 12×12 μέσω αντίστροφων συνελικτικών επιπέδων.

Στη συνέχεια παρατίθενται οι οντότητες και οι υπερπαράμετροι που χρησιμοποιήθηκαν για την εκπαίδευση:

- Ως συνάρτηση ενεργοποίησης ορίζεται η απόλυτη τιμή της Swish όπως περιγράφηκε στο 4.5.
- Για την ενημέρωση των παραμέτρων επιλέγεται ο βελτιστοποιητής Adam με ρυθμό μάθησης 0.001.
- Για την αρχικοποίηση των βαρών επιλέγεται η αρχικοποίηση He.
- Το μέγεθος του batch είναι 128.
- Ως συνάρτηση απωλειών επιλέγεται το Huber Loss.
- Ο μέγιστος αριθμός εποχών εκπαίδευσης είναι 400.
- Πραγματοποιείται Early Stopping, δηλαδή η εκπαίδευση σταματά αν δεν υπάρχει βελτίωση στην απόδοση του μοντέλου για έναν αριθμό διαδοχικών εποχών. Ο αριθμός αυτός καθορίζεται από την υπερπαράμετρο patience που λαμβάνει την τιμή 30.
- Για τα συνελικτικά επίπεδα χρησιμοποιείται πυρήνας μεγέθους (3×3), βήμα 1 και δεν γίνεται χρήση συμπλήρωσης.

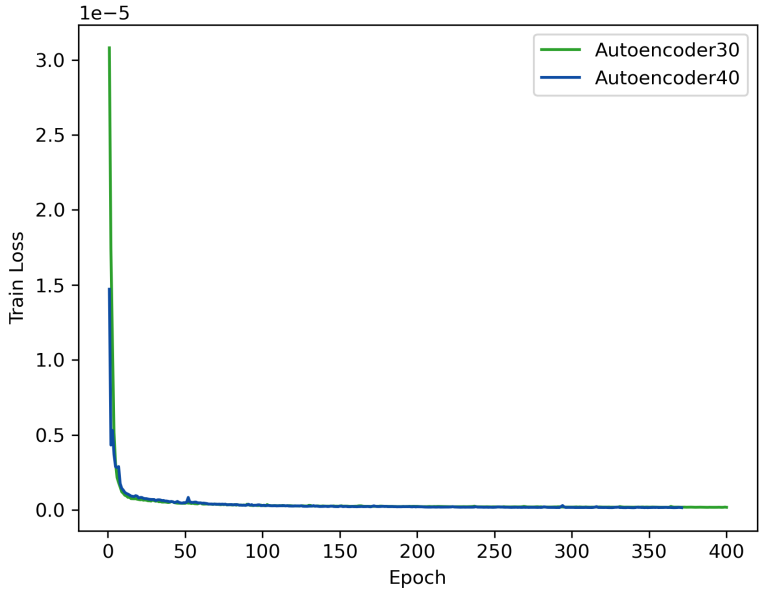
Πίνακας 5.1: Αρχιτεκτονική Κωδικοποιητή

ENCODER	
Layer	Output
Input	(batch, 12, 12, 1)
Conv2D	(batch, 10, 10, 32)
Conv2D	(batch, 8, 8, 64)
Conv2D	(batch, 6, 6, 128)
Flatten	(batch, 4608)
Linear	(batch, 30 ή 40)

Πίνακας 5.2: Αρχιτεκτονική Αποκωδικοποιητή

DECODER	
Layer	Output
Input	(batch, 30 ή 40)
Linear	(batch, 4608)
Unflatten	(batch, 6, 6, 128)
ConvTranspose2D	(batch, 8, 8, 64)
ConvTranspose2D	(batch, 10, 10, 32)
ConvTranspose2D	(batch, 12, 12, 1)

Στο επόμενο διάγραμμα 5.2 παρουσιάζεται η εξέλιξη των συναρτήσεων απωλειών για τους δύο Αυτοκωδικοποιητές $(d = 30, d = 40)$. Παρατηρούμε ότι ο Αυτοκωδικοποιητής 30 εξαντλεί το διαθέσιμο πλήθος εποχών εκπαίδευσης, ενώ ο Αυτοκωδικοποιητής 40 εκτελεί λιγότερες λόγω Early Stopping.



Σχήμα 5.2: Απώλειες Αυτοκωδικοποιητών κατά την εκπαίδευση.

Αντιστρέψιμο Νευρωνικό Δίκτυο Το INN νευρωνικό δίκτυο που θα χρησιμοποιηθεί και στους τρεις τρόπους λειτουργίας αποτελείται από 4 διαδοχικά αφινικά Coupling Layers. Οι s, t μετασχηματισμοί, μοντελοποιούνται από δύο συνελικτικά νευρωνικά δίκτυα, ίδιας αρχιτεκτονικής. Ειδικότερα, η είσοδος στα s, t διάστασης $d/2$ τροφοδοτεί μια σειρά από συνελικτικά επίπεδα μίας διάστασης Conv1d, διατηρώντας τις χωρικές διαστάσεις αμετάβλητες (χρήση συμπλήρωσης). Η αρχιτεκτονική των s, t παρουσιάζεται αναλυτικά στον ακόλουθο πίνακα 5.3:

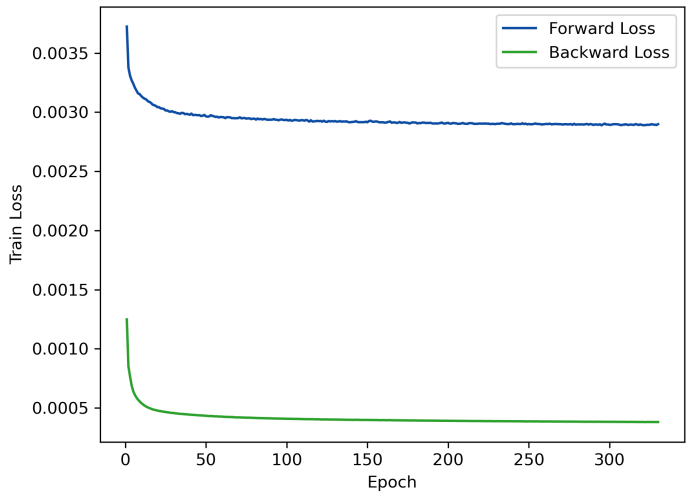
Πίνακας 5.3: Μετασχηματισμοί s, t

Mappings s, t	
Layer	Output
Input	(batch, 15 ή 20)
Unsqueeze	(batch, 15, 1)
Conv1D	(batch, 15, 32)
Conv1D	(batch, 15, 64)
Flatten	(batch, 960)
Linear	(batch, 15)

Στη συνέχεια παρατίθενται οι οντότητες και οι υπερπαράμετροι που χρησιμοποιήθηκαν για την εκπαίδευση:

- Ως συνάρτηση ενεργοποίησης ορίζεται η απόλυτη τιμή της Swish όπως περιγράφηκε στο Κεφάλαιο 4.5.
- Για την ενημέρωση των παραμέτρων επιλέγεται ο βελτιστοποιητής Adam με ρυθμό μάθησης 0.001.
- Για την αρχικοποίηση των βαρών επιλέγεται η αρχικοποίηση He.
- Το μέγεθος του batch είναι 128.
- Ως συναρτήσεις απωλειών χρησιμοποιούνται οι forward και backward losses όπως ορίστηκαν στο Κεφάλαιο 4.4 με $w_1 = 3/4$, $w_2 = 1/4$ και $c = 10^{-6}$.
- Ο μέγιστος αριθμός εποχών εκπαίδευσης είναι 200, 200 και 300 για τα **INN Baseline**, **INN Original** και **INN Extended** αντίστοιχα.
- Πραγματοποιείται Early Stopping με patience 7.
- Για τα συνελικτικά επίπεδα χρησιμοποιείται πυρήνας μεγέθους (3×3), βήμα 1 και χρήση συμπλήρωσης.

Στο ακόλουθο διάγραμμα 5.3 παρατίθεται ενδεικτικά η εξέλιξη τόσο της forward όσο και της backward συνάρτησης απωλειών για τον **INN Extended** τρόπο λειτουργίας.



Σχήμα 5.3: Forward και Backward απώλειες για INN Extended κατά την εκπαίδευση.

Τα μοντέλα αναπτύχθηκαν, εκπαιδεύτηκαν και αξιολογήθηκαν μέσω του προγραμματιστικού περιβάλλοντος PyTorch [84], με τον κώδικα να οργανώνεται σε Jupyter Notebooks [85]. Περισσότερες λεπτομέρειες αναφέρονται στο Παράρτημα Α.

5.3 Αποτελέσματα και Συζήτηση

Σε αυτήν την ενότητα παρουσιάζεται η πειραματική αξιολόγηση των προτεινόμενων μεθόδων για την Εκτίμηση Πίνακα Κίνησης. Οι μετρικές με τις οποίες θα εκτιμηθεί η απόδοση των μοντέλων είναι οι εξής:

- Root Mean Square Error (RMSE):

$$\text{RMSE}(t) = \frac{\|\hat{x}_t - x_t\|_2}{\sqrt{p}} = \sqrt{\frac{1}{p}\sum_{i=1}^{p}(\hat{x}_t(i) - x_t(i))^2}. \tag{5.1}$$

- Normalized Mean Absolute Error (NMAE):

$$\text{NMAE}(t) = \frac{\|\hat{x}_t - x_t\|_1}{\|x_t\|_1} = \frac{\sum_{i=1}^{p}|\hat{x}_t(i) - x_t(i)|}{\sum_{i=1}^{p}|x_t(i)|}. \tag{5.2}$$

- Spatial Relative Error (SRE):

$$\text{SRE}(i) = \frac{\|\hat{x}_{1:N}(i) - x_{1:N}(i)\|_2}{\|x_{1:N}(i)\|_2} = \frac{\sqrt{\sum_{t=1}^{N}(\hat{x}_t(i) - x_t(i))^2}}{\sqrt{\sum_{t=1}^{N}(x_t(i))^2}}. \tag{5.3}$$

- Temporal Relative Error (TRE):

$$\text{TRE}(t) = \frac{\|\hat{x}_t - x_t\|_2}{\|x_t\|_2} = \frac{\sqrt{\sum_{i=1}^{p}(\hat{x}_t(i) - x_t(i))^2}}{\sqrt{\sum_{i=1}^{p}(x_t(i))^2}}. \tag{5.4}$$

όπου $p \equiv n^2$ είναι ο αριθμός των ροών κίνησης, $i = 1, \ldots, p$ περιγράφει κάθε ροή κίνησης, $t = 1, \ldots, N$, δηλώνει κάθε χρονική στιγμή στο σύνολο ελέγχου, x_t είναι ο πραγματικός πίνακας κίνησης (ground truth) σε κάθε χρονικό βήμα ενώ $\hat{x}_t$ είναι αντίστοιχη εκτίμηση του μοντέλου. Το $x_{1:N}(i)$ είναι η ακολουθία των ground truth τιμών για την ροή κίνησης i με εύρος όλες τις χρονικές στιγμές του συνόλου ελέγχου, ενώ $\hat{x}_{1:N}(i)$ είναι αντίστοιχη ακολουθία για τις εκτιμώμενες τιμές. Τέλος τα $\|\cdot\|_1$ και $\|\cdot\|_2$ ισοδυναμούν με την απόλυτη και την Ευκλείδεια νόρμα αντίστοιχα.

Οι μετρικές RMSE και NMAE μοντελοποιούν την μέση απόκλιση των εκτιμώμενων ροών κίνησης από τις πραγματικές τους τιμές, η SRE εκφράζει το σχετικό σφάλμα εκτίμησης για κάθε ροή κίνησης με εύρος όλες τις χρονικές στιγμές του συνόλου ελέγχου ενώ η TRE το σχετικό σφάλμα μεταξύ όλων ρων ροών για μία χρονική στιγμή.

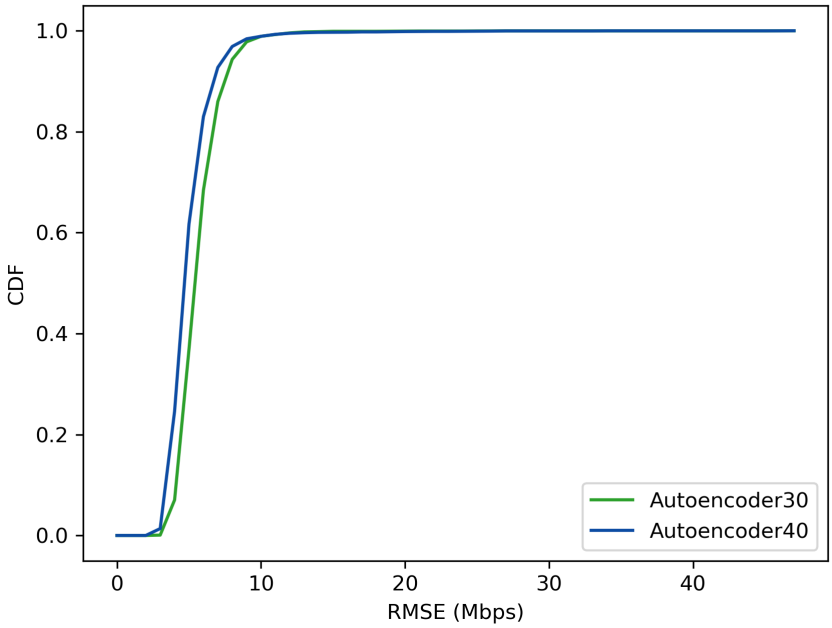
Ως πρώτο δομικό στοιχείο της ροής εργασίας για τις προτεινόμενες μεθόδους, είναι η εκπαίδευση των Αυτοκωδικοποιητών για διαστάσεις κωδικοποίησης $d = 30$ και $d = 40$ για τους τρόπους λειτουργίας (**INN Baseline**,**INN Original**) και **INN Extended** αντίστοιχα. Κατόπιν εκπαίδευσης, τα δύο μοντέλα αξιολογούνται πάνω στο σύνολο ελέγχου και με τις τέσσερις μετρικές. Στον ακόλουθο Πίνακα 5.4 παρουσιάζεται η μέση τιμή, ο διάμεσος και η τυπική απόκλιση των σφαλμάτων για τους δύο Αυτοκωδικοποιητές και για όλες τις μετρικές.

Πίνακας 5.4: Πίνακας σφαλμάτων εκτίμησης Αυτοκωδικοποιητών

Autoencoder $d = 30$			
Error	Mean	Median	Std
RMSE(Mbps)	5.6244	5.3556	1.5479
NMAE	0.1864	0.1811	0.0363
TRE	0.1469	0.1444	0.0367
SRE	0.8206	0.4959	1.3535
Autoencoder $d = 40$			
Error	Mean	Median	Std
RMSE(Mbps)	4.9733	4.6353	1.7099
NMAE	0.1639	0.1589	0.0338
TRE	0.1289	0.1253	0.0351
SRE	0.7529	0.4639	1.1859

Για τη διευκόλυνση της γραφικής αναπαράστασης των σφαλμάτων των προς αξιολόγηση μοντέλων ορίζεται η έννοια της Αθροιστικής Συνάρτησης Κατανομής (Cumulative Distribution Function (CDF)) [41]. Για δοσμένο σύνολο n σημείων $y_1 \leq y_2 \leq \cdots \leq y_n$ η CDF είναι μια συνάρτηση βήματος, η τιμή της οποίας αυξάνεται κατά $\frac{1}{n}$ κάθε φορά που ”συναντά” ένα σημείο. Με άλλα λόγια η τιμή της για ένα σημείο x είναι το κλάσμα των συνολικών παρατηρήσεων με τιμή μικρότερη ή ίση του σημείου x.

Στο ακόλουθο διάγραμμα 5.4 παρουσιάζονται οι CDF κατανομές για το RMSE σφάλμα των δύο Αυτοκωδικοποιητών:



Σχήμα 5.4: RMSE CDF Αυτοκωδικοποιητών για $d = 30$ και $d = 40$

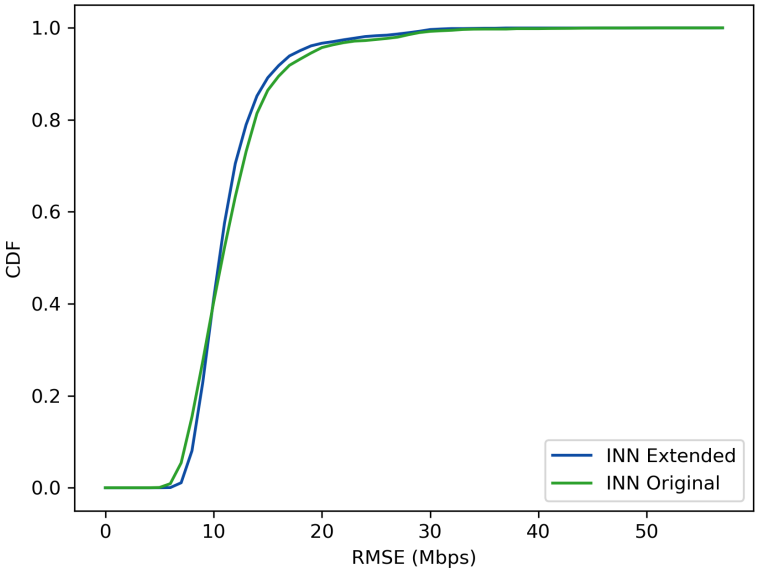
Με βάση τα παραπάνω αποτελέσματα, παρατηρούμε πως η αύξηση της διάστασης στον λανθάνοντα χώρο οδηγεί τον Autencoder40 σε πολύ καλύτερες επιδόσεις σε σχέση με τον Autencoder30. Πιο συγκεκριμένα, ο Αυτοκωδικοποιητής για διάσταση $d = 40$ παρουσιάζει μειωμένες μέσες τιμές σε όλες τις μετρικές σφάλματος. Επιπλέον η CDF του Autencoder40 εμφανίζει αυξημένες τιμές σε σχέση με την CDF του Autencoder30 σε όλο το εύρος των RMSE σφαλμάτων. Αυτό σημαίνει ότι για κάθε πιθανή τιμή του σφάλματος z, ο Autoencoder40 έχει περισσότερα δείγματα από τον Autoencoder30 με τιμή μικρότερη ή ίση του z γεγονός που υποδεικνύει υπεροχή σε όλο το εύρος των RMSE. Η συμπεριφορά αυτή είναι αναμενόμενη καθώς οι Αυτοκωδικοποιητές πραγματοποιούν ελάττωση διάστασης, μια διαδικασία με έμφυτη απώλεια πληροφορίας η οποία μεγιστοποιείται όσο μειώνεται η διάσταση του λανθάνοντος χώρου.

Το επόμενο τμήμα της ροής εργασίας αφορά την εκπαίδευση και την αξιολόγηση των INN μοντέλων κατόπιν μετασχηματισμού των πινάκων κίνησης, σε απεικονίσεις του λανθάνοντος χώρου Z. Μετά το στάδιο της εκπαίδευσης, τα μοντέλα των τριών τρόπων λειτουργίας του INN τροφοδοτούνται με φορτία ζεύξης y από το σύνολο ελέγχου παράγοντας την τελική εκτίμηση του πίνακα κίνησης. Η έξοδός τους συγκρίνεται με την πραγματική τιμή του πίνακα κίνησης και το σφάλμα υπολογίζεται μέσω των μετρικών RMSE, NMAE, SRE, TRE. Στον ακόλουθο Πίνακα 5.5 παρουσιάζονται η μέση τιμή, ο διάμεσος και η τυπική απόκλιση για τις τέσσερις μετρικές και για κάθε τρόπο λειτουργίας του INN.

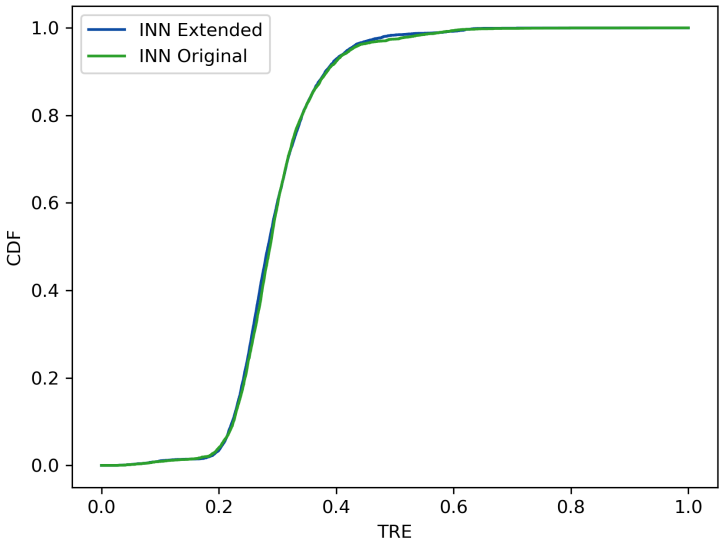
Πίνακας 5.5: Πίνακας σφαλμάτων INN μοντέλων

INN Baseline			
Error	Mean	Median	Std
RMSE(Mbps)	12.0571	10.6099	8.8102
NMAE	0.3210	0.3070	0.0827
TRE	0.2954	0.2835	0.0685
SRE	1.1545	0.6268	1.9876
INN Original			
Error	Mean	Median	Std
RMSE(Mbps)	11.6602	10.8057	4.4497
NMAE	0.3078	0.2964	0.0605
TRE	0.2973	0.2874	0.0763
SRE	0.7765	0.5488	0.8257
INN Extended			
Error	Mean	Median	Std
RMSE(Mbps)	11.3943	10.5109	3.7562
NMAE	0.3141	0.3040	0.0635
TRE	0.2949	0.2838	0.0742
SRE	0.8080	0.5580	1.0561

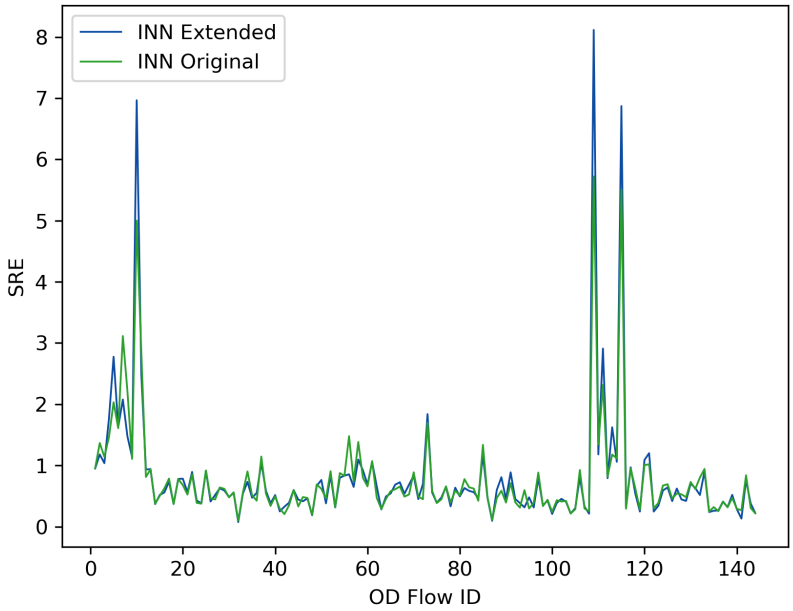
Επιπλέον για τους **INN Original** και **INN Extended** τρόπους λειτουργίας παρουσιάζονται τα CDF διαγράμματα για τα σφάλματα RMSE, TRE αλλά και το διάγραμμα του SRE σφάλματος για τις 144 διαφορετικές ροές κίνησης (διαγράμματα 5.5, 5.6 και 5.7).



Σχήμα 5.5: RMSE CDF για INN Original και INN Extended



Σχήμα 5.6: TRE CDF για INN Original και INN Extended



Σχήμα 5.7: SRE για INN Original και INN Extended

Αρχικά παρατηρούμε πως το **INN Baseline** εμφανίζει αισθητά μειωμένη απόδοση για τις μετρικές RMSE, NMAE και SRE, και σχεδόν ίδια συμπεριφορά για την μετρική TRE με τους **INN Original**, **INN Extended** τρόπους λειτουργίας. Η μεγαλύτερη απόκλιση εντοπίζεται στις μετρικές RMSE και SRE, με το **INN Baseline** όχι μόνο να χαρακτηρίζεται από σημαντικά μεγαλύτερη μέση τιμή του σφάλματος, αλλά και από πολύ υψηλές τιμές στις αντίστοιχες τυπικές αποκλίσεις. Η παρατήρηση αυτή οδηγεί στο συμπέρασμα πως ο βασικός τρόπος λειτουργίας παρουσιάζει πολύ μεγάλο εύρος σφαλμάτων κατά την εκτίμηση των πινάκων κίνησης και άρα έχει μειωμένη σταθερότητα σε σύγκριση με τους υπόλοιπους τρόπους λειτουργίας. Παρόλα αυτά , σε καμία από τις μέσες τιμές των RMSE, NMAE και SRE δεν προκύπτει ακραία απόκλιση από τις αντίστοιχες τιμές των **INN Original** και **INN Extended** μοντέλων.Το γεγονός αυτό σημαίνει πως το baseline μοντέλο, δηλαδή το INN χωρίς την εκμετάλλευση της εκ' κατασκευής αντιστρεψιμότητάς του, παρουσιάζει ήδη αρκετά ικανοποιητική απόδοση. Με άλλα λόγια η χρήση διαδοχικών αφινικών Coupling Layers ως ένα τυπικό feed-forward νευρωνικό δίκτυο δύναται να μοντελοποιήσει αποδοτικά την πολύπλοκη σχέση της μετατροπής των φορτίων ζεύξης y στον πίνακα κίνησης από τον οποίο προήλθαν. Καταλήγουμε λοιπόν πως, αν και εμφανώς πιο αδύναμο, το **INN Baseline** μοντέλο συνιστά μια καλή βάση την οποία οι επόμενοι (και πιο "σωστοί" ως προς την μεθοδολογία τους) τρόποι λειτουργίας θα εξελίξουν με την ενσωμάτωση της αντιστρεψιμότητας του INN.

Στη συνέχεια, τόσο από τον Πίνακα 5.5 όσο και από τα διαγράμματα των σφαλμά-

των, γίνεται αντιληπτό πως ο **INN Extended** τρόπος λειτουργίας παρέχει ελαφρώς καλύτερες εκτιμήσεις των πινάκων κίνησης όταν αξιολογείται ως προς σφάλματα που αφορούν την χρονική συσχέτιση των ροών κίνησης (μετρικές RMSE, TRE) σε σχέση με τον **INN Original** τρόπο λειτουργίας. Η διαφορά τους καθίσταται ιδιαίτερα αισθητή στην βασική μετρική αξιολόγησης RMSE όπου το **INN Extended** έχει μέση τιμή 11.3943 Mbps ενώ το **INN Original** 11.6602 Mbps. Την ίδια εικόνα παρατηρούμε ως προς τον διάμεσο αλλά και την τυπική απόκλιση για την RMSE μετρική η οποία υποδηλώνει πως το **INN Extended** μοντέλο είναι πιο σταθερό και ακριβές στις προβλέψεις του. Η TRE μετρική ακολουθεί την ίδια συμπεριφορά με την RMSE με το extended μοντέλο να αναπαριστά αποδοτικότερα τις χρονικές συσχετίσεις των ροών κίνησης.

Η τάση των RMSE και TRE μετρικών, δεν αντικατοπτρίζεται και στις NMAE και SRE μετρικές, με την δεύτερη να εμφανίζει και την πιο σημαντική απόκλιση (άρα μεγαλύτερο σφάλμα στις χωρικές συσχετίσεις). Το γεγονός αυτό οφείλεται στην παρατήρηση ότι, όπως φαίνεται στο Σχήμα 5.7, ένα μικρό πλήθος ροών κίνησης χαρακτηρίζεται από ακραίες τιμές σε σχέση με την μέση συμπεριφορά της κίνησης. Σε αυτές λοιπόν τις ροές, η απόδοση του **INN Extended** μειώνεται αισθητά και άρα ο μέσος όρος του SRE αυξάνεται (στις υπόλοιπες ροές όμως φαίνεται να έχει συγκρίσιμο και μερικές φορές καλύτερο SRE σφάλμα).

Σε γενικές γραμμές, τόσο η βελτίωση όσο και η μείωση στην απόδοση των δύο κύριων INN μεθόδων είναι σχετικά χαμηλές γεγονός που οφείλεται στην μικρή διαφορά της διάστασης εισόδου/εξόδου του INN (μετάβαση από 30 σε 40). Παρόλα αυτά αναμένεται, με την αύξηση της διάστασης d του INN ως προς το μήκος του διανύσματος του φορτίου ζεύξης y, η διαφορά στην απόδοση των **INN Extended** και **INN Original** μοντέλων να καταστεί πιο έντονη, με τίμημα όμως την αύξηση της πολυπλοκότητας του προβλήματος ελαχιστοποίησης (4.17).

Αν και ο **INN Extended** τρόπος λειτουργίας δεν παρουσιάζει σημαντικά καλύτερες επιδόσεις σε όλες τις μετρικές, πλεονεκτεί αισθητά απέναντι στον **INN Original** καθώς δύναται να χρησιμοποιηθεί και για εργασίες σύνθεσης πινάκων κίνησης. Με άλλα λόγια, το μοντέλο αυτό μπορεί να επιστρατευθεί για την παραγωγή ρεαλιστικών συνθετικών πινάκων κίνησης που να συμμορφώνονται με τα χαρακτηριστικά του δικτύου, ικανοποιώντας τα μοτίβα χώρο-χρονικών συσχετίσεων αλλά και τους περιορισμούς της τοπολογίας. Η σύνθεση πινάκων κίνησης πραγματοποιείται με δειγματοληψία τυχαίων διανυσμάτων w μέσω της Γκαουσιανής κατανομής τα οποία στη συνέχεια συνενώνονται με τα φορτία ζεύξης y για την παραγωγή τεχνητών διανυσμάτων y_e και κατ' επέκταση τον πίνακα κίνησης $\hat{x}$ μέσω της αντίστροφης διάδοσης και αποκωδικοποίησης από το INN και τον Αποκωδικοποιητή αντίστοιχα. Αυτοί οι συνθετικοί πίνακες μπορούν να αξιοποιηθούν στον έλεγχο και στην αξιολόγηση διαφόρων αλγορίθμων διαχείρισης δικτύων και κίνησης μέσω προσομοιώσεων.

Μέχρι στιγμής έχουν παρουσιαστεί όλες οι καινοτόμες μέθοδοι για την Εκτίμηση Πίνακα Κίνησης που ενσωματώνουν INN και έχουν αναλυθεί οι μεταξύ τους συσχετίσεις. Για την καλύτερη κατανόηση αλλά και ανάδειξη της υπεροχής των μεθόδων αυτών, πα-

ρέχονται αποτελέσματα από την σύγκριση των INN τεχνικών με baseline μοντέλα παραγωγής συνθετικών δειγμάτων της βιβλιογραφίας τα οποία έχουν αξιοποιηθεί στο παρελθόν για την Εκτίμηση Πίνακα Κίνησης. Τα μοντέλα αυτά λειτουργούν με παρόμοιο τρόπο όπως το **INN Extended** από την άποψη πως μετασχηματίζουν το ΕΠΚ σε πρόβλημα βελτιστοποίησης σε έναν χαμηλής διάστασης λανθάνοντα χώρο. Η πρώτη μέθοδος με την οποία συγκρίνονται οι INN τεχνικές σχετίζεται με την χρήση ενός συνελικτικού Variational Autoencoder [3] ενώ η δεύτερη μέθοδος χρησιμοποιεί ένα Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) [41].

Το Convolutional VAE αποτελείται από τον Κωδικοποιητή και τον Αποκωδικοποιητή, όπως και ο κλασσικός Αυτοκωδικοποιητής. Η διαφορά τους εντοπίζεται στο γεγονός πως ο VAE δεν στοχεύει στον μετασχηματισμό μεμονωμένων δειγμάτων αλλά στην κατασκευή μιας κατανομής του λανθάνοντος χώρου. Από την κατανομή αυτή, κατόπιν δειγματοληψίας, μπορούν μέσω του Αποκωδικοποιητή να παραχθούν συνθετικά δείγματα και άρα η ΕΠΚ ανάγεται στην εύρεση κατάλληλου διανύσματος στον λανθάνοντα χώρο που ελαχιστοποιεί το MSE(y, Ax). Στην συγκεκριμένη υλοποίηση, ο Κωδικοποιητής αποτελείται από τρία συνελικτικά επίπεδα Conv2d, με την τελική έξοδο να μετατρέπεται σε διάνυσμα και μέσω αυτού να εξάγονται τα διανύσματα της μέσης τιμής και της τυπικής απόκλισης (διάσταση 8 το καθένα) τα οποία θα ορίσουν την Γκαουσιανή κατανομή του λανθάνοντος χώρου. Ο Αποκωδικοποιητής πραγματοποιεί την αντίστροφη διαδικασία με την βοήθεια Conv2dTraspose επιπέδων, λαμβάνοντας ως είσοδο ένα δείγμα από την πολυμεταβλητή Γκαουσιανή κατανομή που έχει παράξει ο Κωδικοποιητής. Ως συνάρτηση ενεργοποίησης χρησιμοποιείται η ReLU. Για την εκπαίδευση του μοντέλου, τα δείγματα του συνόλου εκπαίδευσης μετασχηματίζονται στη μέση τιμή και τυπική απόκλιση μέσω του Κωδικοποιητή, παράγεται ένα δείγμα από την Γκαουσιανή κατανομή και ανακατασκευάζεται μέσω του Αποκωδικοποιητή. Η συνάρτηση απωλειών ορίζεται ως το σφάλμα ανακατασκευής MSE$(x, \hat{x})$ σε συνδυασμό με την ελαχιστοποίηση της Kullback-Leibler [86] απόστασης μεταξύ της παραγόμενης κατανομής και της Γκαουσανής με μέση τιμή 0 και τυπική απόκλιση 1.

Το WGAN-GP μοντέλο αποτελείται από τα νευρωνικά δίκτυα του Γεννήτορα και του Διαχωριστή. Στόχος της εκπαίδευσης είναι ο Γεννήτορας να παράγει, για είσοδο διανύσματα στον λανθάνοντα χώρο, ρεαλιστικά συνθετικά δείγματα (στην περίπτωσή μας πίνακες κίνησης) τα οποία ο διαχωριστής να αδυνατεί να ξεχωρίσει από τα πραγματικά δεδομένα. Ο Γεννήτορας είναι μια ακολουθία γραμμικών επιπέδων που μετασχηματίζει ένα διάνυσμα διάστασης 8 σε έναν πίνακα κίνησης της μορφής $(1 \times 12 \times 12)$. Ο Διαχωριστής λαμβάνει ως είσοδο έναν πίνακα κίνησης και τον προωθεί σε μία σειρά γραμμικών επιπέδων μέχρις ότου η διάσταση εξόδου να έχει μέγεθος 1. Αν η έξοδος του Διαχωριστή έχει την τιμή 1 τότε κατατάσσει τον πίνακα στην είσοδο ως πραγματικό, αλλιώς ως συνθετικό. Τόσο ο Γεννήτορας όσο και ο Διαχωριστής χρησιμοποιούν ReLU ως συνάρτηση ενεργοποίησης. Σε κάθε επανάληψη εκπαίδευσης, ο Γεννήτορας παράγει συνθετικά δείγματα για τα οποία ο Διαχωριστής παράγει εκτιμήσεις. Στη συνέχεια ο Διαχωριστής παράγει εκτιμήσεις και για ένα batch πραγματικών δειγμάτων ούτως ώστε να υπολογιστεί η συνάρτηση απωλειών του. Το Gradient Penalty εμπλουτίζει την συνάρτηση απωλειών

του Διαχωριστή ενσωματώνοντας την παρεμβολή ρεαλιστικών και συνθετικών δειγμάτων ως κανονικοποίηση, εξασφαλίζοντας περιορισμό Lipschitz. Μετά την φάση εκπαίδευσης του Διαχωριστή, ο Γεννήτορας παράγει συνθετικά δείγματα, ο Διαχωριστής παράγει εκτιμήσεις και υπολογίζεται η συνάρτηση απωλειών του Γεννήτορα (ιδανικά θέλει ο Διαχωριστής να ταξινομήσει όλα τα δείγματα ως πραγματικά). Για την εκτίμηση πινάκων κίνησης, απομονώνεται ο Γεννήτορας και επιλύεται πρόβλημα ελαχιστοποίησης του MSE(y, Ax).

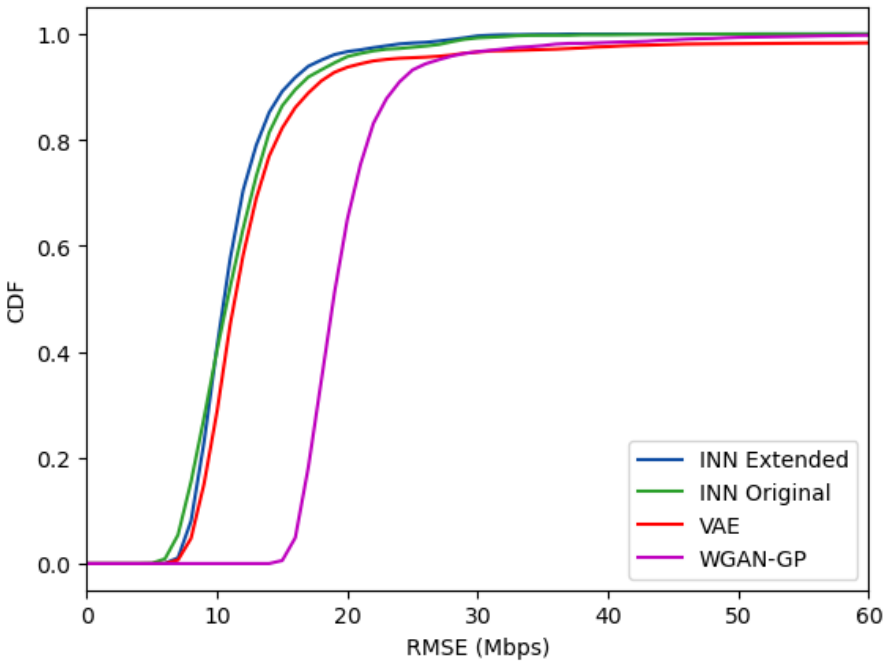
Περισσότερες λεπτομέρειες για την δομή, τις υπερπαραμέρους και τις συναρτήσεις απωλειών μπορούν να βρεθούν στον κώδικα των πειραμάτων. Αξίζει να σημειωθεί πως τόσο τα VAE και WGAN-GP όσο και το **INN Extended** πραγματοποιούν τον ίδιο αριθμό επαναλήψεων (100) ανά πίνακα κίνησης με στόχο την εύρεση κατάλληλου διανύσματος στο λανθάνοντα χώρο που ελαχιστοποιεί τη σχέση MSE(y, Ax).Τα δύο μοντέλα εκπαιδεύονται και αξιολογούνται στο ίδιο σύνολο εκπαίδευσης και ελέγχου με τα INN, με τις μετρικές των RMSE, NMAE, TRE, SRE να παρουσιάζονται στον ακόλουθο πίνακα.

Πίνακας 5.6: Πίνακας σφαλμάτων baseline μοντέλων

Convolutional VAE			
Error	Mean	Median	Std
RMSE(Mbps)	15.2523	11.3264	25.5547
NMAE	0.3361	0.3168	0.0877
TRE	0.3357	0.3109	0.1153
SRE	0.6136	0.5655	0.2647
WGAN-GP			
Error	Mean	Median	Std
RMSE(Mbps)	20.1863	18.9311	7.2145
NMAE	0.6199	0.6183	0.0819
TRE	0.5202	0.5149	0.0852
SRE	0.9866	1.0	0.5117

Συγκρίνοντας τις τιμές των μετρικών με αυτές του Πίνακα 5.5 παρατηρούμε πως οι

INN τεχνικές (ακόμα και η **INN Baseline**) παρουσιάζουν πολύ καλύτερη απόδοση σε σχέση με το Convolutional VAE αλλά και το WGAN-GP μοντέλο. Πιο συγκεκριμένα, οι προτεινόμενες μέθοδοι εμφανίζουν σημαντικά μειωμένα RMSE, TRE και NMAE κατά τη σύγκριση με το Convolutional VAE, με το SRE να είναι η μόνη μετρική στην οποία το VAE αποδίδει καλύτερα. Το WGAN-GP μοντέλο αδυνατεί να συγκλίνει κατά την εκτίμηση του πίνακα κίνησης και χαρακτηρίζεται από τις χειρότερες τιμές σε όλες τις μετρικές κατά την σύγκρισή του τόσο με το VAE όσο και με τις INN ροές εργασίας. Στο Σχήμα 5.8 απεικονίζονται οι CDF κατανομές του RMSE για τα μοντέλα INN Extended, INN Original, Convolutional VAE και WGAN-GP (περιορισμένες για εύρος σφαλμάτων από 0 έως 60 Mbps για λόγους ευκρίνειας) όπου φαίνεται η υπεροχή των INN μοντέλων. Καταλήγουμε λοιπόν στην διαπίστωση πως τα INN είναι πολύ αποτελεσματικά στην ΕΠΚ, και ότι υπερέχουν έναντι των ανταγωνιστικών μοντέλων Μηχανικής Μάθησης.



Σχήμα 5.8: RMSE CDF για INN Extended, INN Original, VAE και WGAN-GP

Τέλος αξίζει να σημειωθεί πως ο στόχος της πειραματικής αξιολόγησης των προτεινόμενων INN τεχνικών είναι η επικύρωση της βιωσιμότητας αλλά και της καταλληλότητας της χρήσης των μοντέλων INN σε μια ροή εργασίας για την Εκτίμηση Πίνακα Κίνησης. Συνεπώς δεν τίθεται ως προτεραιότητα η εύρεση της βέλτιστης αρχιτεκτονικής του δικτύου καθώς και η επιλογή του πιο αποδοτικού συνόλου υπερπαραμέτρων για την εκπαίδευση των νευρωνικών δικτύων. Αυτό σημαίνει πως υπάρχει χώρος για περαιτέρω πειράματα και fine-tuning των υπερπαραμέτρων ούτως ώστε να αξιοποιηθούν οι προτεινόμενες μέθοδοι στο έπακρο.

Κεφάλαιο 6

Επίλογος & Μελλοντικές Επεκτάσεις

6.1 Επίλογος

Σε αυτή τη διπλωματική εργασία μελετάται η χρήση τεχνικών Βαθιάς Μηχανικής Μάθησης για την επίλυση του προβλήματος της Εκτίμησης Πίνακα Κίνησης σε IP δίκτυα κορμού. Αρχικά πραγματοποιείται εκτενής ανασκόπηση της βιβλιογραφίας όπου παρουσιάζονται αναλυτικές και προσεγγιστικές μέθοδοι για την επίλυση της ΕΠΚ, δίνοντας όμως ιδιαίτερη έμφαση στις μεθόδους που ενσωματώνουν μοντέλα Μηχανικής Μάθησης. Με δεδομένο ότι η ΕΠΚ είναι ένα υπό-ορισμένο αντίστροφο πρόβλημα, προτείνεται για πρώτη φορά η χρήση Αντιστρέψιμων Νευρωνικών Δικτύων (INN) με στόχο την επίλυσή της. Τα INN συνιστούν μια κατηγορία μοντέλων Βαθιάς Μηχανικής Μάθησης που είναι εκ κατασκευής σχεδιασμένα για την μοντελοποίηση αντίστροφων προβλημάτων καθώς εκπαιδεύονται πάνω στον γνωστό forward μετασχηματισμό, μαθαίνοντας έμμεσα και την αντίστροφη διαδικασία. Τα μοντέλα αυτά όμως έχουν το μειονέκτημα πως η διάσταση της εισόδου και η διάσταση της εξόδου πρέπει να ταυτίζονται. Στην ΕΠΚ η είσοδος του μοντέλου είναι ο πίνακας κίνησης ο οποίος, αν αναπαρασταθεί ως διάνυσμα, έχει εν γένει σημαντικά μεγαλύτερη διάσταση από το διάνυσμα των φορτίων ζεύξης που αποτελεί και την έξοδο του μοντέλου.

Για την ενσωμάτωση των INN μοντέλων σε μια ροή εργασίας κατάλληλη για την επίλυση του προβλήματος ΕΠΚ, προτείνεται ένα στάδιο μείωσης της διάστασης της εισόδου με τη χρήση ενός προ-εκπαιδευμένου Αυτοκωδικοποιητή. Το στάδιο αυτό συνδυάζεται με τρεις εναλλακτικές μεθόδους που καλύπτουν όλες τις πιθανές χρήσεις των INN σε εφαρμογές Μηχανικής Μάθησης. Η μέθοδος **INN Baseline** χρησιμοποιεί το INN ως ένα τυπικό feed-forward νευρωνικό δίκτυο και λειτουργεί ως μοντέλο αναφοράς για την ικανότητα αναπαράστασης του αντίστροφου μετασχηματισμού από τα αφινικά Coupling Layers του δικτύου. Το **INN Original** συνιστά τον βασικό τρόπο λειτουργίας των INN

καθώς χρησιμοποιεί αμφίδρομη εκπαίδευση και επιλύει το πρόβλημα μέγιστης πιθανοφάνειας (4.7). Τέλος, το **INN Extended** άρει τον περιορισμό πως η διάσταση του INN οφείλει να ταυτίζεται με την διάσταση των φορτίων ζεύξης, ενσωματώνοντας ένα τυχαίο διάνυσμα επέκτασης με τιμές από την Γκαουσιανή κατανομή. Αυτός ο τρόπος λειτουργίας επιτρέπει στο μοντέλο την παραγωγή συνθετικών πινάκων κίνησης υπό την καθοδήγηση του διανύσματος των φορτίων ζεύξης μετατρέποντας την ΕΠΚ σε πρόβλημα βελτιστοποίησης.

Και οι τρεις προτεινόμενες μέθοδοι εκπαιδεύονται και αξιολογούνται σε δημόσια διαθέσιμο σύνολο δεδομένων που περιέχει πίνακες κίνησης από το πραγματικό δίκτυο Abilene. Από τα πειράματα προκύπτει πως το **INN Baseline** συνιστά μια ικανοποιητική βάση ως προς την δυνατότητα κωδικοποίησης του πολύπλοκου αντίστροφου μετασχηματισμού από τα Coupling Layers, ενώ οι πλήρεις τρόποι λειτουργίας **INN Original** και **INN Extended** εμφανίζουν πολύ καλύτερες επιδόσεις, με μικρές μεταξύ τους διαφορές. Το **INN Extended** μοντέλο εμφανίζει όμως το επιπλέον πλεονέκτημα ότι μπορεί να χρησιμοποιηθεί και σε εργασίες σύνθεσης πινάκων κίνησης. Τέλος, τα INN μοντέλα συγκρίνονται με εναλλακτικές προσεγγίσεις Βαθιάς Μηχανικής Μάθησης από την βιβλιογραφία και συγκεκριμένα με τον Συνελικτικό VAE και το WGAN-GP. Από την πειραματική αξιολόγηση προκύπτει η αισθητή υπεροχή των INN ροών εργασίας έναντι των ανταγωνιστικών μοντέλων.

6.2 Μελλοντικές Επεκτάσεις

Στην παρούσα διπλωματική εργασία παρουσιάζεται μια proof-of-concept υλοποίηση ροής εργασίας που αποδεικνύει πως τα INN δύνανται να χρησιμοποιηθούν για την ΕΠΚ. Από την χρήση των INN μοντέλων στην επίλυση του ΕΠΚ προκύπτουν αρκετές πτυχές που απαιτούν πρόσθετη διερεύνηση και επέκταση.

Αρχικά, έχοντας δείξει πως η αντιστρέψιμη αρχιτεκτονική των INN είναι κατάλληλη για την μοντελοποίηση του ΕΠΚ, μια πρώτη κατεύθυνση είναι η ενδελεχής εξερεύνηση εναλλακτικών INN αρχιτεκτονικών. Πιο συγκεκριμένα, αξίζει να κατασκευαστούν ροές εργασίας που να ενσωματώνουν ως INN πυρήνα πολλαπλές από τις κύριες INN υλοποιήσεις όπως τα μοντέλα NICE [49], RealNVP [50] και Glow [51] αλλά και πρωτότυπους μετασχηματισμούς στα αφινικά Coupling Layers ούτως ώστε να προσδιοριστεί η καταλληλότερη αντιστρέψιμη δομή του INN νευρωνικού δικτύου. Επιπλέον, η χαλάρωση της περιοριστικής δομής που επιβάλλουν τα αφινικά Coupling Layers για την εξασφάλιση tractable ορίζουσας του Ιακωβιανού πίνακα, έχει οδηγήσει στην κατασκευή αντιστρέψιμων αρχιτεκτονικών οι οποίες επιστρατεύουν επαναληπτικούς αλγορίθμους για την προσέγγιση της Ιακωβιανής ορίζουσας θυσιάζοντας ακρίβεια για αναπαραστατική ικανότητα. Τέτοια μοντέλα είναι τα i-ResNets [58] και το FFJORD [59] τα οποία αξίζει να δοκιμαστούν πάνω στο πρόβλημα της ΕΠΚ και άρα να εξεταστεί το κατά πόσο είναι απαραίτητη η αυστηρή μορφή των αφινικών επιπέδων για την αποδοτική μοντελοποίηση του προβλήματος.

Ο **INN Extended** τρόπος λειτουργίας επεκτείνει τη διάσταση του INN σε διάνυσμα μεγαλύτερο από το διάνυσμα των φορτίων ζεύξης y, συνενώνοντας το y με ένα Γκαουσιανής κατανομής τυχαίο διάνυσμα w. Στην πειραματική αξιολόγηση, το διάνυσμα y είχε διάσταση $d = 30$ ενώ η επεκτεταμένη του έκδοση $[y, w]$ είχε διάσταση $d = 40$. Το **INN Extended** μοντέλο εμφάνισε σε γενικές γραμμές καλύτερες επιδόσεις έναντι του **INN Original** με τις διαφορές τους όμως να είναι μην είναι σημαντικές λόγω της χαμηλής αύξησης της διάστασης. Καθίσταται λοιπόν απαραίτητη η διερεύνηση της συμπεριφοράς του μοντέλου με την αύξηση της διάστασης του επεκτεταμένου διανύσματος. Η ενέργεια αυτή αναμένουμε να οδηγήσει σε αισθητά βελτιωμένες επιδόσεις καθώς μειώνεται η επίδραση του σταδίου ελάττωσης διάστασης μέσω του Αυτοκωδικοποιητή. Στην ακραία περίπτωση, ακολουθώντας την μέθοδο των Ardizzone et al.[11], το διάνυσμα $[y, w]$ θα έχει διάσταση ίση με αυτή της εισόδου x και άρα το στάδιο του Αυτοκωδικοποιητή θα μπορούσε να παραληφθεί. Επιπλέον, μια εναλλακτική κατεύθυνση για την αύξηση της διάστασης των INN είναι η χρήση conditional INN αρχιτεκτονικών [56] στα οποία η έξοδος του INN δεν σχετίζεται με τα φορτία ζεύξης y παρά μόνο με το λανθάνον διάνυσμα w το οποίο αποκλειστικά καθορίζει την διάσταση του μοντέλου. Σε αυτό το περιβάλλον λειτουργίας τα διανύσματα y ενσωματώνονται ως συνθήκη στα αφινικά επίπεδα τόσο κατά την forward όσο και κατά την backward διάδοση της εισόδου. Τέλος η αύξηση της διάστασης του INN αξίζει να συσχετιστεί και με την ταχύτητα σύγκλισης του βρόχου βελτιστοποίησης που αναμένουμε να λειτουργήσει ως tradeoff μεταξύ ακρίβειας και ταχύτητας στην εκτίμηση των πινάκων κίνησης.

Μια άλλη προσέγγιση στην ΕΠΚ είναι η απευθείας κωδικοποίηση της τοπολογίας του δικτύου στα δεδομένα ούτως ώστε το μοντέλο μάθησης να μην απαιτείται να εξάγει έμμεσα πληροφορίες για την δομή του δικτύου αλλά να εστιάζει αποκλειστικά στην μοντελοποίηση της εκτίμησης. Αξίζει λοιπόν να διερευνηθούν τεχνικές που ενσωματώνουν ένα στάδιο πολυπλεξίας της δομής του δικτύου (πίνακας δρομολόγησης) με τα φορτία ζεύξης y όπως Graph Embedding [87] μέθοδοι, οι οποίες να παρέχουν την εμπλουτισμένη είσοδο στο INN τμήμα της ροής εργασίας. Μοντελοποιώντας το δίκτυο ως γράφο, μια ακόμα επέκταση έγκειται στην εκπαίδευση Graph Neural Networks [88] για την παραγωγή της εμπλουτισμένης με την τοπολογία εισόδου αλλά και για την απευθείας χρήση τους για την εκτίμηση του πίνακα κίνησης. Στο ίδιο πλαίσιο αξίζει να εξεταστεί και ο συνδυασμός των Graph Neural Networks, ως στάδιο προ-επεξεργασίας, με τα INN αλλά και η προσπάθεια κατασκευής Graph Neural Networks με αντιστρέψιμη (ή κατά προσέγγιση αντιστρέψιμη) δομή.

Τέλος, όλες οι τεχνικές INN δικτύων είναι απαραίτητο να εξεταστούν σε περισσότερα σύνολα δεδομένων, τόσο σε συνθετικά όσο και σε πραγματικά δίκτυα. Η ενέργεια αυτή θα εξετάσει την αποτελεσματικότητά τους σε ποικίλα μοτίβα κίνησης και πιθανώς να εντοπίσει αδυναμίες σε συγκεκριμένους τύπους outlier κίνησης. Αξίζει να σημειωθεί πως τα δημόσια διαθέσιμα σύνολα για την ΕΠΚ που αφορούν πραγματικά δίκτυα είναι πολύ λίγα γεγονός που αποτελεί πρόκληση για την ερευνητική κοινότητα ούτως ώστε να εξασφαλιστούν περισσότερα και πιο σύγχρονα σύνολα δεδομένων όπως τα GEANT [89] και MAWI [90] για την ρεαλιστική αξιολόγηση των προτεινομένων μεθόδων.

Συνοψίζοντας, ως μελλοντικές επεκτάσεις αυτής της διπλωματικής εργασίας είναι η δοκιμή εναλλακτικών αρχιτεκτονικών INN, είτε παραδοσιακής μορφής με χρήση αφινικών επιπέδων είτε προσεγγιστικής μορφής για μεγαλύτερη αναπαραστατική ικανότητα. Επιπλέον, αξίζει να διερευνηθεί το tradeoff μεταξύ της επέκτασης της διάστασης του INN μοντέλου, που συνεπάγεται και αύξηση της απόδοσής του, και της ταχύτητας στην παραγωγή εκτιμήσεων. Τέλος προτείνεται η εξέταση μεθόδων που κωδικοποιούν την δομή του δικτύου στα δεδομένα αλλά και η χρήση Graph Neural Networks ως επόμενη κατηγορία μεθόδων Βαθιάς Μηχανικής Μάθησης για την ΕΠΚ. Όλες οι προτεινόμενες μέθοδοι οφείλουν να αξιολογηθούν σε περισσότερα σύνολα δεδομένων με στόχο την εξαγωγή ρεαλιστικών και χωρίς μεροληψία εκτιμήσεων για την απόδοσή τους.

Παράρτημα Α

Περιβάλλον Εκτέλεσης

Σε αυτήν την ενότητα παρουσιάζονται λεπτομέρειες σχετικά με το περιβάλλον εκτέλεσης των πειραμάτων.

Αρχικά το περιβάλλον εργασίας που χρησιμοποιήθηκε είναι το PyTorch [84]. Το PyTorch αποτελεί μια βιβλιοθήκη με μεθόδους Μηχανικής Μάθησης και βασίζεται στην βιβλιοθήκη Torch. Αναπτύχθηκε από την Meta AI και θεωρείται μία από τις πιο δημοφιλείς πλατφόρμες ανάπτυξης αλγορίθμων Μηχανικής Μάθησης μαζί με το TensorFlow [91]. Ο πυρήνας του PyTorch είναι μια δομή που ονομάζεται Tensor, δηλαδή ένας ομοιόμορφων τιμών πολυδιάστατος τετραγωνικός πίνακας που χρησιμοποιείται για την αποθήκευση και την επεξεργασία των δεδομένων από τις εφαρμογές Μηχανικής Μάθησης. Κατά την διάδοση του Tensor εισόδου σε ένα νευρωνικό δίκτυο, το PyTorch κατασκευάζει τον υπολογιστικό γράφο πράξεων/μετασχηματισμών πάνω στα δεδομένα ο οποίος χρησιμοποιείται για την αποδοτική εκτέλεση του Backpropagation για τον υπολογισμό των παραγώγων όλων των στοιχείων του Tensor. Η φύση των Tensors καθιστά το PyTorch ιδανικό για χρήση επιταχυντών υλικού για την ταχύτερη εκπαίδευση και αξιολόγηση των μοντέλων Μηχανικής Μάθησης, όπως οι Κάρτες Γραφικών (GPU). Πιο συγκεκριμένα, το PyTorch δύναται να επιστρατεύσει την GPU ως μονάδα επεξεργασίας μεταφράζοντας τις εντολές του μέσω της παράλληλης προγραμματιστικής πλατφόρμας CUDA (Compute Unified Device Architecture) [92]. Τέλος το PyTorch υποστηρίζει ως κύρια γλώσσα προγραμματισμού την Python [93] ενώ παρέχει υποστήριξη και για C++ [94].

Ο κώδικας των πειραμάτων αναπτύσσεται σε Jupyter Notebooks [85], μια πλατφόρμα η οποία επιτρέπει την οργάνωση του κώδικα σε κελιά, τα οποία μπορούν να εκτελεστούν αυτόνομα και συνδυαστικά.

Τα πειράματα αναπτύχθηκαν και αξιολογήθηκαν στην NVIDIA GeForce GTX 1050 Ti και χρησιμοποίησαν τις εξής βιβλιοθήκες: Python (3.9.7), Pytorch (2.2.1), NumPy (1.20.3) και CUDA (12.1)

Παράρτημα Β

Πηγαίος Κώδικας

Ο πηγαίος κώδικας που χρησιμοποιήθηκε για την ανάπτυξη και πειραματική αξιολόγηση των προτεινόμενων μεθόδων μπορεί να βρεθεί στο παρακάτω αποθετήριο κώδικα:

`https://gitlab.com/ai4networks/invertible-neural-networks`

Στη συνέχεια παρατίθεται ενδεικτικά ο κώδικας που χρησιμοποιήθηκε για την κατασκευή, εκπαίδευση και αξιολόγηση της ροής εργασίας που αφορά τον **INN Extended** τρόπο λειτουργίας.

```
import numpy as np
import math
import time

import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import Dataset, DataLoader
import torch.nn.init as init
import torch.nn.functional as F
torch.manual_seed(42)
```

```
# Define Swish Activation Function
class Swish(nn.Module):
    def __init__(self, beta=1.0):
        super(Swish, self).__init__()

        # Define beta as a trainable parameter
        self.beta = nn.Parameter(torch.tensor(beta))

    def forward(self, x):
        return torch.abs(x * torch.sigmoid(self.beta * x))
```

```
11
12 # Define the autoencoder architecture, target dimension is 40
13 class Autoencoder(nn.Module):
14     def __init__(self):
15         super(Autoencoder, self).__init__()
16
17         # Encoder
18         self.encoder = nn.Sequential(
19             nn.Conv2d(1, 32, kernel_size=3, stride=1, padding=0)
                ,
20             Swish(),
21             nn.Conv2d(32, 64, kernel_size=3, stride=1, padding
                =0),
22             Swish(),
23             nn.Conv2d(64, 128, kernel_size=3, stride=1, padding
                =0),
24             Swish(),
25             nn.Flatten(),
26             nn.Linear(128*36,40),
27             Swish()
28         )
29
30         # Decoder
31         self.decoder = nn.Sequential(
32             nn.Linear(40, 128*36),
33             Swish(),
34             nn.Unflatten(dim=1, unflattened_size=(128, 6, 6)),
35             nn.ConvTranspose2d(128, 64, kernel_size=3, stride=1,
                 padding=0),
36             Swish(),
37             nn.ConvTranspose2d(64, 32, kernel_size=3, stride=1,
                padding=0),
38             Swish(),
39             nn.ConvTranspose2d(32, 1, kernel_size=3, stride=1,
                padding=0),
40             Swish(),
41         )
42
43       # Initialize the weights using He initialization
44         self._initialize_weights()
45
46
47     def _initialize_weights(self):
48         for m in self.modules():
49             if isinstance(m, nn.Conv2d) or isinstance(m, nn.
                ConvTranspose2d) or isinstance(m, nn.Linear):
50                 init.kaiming_normal_(m.weight, mode='fan_out',
                    nonlinearity='relu')
51                 if m.bias is not None:
```

```
                    init.constant_(m.bias, 0)

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x

    def encode(self,x):
      x = self.encoder(x)
      return x

    def decode(self,x):
      x = self.decoder(x)
      return x
```

```
np.random.seed(42)
# Dataset schema and creation

# Create custom dataset class

class TrafficMatrixDataset(Dataset):
    def __init__(self, data):
        self.data = data

    def __len__(self):
        return len(self.data)

    def __getitem__(self, idx):
        return self.data[idx]


# Load traffic matrix from npy file
X_array = np.load('./data/ABILENE/X.npy')
print("Dataset containing all traffic matrices has shape:",
    X_array.shape)


# We define a slice of the dataset that contains 14 continuous
    weeks of traffic matrices and keep samples from weeks 6-19
data_size = 14*24*7*12
week = 24*7*12
threshold = 5*week


# Split the dataset into 80% training samples and 20% testing
    samples and convert the samples in Mbps
X_array = X_array[threshold:(threshold+data_size)]/1000
```

```
X_train = X_array[:math.floor(data_size*0.8)]
X_test =  X_array[math.floor(data_size*0.8):]
print("Final dataset shape:",X_array.shape)
print("Train dataset shape:",X_train.shape)
print("Test dataset shape:",X_test.shape)


min_val = np.min(X_array)
max_val = np.max(X_array)


# Create y and find min max values
# y dataset can be found by the dot product of routing Matrix R and a stacked vector x
R_array = np.load('./data/ABILENE/R.npy') # Routing Matrix
y_array = np.array([np.dot(X_array[i,:,:].reshape(-1,144), R_array.transpose()) for i in range(X_array.shape[0])])
y_array = y_array.reshape(-1,30)
print("Y dataset shape is:",y_array.shape)


# Split y dataset in the same way as for X dataset
y_train = y_array[:math.floor(data_size*0.8)]
y_test =  y_array[math.floor(data_size*0.8):]

min_val_y = np.min(y_array)
max_val_y = np.max(y_array)

# Create tensor X
X_data_train = torch.tensor(X_train, dtype=torch.float32)
X_data_test = torch.tensor(X_test, dtype=torch.float32)

# Create tensor y
y_data_train = torch.tensor(y_train, dtype=torch.float32)
y_data_test = torch.tensor(y_test, dtype=torch.float32)


# Perform min-max scaling
X_data_train = (X_data_train - min_val) / (max_val - min_val)
X_data_test = (X_data_test - min_val) / (max_val - min_val)

y_data_train = (y_data_train - min_val_y) / (max_val_y - min_val_y)
y_data_test = (y_data_test - min_val_y) / (max_val_y - min_val_y)


# Create DataLoader objects for train and test sets

```

```
train_loader_x = DataLoader(dataset=TrafficMatrixDataset(
    X_data_train), batch_size=X_data_train.size(0), shuffle=False
    )
test_loader_x = DataLoader(dataset=TrafficMatrixDataset(
    X_data_test), batch_size=X_data_test.size(0), shuffle=False)
```

```
# Define device to be used for training and testing
device = torch.device('cuda' if torch.cuda.is_available() else '
    cpu')
print(device)
```

```
# Initialize Autoencoder Instance and load the corresponding
    best parameters
model = Autoencoder()
model.to(device)
model.load_state_dict(torch.load('./Models/autoencoder_40_best.
    pth'))
```

```
torch.manual_seed(42)
model.eval()

with torch.no_grad():
    for data in train_loader_x:
        # Each batch has shape (sample_number,12,12) and the
            model has to take input a batch of size (
            sample_number,1,12,12)
        data = data.unsqueeze(1).to(device)
        X_40_train = model.encode(data)

with torch.no_grad():
    for data in test_loader_x:
        # Each batch has shape (sample_number,12,12) and the
            model has to take input a batch of size (
            sample_number,1,12,12)
        data = data.unsqueeze(1).to(device)
        X_40_test = model.encode(data)

print("Latent X train set has shape:",X_40_train.shape)
print("Latent X test set has shape:",X_40_test.shape)

# Save tensors for further use
torch.save(X_40_train, 'X_40_train.pt')
torch.save(X_40_test, 'X_40_test.pt')
```

```
# Deep Neural Layer Architecture (will be used in s, t
    transformations)

class NN_layer(nn.Module):
    def __init__(self,input_size=20,output_size=20):
        super(NN_layer, self).__init__()

        self.nn_layer = nn.Sequential(
           nn.Conv1d(in_channels=1, out_channels=32, kernel_size
               =3, padding="same"),
           Swish(),
           nn.Conv1d(in_channels=32, out_channels=64,
               kernel_size=3, padding="same"),
           Swish(),
           nn.Flatten(),
           nn.Linear(64*input_size,output_size),
           Swish()
       )

      # Initialize the weights using He initialization
        self._initialize_weights()


    def _initialize_weights(self):
        for m in self.modules():
            if isinstance(m, nn.Linear) or isinstance(m, nn.
                Conv1d) :
                 init.kaiming_normal_(m.weight, mode='fan_out',
                     nonlinearity='relu')
                 if m.bias is not None:
                     init.constant_(m.bias, 0)

    def forward(self, x):
        x = x.unsqueeze(1)
        x = self.nn_layer(x)
        return x
```

```python
# Coupling Layer Architecture

class Coupling_layer(nn.Module):
    def __init__(self,rev=False):
        super(Coupling_layer, self).__init__()

        # Dinh et al (2017) affine implementation

        # Forward Pass
        # y1 = x1
        # y2 = x2 * exp(s(x1)) + t(x1)

        # Inverse Pass
        # x1 = y1
        # x2 = (y2 - t(y1)) / exp(s(y1))

        self.s = NN_layer()
        self.t = NN_layer()
        self.rev = rev

    def forward(self, x):

      # Split the each vector in half
      x1, x2 = self.split(x,rev=self.rev)

      # Calculate s,t transformations
      s_out = self.s(x1)
      t_out = self.t(x1)

      exp_s = torch.exp(s_out)

      # Perform Affine calculation
      y1 = x1
      y2 = x2 * exp_s + t_out

      # Calculate the log determinant of the layer as exp(sum(
          s_out))
      sums = torch.sum(s_out, dim=1, keepdim=True)  # Shape: 32
          x1
      exp_sums = torch.exp(sums)
      mean_det = torch.mean(exp_sums, dim=0) # Compute the mean
          across the first dimension (batch size)
      log_mean_det = torch.log(mean_det)

      # Concut y1,y1 and output the result
      y = torch.cat((y1, y2), dim=1)

      return y, log_mean_det
```

```
46
47
48      def inverse(self,y):
49
50        # Split the each vector in half
51        y1, y2 = self.split(y,rev=self.rev)
52
53        # Calculate s,t transformations
54        s_out = self.s(y1)
55        t_out = self.t(y1)
56
57        exp_s = torch.exp(-s_out)
58
59        # Perform Affine calculation
60        x1 = y1
61        x2 = (y2 - t_out)*exp_s
62
63        #Concut x1,x2, perform inverse shuffling and output the
              result
64        x = torch.cat((x1, x2), dim=1)
65
66        return x
67
68
69      def split(self,x,split_point=20,rev=False):
70
71        if(not rev):
72          x1 = x[:, :split_point]
73          x2 = x[:, split_point:]
74
75        else:
76          x2 = x[:, :split_point]
77          x1 = x[:, split_point:]
78
79        return x1,x2
```

```
1  # INN Architecture
2
3  class INN(nn.Module):
4      def __init__(self):
5          super(INN, self).__init__()
6
7
8          self.coupling1 = Coupling_layer()
9          self.coupling2 = Coupling_layer(rev=True)
10         self.coupling3 = Coupling_layer()
11         self.coupling4 = Coupling_layer(rev=True)
```

```
12
13
14      def forward(self, x):
15        x, log_det1 = self.coupling1(x)
16        x, log_det2 = self.coupling2(x)
17        x, log_det3 = self.coupling3(x)
18        y, log_det4 = self.coupling4(x)
19
20        total_log_det = log_det1 + log_det2 + log_det3 + log_det4
21
22        return y, total_log_det
23
24
25      def inverse(self,y):
26        y = self.coupling4.inverse(y)
27        y = self.coupling3.inverse(y)
28        y = self.coupling2.inverse(y)
29        x = self.coupling1.inverse(y)
30
31        # Add 0.02 for regularization
32        x = torch.abs(x)
33        x += 0.02
34        return x
```

```
1  def RMSE_loss(predicted,real):
2    return torch.sqrt(nn.functional.mse_loss(predicted, real))
3
4
5  def NMAE_loss(predicted, real):
6    # Calculate the absolute error between predicted and real
       values
7    abs_error = torch.abs(predicted - real)
8
9    # Calculate the mean absolute error
10   mae = torch.mean(abs_error)
11
12   # Calculate the mean absolute value of real values
13   mean_absolute_value_real = torch.mean(torch.abs(real))
14
15   # Calculate the Normalized Mean Absolute Error (NMAE)
16   nmae = mae / mean_absolute_value_real
17
18   return nmae
19
20
21 def TRE_loss(predicted, real):
22   # Calculate the root mean squared error between predicted and
```

```
    real values
    mse1 = torch.sqrt(nn.functional.mse_loss(predicted, real))

    # Calculate the root mean squared error of the real values
    mse2 = torch.sqrt(torch.mean(torch.square(real)))

    # Calculate the Target Residual Error (TRE) as the ratio of
        mse1 to mse2
    tre = mse1 / mse2

    return tre


def SRE_loss(predicted, real):

    sre_loss = []  # Initialize an empty list to store individual
        SRE losses

    # Flatten the real and predicted tensors
    flat_real = real.view(real.size(0), -1)
    flat_pred = predicted.view(predicted.size(0), -1)

    # Loop through each element in the spatial dimension of the
        flattened tensors
    for i in range(144):
        # Calculate the sum of squared differences between
            predicted and real values
        sum1 = torch.sqrt(torch.square(flat_pred[:, i] - flat_real
            [:, i]).sum())

        # Calculate the square root of the sum of squared real
            values
        sum2 = torch.sqrt(torch.square(flat_real[:, i]).sum())

        # Calculate the SRE for this element and append it to the
            list
        sre_loss.append((sum1 / sum2).item())

    return sre_loss
```

```
torch.manual_seed(42)
X_40_train = torch.load('X_40_train.pt')
X_40_test = torch.load('X_40_test.pt')

min_X_40_train = torch.min(X_40_train)
max_X_40_train = torch.max(X_40_train)

```

```
min_X_40_test = torch.min(X_40_test)
max_X_40_test = torch.max(X_40_test)

min_X_40 = min(min_X_40_train.item(),min_X_40_test.item())
max_X_40 = min(max_X_40_train.item(),max_X_40_test.item())

# Perform Min-Max scaling to X_30 dataset
X_40_train = ((X_40_train - min_X_40) / ((max_X_40 - min_X_40)))
X_40_test = ((X_40_test - min_X_40) / ((max_X_40 - min_X_40)))


min_value = torch.min(X_40_test)
max_value = torch.max(X_40_test)

# Shuffle Dataset
indices = np.random.RandomState(seed=42).permutation(X_40_train.
    size(0))


X_40_train = X_40_train[indices,:]
y_data_train = y_data_train[indices,:]
```

```
torch.manual_seed(42)

def z_distribution(sample_number,vector_size,mean,std):
    z_array = torch.randn(sample_number, vector_size) * std +
        mean
    z_array = torch.clamp(z_array, 0, 1)
    return z_array
```

```
# Construct dataloaders

# X reduced to 40 dataset
batch_size = 128
train_loader_x = DataLoader(dataset=TrafficMatrixDataset(
    X_40_train), batch_size=batch_size, shuffle=False)

# Y dataloader
train_loader_y = DataLoader(dataset=TrafficMatrixDataset(
    y_data_train), batch_size=batch_size, shuffle=False)
```

```
# Forward and Inverse Training
torch.manual_seed(42)

if torch.cuda.is_available():
    # Set seed for GPU operations
    torch.cuda.manual_seed(42)
    torch.cuda.manual_seed_all(42)  # if you are using multiple GPUs
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False  # set this to False for reproducibility


# Initialize the autoencoder
inn_model = INN()
inn_model.to(device)


# Define the loss function and optimizer
criterion = nn.MSELoss()
mse = nn.MSELoss()
optimizer = optim.Adam(inn_model.parameters(),lr=0.001)


# Define training loop's parameters
patience = 7  # Number of epochs to wait for improvement
best_train_loss = float('inf')
best_model_params = None
counter = 0
num_epochs = 500
forward_loss_list = []
backward_loss_list = []
total_elapsed_time = 0

# Training Loop
for epoch in range(num_epochs):
    forward_loss = 0
    backward_loss = 0
    log_det_mean = 0
    inn_model.train()
    start_time = time.time()

    for x_data,y_true in zip(train_loader_x,train_loader_y):

        # Send data to device and sample from z distribution
        x_data = x_data.to(device)
        y_true = y_true.to(device)
        z_true = z_distribution(x_data.size(0),10,torch.mean(
```

```
            y_data_train),torch.std(y_data_train))
        z_true = z_true.to(device)

        # Forward Pass through INN
        optimizer.zero_grad()
        output, log_det = inn_model(x_data)

        # Calculate forward losses
        loss_y = criterion(output[:,:30],y_true)
        loss_z = criterion(output[:,30:],z_true)
        loss = (3/4)*loss_y + (1/4)*loss_z -log_det/1000000

        # Update model parameters
        loss.backward()
        optimizer.step()

        forward_loss += loss.item()

        # Perform inverse pass through INN
        optimizer.zero_grad()
        yz_true = torch.cat((y_true, z_true), dim=1)
        output2 = inn_model.inverse(yz_true)

        # Calculate backward loss and update model parameters
        loss2 = criterion(output2,x_data)
        loss2.backward()
        optimizer.step()
        backward_loss += loss2.item()
        log_det_mean += log_det



    back_total = backward_loss/len(train_loader_x)
    forward_total = forward_loss/len(train_loader_x)
    counter += 1
    backward_loss_list.append(back_total)
    forward_loss_list.append(forward_total)

    end_time = time.time()
    elapsed_time = end_time - start_time
    total_elapsed_time += elapsed_time

    # Store the parameters that produce best backward training
        loss
    if (back_total < best_train_loss):
        best_train_loss = back_total
        torch.save(inn_model.state_dict(), './Models/
            inn_generative_best.pth')
        counter = 0
```

```
93
94
95
96      print('Epoch [{}/{}], Forward Loss: {:.8f}, Backward Loss:
            {:.8f}'.format(epoch+1, num_epochs, forward_total,
            back_total))
97      print("Log det:",log_det_mean.item()/len(train_loader_x))
98      print("Epoch completed in: {:.2f} seconds\n".format(
            elapsed_time))
99      print()
100
101     # Early stopping check
102     if counter > patience and back_total > best_train_loss and
            epoch>300:
103         print("Early stopping triggered.")
104         break
105
106 print("Training completed in: {:.2f} seconds, best train loss is
        : {}\n".format(total_elapsed_time,best_train_loss))
107
108 np.save('./Results/INN_generative/inn_gen_ftrain_loss.npy', np.
        array(forward_loss_list))
109 np.save('./Results/INN_generative/inn_gen_btrain_loss.npy', np.
        array(backward_loss_list))
110 inn_model.load_state_dict(torch.load('./Models/
        inn_generative_best.pth'))
```

```
1  # Find best starting point (parallel version)
2  torch.manual_seed(42)
3
4  if torch.cuda.is_available():
5      # Set seed for GPU operations
6      torch.cuda.manual_seed(42)
7      torch.cuda.manual_seed_all(42)  # if you are using multiple
           GPUs
8      torch.backends.cudnn.deterministic = True
9      torch.backends.cudnn.benchmark = False  # set this to False
           for reproducibility
10
11 R_tensor = torch.tensor(R_array)
12 R_tensor = R_tensor.to(device)
13 model.to(device)
14 mse = nn.MSELoss()
15
16 # The following dataloaders contain test sets of 2d X arrays (
       scaled) and of y dataset
17 test_loader = DataLoader(dataset=TrafficMatrixDataset(
```

```
    X_data_test), batch_size=128, shuffle=False)
test_loader_y = DataLoader(dataset=TrafficMatrixDataset(
    y_data_test), batch_size=128, shuffle=False)


model.eval()
inn_model.eval()

mean_y = torch.mean(y_data_train)
std_y = torch.std(y_data_train)
flag = False # Flag that facilitates the concatenation of each
    batch result

with torch.no_grad():
    for y_sample,x_true in zip(test_loader_y,test_loader):

        # 100 samples to determine good start
        batch_size = y_sample.size(0)
        best_losses = np.full(batch_size, np.inf)
        y_sample = y_sample.to(device)

        batch_z_start = torch.zeros(batch_size, 10)
        batch_z_start = batch_z_start.to(device)

        for i in range(100):

            # Sample z and send it to device
            z_sample = z_distribution(batch_size,10,mean_y,std_y
                )
            z_sample = z_sample.to(device)

            # Concat with y_true and Inverse pass through INN
            yz_sample = torch.cat((y_sample, z_sample), dim=1)
            inn_output = inn_model.inverse(yz_sample)
            inn_output = (inn_output * (max_X_40 - min_X_40) +
                min_X_40)

            # Decode the INN's output to obtain x_pred, and
                remove scaling
            x_pred = model.decode(inn_output)
            x_pred = x_pred.squeeze(1)
            x_pred = x_pred * (max_val - min_val) + min_val

            # Convert X_pred to a stacked vector
            x_pred = x_pred.view(x_pred.size(0), -1)  # -1 means
                infer the size along that dimension

            # Perform dot product with the Routing tensor
            y_pred = torch.matmul(x_pred, R_tensor.t())
```

```
            y_pred = (y_pred - min_val_y)/(max_val_y-min_val_y)

            # Calculate MSE loss for each sample using broadcasting
            mse_losses = F.mse_loss(y_pred, y_sample, reduction='none')  # 'none' to get per-sample losses

            # Calculate mean MSE loss across the second dimension (30) to get a single loss value for each sample
            mse_losses = torch.mean(mse_losses, dim=1)

            # Update best loss and best_z for each sample in batch if its i-th loss is less than current best
            for j in range(batch_size):
                if (mse_losses[j]<best_losses[j]):
                    best_losses[j] = mse_losses[j]
                    batch_z_start[j,:] = z_sample[j,:]

        if(flag):
            z_start =  torch.cat((z_start, batch_z_start), dim=0)
        else:
            z_start = batch_z_start
            flag = True

# Save results
torch.save(z_start, 'z_start.pt')
```

```
# Minimization loop
z_start = torch.load("z_start.pt")
torch.manual_seed(42)
if torch.cuda.is_available():
    # Set seed for GPU operations
    torch.cuda.manual_seed(42)
    torch.cuda.manual_seed_all(42)  # if you are using multiple GPUs
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False  # set this to False for reproducibility

R_tensor = torch.tensor(R_array)
R_tensor = R_tensor.to(device)
model.to(device)
mse = nn.MSELoss()

# Create dataloaders
```

```
test_loader = DataLoader(dataset=TrafficMatrixDataset(
    X_data_test), batch_size=1, shuffle=False)
test_loader_y = DataLoader(dataset=TrafficMatrixDataset(
    y_data_test), batch_size=1, shuffle=False)
test_loader_z = DataLoader(dataset=TrafficMatrixDataset(z_start)
    , batch_size=1, shuffle=False)


model.eval()
inn_model.eval()

count = 0
flag = False

for y_sample,x_true,z_init in zip(test_loader_y,test_loader,
    test_loader_z):

    # Send data to device
    y_sample = y_sample.to(device)
    x_true = x_true * (max_val - min_val) + min_val
    x_true = x_true.to(device)
    z_sample= z_init.clone().detach().requires_grad_(True)
    z_sample = z_sample.to(device)


    batch_size = y_sample.size(0)
    best_losses = np.full(batch_size, np.inf)
    best_z_batch = torch.zeros(batch_size, 10)
    best_i=0

    flag2 = False

    # minimization loop
    for i in range(100):

      # Define Optimizer with current "best z"
      optimizer = optim.Adam([z_sample], lr=0.001)

      # Create input and perform INN inverse pass
      yz_sample = torch.cat((y_sample, z_sample), dim=1)

      inn_output = inn_model.inverse(yz_sample)
      inn_output = (inn_output * (max_X_40 - min_X_40) +
          min_X_40)

      # Decode INN output and remove scaling
      x_pred = model.decode(inn_output)
      x_pred = x_pred.squeeze(1)
      x_pred = x_pred * (max_val - min_val) + min_val
```

```
61
62      # Calculate the "real" RMSE loss between the current
            predicted array with the real testing sample
63      rmse_x = torch.sqrt(mse(x_pred, x_true))
64
65      x_pred = x_pred.view(x_pred.size(0), -1)  # -1 means infer
            the size along that dimension
66
67      # Perform dot product with the Routing tensor
68      y_pred = torch.matmul(x_pred, R_tensor.t())
69      y_pred = (y_pred - min_val_y)/(max_val_y-min_val_y)
70
71      # Find objective's loss
72      optimizer.zero_grad()
73      mse_loss = mse(y_pred,y_sample)
74
75
76      if(not flag2):
77        rmse_x_init = rmse_x
78        y_mse_init = mse_loss
79        flag2 = True
80
81      # Update z vector
82      mse_loss.backward()
83      optimizer.step()
84
85      # Keep the z vector that minimizes objective
86      if(mse_loss.item()<best_losses[0]):
87        best_losses[0] = mse_loss.item()
88        best_z_batch[0,:] = z_sample
89        best_i=i
90        best_rmse = rmse_x
91
92
93    if(flag):
94      z_final = torch.cat((z_final, best_z_batch), dim=0)
95    else:
96      z_final = best_z_batch
97      flag = True
98
99    print(f"Sample: {count}, min iteration is {best_i}, initial
          y_loss*10^5 and x_loss is {y_mse_init.item()*(10**5)} and
           {rmse_x_init.item()}")
100   print(f"Sample: {count}, best y and x loss are {best_losses
          [0]*(10**5)} and {best_rmse.item()}")
101   print()
102
103   count+=1
104
```

```
# Save results
torch.save(z_final, 'z_final.pt')
```

```
# Final test

z_final = torch.load("z_final.pt")

torch.manual_seed(42)
model.to(device)
mse = nn.MSELoss()

# Dataloader creation with batch size 1
test_loader = DataLoader(dataset=TrafficMatrixDataset(
    X_data_test), batch_size=1, shuffle=False)
test_loader_y = DataLoader(dataset=TrafficMatrixDataset(
    y_data_test), batch_size=1, shuffle=False)
test_loader_z = DataLoader(dataset=TrafficMatrixDataset(z_final)
    , batch_size=1, shuffle=False)


model.eval()
inn_model.eval()
total_loss_rmse = []
total_loss_nmae = []
total_loss_tre = []

with torch.no_grad():
    for y_sample,x_true,z_sample in zip(test_loader_y,
        test_loader,test_loader_z):

        # Send data to device
        y_sample = y_sample.to(device)
        x_true = x_true.to(device)
        z_sample = z_sample.to(device)

        # Create input and inverse pass
        yz_sample = torch.cat((y_sample, z_sample), dim=1)

        inn_output = inn_model.inverse(yz_sample)
        inn_output = (inn_output * (max_X_40 - min_X_40) +
            min_X_40)

        # Decode INN's output
        x_pred = model.decode(inn_output)
        x_pred = x_pred.squeeze(1)

        # Remove Scaling
```

```
        x_pred = x_pred * (max_val - min_val) + min_val
        x_true = x_true * (max_val - min_val) + min_val

        total_loss_rmse.append(RMSE_loss(x_pred,x_true).item())
        total_loss_nmae.append(NMAE_loss(x_pred,x_true).item())
        total_loss_tre.append(TRE_loss(x_pred,x_true).item())



# Perform the same process where batch is the whole test set
test_loader = DataLoader(dataset=TrafficMatrixDataset(
    X_data_test), batch_size=X_data_test.size(0), shuffle=False)
test_loader_y = DataLoader(dataset=TrafficMatrixDataset(
    y_data_test), batch_size=X_data_test.size(0), shuffle=False)
test_loader_z = DataLoader(dataset=TrafficMatrixDataset(z_final)
    , batch_size=X_data_test.size(0), shuffle=False)

with torch.no_grad():
    for y_sample,x_true,z_sample in zip(test_loader_y,
        test_loader,test_loader_z):

        y_sample = y_sample.to(device)

        x_true = x_true.to(device)
        z_sample = z_sample.to(device)

        yz_sample = torch.cat((y_sample, z_sample), dim=1)

        inn_output = inn_model.inverse(yz_sample)

        inn_output = (inn_output * (max_X_40 - min_X_40) +
            min_X_40)


        x_pred = model.decode(inn_output)
        x_pred = x_pred.squeeze(1)

        x_pred = x_pred * (max_val - min_val) + min_val
        x_true = x_true * (max_val - min_val) + min_val
        total_loss_sre = SRE_loss(x_pred,x_true)


total_loss_rmse = np.array(total_loss_rmse)
total_loss_nmae = np.array(total_loss_nmae)
total_loss_tre = np.array(total_loss_tre)
total_loss_sre = np.array(total_loss_sre)

# Save losses for ploting
np.save('./Results/INN_generative/inn_gen_rmse_loss.npy',
```

```
    total_loss_rmse)
np.save('./Results/INN_generative/inn_gen_nmae_loss.npy',
    total_loss_nmae)
np.save('./Results/INN_generative/inn_gen_tre_loss.npy',
    total_loss_tre)
np.save('./Results/INN_generative/inn_gen_sre_loss.npy',
    total_loss_sre)

# Calculate the requested metrics
print("Mean loss (RMSE) is", np.mean(total_loss_rmse))
print("Mean loss (NMAE) is", np.mean(total_loss_nmae))
print("Mean loss (TRE) is", np.mean(total_loss_tre))
print("Mean loss (SRE) is", np.mean(total_loss_sre))
print()
print("Median of loss (RMSE) is", np.median(total_loss_rmse))
print("Median of loss (NMAE) is", np.median(total_loss_nmae))
print("Median of loss (TRE) is", np.median(total_loss_tre))
print("Median of loss (SRE) is", np.median(total_loss_sre))
print()
print("Std of loss (RMSE) is", np.std(total_loss_rmse))
print("Std of loss (NMAE) is", np.std(total_loss_nmae))
print("Std of loss (TRE) is", np.std(total_loss_tre))
print("Std of loss (SRE) is", np.std(total_loss_sre))
print()
```

Βιβλιογραφία

[1] Cisco Systems, Inc. *Cisco Annual Internet Report (2018-2023)*. 2023. URL: https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/index.html.

[2] Paul Tune, Matthew Roughan, H Haddadi και O Bonaventure. "Internet traffic matrices: A primer". Στο: *Recent Advances in Networking* 1 (2013), σσ. 1–56.

[3] Grigorios Kakkavas, Michail Kalntis, Vasileios Karyotis και Symeon Papavassiliou. "Future Network Traffic Matrix Synthesis and Estimation Based on Deep Generative Models". Στο: *2021 International Conference on Computer Communications and Networks (ICCCN)*. 2021, σσ. 1–8. DOI: 10.1109/ICCCN52240.2021.9522222.

[4] Grigorios Kakkavas, Despoina Gkatzioura, Vasileios Karyotis και Symeon Papavassiliou. "A Review of Advanced Algebraic Approaches Enabling Network Tomography for Future Network Infrastructures". en. Στο: *Future Internet* 12.2 (Ιαν. 2020), σ. 20. ISSN: 1999-5903. DOI: 10.3390/fi12020020. URL: https://www.mdpi.com/1999-5903/12/2/20.

[5] Yann LeCun, Yoshua Bengio και Geoffrey Hinton. "Deep learning". en. Στο: *Nature* 521.7553 (Μάι. 2015), σσ. 436–444. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539.

[6] Grigorios Kakkavas, Adamantia Stamou, Vasileios Karyotis και Symeon Papavassiliou. "Network Tomography for Efficient Monitoring in SDN-Enabled 5G Networks and Beyond: Challenges and Opportunities". Στο: *IEEE Communications Magazine* 59.3 (2021), σσ. 70–76. DOI: 10.1109/MCOM.001.2000458.

[7] Matthew Roughan. "Robust Network Planning". Στο: *Guide to Reliable Internet Services and Applications*. Επιμέλεια υπό Charles R. Kalmanek, Sudip Misra και Yang (Richard) Yang. London: Springer London, 2010,

σσ. 137–177. ISBN: 978-1-84882-828-5. DOI: 10.1007/978-1-84882-828-5_5. URL: https://doi.org/10.1007/978-1-84882-828-5_5.

[8] Hong Huang, Hussein Al-Azzawi και Hajar Brani. *Network Traffic Anomaly Detection*. 2014. arXiv: 1402.0856 [cs.CR]. URL: https://arxiv.org/abs/1402.0856.

[9] Matthew Roughan κ.ά. "Experience in measuring backbone traffic variability: models, metrics, measurements and meaning". Στο: *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment*. IMW '02. Marseille, France: Association for Computing Machinery, 2002, σσ. 91–92. ISBN: 158113603X. DOI: 10.1145/637201.637213. URL: https://doi.org/10.1145/637201.637213.

[10] Rick Hofstede κ.ά. "Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX". Στο: *IEEE Communications Surveys & Tutorials* 16.4 (2014), σσ. 2037–2064. DOI: 10.1109/COMST.2014.2321898.

[11] Lynton Ardizzone κ.ά. "Analyzing Inverse Problems with Invertible Neural Networks". Στο: (2018). DOI: 10.48550/ARXIV.1808.04730. URL: https://arxiv.org/abs/1808.04730.

[12] Y. Zhang. *Abilene network topology data and traffic traces*. Online. 2004. URL: https://www.cs.utexas.edu/~yzhang/research/AbileneTM/.

[13] A. Coates, A.O. Hero Iii, R. Nowak και Bin Yu. "Internet tomography". Στο: *IEEE Signal Processing Magazine* 19.3 (Μάι. 2002), σσ. 47–65. ISSN: 10535888. DOI: 10.1109/79.998081. URL: http://ieeexplore.ieee.org/document/998081/.

[14] Grigorios Kakkavas, Vasileios Karyotis και Symeon Papavassiliou. "A Distance-based Agglomerative Clustering Algorithm for Multicast Network Tomography". Στο: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. 2020, σσ. 1–7. DOI: 10.1109/ICC40277.2020.9149412.

[15] R. Caceres, N.G. Duffield, J. Horowitz και D.F. Towsley. "Multicast-based inference of network-internal loss characteristics". Στο: *IEEE Transactions on Information Theory* 45.7 (1999), σσ. 2462–2480. DOI: 10.1109/18.796384.

[16] F. Lo Presti, N.G. Duffield, J. Horowitz και D. Towsley. "Multicast-based inference of network-internal delay distributions". Στο: *IEEE/ACM Transactions on Networking* 10.6 (2002), σσ. 761–775. DOI: 10.1109/TNET.2002.805026.

[17] Bikash Kumar Dey, D. Manjunath και Supriyo Chakraborty. “Estimating Network Link Characteristics using Packet-Pair Dispersion: A Discrete Time Queueing Theoretic View”. Στο: *CoRR* abs/0911.3528 (2009). arXiv: 0911.3528. URL: http://arxiv.org/abs/0911.3528.

[18] Grigorios Kakkavas, Vasileios Karyotis και Symeon Papavassiliou. “Topology Inference and Link Parameter Estimation Based on End-to-End Measurements”. Στο: *Future Internet* 14.2 (2022). ISSN: 1999-5903. DOI: 10.3390/fi14020045. URL: https://www.mdpi.com/1999-5903/14/2/45.

[19] R.M. Castro, M.J. Coates και R.D. Nowak. “Likelihood based hierarchical clustering”. Στο: *IEEE Transactions on Signal Processing* 52.8 (2004), σσ. 2308–2321. DOI: 10.1109/TSP.2004.831124.

[20] Mark Coates κ.ά. “Maximum likelihood network topology identification from edge-based unicast measurements”. en. Στο: *ACM SIGMETRICS Performance Evaluation Review* 30.1 (Ιούν. 2002), σσ. 11–20. ISSN: 0163-5999. DOI: 10.1145/511399.511337. URL: https://dl.acm.org/doi/10.1145/511399.511337.

[21] M. Coates και R. Nowak. “Networks for networks: Internet analysis using graphical statistical models”. Στο: *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No.00TH8501)*. Τόμ. 2. 2000, 755–764 vol.2. DOI: 10.1109/NNSP.2000.890155.

[22] J. D. Case, M. Fedor, M. L. Schoffstall και J. R. Davin. *A simple network management protocol (SNMP)*. Technical Report RFC 1157. IETF, Μάι. 1990. URL: http://www.ietf.org/rfc/rfc1157.txt.

[23] Y. Vardi. “Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data”. en. Στο: *Journal of the American Statistical Association* 91.433 (Μαρ. 1996), σσ. 365–377. ISSN: 0162-1459, 1537-274X. DOI: 10.1080/01621459.1996.10476697. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1996.10476697.

[24] Scott Vander Wiel Jin Cao Drew Davis και Bin Yu. “Time-Varying Network Tomography: Router Link Data”. Στο: *Journal of the American Statistical Association* 95.452 (2000), σσ. 1063–1075. DOI: 10.1080/01621459.2000.10474303. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.2000.10474303. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.2000.10474303.

[25] Yin Zhang, Matthew Roughan, Nick Duffield και Albert Greenberg. “Fast accurate computation of large-scale IP traffic matrices from link loads”. en. Στο: *Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. San Diego CA USA: ACM, Ιούν. 2003, σσ. 206–217. ISBN: 9781581136647. DOI: 10.1145/781027.781053. URL: https://dl.acm.org/doi/10.1145/781027.781053.

[26] Augustin Soule κ.ά. “How to identify and estimate the largest traffic matrix elements in a dynamic environment”. en. Στο: *Proceedings of the joint international conference on Measurement and modeling of computer systems*. New York NY USA: ACM, Ιούν. 2004, σσ. 73–84. ISBN: 9781581138733. DOI: 10.1145/1005686.1005698. URL: https://dl.acm.org/doi/10.1145/1005686.1005698.

[27] Konstantina Papagiannaki, Nina Taft και Anukool Lakhina. “A distributed approach to measure IP traffic matrices”. Στο: *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*. IMC ’04. Taormina, Sicily, Italy: Association for Computing Machinery, 2004, σσ. 161–174. ISBN: 1581138210. DOI: 10.1145/1028788.1028808. URL: https://doi.org/10.1145/1028788.1028808.

[28] Augustin Soule κ.ά. “Traffic matrices: balancing measurements, inference and modeling”. Στο: *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. SIGMETRICS ’05. Banff, Alberta, Canada: Association for Computing Machinery, 2005, σσ. 362–373. ISBN: 1595930221. DOI: 10.1145/1064212.1064259. URL: https://doi.org/10.1145/1064212.1064259.

[29] Warren S. McCulloch και Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. en. Στο: *The Bulletin of Mathematical Biophysics* 5.4 (Δεκ. 1943), σσ. 115–133. ISSN: 0007-4985, 1522-9602. DOI: 10.1007/BF02478259. URL: http://link.springer.com/10.1007/BF02478259.

[30] Dingde Jiang κ.ά. “Joint time–frequency sparse estimation of large-scale network traffic”. Στο: *Computer Networks* 55.15 (2011), σσ. 3533–3547. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2011.06.027. URL: https://www.sciencedirect.com/science/article/pii/S138912861100257X.

[31] A. Omidvar και H. S. Shahhoseini. “Intelligent IP traffic matrix estimation by neural network and genetic algorithm”. Στο: *2011 IEEE 7th International*

Symposium on Intelligent Signal Processing. 2011, σσ. 1–6. DOI: 10.1109/WISP.2011.6051689.

[32] Dingde Jiang και Guangmin Hu. "A Novel Approach to Large-Scale IP Traffic Matrix Estimation Based on RBF Neural Network". Στο: *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*. ISSN: 2161-9654. Οκτ. 2008, σσ. 1–4. DOI: 10.1109/WiCom.2008.1068. URL: https://ieeexplore.ieee.org/document/4678976/.

[33] Dingde Jiang κ.ά. "How to reconstruct end-to-end traffic based on time-frequency analysis and artificial neural network". Στο: *AEU - International Journal of Electronics and Communications* 68.10 (2014), σσ. 915–925. ISSN: 1434-8411. DOI: https://doi.org/10.1016/j.aeue.2014.04.011. URL: https://www.sciencedirect.com/science/article/pii/S1434841114001071.

[34] Syed Saiq Hussain, Muhammad Arif Sultan, Sameer Qazi και Mehmood Ameer. "Intelligent Traffic Matrix Estimation Using LevenBerg-Marquardt Artificial Neural Network of Large Scale IP Network". Στο: *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*. 2019, σσ. 1–5. DOI: 10.1109/MACS48846.2019.9024765.

[35] Haifeng Zhou, Liansheng Tan, Qian Zeng και Chunming Wu. "Traffic matrix estimation: A neural network approach with extended input and expectation maximization iteration". Στο: *Journal of Network and Computer Applications* 60 (2016), σσ. 220–232. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2015.11.013. URL: https://www.sciencedirect.com/science/article/pii/S1084804515002854.

[36] Laisen Nie, Dingde Jiang, Lei Guo και Shui Yu. "Traffic matrix prediction and estimation based on deep learning in large-scale IP backbone networks". Στο: *Journal of Network and Computer Applications* 76 (2016), σσ. 16–22. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2016.10.006. URL: https://www.sciencedirect.com/science/article/pii/S1084804516302351.

[37] Laisen Nie κ.ά. "Traffic Matrix Prediction and Estimation Based on Deep Learning for Data Center Networks". Στο: *2016 IEEE Globecom Workshops (GC Wkshps)*. 2016, σσ. 1–6. DOI: 10.1109/GLOCOMW.2016.7849067.

[38] Y. Lecun, L. Bottou, Y. Bengio και P. Haffner. "Gradient-based learning applied to document recognition". Στο: *Proceedings of the IEEE* 86.11 (1998), σσ. 2278–2324. DOI: 10.1109/5.726791.

[39] Mohsen Emami, Reza Akbari, Reza Javidan και Ali Zamani. “A new approach for traffic matrix estimation in high load computer networks based on graph embedding and convolutional neural network”. en. Στο: *Transactions on Emerging Telecommunications Technologies* 30.6 (Ιούν. 2019), e3604. ISSN: 2161-3915, 2161-3915. DOI: 10.1002/ett.3604. URL: https://onlinelibrary.wiley.com/doi/10.1002/ett.3604.

[40] Harshvardhan GM, Mahendra Kumar Gourisaria, Manjusha Pandey και Siddharth Swarup Rautaray. “A comprehensive survey and analysis of generative models in machine learning”. Στο: *Computer Science Review* 38 (2020), σ. 100285. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2020.100285. URL: https://www.sciencedirect.com/science/article/pii/S1574013720303853.

[41] Shenghe Xu, Murali Kodialam, T. V. Lakshman και Shivendra S. Panwar. “Learning Based Methods for Traffic Matrix Estimation From Link Measurements”. Στο: *IEEE Open Journal of the Communications Society* 2 (2021), σσ. 488–499. DOI: 10.1109/OJCOMS.2021.3062636.

[42] Grigorios Kakkavas, Nikolaos Fryganiotis, Vasileios Karyotis και Symeon Papavassiliou. “Generative Deep Learning Techniques for Traffic Matrix Estimation From Link Load Measurements”. Στο: *IEEE Open Journal of the Communications Society* 5 (2024), σσ. 1029–1046. DOI: 10.1109/OJCOMS.2024.3358740.

[43] Ashish Vaswani κ.ά. “Attention Is All You Need”. Στο: (2017). DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[44] Xingjian SHI κ.ά. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”. Στο: *Advances in Neural Information Processing Systems*. Επιμέλεια υπό C. Cortes κ.ά. Τόμ. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf.

[45] Alessio Sacco, Flavio Esposito και Guido Marchetto. “Completing and Predicting Internet Traffic Matrices Using Adversarial Autoencoders and Hidden Markov Models”. Στο: *IEEE Transactions on Network and Service Management* 20.3 (2023), σσ. 2244–2258. DOI: 10.1109/TNSM.2023.3270166.

[46] Xinyu Yuan κ.ά. “Traffic Matrix Estimation based on Denoising Diffusion Probabilistic Model”. Στο: *2023 IEEE Symposium on Computers and Communications (ISCC)*. 2023, σσ. 316–322. DOI: 10.1109/ISCC58397.2023.10218016.

[47] Diederik P Kingma και Max Welling. “Auto-Encoding Variational Bayes”. Στο: (2013). DOI: 10.48550/ARXIV.1312.6114. URL: https://arxiv.org/abs/1312.6114.

[48] Ian Goodfellow κ.ά. “Generative Adversarial Nets”. Στο: *Advances in Neural Information Processing Systems*. Επιμέλεια υπό Z. Ghahramani κ.ά. Τόμ. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[49] Laurent Dinh, David Krueger και Yoshua Bengio. “NICE: Non-linear Independent Components Estimation”. Στο: (2014). DOI: 10.48550/ARXIV.1410.8516. URL: https://arxiv.org/abs/1410.8516.

[50] Laurent Dinh, Jascha Sohl-Dickstein και Samy Bengio. “Density estimation using Real NVP”. Στο: (2016). DOI: 10.48550/ARXIV.1605.08803. URL: https://arxiv.org/abs/1605.08803.

[51] Durk P Kingma και Prafulla Dhariwal. “Glow: Generative Flow with Invertible 1x1 Convolutions”. Στο: *Advances in Neural Information Processing Systems*. Επιμέλεια υπό S. Bengio κ.ά. Τόμ. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf.

[52] Vinod Nair και Geoffrey E. Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines”. Στο: *Proceedings of the 27th International Conference on Machine Learning (ICML)*. Haifa, Israel: Omnipress, 2010, σσ. 807–814. ISBN: 9781605589077. URL: https://www.cs.toronto.edu/~hinton/absps/reluICML.pdf.

[53] Andrew L. Maas, Awni Y. Hannun και Andrew Y. Ng. “Rectifier Nonlinearities Improve Neural Network Acoustic Models”. Στο: *Proceedings of the 30th International Conference on Machine Learning (ICML)*. Atlanta, Georgia, USA, 2013. URL: http://robotics.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.

[54] Arthur Gretton κ.ά. “A Kernel Two-Sample Test”. Στο: *Journal of Machine Learning Research* 13.Mar (2012), σσ. 723–773. URL: https://www.jmlr.org/papers/volume13/gretton12a/gretton12a.pdf.

[55] Muhammad Asim κ.ά. “Invertible generative models for inverse problems: mitigating representation error and dataset bias”. Στο: *Proceedings of the 37th International Conference on Machine Learning*. Επιμέλεια υπό Hal Daumé III και Aarti Singh. Τόμ. 119. Proceedings of Machine

Learning Research. PMLR, 13–18 Jul 2020, σσ. 399–409. URL: https://proceedings.mlr.press/v119/asim20a.html.

[56] Lynton Ardizzone κ.ά. "Guided Image Generation with Conditional Invertible Neural Networks". Στο: (2019). DOI: 10.48550/ARXIV.1907.02392. URL: https://arxiv.org/abs/1907.02392.

[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. "Deep Residual Learning for Image Recognition". Στο: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Ιούν. 2016.

[58] Jens Behrmann κ.ά. "Invertible Residual Networks". Στο: *Proceedings of the 36th International Conference on Machine Learning*. Επιμέλεια υπό Kamalika Chaudhuri και Ruslan Salakhutdinov. Τόμ. 97. Proceedings of Machine Learning Research. PMLR, Σεπτ. 2019, σσ. 573–582. URL: https://proceedings.mlr.press/v97/behrmann19a.html.

[59] Will Grathwohl κ.ά. "FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models". Στο: (2018). DOI: 10.48550/ARXIV.1810.01367. URL: https://arxiv.org/abs/1810.01367.

[60] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt και David K Duvenaud. "Neural Ordinary Differential Equations". Στο: *Advances in Neural Information Processing Systems*. Επιμέλεια υπό S. Bengio κ.ά. Τόμ. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

[61] M.F. Hutchinson. "A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines". Στο: *Communications in Statistics - Simulation and Computation* 18.3 (1989), σσ. 1059–1076. DOI: 10.1080/03610918908812806. eprint: https://doi.org/10.1080/03610918908812806. URL: https://doi.org/10.1080/03610918908812806.

[62] Jakob Kruse, Lynton Ardizzone, Carsten Rother και Ullrich Köthe. "Benchmarking Invertible Architectures on Inverse Problems". Στο: (2021). DOI: 10.48550/ARXIV.2101.10763. URL: https://arxiv.org/abs/2101.10763 (επίσκεψη 08/08/2024).

[63] Durk P Kingma κ.ά. "Improved Variational Inference with Inverse Autoregressive Flow". Στο: *Advances in Neural Information Processing Systems*. Επιμέλεια υπό D. Lee κ.ά. Τόμ. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/ddeebdeefdb7e7e7a697e1c3e3d8ef54-Paper.pdf.

[64] Yunfei Teng και Anna Choromanska. “Invertible Autoencoder for Domain Adaptation”. Στο: *Computation* 7.2 (2019). ISSN: 2079-3197. DOI: 10.3390/computation7020020. URL: https://www.mdpi.com/2079-3197/7/2/20.

[65] Kihyuk Sohn, Honglak Lee και Xinchen Yan. “Learning Structured Output Representation using Deep Conditional Generative Models”. Στο: *Advances in Neural Information Processing Systems*. Επιμέλεια υπό C. Cortes κ.ά. Τόμ. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf.

[66] Laurens Van Der Maaten, Eric O Postma, H Jaap Van Den Herik κ.ά. “Dimensionality reduction: A comparative review”. Στο: *Journal of Machine Learning Research* 10.66-71 (2009), σ. 13.

[67] Dor Bank, Noam Koenigstein και Raja Giryes. “Autoencoders”. Στο: *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*. Επιμέλεια υπό Lior Rokach, Oded Maimon και Erez Shmueli. Cham: Springer International Publishing, 2023, σσ. 353–374. ISBN: 978-3-031-24628-9. DOI: 10.1007/978-3-031-24628-9_16. URL: https://doi.org/10.1007/978-3-031-24628-9_16.

[68] Xiaojiao Mao, Chunhua Shen και Yu-Bin Yang. “Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections”. Στο: *Advances in Neural Information Processing Systems*. Επιμέλεια υπό D. Lee κ.ά. Τόμ. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/0ed9422357395a0d4879191c66f4faa2-Paper.pdf.

[69] Dana H. Ballard. “Modular learning in neural networks”. Στο: *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1*. AAAI’87. Seattle, Washington: AAAI Press, 1987, σσ. 279–284. ISBN: 0934613427.

[70] Stuart Russell και Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd. Upper Saddle River: Pearson, 1 Δεκ. 2009, σ. 1152. ISBN: 978-0-13-604259-4.

[71] Peter J. Huber. “Robust Estimation of a Location Parameter”. Στο: *Breakthroughs in Statistics: Methodology and Distribution*. Επιμέλεια υπό Samuel Kotz και Norman L. Johnson. New York, NY: Springer New York, 1992, σσ. 492–518. ISBN: 978-1-4612-4380-9. DOI: 10.1007/978-1-4612-4380-9_35. URL: https://doi.org/10.1007/978-1-4612-4380-9_35.

[72] Raúl Rojas. “The Backpropagation Algorithm”. Στο: *Neural Networks: A Systematic Introduction*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, σσ. 149–182. ISBN: 978-3-642-61068-4. DOI: 10.1007/978-3-642-61068-4_7. URL: https://doi.org/10.1007/978-3-642-61068-4_7.

[73] Diederik P. Kingma και Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

[74] Jun Han και Claudio Moraga. “The influence of the sigmoid function parameters on the speed of backpropagation learning”. Στο: *From Natural to Artificial Neural Computation*. Επιμέλεια υπό José Mira και Francisco Sandoval. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, σσ. 195–201. ISBN: 978-3-540-49288-7.

[75] B.L. Kalman και S.C. Kwasny. “Why tanh: choosing a sigmoidal function”. Στο: *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*. Τόμ. 4. 1992, 578–581 vol.4. DOI: 10.1109/IJCNN.1992.227257.

[76] Sepp Hochreiter. “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions”. Στο: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 06.02 (1998), σσ. 107–116. DOI: 10.1142/S0218488598000094. eprint: https://doi.org/10.1142/S0218488598000094. URL: https://doi.org/10.1142/S0218488598000094.

[77] Scott C. Douglas και Jiutian Yu. “Why RELU Units Sometimes Die: Analysis of Single-Unit Error Backpropagation in Neural Networks”. Στο: *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. 2018, σσ. 864–868. DOI: 10.1109/ACSSC.2018.8645556.

[78] Prajit Ramachandran, Barret Zoph και Quoc V. Le. *Searching for Activation Functions*. 2017. arXiv: 1710.05941 [cs.NE]. URL: https://arxiv.org/abs/1710.05941.

[79] George Philipp, Dawn Song και Jaime G. Carbonell. *The exploding gradient problem demystified - definition, prevalence, impact, origin, tradeoffs, and solutions*. 2018. arXiv: 1712.05577 [cs.LG]. URL: https://arxiv.org/abs/1712.05577.

[80] Xavier Glorot και Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. Στο: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Επιμέλεια υπό Yee Whye Teh και Mike Titterington. Τόμ. 9. Proceedings of Machine

Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, σσ. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a.html.

[81] Kaiming He, Xiangyu Zhang, Shaoqing Ren και Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". Στο: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Δεκ. 2015.

[82] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2017. arXiv: 1609.04747 [cs.LG]. URL: https://arxiv.org/abs/1609.04747.

[83] Jarek Duda. *SGD momentum optimizer with step estimation by online parabola model*. 2019. arXiv: 1907.07063 [cs.LG]. URL: https://arxiv.org/abs/1907.07063.

[84] Nikhil Ketkar και Jojo Moolayil. "Introduction to PyTorch". Στο: *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch*. Berkeley, CA: Apress, 2021, σσ. 27–91. ISBN: 978-1-4842-5364-9. DOI: 10.1007/978-1-4842-5364-9_2. URL: https://doi.org/10.1007/978-1-4842-5364-9_2.

[85] T. Kluyver κ.ά. "Jupyter Notebooks – a publishing format for reproducible computational workflows". Στο: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Επιμέλεια υπό F. Loizides και B. Schmidt. IOS Press, 2016, σσ. 87–90.

[86] S. Kullback και R. A. Leibler. "On Information and Sufficiency". Στο: *The Annals of Mathematical Statistics* 22.1 (1951), σσ. 79–86. ISSN: 00034851. URL: http://www.jstor.org/stable/2236703 (επίσκεψη 24/08/2024).

[87] HongYun Cai, Vincent W. Zheng και Kevin Chen-Chuan Chang. "A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications". Στο: *IEEE Transactions on Knowledge and Data Engineering* 30.9 (2018), σσ. 1616–1637. DOI: 10.1109/TKDE.2018.2807452.

[88] Franco Scarselli κ.ά. "The Graph Neural Network Model". Στο: *IEEE Transactions on Neural Networks* 20.1 (2009), σσ. 61–80. DOI: 10.1109/TNN.2008.2005605.

[89] Steve Uhlig, Bruno Quoitin, Jean Lepropre και Simon Balon. “Providing public intradomain traffic matrices to the research community”. Στο: *SIGCOMM Comput. Commun. Rev.* 36.1 (Ιαν. 2006), σσ. 83–86. ISSN: 0146-4833. DOI: 10.1145/1111322.1111341. URL: https://doi.org/10.1145/1111322.1111341.

[90] K. Cho, K. Mitsuya και A. Kato. “Traffic data repository at the WIDE project”. Στο: *Proceedings of the USENIX Annual Technical Conference (USENIX ATC)*. 2000, σσ. 263–270.

[91] Martín Abadi κ.ά. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016. arXiv: 1603.04467 [cs.DC]. URL: https://arxiv.org/abs/1603.04467.

[92] David Luebke. “CUDA: Scalable parallel programming for high-performance scientific computing”. Στο: *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. 2008, σσ. 836–838. DOI: 10.1109/ISBI.2008.4541126.

[93] Python Software Foundation. *Python Programming Language*. 2024. URL: https://www.python.org/.

[94] Standard C++ Foundation. *The C++ Programming Language*. 2024. URL: https://isocpp.org/.