ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Εφαρμογή Τεχνικών Ανάλυσης Μεγάλων Δεδομένων σε Προβλήματα Μεταφορών και Αστροφυσικής

Συγγραφέας:

Νεκταρία Χ. ΑΝΤΩΝΙΟΥ

Επιβλέποντες καθηγητές:

Δρ. Νικόλαος ΧΡΗΣΤΑΚΗΣ

Καθ. Ανδρέας ΖΕΖΑΣ

Σχολή Θετικών και Τεχνολογικών Επιστημών

Τμήμα Φυσικής

Ηράκλειο, 23, Ιουνίου 2017

«Δεν μιλάω όπως γράφω, δεν γράφω όπως σκέφτομαι, δεν σκέφτομαι όπως θα έπρεπε να σκέφτομαι και γι’ αυτό τα πάντα προχωρούν μέσα σε βαθύ σκοτάδι.»

Γιοχάνες Κέπλερ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Σχολή Θετικών και Τεχνολογικών Επιστημών

Τμήμα Φυσικής

Εφαρμογή Τεχνικών Ανάλυσης Μεγάλων Δεδομένων σε Προβλήματα Μεταφορών και Αστροφυσικής

A Big Data Analytics Application in Transportation and Astrophysics

Νεκταρία Χ. Αντωνίου
Πτυχίο Φυσικής

Περίληψη

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η επεξεργασία και ανάλυση «Μεγάλων Δεδομένων». Για τον σκοπό αυτό δημιουργήθηκε ένα ολοκληρωμένο υπολογιστικό πλαίσιο επεξεργασίας και ανάλυσης Μεγάλων Δεδομένων. Το πλαίσιο αυτό είναι σε θέση να εκτελεί τόσο την προεπεξεργασία των δεδομένων, όσο και την ανάλυση τους για εξαγωγή αποτελεσμάτων. Για το λόγο αυτό, στο υπολογιστικό πλαίσιο η διαδικασία αποτελείται από δύο στάδια: το στάδιο της προεπεξεργασίας και προετοιμασίας των δεδομένων και το στάδιο της ανάλυσής τους. Στο πρώτο στάδιο, γίνεται χρήση της πλατφόρμας Hadoop, ενώ το δεύτερο στάδιο υλοποιείται με τη βοήθεια Τεχνητών Νευρωνικών Δικτύων (ΤΝΔ). Αρχικά, γίνεται αναλυτική περιγραφή των εργαλείων που συνθέτουν το υπολογιστικό πλαίσιο ανάλυσης Μεγάλων Δεδομένων. Στη συνέχεια παρουσιάζεται η εφαρμογή του υπολογιστικού πλαισίου σε δύο προβλήματα, ένα πρόβλημα μεταφορών και ένα πρόβλημα αστροφυσικής. Στο πρώτο πρόβλημα, από ένα μεγάλο όγκο δεδομένων από όλα τα αεροδρόμια των ΗΠΑ ανακτώνται δεδομένα για ένα συγκεκριμένο αεροδρόμιο και αυτά αναλύονται, έτσι ώστε να υπολογιστεί για ένα μεγάλο χρονικό διάστημα (18 ετών) η μέση καθυστέρηση πτήσεων ανά ημέρα και αυτά τα αποτελέσματα να χρησιμοποιηθούν για να προβλεφθεί ο συνολικός χρόνος που απαιτείται για την εξομάλυνση και την επιστροφή των λειτουργιών του αεροδρομίου στους

προγραμματισμένους τους χρόνους. Στο δεύτερο πρόβλημα, βασιζόμενοι στα χρώματα των γαλαξιών, επιχειρήθηκε μέσω των ΤΝΔ η πρόβλεψη της μορφολογίας γαλαξιών που έχουν παρατηρηθεί με την χρήση του τηλεσκόπιου SDSS («Sloan Digital Sky Survey»). Τα εξαγόμενα αποτελέσματα αποτιμώνται και εξάγονται συμπεράσματα για την ικανότητα του υπολογιστικού πλαισίου που παρουσιάστηκε να ανακτήσει και να αναλύσει επιτυχώς Μεγάλα Δεδομένα. Τέλος, δίδονται κατευθύνσεις για περαιτέρω μελέτη και βελτίωση του υλοποιηθέντος πλαισίου.

The main subject of this thesis is the development of a complete computational framework for «Big Data» analytics. This framework is able to perform the pre-processing and manipulation of data, as well as their analysis. For this reason, the computational framework requires two stages: the pre-processing and preparation stage of the data and the stage of data analysis. The first stage utilizes Hadoop platform, while the second stage utilizes Artificial Neural Networks (ANN). Initially, a detailed description of the tools that are used in the computational framework is given. Then, the application of the presented framework is given on a transportation problem and an astrophysics problem. For the transportation problem, a large amount of data from all US airports were retrieved and data for one particular airport were extracted, in order to identify mean flight delays per day. Then, these data were used to predict the time required for airport operations to return back to their scheduled times (i.e., normalization time). For the second problem, an attempt was made, via ANN, to predict the morphology of galaxies based on spectral data from Sloan Digital Sky Survey (SDSS). The results from these two applications are presented and discussed extensively. Finally, conclusions are drawn on the ability of the computational framework to handle successfully problems related to Big Data analytics and suggestions for its further improvement are made.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Αναπληρωτή Καθηγητή του Τμήματος Φυσική του Πανεπιστημίου Κρήτης και υπεύθυνο καθηγητή μου κ. Ζέζα Ανδρέα για την ευκαιρία που μου έδωσε να συνεργαστώ μαζί του. Επίσης, ευχαριστώ θερμά τον επισκέπτη καθηγητή κ. Χρηστάκη Νικόλαο, ο οποίος στάθηκε σημαντικός αρωγός στην προσπάθειά μου και με υποστήριξε σε κάθε φάση της πορείας μου. Τέλος θα ήθελα να ευχαριστήσω τον Καθηγητή και Πρόεδρο του Τμήματος Φυσικής κ. Παπαδάκη Ιωσήφ αλλά και την κ. Παυλίδου Βασιλική, Επίκουρη Καθηγήτρια του Τμήματος Φυσικής που δέχτηκαν να είναι μέλη της τριμελής επιτροπής αξιολόγησης της μεταπτυχιακής μου εργασίας.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες

ΤΝΔ : Τεχνητά Νευρωνικά Δίκτυα
NAS: National Airpot System – Εθνικό Σύστημα Αεροπορίας
SDSS: Sloan Digital Sky Survey

Σύμβολα

BT	λόγος της έντασης της ακτινοβολίας του σφαιροειδούς προς τη συνολική ένταση
F	ένταση ακτινοβολίας (W / m^2)
h	κανονικοποιημένη τιμή εισόδου-εξόδου μεταξύ νευρώνων ΤΝΔ μέσω συνάρτησης ενεργοποίησης
I_{del}	αριθμός ημερών που απαιτείται για ομαλοποίηση κατάστασης καθυστέρησης
I_{eff}	δείκτης απόδοσης ΤΝΔ
m	φαινόμενο μέγεθος ουράνιου σώματος
N_i	αριθμός τιμών εισόδου στο ΤΝΔ
N_o	αριθμός τιμών εξόδου στο ΤΝΔ
N_p	αριθμός μοτίβων στο ΤΝΔ
N_w	συνολικός αριθμός συναπτικών βαρών στο ΤΝΔ
T_{cr}	κρίσιμος χρόνος πάνω από τον οποίο μία πτήση λογίζεται ως καθυστερημένη (=15 λεπτά)
T_{del}	μέσος χρόνος καθυστέρησης πτήσης ανά ημέρα
w	συναπτικό βάρος
α	ταχύτητα εκμάθησης ΤΝΔ
ϵ	ελλειπτικότητα γαλαξία
η	ρυθμός εκμάθησης ΤΝΔ

Αφιερωμένο στον παππού μου,

που έφυγε νωρίς.

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 «*Μεγάλα Δεδομένα*»

Με την πάροδο των χρόνων και τη συνεχή εξέλιξη της τεχνολογίας ο όγκος πληροφορίας αυξάνεται με πολύ γρήγορους ρυθμούς και με μεγάλη ποικιλία ως προς το είδος της. Όλη αυτή η πληροφορία περνάει συχνά απαρατήρητη με αποτέλεσμα να χάνουμε δεδομένα που θα μπορούσαν να βοηθήσουν στην εξέλιξη της ανθρωπότητας. Καθώς λοιπόν παρατηρείται ότι χρειάζεται η εκμετάλλευση της πληροφορίας αυτής ολοένα και περισσότεροι τομείς της ανθρωπότητας ασχολούνται με το θέμα αυτό. Η «έκρηξη πληροφορίας» οδήγησε σε αυτό που αποκαλούμε «Μεγάλα Δεδομένα» ή αλλιώς «*Big Data*».

Με τον όρο Μεγάλα Δεδομένα εννοούμε τα σύνολα δεδομένων των οποίων το μέγεθος είναι πέρα των δυνατοτήτων κοινών εργαλείων λογισμικού και υλικού για τη διαχείριση και επεξεργασία τους μέσα σε ένα αποδεκτό χρονικό διάστημα. (Zikopoulos et al., 2012, Zikopoulos et al., 2013)

Η πρώτη εμφάνιση του όρου έγινε το 1997 απο επιστήμονες της NASA. Ανέφεραν ότι αδυνατούσαν να αναπαραστήσουν γραφικά τα σύνολα δεδομένων που κατείχαν, καθώς ήταν τόσο μεγάλα που ήταν ακατόρθωτο να τα αποθηκεύσουν στη κύρια μνήμη, στον τοπικό δίσκο και σε εξωτερικό σκληρό δίσκο. Έτσι δήλωσαν ότι αντιμετώπιζαν πρόβλημα Μεγάλων Δεδομένων. (Cox and Ellsworth, 1997)

Εάν και η έννοια των Μεγάλων Δεδομένων εμφανίστηκε πρόσφατα, τα θεμέλια πάνω στα οποία στηρίχθηκε η εμφάνιση και η εξέλιξη τους, τέθηκαν πολλά χρόνια πριν. Πολύ πριν την εμφάνιση των ηλεκτρονικών υπολογιστών, υπήρξε η ανάγκη να αποθηκεύουμε, αναπαράγουμε και αναλύουμε τις πληροφορίες που κατείχαμε. Ας δούμε όμως τι μας οδήγησε στη σημερινή εποχή των Μεγάλων Δεδομένων.

1.2 Η Ιστορία των Μεγάλων Δεδομένων

Σχήμα 1.1: Η εξέλιξη των Μεγάλων Δεδομένων.

Πηγή: https://www.dezyre.com/article/big-data-timeline-series-of-big-data-evolution/160

Η ιστορία των Μεγάλων Δεδομένων μπορεί να πει κάποιος ότι ξεκινάει χιλιάδες χρόνια πριν (Marr, 2015). Οι πρώτες προσπάθειες καταγραφής δεδομένων χρονολογούνται περίπου 18.000 χρόνια π.Χ., όταν οι άνθρωποι χρησιμοποιούσαν ψηλές ράβδους («*tally sticks*») τις οποίες χάραζαν για να καταγράφουν με το τρόπο αυτό τα αποθέματα τροφής και να παρακολουθούν την επιχειρηματική δραστηριότητα.

Χιλιάδες χρόνια αργότερα, γύρω στο 2000 π.Χ. οι αρχαίοι Βαβυλώνιοι είχαν αναπτύξει πολύ το εμπόριο και χρειάζονταν κάτι να τους βοηθά στους υπολογισμούς τους. Η ανάγκη αυτή τους οδήγησε στο να δημιουργήσουν τον πρώτο υπολογιστή, που δεν ήταν άλλος απο τον γνωστό άβακα. Γύρω στο 300 π.Χ. δημιουργείται η βιβλιοθήκη της Αλεξάνδρειας, η οποία αποτελούσε το μεγαλύτερο κέντρο αποθήκευσης δεδομένων του κόσμου. Το 1900 μ.Χ. ανακαλύπτεται σε ναυάγιο ανοικτά του νησιού Αντικύθηρα ένα αρχαίο τέχνημα που πιστεύεται ότι ήταν ο πρώτος αναλογικός υπολογιστής, ο οποίος χρονολογείται μεταξύ του 200 και 100 π.Χ..

Το 1663, κάνει την εμφάνιση της η στατιστική. Ο John Graunt διεξάγει το πρώτο καταγεγραμμένο πείραμα στατιστικής ανάλυσης, προκειμένου να περιορίσει την εξάπλωση της πανούκλας στην Ευρώπη. Το 1865 χρησιμοποιείται ο όρος «επιχειρηματική ευφυΐα» από τον Richard Millar Devens, στην «*Encyclopaedia of Commercial and Business Anecdotes*», περιγράφοντας πώς ο τραπεζίτης Henry Furnese πλεονεκτούσε έναντι των ανταγωνιστών του λόγω της συλλογής και ανάλυσης

πληροφοριών σχετικά με τις επιχειρηματικές του δραστηριότητες. Αυτό πιστεύεται ότι είναι η πρώτη ανάλυση δεδομένων που χρησιμοποιήθηκε για εμπορικούς σκοπούς. Το 1880 ο Herman Hollerith δημιουργεί μια μηχανή πινακοποίησης η οποία χρησιμοποιεί διάτρητες κάρτες για να μειώσει το φόρτο εργασίας στις απογραφές των Η.Π.Α.. Το 1881 γίνεται γνωστό το «*Hollerith Tabulating Machine»* και ιδρύεται η εταιρεία του Herman Hollerith, η οποία στο μέλλον θα γίνει γνωστή ως *IBM (International Business Machines).*

Το 1926 είναι η πρώιμη φάση της σύγχρονης αποθήκευσης δεδομένων. Ο Nikola Tesla προέβλεψε ότι ο άνθρωπος θα μπορεί να έχει πρόσβαση και να αναλύει τεράστιες ποσότητες δεδομένων χρησιμοποιώντας μια μικρή συσκευή που θα χωράει στη τσέπη του. Δύο χρόνια αργότερα, ο Fritz Pfleumer εφευρίσκει ένα τρόπο μαγνητικής αποθήκευσης δεδομένων, γεγονός το οποίο αποτέλεσε βάση της σύγχρονης τεχνολογίας ψηφιακής αποθήκευσης δεδομένων. Το 1958 εμφανίζεται για πρώτη φορά η έννοια της επιχειρηματικής ευφυΐας, ενώ το 1965 η κυβέρνηση των Η.Π.Α. σχεδιάζει τη κατασκευή του μεγαλύτερου κέντρου αποθήκευσης δεδομένων, το οποίο θα καταγράφει 742 εκατομμύρια αποτυπώματα πολιτών, τα οποία θα είναι καταγεγραμμένα σε μαγνητική ταινία.

Πέντε χρόνια αργότερα, η IBM εισάγει το σχεσιακό μοντέλο βάσης δεδομένων, κάτι που σημαίνει ότι πλέον ο καθένας μπορούσε να χειρίζεται μια βάση δεδομένων. Το 1976 έχουμε τη χρήση Συστημάτων Σχεδιασμού Απαιτήσεων Υλικών («*Material Requirements Planning Systems»*) σε ολόκληρο τον επιχειρηματικό κόσμο.

Και έτσι η πλέον η καταγραφή δεδομένων στις μεγάλες επιχειρήσεις γίνεται με τη χρήση υπολογιστή. Το 1989 γίνεται ενδεχομένως η πρώτη χρήση του όρου Μεγάλα Δεδομένα με τον τρόπο που χρησιμοποιείται και σήμερα από τον Erik Larson -διεθνούς φήμης συγγραφέας- σε ένα άρθρο του στο περιοδικό Harpers. Οι επόμενες χρονιές κρίνονται καθοριστικές για την εξέλιξη του όρου Μεγάλα Δεδομένα. Το 1991, κάνει την εμφάνιση του το Διαδίκτυο επιτρέποντας με τον τρόπο αυτό την εκμετάλλευση όσων πληροφοριών βρίσκονταν ήδη σε κάποιο ιστότοπο και τη φόρτωση πληροφοριών από τον οποιονδήποτε. Το 1997 ο Michael Lesk, κάνει μια δημοσίευση με τίτλο «*How Much Information is there in the World ?»* (Lesk, 1997).

Το 1998, η *Google* παρουσιάζει τη μηχανή αναζήτησης της, η οποία μετατρέπεται στη πιό δημοφιλή παγκοσμίως. Το 2001, ο Doug Laney στη δημοσίευση του με τίτλο «*3D Management: Controlling Data Volume, Velocity and Variety» προσδιορίζει τα τρία χαρακτηριστικά των Μεγάλων Δεδομένων. Το μοντέλο «3V» (Volume, Velocity, Variety)*. Το 2005 κάνει την εμφάνιση του το *WEB 2.0* (Ιστός 2.0). Την
ίδια χρονία αναπτύσσεται από την Apache το Hadoop, ένα ανοιχτού κώδικα λογισμικό επεξεργασίας Μεγάλων Δεδομένων. Το 2008, οι επεξεργαστές κεντρικών μονάδων (*CPU-Central Process Unit*) επεξεργάζονται πλέον 9.57 zettabytes πληροφοριών.

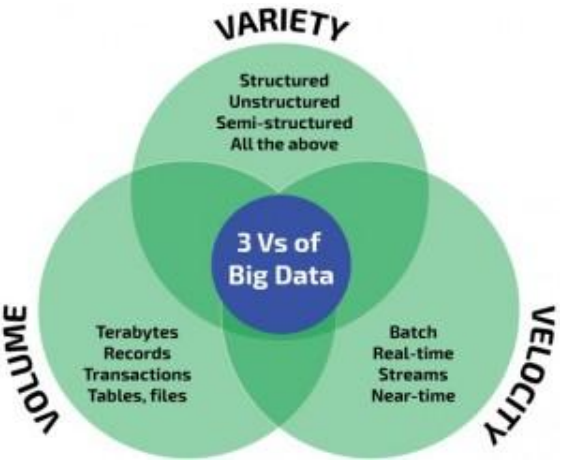
Την επόμενη χρονιά, η μέση Αμερικάνικη εταιρεία με πάνω από 1,000 εργαζόμενους αποθηκεύει περισσότερα απο 200 terabytes δεδομένων, σύμφωνα με την έκθεση «*Big Data: The Next Frontier for*

Innovation, Competition and Productivity by McKinsey Global Institute». Το 2010, ο διευθυντής της *Google* Eric Schmidt δηλώνει ότι: «Τα δεδομένα που παράγονται ανά δύο ημέρες παγκοσμίως, είναι περισσότερα από τα δεδομένα που δημιουργήθηκαν από την αρχή του ανθρώπινου πολιτισμού μέχρι και το 2003».

Σύμφωνα με τα παραπάνω τα Μεγάλα Δεδομένα, υπήρχαν στη ζωή του ανθρώπου εδώ και πολλά χρόνια και αποτελούν μέρος μίας μακράς εξελικτικής διαδικασίας καταγραφής και χρήσης δεδομένων. Τα Μεγάλα Δεδομένα είναι ένα ακόμη βήμα που θα φέρει την αλλαγή στον τρόπο που λειτουργούν οι επιχειρήσεις και η κοινωνία. Παρακάτω θα αναλύσουμε τα ενδιαφέροντα χαρακτηριστικά των Μεγάλων Δεδομένων, καθώς και τις εφαρμογές τους.

1.3 Τα Χαρακτηριστικά των Μεγάλων Δεδομένων

Ο όρος «Μεγάλα Δεδομένα» έχει προσδιοριστεί με διάφορους τρόπους αλλά όλοι έχουν παρόμοιο νόημα. Το 2001 ο Doug Laney προσδιόρισε πρώτος τον όρο αυτό, στην έκθεση του με τίτλο «*3-D Data Management: Controlling Data Volume, Velocity and Variety»*. (Laney, 2001) Η έννοια των Μεγάλων Δεδομένων προσδιορίζεται με το μοντέλο «*3V*». (Σχήμα 1.2)



Σχήμα 1.2: Το Μοντέλο «3Vs».

Πηγή:http://sci2s.ugr.es/sites/default/files/files/TematicWebSites/BigData/v3.png

Το πρώτο «*V*» σχετίζεται με τον όρο *Volume*, ο οποίος αναφέρεται στο τεράστιο μέγεθος που έχουν τα σύνολα δεδομένων (*datasets*). Το δεύτερο «*V*» αφορά τον όρο *Velocity*, που σχετίζεται με την ταχύτητα που αποκτούνται αλλά και χρησιμοποιούνται τα δεδομένα. Το τρίτο «*V*» αφορά τον όρο *Variety*, ο οποίος αναφέρεται στην ποικιλία των δεδομένων που είναι διαθέσιμα προς συλλογή και ανάλυση. Έτσι λοιπόν η αξία της πληροφορίας έγκειται τόσο σε δομημένα όσο και σε μη δομημένα και ημιδομημένα δεδομένα.

Ας προχωρήσουμε όμως τώρα στην ανάλυση καθενός από αυτά τα χαρακτηριστικά γνωρίσματα που καθορίζουν τη δυναμικότητα των Μεγάλων Δεδομένων ξεχωριστά.

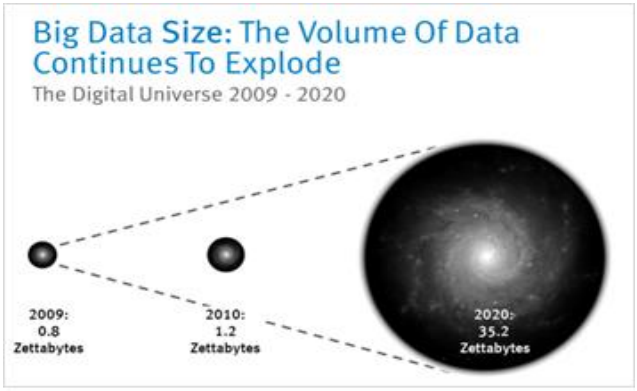
1.3.1 Όγκος (*Volume*)

Ο Όγκος των δεδομένων που αποθηκεύονται σήμερα αυξάνεται αλματωδώς καθημερινά. Με την πάροδο των ετών η αναλογική πληροφορία -έντυπης μορφής πληροφορία- άρχισε να αντικαθίσταται όλο και περισσότερο με ψηφιακή. Πλέον, έχουμε τεράστιες ποσότητες δεδομένων σε μορφή «*video*», ήχου και εικόνων, τόσο σε επιστημονικές εφαρμογές όσο και στα ευρέως διαδεδομένα κοινωνικά δίκτυα. Σε ένα κόσμο, όπου είναι δυνατόν να αποθηκευτούν και να αποκωδικοποιηθούν σχεδόν τα πάντα, από ατομικές πληροφορίες μικρότερης σημασίας μέχρι περιβαλλοντικά, οικονομικά, ιατρικά, πολιτικά δεδομένα, η εξόρυξη σημαντικών πληροφοριών από ένα πλήθος δεδομένων αποτελεί σημαντική συνιστώσα. Όσον αφορά τον Όγκο ως χαρακτηριστικό παρατηρούμε ότι γίνεται χρήση Κρυφής Μνήμης (*Data Cache*), όπου εκεί συνήθως αποθηκεύονται και επεξεργάζονται τα μεγάλου Όγκου δεδομένα πριν αποθηκευθούν στις Βάσεις Δεδομένων. Προκειμένου να συνειδητοποιήσουμε αυτή την αλματώδη αύξηση που λαμβάνει χώρα, ενδεικτικά μπορεί να αναφερθεί ότι το έτος 2010 ο όγκος των δεδομένων που αποθηκευόταν ανερχόταν στα 1,2 zettabytes, ενώ το 2020 ο αντίστοιχος όγκος αναμένεται να ανέλθει στα 35,2 zettabytes. (Σχήμα 1.3)

1.3.2 Ταχύτητα (*Velocity*)

Όπως ακριβώς έχουν αλλάξει τα δεδομένα όσον αφορά τον Όγκο των δεδομένων, με την τεχνολογική εξέλιξη και τη διεύρυνση των διαδικτυακών δυνατοτήτων, κάτι ανάλογο ισχύει και με την Ταχύτητα. Η Ταχύτητα σύμφωνα με μια συμβατική διατύπωση που μπορεί να αποδοθεί όσον αφορά τη συγκεκριμένη συνιστώσα, είναι το μέγεθος μέσω του οποίου μπορεί να περιγραφεί το πόσο γρήγορα εισέρχονται και ανανεώνονται τα ήδη υπάρχοντα δεδομένα και με τον χρόνο που απαιτείται για την επεξεργασία και ανάλυση τους κατά την είσοδο των δεδομένων στο σύστημα. Όσον αφορά το πρώτο μέρος, το ζητούμενο είναι πώς το σύστημα θα δεχθεί, θα φιλτράρει, θα διαχειριστεί και θα αποθηκεύσει τα δεδομένα που έρχονται συνεχώς και ταχύτατα. Το δεύτερο μέρος, αφορά τον απαιτούμενο ρυθμό ώστε να γίνει εξαγωγή πληροφορίας από τα εισερχόμενα δεδομένα. Επιπλέον, δεν

αρκεί μόνο να μπορούμε να αναλύουμε τα δεδομένα και να εξάγουμε πληροφορία απο αυτά σε πραγματικό χρόνο, αλλά είναι απαραίτητο να μπορούμε να εκτελούμε και όλες τις λειτουργίες που ενεργοποιούνται από αυτά σε πραγματικό χρόνο, ώστε η όλη διαδικασία να μην καθυστερεί. Ένα παράδειγμα στο οποίο κρίνεται αναγκαίο αυτό, είναι το χρηματιστήριο. Δεν αρκεί να μπορεί να γίνεται γρήγορη καταγραφή τιμών αλλά και να εκτελούνται γρήγορα και οι κατάλληλες διαδικασίες σύμφωνα με τις τιμές αυτές, όπως η αγοραπωλησία μετοχών όταν αυτές ξεπερνούν ένα όριο τιμών.



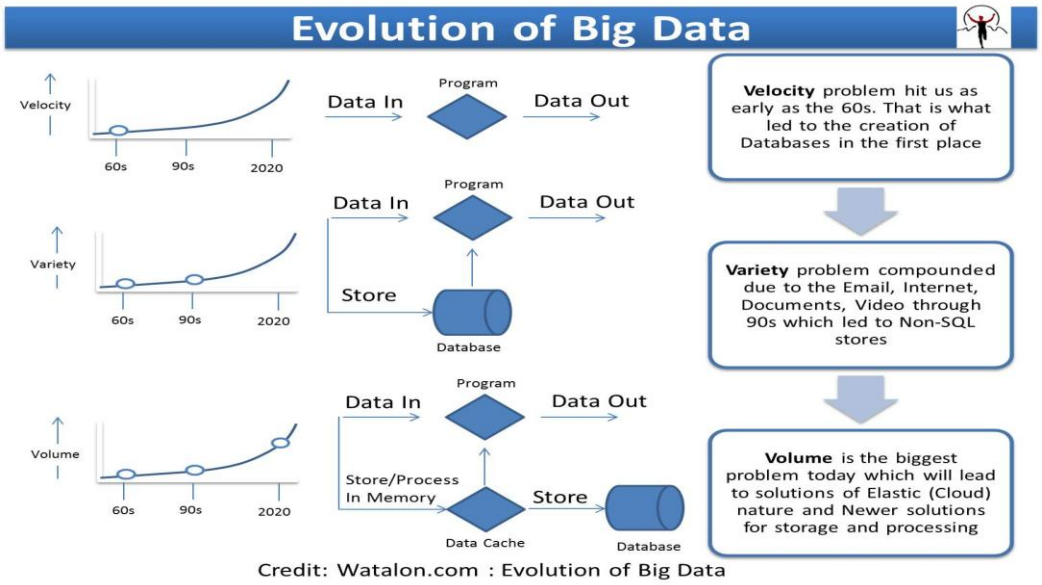
Σχήμα 1.3: Η εξέλιξη του Όγκου των Μεγάλων Δεδομένων.

Πηγή: http://3.bp.blogspot.com/-pCJvZ_IxknI/Vpdy0mcxVkI/AAAAAAAACBs/uZzMZ4Ha2BY/s1600/Key%2BCharacteristics%2Bof%2BBig%2BData.png

1.3.3 Ποικιλία (*Variety*)

Με τον όρο αυτό αναφερόμαστε στη ποικιλία πληροφοριών που μπορεί κάποια αυτόνομη μονάδα ή οργανισμός να καταχωρεί, να επεξεργάζεται και να συνδυάζει δεδομένα διαφορετικών πηγών. Αυτό μας φέρνει αντιμέτωπους όχι μόνο με διαφορετικούς τύπους δεδομένων, αλλά και με διαφορετική δομή μεταξύ ίδιων τύπων. Έτσι δημιουργείται σε πρώτη φάση η ανάγκη να ενσωματωθούν δεδομένα αυστηρώς δομημένα, ημιδομημένα και μη δομημένα. Και σε δεύτερη φάση, ακόμα και αν οι πηγές μας χρησιμοποιούν αυστηρή δόμηση δεδομένων, πιθανόν να χρησιμοποιούν διαφορετική σημασιολογία, ή να υπάρχει ασυμβατότητα μεταξύ τους.

Στο παρακάτω σχήμα (Σχήμα 1.4) απεικονίζονται τα στάδια εξέλιξης των τριών χαρακτηριστικών που είδαμε πιο πάνω και οι προβλέψεις αυτών εώς το 2020.



Σχήμα 1.4: Η εξέλιξη των «3V».

Πηγή: http://watalon.com/wp-content/uploads/2012/02/Evolution_of_Big_Data_Technology.jpg

Από τότε που δόθηκε ο πρώτος ορισμός και το μοντέλο των τριών χαρακτηριστικών, πολλοί που ασχολούνται με τα Μεγάλα Δεδομένα έχουν προσπαθήσει να ενσωματώσουν και άλλα χαρακτηριστικά σε αυτό το μοντέλο. Όπως η Ειλικρίνεια (*Veracity*), Μεταβλητότητα (*Variability*), Αξία (*Value*), Απεικόνιση (*Visualization*) (πχ. Gani et al., 2016). Ωστόσο μόνο τα τρία πρώτα Όγκος, Ταχύτητα, Ποικιλία είναι μέτρα που δείχνουν το μέγεθος των δεδομένων.

1.4 Ποιά Δεδομένα θεωρούνται Μεγάλα;

Σύμφωνα με τους Devlin et al. (2012), έχουν προσδιοριστεί τέσσερις κατηγορίες πληροφοριών που αποτελούν Μεγάλα Δεδομένα:

I) Δεδομένα που δημιουργούνται απο μηχανές (*Machine-generated data*)

II) Δεδομένα καταγραφής του υπολογιστή (*Computer log data*)

III) Δεδομένα μέσων κοινωνικής δικτύωσης

IV) Δεδομένα πολυμέσων

Ας δούμε τώρα την κάθε κατηγορία ξεχωριστά.

1.4.1 Δεδομένα που δημιουργούνται απο μηχανές (*Machine-generated data*)

Σε αυτή την κατηγορία δεδομένων περιέχονται εκείνα που δημιουργούνται από μηχανές (*Machine-generated data*). Για παράδειγμα, στην κατηγορία αυτή συμπεριλαμβάνονται δεδομένα γεωγραφικής θέσης από αντικείμενα (*geolocation data*) όπως κινητές συσκευές. Επίσης περιέχονται και δεδομένα που σχετίζονται με την ταυτοποίηση μέσω ραδιοσυχνοτήτων (*RFID -Radio Frequency Identification Data*).

1.4.2 Δεδομένα καταγραφής του υπολογιστή (*Computer log data*)

Σε αυτή την κατηγορία Μεγάλων Δεδομένων συμπεριλαμβάνονται τα δεδομένα καταγραφής του υπολογιστή, όπως για παράδειγμα ο αριθμός των πατημάτων στο ποντίκι του υπολογιστή (*click*) κατά τη διάρκεια μίας επίσκεψης σε μία ιστοσελίδα (*clickstreams*).

1.4.3 Δεδομένα μέσων κοινωνικής δικτύωσης

Η τρίτη κατηγορία, που είναι ευρέως γνωστή αφορά τις γραπτές πληροφορίες που προέρχονται από τα μέσα κοινωνικής δικτύωσης, όπως το *Facebook* και το *Twitter*.

1.4.4 Δεδομένα Πολυμέσων

Στην τελευταία αυτή κατηγορία Μεγάλων Δεδομένων ανήκουν τα δεδομένα που αντλούνται από πολυμέσα, απο πηγές όπως το *YouTube*, το *Flickr* και άλλες παρόμοιες ιστοσελίδες.

Μέσω αυτής της απλής κατηγοριοποίησης Μεγάλων Δεδομένων γίνεται αμέσως προφανές ότι τα δεδομένα αυτά δεν είναι ομοιογενή. Αυτή τους η διαφορετικότητα είναι πιθανόν να απαιτεί διαφορετικές προσεγγίσεις για την διαχείριση και την επεξεργασία τους.

1.5 Οι τυποί στους οποίους διακρίνονται τα Μεγάλα Δεδομένα

Ο Bill Vorhies (2013) εξηγεί ότι υπάρχουν τρείς τύποι Μεγάλων Δεδομένων, τα δομημένα (*structured*), τα μη δομημένα (*unstructured*) και τα ημιδομημένα (*semi-structured*).

1.5.1 Δομημένα Δεδομένα

Η τρέχουσα αποθήκη δεδομένων περιέχει μόνο δομημένα δεδομένα. Ονομάζονται δομημένα γιατί όταν εισάγονται σε μία σχεσιακή βάση δεδομένων, διατηρούν μια δομή (*structure)*. Οπότε είναι γνωστό το τι σημαίνουν και το που βρίσκονται τα δεδομένα αυτά αλλά και το πώς συνδέονται με τα άλλα δεδομένα που υπάρχουν. Τέτοια δεδομένα είναι ένα κείμενο (π.χ. το όνομα ενός ατόμου) και αριθμοί (π.χ. η ηλικία ενός ατόμου).

1.5.2 Μη Δομημένα Δεδομένα

Μη δομημένα δεδομένα θεωρούνται τα δεδομένα που δεν έχουν μία συγκεκριμένη δομή. Σε αυτή την κατηγορία περιλαμβάνονται δεδομένα που περιέχουν ελεύθερο κείμενο όπως έγγραφα που παράγονται σε μία εταιρεία, αρχεία ήχου, εικόνες και βίντεο. Εάν το αντικείμενο που πρόκειται να αποθηκευτεί δεν φέρει καμία ετικέτα -σχετικά με τα δεδομένα- και δεν έχει συγκεκριμένο σχήμα, οντότητα, γλωσσάριο ή σταθερή οργάνωση θεωρείται μη δομημένο. Ωστόσο, στην ίδια κατηγορία με τα μη δομημένα δεδομένα υπάρχουν πολλοί τύποι δεδομένων που έχουν τουλάχιστον κάποια οργάνωση και αυτά αποκαλούνται ημιδομημένα δεδομένα.

1.5.3 Ημιδομημένα Δεδομένα

Η διαχωριστική γραμμή μεταξύ μη δομημένων και ημιδομημένων δεδομένων είναι λίγο ασαφής. Εάν τα δεδομένα δεν έχουν καμία δομή-ανοργάνωτα- ή φέρουν ετικέτα (π.χ. *XML*[1]) τότε είναι πιο εύκολο να οργανωθούν και να αναλυθούν και αφού είναι πιο εύκολα προσβάσιμα για ανάλυση μπορούν να γίνουν πολυτιμότερα.

1.6 Ανάλυση των Μεγάλων Δεδομένων

Στο παρακάτω σχήμα (Σχήμα 1.5) παρουσιάζονται τα στάδια ανάλυσης των Μεγάλων Δεδομένων.

Σχήμα 1.5: Στάδια Ανάλυσης Μεγάλων Δεδομένων.

Πηγή: https://www.datameer.com/wp-content/uploads/2015/07/img-5-steps.png

[1] *XML* Στην πληροφορική, *Extensible Markup Language* (*XML*) είναι μια γλώσσα σήμανσης που ορίζει ένα σύνολο κανόνων για τα έγγραφα κωδικοποίησης σε μια μορφή που είναι αναγνώσιμη τόσο από τον άνθρωπο όσο και από τη μηχανή.

Αναλυτικά τα στάδια ανάλυσης περιγράφονται πιο κάτω:

I) Ανάκτηση/Καταγραφή Δεδομένων

Το πρώτο βήμα για την ανάλυση των Μεγάλων Δεδομένων είναι η συλλογή των προς επεξεργασία δεδομένων. Είναι σημαντικό η συλλογή αυτών να γίνει σωστά ώστε τα δεδομένα που θα συλλεχθούν να μπορούν να δώσουν τα συμπεράσματα που ζητούνται, και ταυτόχρονα να μην είναι περισσότερα απ' όσα αρκούν για την εξαγωγή τους. Αρχικά χρειάζεται να γίνει φιλτράρισμα των προς επεξεργασία δεδομένων. Στις περισσότερες περιπτώσεις, είναι δύσκολο να αναγνωρίσουμε την «καθαρότητα» των δεδομένων που λαμβάνουν οι διάφορες πηγές.

Έτσι κρίνεται απαραίτητο να αναπτυχθούν τεχνικές οι οποίες να μπορούν να κρίνουν πότε η ύπαρξη «εξωπραγματικών» δεδομένων οφείλεται σε πραγματικές καταστάσεις και πότε αποτελούν μη έγκυρες μετρήσεις. Επίσης είναι συχνό φαινόμενο να παρατηρείται επικάλυψη μεταξύ των δεδομένων που λαμβάνουμε από διαφορετικές πηγές. Σε αυτήν την περίπτωση, είναι σημαντικό κάτι τέτοιο να γίνεται αντιληπτό ώστε να μην αποθηκεύεται παραπάνω πληροφορία άσκοπα. Από τα παραπάνω κρίνεται απαραίτητο το κατάλληλο «φιλτράρισμα» των δεδομένων ώστε να μειώνεται στο ελάχιστο ο όγκος των δεδομένων, χωρίς όμως να χάνεται πολύτιμη πληροφορία.

Μάλιστα στο στάδιο αυτό είναι σημαντικό τα παραπάνω να γίνονται και σε σύντομο χρονικό διάστημα. Καθώς τα δεδομένα που συλλέγονται από την παραπάνω διαδικασία εξακολουθούν να είναι ακόμη μεγάλα σε όγκο, κρίνεται αναγκαίο παράλληλα με τη συλλογή τους να συλλέγεται και πληροφορία που να περιγράφει τα δεδομένα αυτά και τον τρόπο που εκείνα συλλέχτηκαν ώστε να γίνεται πιο εφικτή η επεξεργασία τους. Και καθώς τα δεδομένα αυτά θα περάσουν από αρκετά στάδια μέχρι να εξάγουμε τα τελικά αποτελέσματα, είναι σημαντικό να υπάρχει τρόπος ώστε οι περιγραφές» αυτές να διατηρούνται έως το τέλος της ανάλυσης και να μην «χάνονται» κατά τη διάρκεια της όλης διαδικασίας.

II) Καθάρισμα/Εξαγωγή

Σε αυτό το στάδιο, έχουμε συλλέξει τα δεδομένα και ενδιαφερόμαστε να τα προετοιμάσουμε για την ανάλυση που ακολουθεί. Αρχικά μετατρέπουμε τα δεδομένα σε κατάλληλη μορφοποίηση (*format*). Τα δεδομένα που συλλέγουμε μπορεί να είναι σε διάφορες μορφοποιήσεις όπως κείμενο, εικόνα, βίντεο, ήχο. Οπότε κρίνεται απαραίτητο να υπάρχει μια διαδικασία «εξόρυξης» των απαιτούμενων πληροφοριών απο τα παραπάνω δεδομένα και να τα μετατρέπει σε κατάλληλη μορφοποίηση για την επεξεργασία που θα ακολουθήσει. Σε αυτή τη φάση γίνεται επίσης αναγνώριση του «θορύβου» στα αρχικά δεδομένα.

Πολλά από τα δεδομένα περιέχουν παραποιημένη πληροφορία, οπότε σε τέτοιες περιπτώσεις θα πρέπει κατά τη διαδικασία της ανάλυσης να λαμβάνεται υπόψη το αντίστοιχο σφάλμα στα τελικά αποτελέσματα.

III) Ομαδοποίηση δεδομένων/Σχεδίαση βάσης δεδομένων

Το τελευταίο στάδιο της προετοιμασίας πριν την ανάλυση. Εδώ θα πρέπει τα δεδομένα που προέρχονται από διαφορετικά σύνολα να τροποποιηθούν με τέτοιο τρόπο ώστε να μπορούν να αναλυθούν. Και φυσικά αυτό θα πρέπει να μπορεί να γίνει αυτοματοποιημένα. Είναι επίσης σημαντικό να γίνει η ορθή επιλογή της βάσης δεδομένων που θα χρησιμοποιηθεί. Και αφού η διαδικασία σχεδίασης της βάσης δεδομένων είναι κρίσιμη, θα πρέπει να γίνει νωρίτερα μια σχετική έρευνα ούτως ώστε να επιλέξουμε την κατάλληλη κάθε φορά.

IV) Ανάλυση

Η υποδομή που απαιτείται για την ανάλυση των Μεγάλων Δεδομένων πρέπει να υποστηρίζει εργαλεία στατιστικής ανάλυσης και εξόρυξης πληροφορίας σε μια ποικιλία τύπων δεδομένων που είναι αποθηκευμένα σε διαφορετικά συστήματα. Τις περισσότερες φορές, ο Όγκος των Μεγάλων Δεδομένων οδηγεί σε σημαντικές ανακρίβειες στην ποιότητα της πληροφορίας.

Παρ'όλα αυτά, η κατάλληλη ανάλυση επιτρέπει τον εντοπισμό μοτίβων και βοηθά σημαντικά στη «εξόρυξη» σημαντικών πληροφοριών.

V) Παρουσίαση Αποτελεσμάτων

Αφού έχει προηγηθεί η ανάλυση των δεδομένων θα ακολουθήσει η παρουσίαση των αποτελεσμάτων, η οποία δεν είναι μια απλή διαδικασία. Τα αποτελέσματα από μόνα τους δεν έχουν ιδιαίτερη αξία. Για να είναι ουσιώδη χρειάζεται να παρέχουν στο χρήστη λεπτομερή και κατανοητή περιγραφή αυτών, όπως το τι σημαίνουν και πώς προέκυψαν. Επίσης είναι αναγκαίο να δίνεται στο τέλος πληροφορία σχετικά με τις παραδοχές που έγιναν στα διάφορα στάδια της ανάλυσης των Μεγάλων Δεδομένων.

1.7 Οι Εφαρμογές των Μεγάλων Δεδομένων

Η μεγάλη τεχνολογική και επιχειρηματική πρόκληση της εποχής είναι η αποτελεσματική διαχείριση του τεράστιου όγκου δεδομένων. Η πραγματική επανάσταση δεν βρίσκεται στις μηχανές επεξεργασίας δεδομένων, αλλά στα ίδια τα δεδομένα και πώς αυτά χρησιμοποιούνται. Πώς όμως αντιμετώπισε ο κόσμος την εμφάνιση των Μεγάλων Δεδομένων και πώς αυτά επηρέασαν τη ζωή τους;

Κάθε πτυχή της ζωής μας προβλέπεται ότι θα επηρεαστεί από τα Μεγάλα Δεδομένα. Ωστόσο, υπάρχουν ορισμένοι τομείς όπου τα Μεγάλα Δεδομένα έχουν ήδη κάνει τη διαφορά σήμερα. Πιο κάτω ακολουθεί περιγραφή μερικών από αυτών.

1.7.1 Μεγάλα Δεδομένα και Σύστημα Υγείας

Παραδείγματα Μεγάλων Δεδομένων στο Σύστημα Υγείας αποτελούν οι φάκελοι των ασθενών, πληροφορίες για την ασφάλιση τους, όπως και τα σχέδια υγείας. Αναλύοντας σε γρήγορο χρόνο μεγάλο όγκο πληροφοριών, τόσο δομημένων όσο και μη δομημένων, οι παροχές υγείας μπορούν να παρέχουν πολύ πιο γρήγορα και με λιγότερο κόπο και κόστος την πιο πιθανή διάγνωση και την κατάλληλη θεραπεία της.

1.7.2 Μεγάλα Δεδομένα και Λιανική Πώληση

Η εξυπηρέτηση πελατών έχει εξελιχθεί πολύ τα τελευταία χρόνια, καθώς οι καταναλωτές περιμένουν από τις επιχειρήσεις να καταλαβαίνουν τι ακριβώς χρειάζονται, όταν το χρειάζονται. Η ανάλυση των Μεγάλων Δεδομένων βοηθά με αυτόν τον τρόπο τις εταιρείες λιανικής πώλησης να πληρούν αυτές τις απαιτήσεις.

Συλλέγοντας το πλήθος των πληροφοριών από τις συναλλαγές με τους πελάτες τους, οι επιχειρήσεις μπορούν να αναδείξουν τις τάσεις και ροπές της καταναλωτικής συμπεριφοράς και με τον τρόπο αυτό ικανοποιούν και τους πελάτες αυξάνουν τα έσοδά τους.

1.7.3 Μεγάλα Δεδομένα και Τραπεζικές/ Οικονομικές Συναλλαγές

Μέσα σε όλη αυτή την πληθώρα δεδομένων, οι τράπεζες χρειάζεται να βρίσκουν καινοτόμους τρόπους διαχείρισης των Μεγάλων Δεδομένων. Είναι πολύ σημαντικό να κατανοήσουν τους πελάτες τους και να τους εξασφαλίζουν την ασφάλεια των συναλλαγών τους. Οι συναλλαγές υψηλής συχνότητας (*HFT - «High-Frequency Trading»*) είναι επίσης μια περιοχή όπου τα Μεγάλα Δεδομένα βρίσκουν μεγάλη χρήση σήμερα. Αλγόριθμοι Μεγάλων Δεδομένων χρησιμοποιούνται για τη λήψη επενδυτικών αποφάσεων. Η πλειοψηφία των μετοχών που ανταλλάσσονται πραγματοποιείται μέσω αλγορίθμων Μεγάλων Δεδομένων. Αυτοί λαμβάνουν κυρίως υπόψη τους δεδομένα που λαμβάνουν από τα μέσα κοινωνικής δικτύωσης και τις ειδησεογραφικές ιστοσελίδες προκειμένου να σχηματίσουν αποφάσεις και να διεκπεραιώσουν με τον τόπο αυτό τις αγορές και τις πωλήσεις τους σε κλάσματα του δευτερολέπτου.

1.7.4 Μεγάλα Δεδομένα και Εκπαίδευση

Η Τεχνολογία παρέχει την δυνατότητα μάθησης μέσω προβλέψεων και διαγνωστικών αξιολογήσεων. Όπως είπε ο D. M. West «Τα Μεγάλα Δεδομένα μπορούν να υποστηρίξουν το κλασσικό εκπαιδευτικό σύστημα βοηθώντας τους δασκάλους να αναλύσουν τις γνώσεις των μαθητών και να προσδιορίσουν ποιές τεχνικές εκπαίδευσης είναι πιο αποτελεσματικές για κάθε μαθητή ξεχωριστά». Με τον τρόπο αυτό οι καθηγητές είναι σε θέση να μάθουν νέες τεχνικές και μεθόδους για την εκπαίδευση (West, 2012).

1.7.5 Μεγάλα Δεδομένα και Καιρός

Μία καθημερινή χρήση που προκύπτει από τη διαχείριση των Μεγάλων Δεδομένων είναι η εφαρμογή WeatherSignal, (http://weathersignal.com) η οποία χχρησιμοποιεί αισθητήρες κινητών τηλεφώνων για να μετρήσει τις τοπικές ατμοσφαιρικές συνθήκες και ακολούθως ενημερώνονται οι χάρτες της εφαρμογής. Η εφαρμογή αυτή έχει κυκλοφορήσει από την *OpenSignal* (http://opensignal.com) το Μάϊο του 2013.

1.7.6 Μεγάλα Δεδομένα και Βελτίωση Επιστήμης και Έρευνας

Η Επιστήμη και η Έρευνα βρίσκονται σε ένα μεταβατικό στάδιο λόγω των δυνατοτήτων που τους προσφέρουν τα Μεγάλα Δεδομένα. Παράδειγμα αποτελεί το *CERN* (*European Organization for*

Nuclear Research), το Ελβετικό εργαστήριο πυρηνικής φυσικής με το μεγαλύτερο και μέγιστης ενέργειας επιταχυντή Αδρονίων (*LHC*[2]) στο κόσμο. Λόγω των πειραμάτων που πραγματοποιούνται δημιουργούνται τεράστιες ποσότητες δεδομένων. Το κέντρο δεδομένων του *CERN* διαθέτει 65.000 επεξεργαστές για να αναλύσει 30 petabyte δεδομένων κάθε χρόνο. Ωστόσο, χρησιμοποιεί την υπολογιστική δύναμη χιλιάδων υπολογιστών κατανεμημένοι σε 150 κέντρα δεδομένων σε όλο τον κόσμο, προκειμένου να αναλύουν τα δεδομένα. (Mascetti et al., 2015)

Ένα ακόμη παράδειγμα αποτελεί το Ευρωπαικό Ινστιτούτο Βιοπληροφορικής (EBI) στο *Hinxton* του Ηνωμένου Βασιλείου που είναι μία από τις μεγαλύτερες αποθήκες δεδομένων βιολογικών δειγμάτων στον κόσμο. (http://www.ebi.ac.uk/)

Ένας ακόμη τομέας της επιστήμης που ασχολείται με την διαχείριση Μεγάλων Δεδομένων είναι η Τηλεπισκόπηση (Remote Sensing). Η Τηλεπισκόπηση είναι μια επιστήμη που μελετά τα αντικείμενα ή τα φαινόμενα μιας περιοχής από απόσταση. Παραδείγματα μορφών καταγραφής δεδομένων από απόσταση αποτελούν οι αεροφωτογραφίες, οι δορυφορικές εικόνες και οι εικόνες «ραντάρ». Η δορυφορική τηλεπισκόπηση συνεισφέρει σε πολλές εφαρμογές, όπως τη μετεωρολογία, τη σεισμολογία και την ωκεανογραφία. Για παράδειγμα στο σύμπλεγμα υπερυπολογιστών του Κέντρου Κλιματικής Προσομοίωσης της NASA διαχειρίστηκαν 32 petabyte δεδομένων από παρατηρήσεις και προσομοιώσεις που σχετίζονται με το κλίμα. (Chen and Zhang, 2014)

1.7.7 Μεγάλα Δεδομένα και Αθλητισμός

Τα πιο δημοφιλή σύγχρονα αθλήματα σήμερα, στηρίζονται στη διαχείρηση Μεγάλων Δεδομένων. Χαρακτηριστικό παράδειγμα αποτελεί η εφαρμογή *IBM SlamTracker* για τουρνουά αντισφαίρισης, η οποία χρησιμοποιεί αναλύσεις βίντεο για να παρακολουθεί την απόδοση του κάθε παίκτη. (http://www.usopen.org/index.html?promo=topnav) Επιπλέον, πολλές μεγάλες αθλητικές ομάδες παρακολουθούν τους αθλητές έξω από το αθλητικό περιβάλλον, χρησιμοποιώντας έξυπνη τεχνολογία για να παρατηρούν τη διατροφή, τον ύπνο όπως και τη συναισθηματική ευεξία των αθλητών τους παρακολουθώντας τις συνομιλίες και τις αναρτήσεις τους στα διάφορα κοινωνικά δίκτυα. Με τον τρόπο αυτό μπορούν να προβλέπουν και να διασφαλίζουν την επιτυχία της ομάδας τους.

[2] *LHC =Large Hadron Collider*

1.8 Συστήματα Προ-επεξεργασίας Μεγάλων Δεδομένων

Σήμερα υπάρχουν πολλές ανοιχτού κώδικα βάσεις δεδομένων για την ανάκτηση, επεξεργασία και αποθήκευση Μεγάλων Δεδομένων. Το *Hadoop* (http://hadoop.apache.org/) αναδεικνύεται ως το κύριο σύστημα για την ανάλυση Μεγάλων Δεδομένων και είναι η επέκταση της εμβέλειας των σχεσιακών βάσεων δεδομένων σε λιγότερο δομημένα δεδομένα. Τα συστήματα αυτά έχουν δημιουργήσει ένα διαιρεμένο φάσμα λύσεων που αποτελείται από *NoSQL (Not only Structured Query Language)* λύσεις για ευέλικτο και εξειδικευμένο προγραμματισμό και από *SQL (Structured Query Language)* λύσεις, δηλαδή τη διαχείριση, ασφάλεια και αξιοπιστία των σχεσιακών συστημάτων διαχείρισης βάσεων δεδομένων. Μια εκτενής περιγραφή και σύγκριση των *NoSQL και SQL* βάσεων δίδεται από τους Moniruzzaman and Hossain (2013).

Τα *NoSQL* συστήματα είναι σχεδιασμένα για να αποκτούν όλα τα δεδομένα, χωρίς την κατηγοριοποίηση και την ανάλυση τους κατά την είσοδο στο σύστημα, και ως εκ τούτου τα δεδομένα ποικίλουν. Ενώ στα *SQL* συστήματα, τοποθετούνται τυπικά δεδομένα σε σαφώς καθορισμένες δομές οι *NoSQL* βάσεις δεδομένων είναι εφαρμογές *OLTP (On-Line transaction processing)*, οι οποίες έχουν βελτιστοποιηθεί για τη γρήγορη συλλογή δεδομένων. Οι δομές των δεδομένων που χρησιμοποιούνται από τις *NoSQL* βάσεις είναι διαφορετικές από εκείνες που χρησιμοποιούνται από τις αντίστοιχες σχεσιακές, καθιστώντας έτσι ορισμένες διεργασίες σε *NoSQL* βάσεις δεδομένων πιο γρήγορες.

Ωστόσο, λόγω της μεταβαλλόμενης φύσης των δεδομένων σε μία βάση *NoSQL*, κάθε προσπάθεια οργάνωσης τους απαιτεί γνώσεις προγραμματισμού για τη διερμηνεία της λογικής αποθήκευσης που χρησιμοποιείται. Αυτό, σε συνδυασμό με την απουσία υποστήριξης σε πολύπλοκα ερωτήματα, καθιστά δύσκολο στους χρήστες να αξιοποιήσουν τη χρησιμότητα μιας *NoSQL* βάσης δεδομένων. Ο συνδυασμός των δύο, *NoSQL* και *SQL* βάσεις δεδομένων αξιοποιεί στο έπακρο τις απαιτήσεις για επιχειρηματική χρήση. Στο επόμενο κεφάλαιο θα ασχοληθούμε πιο αναλυτικά με αυτό.

ΚΕΦΑΛΑΙΟ 2

ΥΠΟΛΟΓΙΣΤΙΚΟ ΠΛΑΙΣΙΟ ΑΝΑΛΥΣΗΣ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑΣ ΜΕΓΑΛΩΝ ΔΕΔΟΜΕΝΩΝ

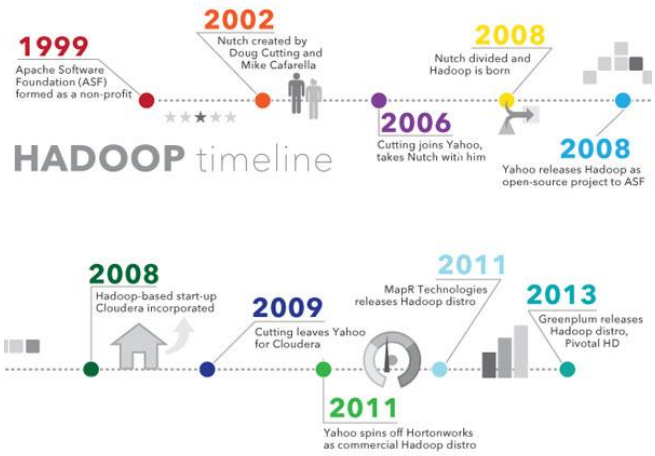
Στα πλαίσια της παρούσας διπλωματικής εργασίας κατασκευάσαμε και εφαρμόσαμε ένα πλήρες υπολογιστικό πλαίσιο επεξεργασίας και ανάλυσης Μεγάλων Δεδομένων. Το πλαίσιο αυτό πρέπει να είναι σε θέση να εκτελεί τόσο την προεπεξεργασία των δεδομένων, όσο και την ανάλυσή τους για την εξαγωγή αποτελεσμάτων. Για το λόγο αυτό, το υπολογιστικό πλαίσιο αποτελείται από 2 διακριτά τμήματα: ένα για την προεπεξεργασία και την προετοιμασία των δεδομένων και ένα δεύτερο για την ανάλυσή τους. Για το πρώτο τμήμα, έπειτα απο εκτενή βιβλιογραφική ανασκόπηση για την εύρεση της κατάλληλης πλατφόρμας, καταλήξαμε στην πλατφόρμα Hadoop. Για το δεύτερο τμήμα αποφασίστηκε να επιλεγούν τα Τεχνητά Νευρωνικά Δίκτυα, λόγω της υπάρχουσας τεχνογνωσίας στον συγκεκριμένο τομέα (πρβλ. Christakis et al., 2011). Στο παρόν κεφάλαιο δίδεται μια αναλυτική περιγραφή του Hadoop και των Τεχνητών Νευρωνικών Δικτύων. Αυτό που πρέπει να σημειωθεί είναι ότι για την παρούσα εργασία, η υλοποίηση τόσο του Hadoop όσο και των Νευρωνικών Δικτύων έγινε σειριακά για να αναδειχθούν καταρχήν οι δυνατότητες αυτού του υπολογιστικού πλαισίου. Στη συνέχεια, σκοπεύεται να πραγματοποιηθεί παράλληλη υλοποίηση του πλαισίου αυτού για να επιτραπεί με αυτόν τον τρόπο βελτιστοποίηση τόσο στη διαχείριση όσο και στην ανάλυση των εισαγομένων δεδομένων.

2.1 Εισαγωγή στο Hadoop

Κατά τη διάρκεια των τελευταίων ετών, η βιομηχανία των Μεγάλων Δεδομένων έχει γνωρίσει τεράστια άνοδο με αποτέλεσμα οι περισσότερες από τις παγκόσμιες εταιρίες να επενδύουν όλο και περισσότερο στην έρευνα και την προσέγγισή τους. Ωστόσο, το αληθινό κίνητρο πίσω από τις μεγάλες επενδύσεις των εταιριών στο συγκεκριμένο τομέα δεν είναι μόνο η συλλογή των δεδομένων αλλά η ανάλυση, η αποσαφήνισή και η ορθή εκμετάλλευση τους. Για το λόγο αυτό αναπτύσσονται διάφορες τεχνολογίες όπως το Hadoop και το Spark, γνωστό και ως ξάδερφο του Hadoop. Το Hadoop σήμερα είναι η πιο διαδεδομένη υλοποιήση του MapReduce και χρησιμοποιείται για διδακτικούς σκοπούς σε αρκετά πανεπιστήμια, αλλά και σε μεγάλους οργανισμούς για την επεξεργασία Μεγάλων Δεδομένων . Κάποιοι απο τους οργανισμούς αυτούς που διατηρούν συστοιχίες υπολογιστών (clusters) για την εκτέλεση Hadoop εργασιών είναι: Yahoo!, Amazon, Cornell University Web Lab, Facebook, Google, IBM, New York Times.Το Hadoop γνώρισε τεράστια επιτυχία τα τελευταία χρόνια και χρησιμοποιείται ευρέως σε τομείς όπως η οικονομία, τα μέσα μαζικής ενημέρωσης, η υγειονομική περίθαλψη, οι υπηρεσίες πληροφοριών, το λιανικό εμπόριο και άλλες βιομηχανίες με απαιτήσεις μεγάλων δεδομένων. Είναι σχεδιασμένο για την επεξεργασία μεγάλου όγκου δεδομένων από terabytes σε petabytes. Το Hadoop είναι η κινητήρια δύναμη πίσω από την ανάπτυξη της βιομηχανίας των Μεγάλων Δεδομένων και χωρίς αμφιβολία είναι το «biggest thing» αυτή τη στιγμή αλλάζοντας ριζικά τον τρόπο με τον οποίο οι επιχειρήσεις αναλύουν, επεξεργάζονται και αποθηκεύουν δεδομένα.

2.2 Ιστορική αναδρομή

Το 2002 οι Doug Cutting και Mike Cafarella έφτιαξαν μία μηχανή αναζήτησης ανοιχτού κώδικα, το Nutch. Η ιδέα εμπνεύστηκε από την ίδια ιδέα που δημιουργήθηκε η μηχανή αναζήτησης Google και βασιζόταν στην επεξεργασία και αποθήκευση δεδομένων με ένα κατανεμημένο και αυτοματοποιημένο τρόπο έτσι ώστε σχετικά αποτελέσματα αναζήτησης στον ιστό να επιστρέφονται γρηγορότερα. Το 2006 ο Cutting εντάχθηκε στη Yahoo!, παίρνοντας μαζί του το έργο Nutch και οδηγώντας στην απόσπαση του τμήματος κατανεμημένου υπολογισμού και επεξεργασίας με την ονομασία Hadoop το 2008. Η ονομασία της πλατφόρμας αυτής προήλθε από ένα παιχνίδι-ελέφαντα του γιου του Cutting. Ουσιαστικά, το Hadoop ήταν η απόρροια της προσπάθειας από το Yahoo! να αντιμετωπίσει το πρόβλημα των υπέρογκων συνόλων δεδομένων, σπάζοντάς τα σε μικρότερα κομμάτια τα οποία θα μπορούσαν να υφίστανται παράλληλη επεξεργασία. Σήμερα, το πλαίσιο και το οικοσύστημα των τεχνολογιών του Hadoop διαχειρίζονται και συντηρούνται από το Apache Software Foundation (ASF) και μία διεθνή κοινότητα προγραμματιστών. (Σχήμα 2.1) (White, 2012)



Σχήμα 2.1: Ιστορική Αναδρομή.

Πηγή: https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcQw2PEofK8nrg-pt6_DOdWr0yVixKMAIohVq-vu2nImerkOuzml

2.3 Ορισμός

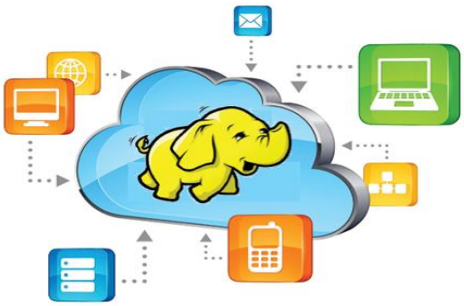
Το Hadoop είναι ένα προγραμματιστικό πλαίσιο ανοιχτού κώδικα (open-source framework), γραμμένο σε Java, για την κατανεμημένη αποθήκευση και επεξεργασία μεγάλων συνόλων, δομημένων και μη-δομημένων, δεδομένων σε συστοιχίες υπολογιστών (clusters) με τη χρήση απλών μοντέλων προγραμματισμού. (Apachee Hadoop, 2014) (Σχήμα 2.2)

Σχήμα 2.2: Λογότυπο Hdoop.

Πηγή: http://hadoop.apache.org

2.4 Σχεδιασμός

Σε αντίθεση με τα παραδοσιακά συστήματα, το Hadoop επιτρέπει σε πολλαπλού τύπου αναλυτικές μονάδες εργασίας (multiple types of analytic workloads) να έχουν πρόσβαση στα ίδια, δομημένα και μη-δομημένα δεδομένα, την ίδια χρονική στιγμή. Όλες οι μονάδες του Hadoop έχουν σχεδιαστεί με τη παραδοχή ότι οι αποτυχίες του υλικού (hardware failures) είναι συχνές και θα πρέπει αυτόματα να αντιμετωπίζονται από το πλαίσιο λογισμικού (framework). Έτσι, η βιβλιοθήκη από μόνη της είναι σχεδιασμένη να εντοπίζει και να χειρίζεται τυχόν αποτυχίες σε επίπεδο εφαρμογής, προσφέροντας υψηλής διαθεσιμότητας υπηρεσίες πάνω από συστοιχίες υπολογιστών (clusters), καθένας από τους οποίους είναι επιρρεπής σε αποτυχίες. (Turner, 2011) (Σχήμα 2.3)



Σχήμα 2.3: Αλληλεπίδραση του Hadoop.

Πηγή: http://www.xoomtrainings.com/blog/why-hadoop-and-what-skills-should-i-learn

2.5 Τρόποι λειτουργίας του Hadoop

Οι τρόποι λειτουργίας του Hadoop είναι τρείς:

- Local mode (standalone)
- Pseudo-distributed mode
- Fully-distributed mode

Το **Local mode (standalone)** είναι η προεπιλεγμένη λειτουργία του Hadoop, με την εγκατάσταση της εφαρμογής αυτής. Στην κατάσταση αυτή το Hadoop εκτελείται στο τοπικό μηχάνημα και δεν υπάρχει

ανάγκη να επικοινωνήσει με άλλους κόμβους. Χρησιμοποιείται κυρίως για ανάπτυξή και έλεγχο των προγραμμάτων MapReduce.

Το **Pseudo-distributed mode** τρέχει σε «συστοιχία ενός κόμβου». Χρησιμοποιείται για τους ίδιους λόγους όπως και πιο πάνω αλλά και για άλλους λόγους όπως τον έλεγχο χρησιμοποιημένης μνήμης.

Το **Fully-distributed mode** τρέχει σε μία συστοιχία κόμβων. Το Hadoop είναι σχεδιασμένο για να λειτουργεί ταυτόχρονα σε ένα μεγάλο αριθμό κόμβων, που δεν μοιράζονται την ίδια μνήμη. Έτσι, για την αποθήκευση των δεδομένων, τα αρχεία χωρίζονται σε μεγάλα τεμάχια (blocks) και διανέμονται σε πολλαπλούς κόμβους μιας συστοιχίας υπολογιστών (clusters), με το Hadoop να παρακολουθεί πού τοποθετούνται τα δεδομένα. Αν ένας κόμβος βγει εκτός σύνδεσης ή υποστεί κάποια βλάβη, τα δεδομένα που βρίσκονται αποθηκευμένα σε αυτόν δεν χάνονται αλλά μπορούν να επανακτηθούν από κάποιο άλλο αντίγραφο, αποφεύγοντας με τον τρόπο αυτό πιθανά σφάλματα υλικού.

2.6 Βασικό πλαίσιο

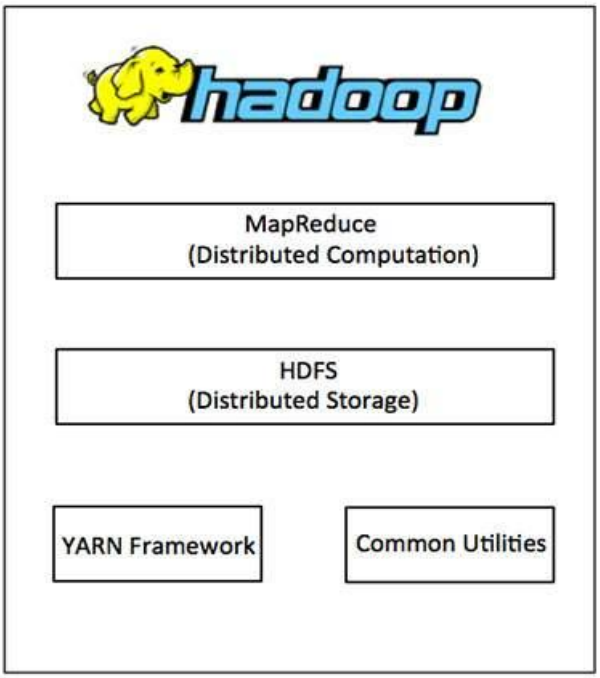
Ο πυρήνας του Apache Hadoop αποτελείται από τα εξής τέσσερα συστατικά: (Apache, 2008) (Σχήμα 2.4)

- Hadoop MapReduce: Ένα προγραμματιστικό μοντέλο, βασισμένο στο YARN, για την παράλληλη επεξεργασία μεγάλων συνόλων δομημένων και αδόμητων δεδομένων.

- Hadoop Distributed File System (HDFS): Ένα κατανεμημένο σύστημα αρχείων σε Java που παρέχει πρόσβαση, επεκτασιμότητα (scalability) και αξιόπιστη αποθήκευση δεδομένων σε πολλαπλούς κόμβους.

- Hadoop YARN (Yet Another Resource Negotiator): Ένα πλαίσιο για τη διαχείριση των πόρων της συστοιχίας υπολογιστών και τον προγραμματισμό των εργασιών.

- Hadoop Common: Αποτελείται από Java βιβλιοθήκες και διάφορα βοηθητικά προγράμματα που είναι απαραίτητα για τη λειτουργία των υπολοίπων συστατικών του Hadoop.



Σχήμα 2.4: Βσικό Πλαίσιο Hadoop.

Πηγή:https://www.tutorialspoint.com/hadoop/hadoop_introduction.htm

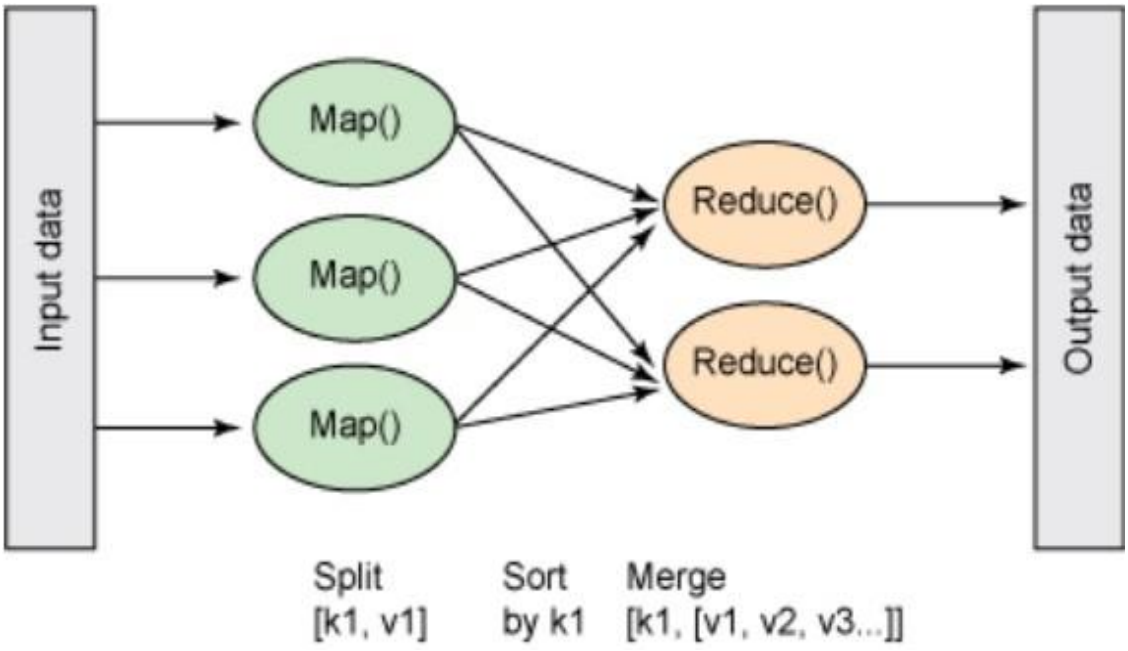
2.7 Αρχιτεκτονική

Το βασικό πλαίσιο του Hadoop, όπως αναφέρθηκε πιο πάνω αποτελείται απο τέσσερα βασικά συστατικά. Οπότε σε αυτή τη φάση θα γίνει μια εκτενέστερη περιγραφή των συστατικών αυτών.

2.7.1 MapReduce

Το MapReduce είναι ένα προγραμματιστικό μοντέλο για την επεξεργασία και παραγωγή μεγάλων συνόλων δεδομένων σε ένα δίκτυο υπολογιστών (cluster). Το μοντέλο αυτό είναι αρκετά απλό στη χρήση του και ευρέως διαδεδομένο. Η ιδέα του MapReduce είναι βασισμένη στον να παίρνει σαν είσοδο ένα σύνολο από ζευγάρια «κλειδί εισόδου-τιμή» και να παράγει ένα σύνολο από ζευγάρια «τιμή εξόδου-αποτέλεσμα». Αυτά τα δύο εκφράζονται σαν δύο συναρτήσεις, την συνάρτηση map και την

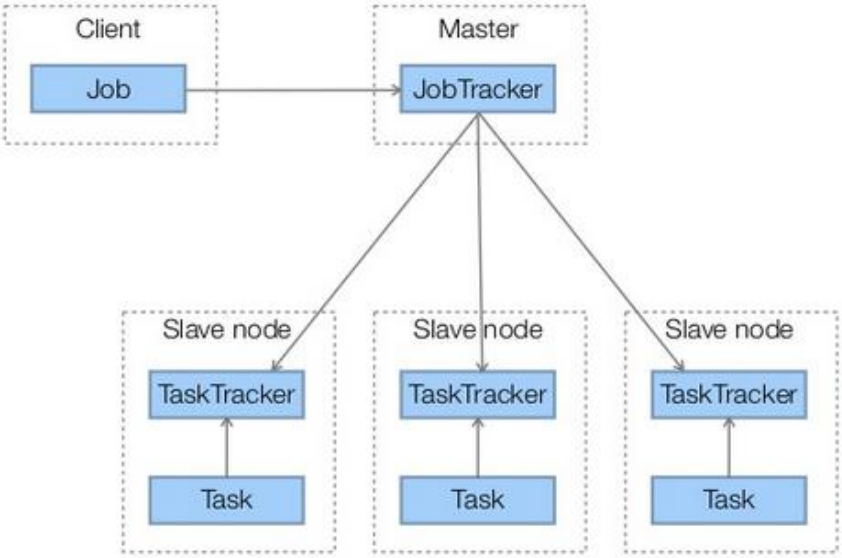
συνάρτηση reduce. Η συνάρτηση map δέχεται σαν είσοδο ένα ζεύγος «κλειδί -τιμή» και η έξοδος της συνάρτησης αυτής ταξινομημένη με βάση το κλειδί, είναι η είσοδος της συνάρτησης reduce. Οπότε η συνάρτηση reduce, παίρνει σαν είσοδο την έξοδο της συνάρτησης map σε μορφή «κλειδί-ενδιάμεση τιμή» και την επεξεργάζεται. Συνήθως για κάθε κλειδί έχουμε μία τιμή στην έξοδο. Στο Σχήμα 2.5 που ακολουθεί περιγράφεται η διαδικασία MapReduce. (Murthy et al., 2014)



Σχήμα 2.5: Διάγραμμα MapReduce

Πηγή:http://www.ibm.com/developerworks/cloud/library/cl-openstack-deployhadoop

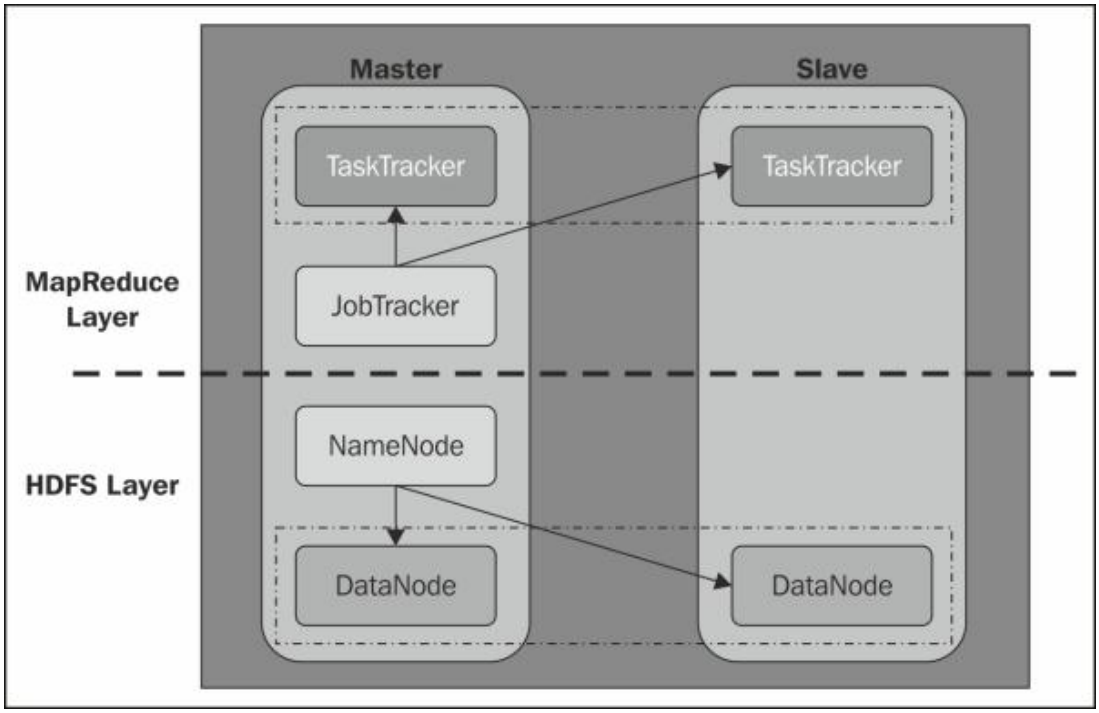
Ο μηχανισμός του MapReduce ακολουθεί το μοντέλο αρχιτεκτονικής «Master/Slave». Όπου ο κεντρικός Master node (JobTracker), αναλαμβάνει τον διαμερισμό των εργασιών στους υπόλοιπους κόμβους Slave node (TaskTrackers). Ο χρήστης καλεί τον JobTracker να αρχίσει την εργασία επεξεργασίας των δεδομένων, ο JobTracker διαμοιράζει την εργασία και αναθέτει διαφορετικές map και reduce εργασίες σε κάθε TaskTracker στη συστοιχία. Οι TaskTrackers απλά εκτελούν τις εργασίες που του αναθέτει ο JobTracker. (Σχήμα 2.6)



Σχήμα 2.6: Αναπαράσταση MapReduce διαδικασίας.

2.7.2 HDFS

Το HDFS είναι ένα κατανεμημένο σύστημα αρχείων γραμμένο σε JAVA που χρησιμοποιεί επίσης το μοντέλο αρχιτεκτονικής «Master/Slave» και είναι ιδανικό για αποθήκευση μεγάλων αρχείων. Το HDFS είναι παρόμοιο με το Google File System(GFS). Επιτυγχάνει να είναι αξιόπιστο αποθηκεύοντας τα δεδομένα σε περισσότερους από ένα κόμβους. Οι κόμβοι επικοινωνούν μεταξύ τους για αντιγραφή ή μετακίνηση των δεδομένων. Το σύστημα αρχείων HDFS χρησιμοποιεί ένα κεντρικό κόμβο, τον Name node, ο οποίος είναι και το μοναδικό σημείο αποτυχίας (single point of failure), που κρατά τις πληροφορίες για το που βρίσκεται κάθε δεδομένο στο HDFS. Αν αυτός δεν είναι διαθέσιμος τότε δεν υπάρχει πρόσβαση στο σύστημα αρχείων. Επιπλέον χρησιμοποιεί ένα ακόμα κόμβο, τον Secondary Name node, ο οποίος κρατά αντίγραφα των φακέλων του Name node και μαζί με τα αρχεία ιστορικού (logs) του Name node επαναφέρει το σύστημα αρχείων μετά από τυχόν αποτυχία. Οι υπόλοιποι κόμβοι ονομάζονται Data nodes και απλά αποθηκεύουν δεδομένα. Χρησιμοποιείται σαν είσοδος δεδομένων και έξοδος αποτελεσμάτων από το MapReduce. Στο Σχήμα 2.7 φαίνεται η δομή που έχει το MapReduce και HDFS.



Σχήμα 2.7: Δομή MapReduce και HDFS.

2.7.3 YARN

Στη $2^{η}$ γενιά του Hadoop, το MapReduce έχει υποστεί μια πλήρη αναμόρφωση και πλέον διαμορφώνεται σε αυτό που αναφέρεται ως MapReduce 2.0 ή YARN. Η βασική ιδέα πίσω από το YARN είναι να διαχωριστούν οι δύο κύριες λειτουργίες του JobTracker, η διαχείριση πόρων και ο χρονοπρογραμματισμός, σε ξεχωριστές οντότητες. Το YARN είναι η προϋπόθεση για το επιχειρηματικό Hadoop και παρέχει διαχείριση πόρων και μια κεντρική πλατφόρμα όπου προσφέρονται συνεπείς εργασίες, ασφάλεια και εργαλεία διαχείρισης δεδομένων σε συμπλέγματα Hadoop. (Hortonworks, 2015)

2.7.4 Common

Στο πακέτο Hadoop Common περιλαμβάνονται τα απαραίτητα εργαλεία για εγκατάσταση και εκκίνηση του Hadoop σε συστήματα ενός ή μερικών χιλιάδων κόμβων. Στην περίπτωση συστημάτων

ενός μόνο κόμβου παρέχονται μηχανισμοί για την εκτέλεση απλών εργασιών χρησιμοποιώντας μόνο το MapReduce και το HDFS. Σε συστοιχίες υπολογιστών που απαρτίζονται από περισσότερα μηχανήματα προσφέρονται εργαλεία για την εκτέλεση απαιτητικών, κατανεμημένων εφαρμογών με χρήση των πακέτων MapReduce, HDFS και Yarn.

2.8 Προεπεξεργασία Δεδομένων

Δεδομένου ότι οι περισσότερες βάσεις δεδομένων του πραγματικού κόσμου επηρεάζονται σε μεγάλο βαθμό από αρνητικά στοιχεία, όπως η ύπαρξη θορύβου, οι ελλείπουσες τιμές, τα ασυνεπή και περιττά δεδομένα, είναι πολύ σημαντικό να κάνουμε μια προ-επεξεργασία στα δεδομένα για να επιτύχουμε καλύτερη ανάλυση. (Fernández, 2015). Σήμερα, η ανάγκη αυτή είναι ακόμη πιο σημαντική λόγω των Μεγάλων Δεδομένων. Υπάρχουν αρκετά λογισμικά για την επεξεργασία των δεδομένων πριν από τη φόρτωση τους στο Hadoop. Επίσης υπάρχουν πολλά προγράμματα και γλώσσες που μπορούν να βοηθήσουν στην ακριβέστερη προ-επεξεργασία τεράστιων ποσοτήτων δεδομένων, όπως SQL ή SAS. Επιπλέον, είναι δυνατή η προ-επεξεργασία των δεδομένων εντός του πλαισίου Hadoop, χρησιμοποιώντας γλώσσες προγραμματισμού όπως είναι το HIVE, Pig και Jaql. Ανεξάρτητα όμως από τον τρόπο με τον οποίο τα δεδομένα φιλτράρονται, καθαρίζονται και ενσωματώνονται, η προεπεξεργασία δεδομένων αποτελεί ένα σημαντικό στάδιο για την επιτυχή διαχείριση μεγάλων δεδομένων.

2.9 Το Οικοσύστημα

Ο όρος Hadoop όμως δεν αναφέρεται μόνο στο βασικό πλαίσιο που περιγράφηκε παραπάνω, αλλά και στο οικοσύστημα, τη συλλογή επιπρόσθετων πακέτων λογισμικού που μπορούν να εγκατασταθούν παράλληλα με το Hadoop ή στην κορυφή του. Τα εργαλεία αυτά μπορούν να ταξινομηθούν σε Υπηρεσίες Δεδομένων (Hadoop Data Services) και σε Λειτουργικές Υπηρεσίες (Hadoop Operational Services) με βάση την λειτουργικότητα που προσφέρουν. (Lam, 2010)

Υπηρεσίες Δεδομένων, είναι τα εργαλεία εκείνα που επιτρέπουν στους χρήστες να χειρίζονται και να επεξεργάζονται τα δεδομένα με πιο εύκολο τρόπο. Μερικά από αυτά είναι:

- Apache Hive

  Το Hive είναι μια γλώσσα προγραμματισμού που δημιουργήθηκε αρχικά απο το Facebook και αργότερα το ανέλαβε και το ανέπτυξε περαιτέρω η Apache Software Foundation . Είναι παρόμοια με την SQL, και έτσι διευκολύνει τους χρήστες που γνωρίζουν SQL να γράφουν

Hive Query Language (HQL). Οι HQL δηλώσεις διασπώνται από την υπηρεσία Hive σε MapReduce εργασίες και εκτελούνται στο Hadoop. Βρίσκεται στην κορυφή του Hadoop και είναι υπεύθυνο για την προεπεξεργασία των Μεγάλων Δεδομένων για να κάνει ευκολότερη την ανάλυση τους. (Apache, 2017)

- Apache Pig

  Η γλώσσα προγραμματισμού Pig αρχικά αναπτύχθηκε από την Yahoo! και αργότερα το ανέλαβε και το ανέπτυξε περαιτέρω η Apache Software Foundation . Το Pig επιτρέπει τη συγγραφή σύνθετων MapReduce εργασιών, χρησιμοποιώντας μια γλώσσα που ονομάζεται Pig Latin. Το Pig διασπά την Pig Latin σε MapReduce και έτσι μπορούν να εκτελεστούν μέσα στο HDFS του Hadoop . (Apache, 2016)

- Jaql

  Η Jaql παραχωρήθηκε από την IBM στη κοινότητα ανοιχτού κώδικα. Είναι μια γλώσσα προγραμματισμού που βασίζεται σε Javascript Object Notation (JSON), αλλά υποστηρίζει πολύ περισσότερες γλώσσες από αυτή. Μας επιτρέπει να επεξεργαστούμε δομημένα και μη δομημένα δεδομένα.

Λειτουργικές Υπηρεσίες είναι τα εργαλεία εκείνα τα οποία έχουν δημιουργηθεί για να βοηθήσουν στις λειτουργίες και τη διαχείριση μιας συστοιχίας Hadoop, όπως:

- Apache Oozie

  Το Oozie είναι ένα έργο ανοιχτού κώδικα του Apache, που απλοποιεί τη ροή εργασιών και των συντονισμό τους. Προσφέρει στον χρήστη την δυνατότητα να ορίζει ενέργειες και τις εξαρτήσεις μεταξύ αυτών, έτσι ώστε η κατάλληλη ενέργεια να εκτελεστεί την κατάλληλη χρονική στιγμή. μία web εφαρμογή σε Java, που χρησιμοποιείται για το χρονοπρογραμματισμό των εργασιών του Hadoop. Συνδυάζει πολλαπλές εργασίες διαδοχικά σε μία λογική μονάδα εργασίας και υποστηρίζει Hadoop εργασίες για το MapReduce, Pig, Hive και Sqoop. (Apache, 2017)

- Apache ZooKeeper

  Το ZooKeeper είναι ένα έργο του Apache που παρέχει λειτουργικές υπηρεσίες σε μια συστοιχία Hadoop που επιτρέπουν στις κατανεμημένες διαδικασίες του να συντονιστούν πιο εύκολα μεταξύ τους. (Apache, 2016)

Αναλυτική περιγραφή της διαδικασίας ψευδο-κατανεμημένης εγκατάστασης των Hadoop και Hive σε έναν επεξεργαστή δίδεται στο Παράρτημα Α.

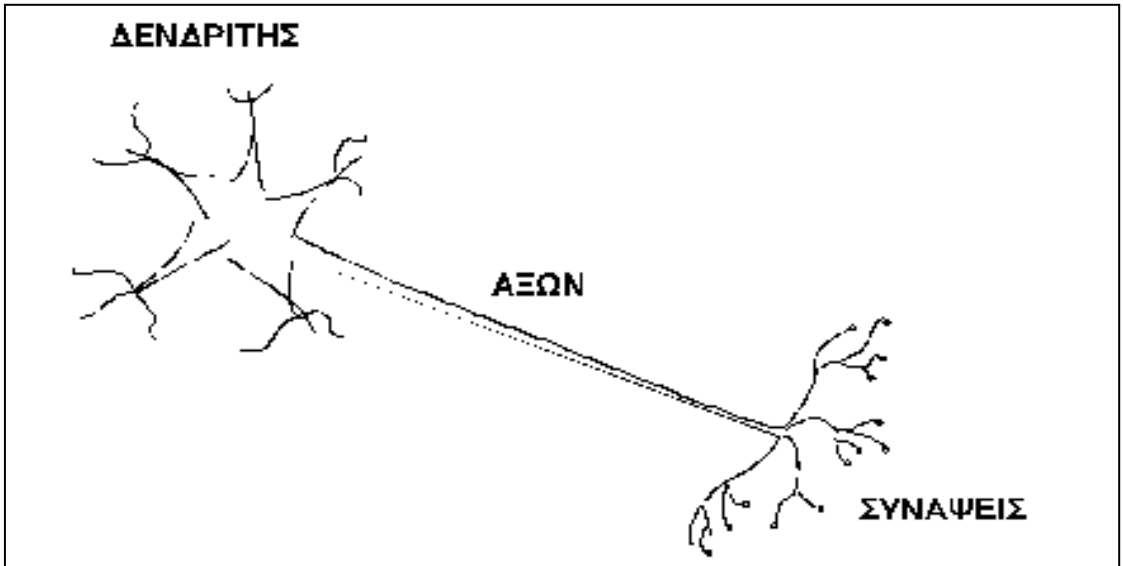
2.10 Εισαγωγή στα Τεχνητά Νευρωνικά Δίκτυα

Τα τελευταία χρόνια παρατηρείται έντονο ενδιαφέρον για τα συστήματα τεχνητών νευρωνικών δικτύων (ΤΝΔ) και την καταλληλότητα τους για τη λύση πολλών πραγματικών προβλημάτων. Κατά την προσομοίωση σύνθετων συστημάτων που εμφανίζουν μη-γραμμική συμπεριφορά οι παραδοσιακές μέθοδοι αποδείχθηκαν ανεπαρκείς είτε λόγο της απλής περιγραφής του προβλήματος είτε λόγω της έλλειψης διαθέσιμων υπολογιστικών πόρων. (Christakis et al., 2011) Ένα ΤΝΔ είναι ένα μαθηματικό μοντέλο επεξεργασίας πληροφορίας που η λειτουργία του είναι εμπνευσμένη από τον τρόπο με τον οποίο οι βιολογικοί νευρώνες επεξεργάζονται την πληροφορία. Το βασικό στοιχείο ενός νευρωνικού δικτύου είναι ο νευρώνας. Ένα ΤΝΔ αποτελείται από ένα μεγάλο αριθμό διασυνδεδεμένων στοιχείων επεξεργασίας (νευρώνες) τα οποία συνεργάζονται για την επίλυση συγκεκριμένων προβλημάτων. Τα ΤΝΔ, όπως και οι άνθρωποι, μαθαίνουν από παραδείγματα και ρυθμίζονται προκειμένου να μπορούν να χρησιμοποιηθούν σε συγκεκριμένες εργασίες μέσα από μια διαδικασία εκπαίδευσης. Η εκπαίδευση στα βιολογικά συστήματα συνεπάγεται αναπροσαρμογές των συναπτικών συνδέσεων που υπάρχουν μεταξύ των νευρώνων. Αντίστοιχα για τα ΤΝΔ, η διαδικασία της εκπαίδευσης ρυθμίζει κατάλληλα τις διασυνδέσεις των νευρώνων μέσω μίας μεταβλητής που ονομάζεται «συναπτικό βάρος».

2.10.1 Από τα Βιολογικά στα Τεχνητά Νευρωνικά Δίκτυα

Είναι γνωστό ότι ο εγκέφαλος του ανθρώπου έχει μια από τις πιο περίπλοκες δομές που συναντά κανείς στον φυσικό κόσμο και ακόμη δεν είναι πλήρως γνωστό πώς αυτός εκπαιδεύεται για να επεξεργάζεται τις πληροφορίες και πως ακριβώς λειτουργεί. Στον ανθρώπινο εγκέφαλο, θεωρούμε ότι η βασική μονάδα δόμησης είναι ένα κύτταρο που ονομάζεται νευρώνας. Ο ανθρώπινος εγκέφαλος αποτελείται από ένα πολύ μεγάλο αριθμό νευρώνων, της τάξης του 10^{10}. Ένας νευρώνας αποτελείται από το κυρίως σώμα, τον άξονα και τους δενδρίτες. (Σχήμα 2.8) Κάθε νευρώνας συνδέεται με πολλούς άλλους νευρώνες με συνδέσεις που ονομάζονται συνάψεις. Ο ρόλος του νευρώνα σε ένα νευρωνικό δίκτυο είναι να λαμβάνει όλα τα εισερχόμενα σήματα απο τους άλλους νευρώνες, να τα επεξεργάζεται με κατάλληλο τρόπο και να μεταδίδει το επεξεργασμένο σήμα σε άλλους νευρώνες, ούτως ώστε το σήμα να διαδίδεται μέσω ενός τεράστιου αριθμού νευρώνων. Οπότε ένας τυπικός νευρώνας συλλέγει σήματα από άλλους νευρώνες μέσα από μια σειρά από δομές που ονομάζονται δενδρίτες. Ο νευρώνας αποτελείται από ένα μακρύ και λεπτό άξονα ο οποίος χωρίζεται σε χιλιάδες διακλαδώσεις. Όταν ένας νευρώνας διεγείρεται τότε στέλνει ένα παλμό ηλεκτρικής ενέργειας κατά μήκος του άξονα του,

μεταδίδοντας τη διέγερση σε άλλους διασυνδεδεμένους νευρώνες. (Plerou, 2012). Η επανάληψη αυτής της διαδικασίας οδηγεί στην εκπέδευση του βιολογικού νευρωνικού δικτύου και μέσα απο την εκπαίδευση αυτή γίνεται τροποποίηση του κατά πόσο η έξοδος του ενός νευρώνα επηρεάζει τον άλλο.



Σχήμα 2.8: Απλοϊκή αναπαράσταση βιολογικού νευρώνα.

Πηγή:http://www.thalys.gr/pagesgr/business%20management/Ltlearn/learning_23_gr.htm

2.10.2 Ιστορική αναδρομή

Το επιστημονικό ενδιαφέρον για τα νευρωνικά δίκτυα είχε δημιουργηθεί πολύ πριν την δημιουργια των ηλεκτρονικών υπολογιστών. Συγκεκριμένα το 1943 οι McCulloch και Pitts ανέπτυξαν μοντέλα νευρωνικών δικτύων βασιζόμενοι στις γνώσεις που είχαν στην νευρολογία. Τα μοντέλα αυτά παρήγαγαν αρκετές υποθέσεις σχετικά με το πώς λειτουργούν οι νευρώνες. Τα δίκτυά τους βασίζονταν σε απλούς νευρώνες που θεωρούνταν ότι είναι δυαδικές διατάξεις με σταθερά όρια. Τα αποτελέσματα των μοντέλων τους ήταν απλές λογικές συναρτήσεις όπως «α ή β» και «α και β». Μια ακόμη προσπάθεια πραγματοποιήθηκε το 1954 με τη χρήση προσομοιώσεων σε υπολογιστή από τους Farley και Clark και αργότερα το 1956 από τους Rochester, Holland, Haibit και Duda. Η πρώτη ομάδα εκ αυτών διατηρούσε στενή επαφή με νευροεπιστήμονες στο Πανεπιστήμιο McGill. Έτσι κάθε φορά που αντιμετώπιζαν κάποιο πρόβλημα και τα μοντέλα τους δεν λειτουργούσαν, ζητούσαν την βοήθεια των νευροεπιστημόνων. Αυτή η αλληλεπίδραση καθιέρωσε μια τάση ανταλλαγής γνώσεων και μεθοδολογιών ανάμεσα στις δυο αυτές επιστήμες η οποία συνεχίζεται μέχρι σήμερα. (Sutton and Barto, 1998) Με την συνεργασία νευροεπιστημόνων, ψυχολόγων και μηχανικών πραγματοποιήθηκε ανάπτυξη των νευρωνικών δικτύων, με την χρήση εξελιγμένων προσομοιώσεων. Το 1958 ο Rosenblatt

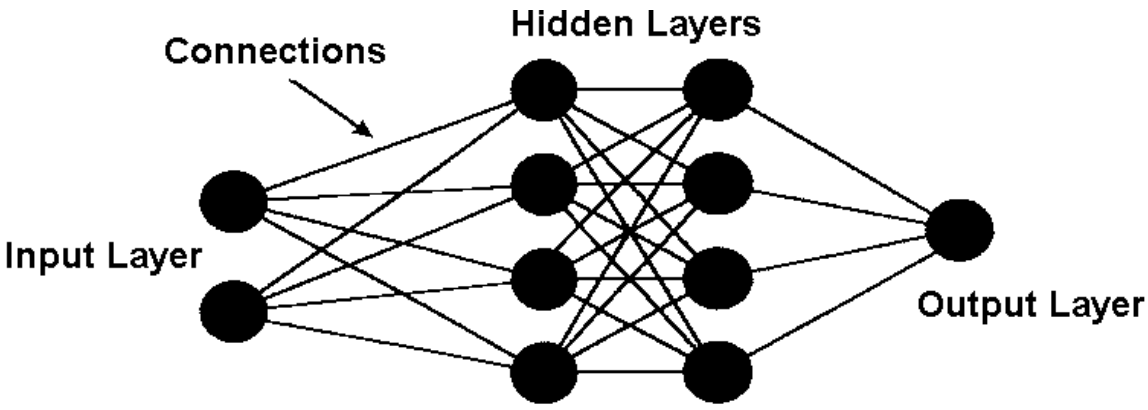
έδειξε μεγάλο ενδιαφέρον σε αυτόν τον τομέα, σχεδιάζοντας και αναπτύσσοντας το νευρωνικό δίκτυο γνωστό και ως «Perceptron». (Rosenblatt, 1957) Ένα άλλο σύστημα ήταν το ADALINE (ADAptive LInear Element), το οποίο αναπτύχθηκε το 1960 από τους Widrow και Hoff. Η ADALINE ήταν μια αναλογικού τύπου ηλεκτρονική συσκευή. Η μέθοδος που χρησιμοποιούσε για την εκμάθηση ήταν διαφορετική από εκείνη που χρησιμοποιούσε το Perceptron. Το 1969 οι Minsky και Papert (Minsky and Papert, 1969) έγραψαν ένα βιβλίο στο οποίο τόνιζαν τους περιορισμούς που παρουσίαζαν τα Perceptrons σε πολυστρωματικά συστήματα. Η δημοσίευση αυτή είχε τόσο σημαντικό αντίκτυπο στην ερευνητική κοινότητα που οδήγησε στην δραματική μείωση της χρηματοδότησης της έρευνας σχετικά με τα νευρωνικά δίκτυα. Το 1974 ο Paul Werbos ανέπτυξε και χρησιμοποίησε τη μέθοδο εκμάθησης οπίσθιας τροφοδότησης (back-propagation). Παρόλα αυτά πέρασαν αρκετά χρόνια μέχρι να διαδοθεί αυτή η προσέγγιση. Σήμερα, τα δίκτυα εκπαίδευσης οπίσθιας τροφοδότησης είναι ίσως η πιο διαδεδομένη εφαρμογή των τεχνητών νευρωνικών δικτύων. Ο Fukushima (Kunihiko,1975) ανέπτυξε ένα, εκπαιδευόμενο σε βήματα, νευρωνικό δίκτυο πολλαπλών επιπέδων για ερμηνεία χειρόγραφων χαρακτήρων. Το αρχικό δίκτυο εκδόθηκε το 1975 και ονομαζόταν Cognitron. Η πρόοδος κατά τα τέλη του 1970 και στις αρχές της δεκαετίας του 1980 ήταν σημαντική για την επαναπυροδότηση του ενδιαφέροντος στον τομέα των νευρωνικών δικτύων. Διάφοροι παράγοντες έπαιξαν ρόλο σε αυτό. Σημαντικό ρόλο έπαιξαν διάφορα βιβλία και συνέδρια που προωθούσαν τα ΤΝΔ και την προσφορά αυτών σε διάφορες ειδικότητες προτείνοντας εξειδικευμένες τεχνικές. Επίσης τα μέσα μαζικής ενημέρωσης παρουσίασαν μεγάλο ενδιαφέρον για αυτή την δραστηριότητα και βοήθησαν ουσιαστικά στη διάδοση της συγκεκριμένης τεχνολογίας. Ακολούθως εμφανίστηκαν τα πρώτα Ακαδημαϊκά προγράμματα και σχετικά μαθήματα εφαρμόστηκαν στα περισσότερα μεγάλα Πανεπιστήμια σε ΗΠΑ και Ευρώπη. Στις μέρες μας παρατηρείται σημαντική δραστηριότητα στον τομέα αυτό, και το μέλλον του είναι πολλά υποσχόμενο. (Strickland, 2016)

2.10.3 Εισαγωγή στους Τεχνητούς Νευρώνες

Κατά αναλογία με τα βιολογικά νευρωνικά δίκτυα, τα τεχνητά νευρωνικά δίκτυα είναι δίκτυα που αποτελούνται από απλούς υπολογιστικούς κόμβους (νευρώνες), διασυνδεδεμένους μεταξύ τους. Οι *νευρώνες* είναι τα δομικά στοιχεία του δικτύου. Κάθε τέτοιος κόμβος δέχεται ένα σύνολο αριθμητικών εισόδων από διαφορετικές πηγές (είτε από άλλους νευρώνες, είτε από το περιβάλλον), πραγματοποιεί έναν υπολογισμό με βάση αυτές τις εισόδους και παράγει μία αριθμητική έξοδο. (Rahman et al., 2016). Η συγκεκριμένη έξοδος είτε κατευθύνεται στο περιβάλλον, είτε τροφοδοτείται ως είσοδος σε άλλους νευρώνες του δικτύου.

Οι νευρώνες βρίσκονται κατανεμημένοι σε τριών ειδών επίπεδα (layers): (Σχήμα 2.9)

- το επίπεδο *εισόδου (input-layer)*
- το επίπεδο *εξόδου* (output-layer)
- *τα επίπεδο υπολογισμών ή κρυμμένα επίπεδα (hidden-layers)*



Σχήμα 2.9: Αναπαράσταση τεχνητού νευρωνικού δικτύου. Με ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και δύο κρυμμένα επίπεδα.

Οι νευρώνες εισόδου δεν πραγματοποιούν κανέναν υπολογισμό ενώ οι νευρώνες εξόδου διοχετεύουν στο περιβάλλον τις τελικές αριθμητικές τιμές εξόδου του δικτύου. Κατά αναλογία με τους βιολογικούς νευρώνες, στις συνδέσεις μεταξύ των νευρώνων στα τεχνητά νευρωνικά δίκτυα αντιστοιχεί ένας πραγματικός αριθμός, που ονομάζεται συντελεστής βάρους ή συναπτικό βάρος. Το συναπτικό βάρος χρησιμοποιείται για την τροποποίηση των δεδομένων εισόδου του νευρώνα όπως και οι συνάψεις μεταξύ των βιολογικών νευρώνων. Κάθε νευρώνας συνδέεται με όλους τους νευρώνες των προηγούμενων και των επόμενων επιπέδων. Οι νευρώνες του ιδίου επιπέδου δεν αλληλεπιδρούν ποτέ μεταξύ τους.

2.10.4 Αρχιτεκτονικές των ΤΝΔ

Η αρχιτεκτονική των ΤΝΔ εξαρτάται από τα είδη των συνδέσεων και τον τρόπο διάδοσης των δεδομένων μεταξύ των νευρώνων. Διακρίνουμε δυο τρόπους διάδοσης στα ΤΝΔ, πρόσθιας τροφοδότησης και ανατροφοδότησης.

Στην **πρόσθια τροφοδότηση** το σήμα μεταφέρεται μόνο προς μία κατεύθυνση, από την είσοδο του νευρωνικού δικτύου προς την έξοδο. Έτσι το σήμα διαδίδεται ώστε να μην υπάρχει νευρώνας που η έξοδος του να είναι είσοδος κάποιου νευρώνα του ιδίου ή του προηγούμενου επιπέδου. Η έξοδος ενός επιπέδου επηρεάζει μόνο τους νευρώνες του επόμενου επιπέδου. Τα **Perceptrons** αποτελούν την απλούστερη μορφή δικτύου πρόσθιας τροφοδότησης.

Στα δίκτυα με **ανατροφοδότηση** τα σήματα κατευθύνονται και προς τις δύο κατευθύνσεις, αφού υπάρχουν βρόγχοι στο δίκτυα. Τα δίκτυα αυτά είναι εν γένει πολύ ισχυρά και ιδιαίτερα πολύπλοκα. Η κατάσταση τους αλλάζει συνεχώς μέχρι να φτάσουν σε μια κατάσταση ισορροπίας, στην οποία παραμένουν μέχρι να αλλάξει η είσοδος και να βρεθεί άλλο σημείο ισορροπίας.

2.10.5 Δυνατότητες των ΤΝΔ

Το έντονο ενδιαφέρον που παρατηρείται για τα ΤΝΔ προκύπτει κυρίως από τη δυνατότητα τους να επιλύουν περίπλοκα υπολογιστικά προβλήματα του πραγματικού κόσμου. Μερικές από τις δυνατότητες των ΤΝΔ είναι:

- **Αντικατάσταση μαθηματικού μοντέλου.** Η αξία των ΤΝΔ βασίζεται κυρίως στο γεγονός ότι δεν χρειάζεται απαραίτητα τη δημιουργία κάποιου μαθηματικού μοντέλου να περιγράφει το προς εξέταση πρόβλημα. Οπότε σε περιπτώσεις που η δημιουργία κάποιου μαθηματικού μοντέλου είναι ιδιαίτερα δύσκολη για τη περιγραφή του προβλήματος τότε τα ΤΝΔ καλούνται να πάρουν θέση. Με την διαδικασία της εκπαίδευσης του ΤΝΔ τα δεδομένα εξόδου θα πρέπει να ταυτίζονται με τα επιθυμητά δεδομένα εξόδου, με αποτέλεσμα να δημιουργείται μια συσχέτιση μεταξύ των δεδομένων εισόδου και εξόδου, χωρίς τη χρήση κάποιου προκαθορισμένου μαθηματικού μοντέλου.

- **Προσαρμογή.** Τα ΤΝΔ έχουν την δυνατότητα να μεταβάλλουν την απόκριση τους, μεταβάλλοντας τα συναπτικά βάρη των νευρώνων, ανάλογα με τις τιμές εισόδου. Οπότε σε περιβάλλοντα που αλλάζουν συνεχώς, τα ΤΝΔ μπορούν να εφαρμοστούν και να προσαρμοστούν ανάλογα, με τα καινούργια δεδομένα εισόδου και εξόδου.

- **Ανεκτικότητα σε σφάλματα.** Τα ΤΝΔ που πραγματοποιούνται σε υλικό (hardware) έχουν την δυνατότητα της ανεκτικότητας σε σφάλματα, διότι η απόδοση του συστήματος μειώνεται ομαλά σε περιπτώσεις που εμφανίζεται κάποιο λάθος. Οπότε σε περίπτωση που καταστραφεί κάποιος νευρώνας, το ΤΝΔ θα συνεχίσει να λειτουργεί αλλά με μειωμένη απόδοση.

- **Ανεκτικότητα στο θόρυβο.** Το ΤΝΔ είναι ανεκτικό στο θόρυβο, δηλαδή εάν εκπαιδευτεί να αναγνωρίζει ένα συγκεκριμένο πρότυπο μέσα στα δεδομένα, σε περίπτωση που τα εν λόγω δεδομένα έχουν υποστεί κάποια αλλοίωση (θόρυβο) τότε η ταξινόμηση τους από το ΤΝΔ δεν επηρεάζεται.

2.10.6 Γενικευμένο ΤΝΔ πρόσθιας τροφοδότησης

Τα ΤΝΔ πρόσθιας τροφοδότησης αποτελούν τον πιο διαδεδομένο τύπο ΤΝΔ. Πρέπει να σημειωθεί ότι ο βέλτιστος αριθμός κρυμμένων επιπέδων καθώς και ο αριθμός των νευρών σε κάθε ένα από αυτά είναι περισσότερο θέμα «τέχνης» παρά επιστήμης. Ενώ ο αριθμός των νευρώνων στα επίπεδα εισόδου και εξόδου καθορίζονται από τη φυσική του προβλήματος. Οι Kavzoglou and Mather (2003) υποστηρίζουν ότι η επιλογή των κρυμμένων επιπέδων αλλά και ο αριθμός των νευρώνων σε αυτά εξαρτώνται από την πολυπλοκότητα του προβλήματος και τον αριθμό των μοτίβων που χρησιμοποιούνται στην εκπαίδευση. Λέγοντας μοτίβα, αναφέρονται σε ένα διάνυσμα εισόδου-εξόδου. Σύμφωνα με τον Bishop (1995) ένα ΤΝΔ πρόσθιας τροφοδότησης με ένα ή περισσότερα κρυμμένα επίπεδα, μπορεί να μάθει οποιαδήποτε συνεχή απεικόνιση, με μία αυθαίρετη ακρίβεια. Η ύπαρξη περισσότερων του ενός επιπέδου μπορεί να είναι ωφέλιμη για κάποια προβλήματα αλλά συνήθως ένα κρυμμένο επίπεδο είναι αρκετό. Παρόλα αυτά φαίνεται ότι το αποτέλεσμα της αύξησης του αριθμού των κρυμμένων επιπέδων στο Νευρωνικό Δίκτυο το κάνει πιο «έξυπνο», ενώ το αποτέλεσμα της αύξησης του αριθμού των νευρώνων στα κρυμμένα επίπεδα κάνει το δίκτυο πιο «ακριβές».

Υπάρχουν διάφοροι τύποι με τους οποίους μπορεί κανείς να υπολογίσει τον απαραίτητο αριθμό νευρώνων ανα κρυμμένο επίπεδο και αυτοί προέκυψαν από διαισθητική ανάλυση. Σύμφωνα με τον Kavzoglou and Mather (2003) αυτοί διακρίνονται στους ακόλουθους τύπους (Πίνακας 2.1):

Τύπος	Πηγή
$2 \times N_i$ή$3 \times N_i$	Kanellopoulos amd Wilkinson (1997)
$3 \times N_i$	Hush (1989)
$2 \times N_i + 1$	Heeht-Nielsen (1987)
$2 \times \frac{N_i}{3}$	Wang (1994)
$\frac{(N_i + N_o)}{2}$	Ripley (1993)
$\frac{N_p}{[r \times (N_i + N_o)]}$	Garson (1998)
$\frac{2 + N_o \times N_i + \frac{1}{2} \times N_o \times (N_i^2 + N_i) - 3}{N_i + N_o}$	Paola (1994)

Πίνακας 2.1: Τύποι που προέκυψαν από διαισθητική ανάλυση, για τον υπολογισμό του αριθμού των νευρώνων στα κρυμμένα επίπεδα.

N_i: αριθμός τιμών εισόδου, N_o: αριθμός τιμών εξόδου, N_p: αριθμός μοτίβων
r ~[5,10] ανάλογα με το θόρυβο στα δεδομένα

Αξίζει να σημειωθεί ότι στην παρούσα εργασία χρησιμοποιήθηκαν διάφοροι αριθμοί νευρώνων στα κρυμμένα επίπεδα, έτσι ώστε να βρεθεί ο συνδυασμός που ελαχιστοποιεί το σφάλμα.

Κάθε νευρώνας μπορεί να θεωρηθεί σαν ένα «μαύρο κουτί» το οποίο εκτελεί εσωτερικά δύο βασικές λειτουργίες:

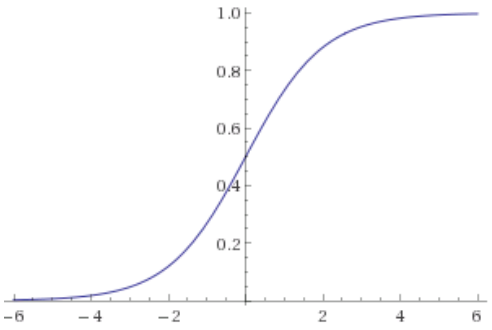
i. Ένα γραμμικό συνδυασμό που υπολογίζει ένα άθροισμα των σημάτων εισόδου.

ii. Μια κλιμάκωση που περιορίζει τα αποτελέσματα του αθροίσματος σε ένα διάστημα, μέσω της συνάρτησης ενεργοποίησης.

Η συνάρτηση ενεργοποίησης μπορεί να είναι βηματική (step transfer function), γραμμική (linear transfer function) ή μη-γραμμική (non-linear transfer function). Η συνάρτηση ενεργοποίησης που χρησιμοποιείται ευρέως στα ΤΝΔ είναι μια μη-γραμμική συνάρτηση και συγκεκριμένα μια σιγμοειδής συνάρτηση (sigmoid function). Η γραφική της απεικόνιση είναι της μορφής S και είναι μια αύξουσα διαφορίσιμη συνάρτηση. Μία έκφραση της σιγμοειδούς συνάρτησης είναι η λογιστική συνάρτηση της οποίας το εύρος των αποτελεσμάτων βρίσκεται στην περιοχή τιμών [0,1]. (Σχήμα 2.10) Ο τύπος της λογιστικής σιγμοειδούς συνάρτηση είναι (εξίσωση 2.1):

$$f(x) = \frac{1}{1+e^{-x}} \qquad \text{(εξίσωση 2.1)}$$

Όπως φαίνεται και από το Σχήμα 2.10 που ακολουθεί η συγκεκριμένη συνάρτηση ανάλογα με την τιμή του τοπικού νευρώνα δίνει στην έξοδο της τιμές που βρίσκονται μέσα στα όρια [0,1].

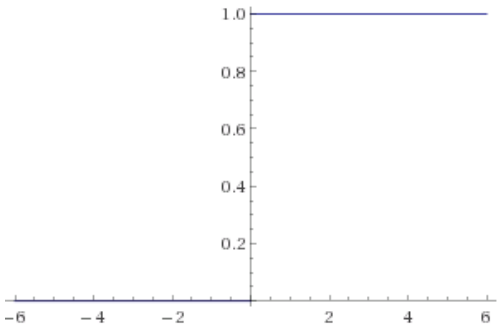


Σχήμα 2.10: Λογιστική σιγμοειδής συνάρτηση.

Ως συνάρτηση ενεργοποίησης όπως είπαμε και πιό πάνω μπορεί να έχουμε μια βηματική συνάρτηση. Μια βηματική συνάρτηση, μπορεί να περιγραφεί απο τον ακόλουθω τύπο (εξίσωση 2.2),

$$f(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases} \qquad \text{(εξίσωση 2.2)}$$

Η λειτουργία της βηματικής εξίσωσης είναι απλή. Όταν το τοπικό πεδίο έχει αρνητική τιμή, τότε η συνάρτηση ενεργοποίησης δίνει την τιμή 0 στην έξοδο του νευρώνα. Αντίθετα όταν το τοπικό πεδίο είναι θετικό ή μηδέν, τότε η συνάρτηση ενεργοποίησης δίνει τιμή ίση με τη μονάδα στην έξοδο του νευρώνα. Το βασικό μειονέκτημα της συνάρτησης αυτής είναι ότι η παράγωγός της απειρίζεται, πράγμα που δεν είναι επιθυμητό στα ΤΝΔ. Στο σχήμα που ακολουθεί (Σχήμα 2.11) φαίνεται μια βηματική συνάρτηση.

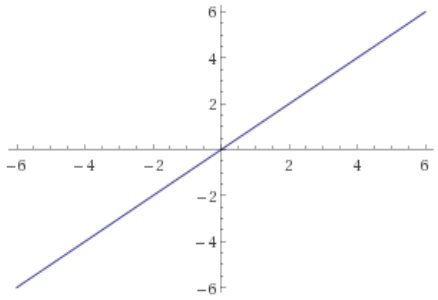


Σχήμα 2.11: Βηματική συνάρτηση.

Η γραμμική συνάρτηση ενεργοποίησης μπορεί να περιγραφεί από τον παρακάτω τύπο (εξίσωση 2.3):

$$f(x) = x \qquad \text{(εξίσωση 2.3)}$$

ή οποιαδήποτε άλλη γραμμική συνάρτηση. Όπως φαίνεται και από τον τύπο της, η γραμμική συνάρτηση δίνει ως έξοδο του νευρώνα την τιμή του τοπικού πεδίου. Δεν χρησιμοποιούνται ευρέως στα ΤΝΔ, αφού η έξοδος του νευρώνα θα είναι ευθέως ανάλογη της εισόδου. Στο σχήμα που ακολουθεί (Σχήμα 2.12) φαίνεται μια γραμμική συνάρτηση.

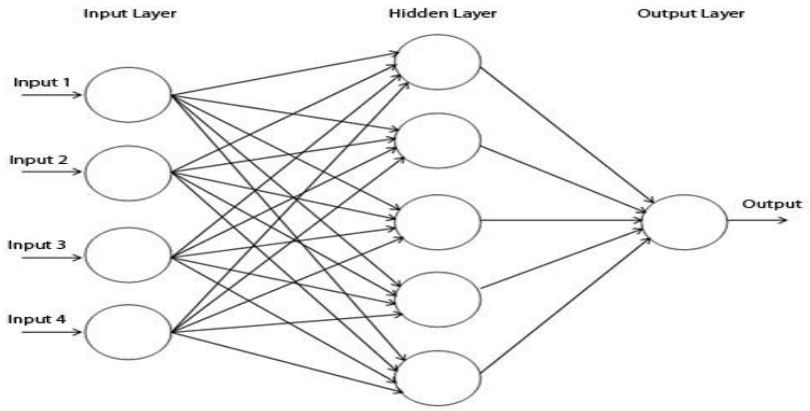


Σχήμα 2.12: Γραμμική συνάρτηση.

Ένα ΤΝΔ μπορεί να θεωρηθεί ως μια γενική προσέγγιση, που έχει την ικανότητα να προσομοιώνει προβλήματα που μπορούν να αναχθούν στην ακόλουθη μορφή (εξίσωση 2.4):

$$z = z(x_1, x_2, \ldots, x_n) \qquad \text{(εξίσωση 2.4)}$$

Το παρόν ΤΝΔ χρησιμοποιεί τη σιγμοειδή συνάρτηση ως συνάρτηση ενεργοποίησης και όλα τα δεδομένα εισόδου-εξόδου κανονικοποιούνται στο διάστημα [0.1]. Για να μπορέσουμε να καταλάβουμε πώς οι τιμές διαδίδονται μέσω διαδοχικών επιπέδων, μπορούμε να φανταστούμε ότι έχουμε ένα επίπεδο εισόδου με 4 νευρώνες, ένα κρυμμένο επίπεδο με 5 νευρώνες και ένα νευρώνα στην έξοδο, όπως φαίνεται στο Σχήμα 2.13, που τροφοδοτούν στο σύστημα τις τιμές x_i (i = 1, 2, 3, 4).



Σχήμα 2.13: Παράδειγμα Νευρωνικού Δικτύου πρόσθιας τροφοδότησης. Με τέσσερεις νευρώνες στο επίπεδο εισόδου, ένα κρυμμένο επίπεδο με πέντε νευρώνες και ένα νευρώνα στο επίπεδο εξόδου.

Πηγή: https://www.codeproject.com/KB/dotnet/predictor/network.jpg

Καθένας από τους νευρώνες εισόδου θα περάσει αυτή την τιμή πολλαπλασιασμένη με το αντίστοιχο συναπτικό βάρος με το οποίο συνδέεται καθένας από τους νευρώνες του επιπέδου εισόδου με τους νευρώνες του κρυμμένου επιπέδου (βάρη w_{ij}, i=1,2,3,4 και j=1,2,3,4,5). Οπότε, καθένας από τους νευρώνες j του κρυμμένου επίπεδου δέχεται σήμα ίσο με (εξίσωση 2.5):

$$h_j^{\text{in}} = \sum_{i=1}^{4} x_i \,.\, w_{ij} \qquad \text{j=1,…, 5} \qquad \text{(εξίσωση 2.5)}$$

Αυτά τα σήματα θα περάσουν στη συνέχεια μέσω της σιγμοειδούς συνάρτησης ενεργοποίησης και θα δώσουν την ακόλουθη τιμή εξόδου για κάθε νευρώνα του κρυμμένου επιπέδου (εξίσωση 2.6):

$$h_j\left(h_j^{\text{in}}\right) = \frac{1}{1+e^{-h_j^{\text{in}}}} \qquad \text{j=1,…,5} \qquad \text{(εξίσωση 2.6)}$$

Στη συνέχεια, οι τιμές από το κρυμμένο επίπεδο περνάνε με αντίστοιχη διαδικασία στο νευρώνα εξόδου, δηλαδή πρώτα αθροίζονται οι τιμές h_j πολλαπλασιασμένες με το αντίστοιχο συναπτικό βάρος και μετά το άθροισμα περνάει μέσα από τη σιγμοειδή συνάρτηση, για να δώσει έτσι την τιμή εξόδου. Εάν υπήρχαν περισσότερα κρυμμένα επίπεδα, η πιο πάνω διαδικασία επαναλαμβάνεται για κάθε νευρώνα του νέου επιπέδου, μέχρι οι τιμές να φτάσουν στο επίπεδο εξόδου. Με αυτό τον τρόπο το ΤΝΔ μπορεί να θεωρηθεί σαν ένα «δέντρο», το οποιό διαδίδει προς τα εμπρός την είσοδο.

Κάθε ένας απο τους κλάδους αυτού του «δέντρου», είναι ουσιαστικά ένα συναπτικό βάρος που καθορίζει το μέγεθος της σύνδεσης κάθε νευρώνα σε ενα δεδομένο επίπεδο με κάθε νευρώνα του επόμενου επιπέδου. Έτσι αν είναι γνωστά όλα τα συνοπτικά βάρη του δικτύου και έχουμε ενα δεδομένο συνδιασμό τιμών εισόδου, τότε μια μοναδική τιμή εξόδου.

Η διαδικασία που περιγράφηκε παραπάνω αναφέρεται στον προγνωστικό τρόπο των Νευρωνικών Δικτύων με τροφοδοσία προς τα εμπρός, στην περίπτωση του οποίου θα πρέπει να γνωρίζουμε όλα τα συναπτικά βάρη. Στα προβλήματα μας όμως οι τιμές αυτές ήταν άγνωστες, με αποτέλεσμα να χρειαζόμαστε τη χρήση κάποιου άλλου τρόπου. Έτσι χρησιμοποιήσαμε τα τροφοδοτούμενα προς τα εμπρός νευρωνικά δίκτυα με τη χρήση του αλγορίθμου οπίσθιας διάδοσης. Τα ΤΝΔ πρόσθιας τροφοδότησης - οπίσθιας διάδοσης αποτελούν το πιο διαδεδομένο τύπο ΤΝΔ και η περιγραφή τους δίδεται στη συνέχεια.

2.10.7 ΤΝΔ τροφοδοτούμενο προς τα εμπρός με αλγόριθμο οπίσθιας διάδοσης (Feed Forward Back Propagation)

Ένα «Feed Forward Back Propagation» δίκτυο εκπαιδεύεται από ένα επαρκή αριθμό μοτίβων, δηλαδή διανυσμάτων εισόδου-εξόδου. Η διαδικασία εκπαίδευσης αρχίζει με μια τυχαία κατανομή των τιμών των συνοπτικών βαρών του δικτύου. Μέσω της διαδικασίας που περιγράφηκε παραπάνω παράγεται ένα μοναδικό και «λανθασμένο» διάνυσμα εξόδου, στην έξοδο του Νευρωνικού Δικτύου, για ένα συγκεκριμένο μοτίβο. Ωστόσο, επειδή είναι γνωστό το διάνυσμα που αντιστοιχεί στο διάνυσμα εισόδου του συγκεκριμένου μοτίβου, μπορεί να υπολογιστεί ένα μέτρο του σφάλματος μέσω της συνάρτησης σφάλματος που δίνεται απο τον ακόλουθο τύπο (εξίσωση 2.7):

$$E = \frac{1}{2}\sum_{k=1}^{K}(y_k - t_k)^2 \qquad \text{(εξίσωση 2.7)}$$

όπου το K αντιστοιχεί στον αριθμό των νευρώνων εξόδου, και τα y_k, t_k αντιπροσωπεύουν την τιμή εξόδου και την τιμή επιθυμητής εξόδου, αντίστοιχα.

Ο στόχος του αλγόριθμου οπίσθιας διάδοσης είναι να βρούμε την συγκεκριμένη κατανομή βαρών που ελαχιστοποιεί την συνάρτηση απόδοσης. (Schmidhuber, 2014) Πιο κάτω ακολουθεί αναλυτική περιγραφή που εξηγεί βήμα προς βήμα τον αλγόριθμο αυτό. Για να γίνει αυτό σχετικά εύκολα, χωρίς απώλεια της γενικότητας, θεωρείται ότι το Νευρωνικό Δίκτυο αποτελείται από ένα μόνο κρυμμένο επίπεδο.

Βήμα 1

Υπολογίζεται η παράγωγος της εξίσωσης 2.7,

$$\frac{\partial E}{\partial y_k} = y_k - t_k \qquad \text{(εξίσωση 2.8)}$$

Η παράγωγος αυτή αντιπροσωπεύει το ρυθμό μεταβολής της συνάρτησης σφάλματος.

Βήμα 2

Ακολούθως η αλλαγή αυτή μεταδίδεται προς τα πίσω στην είσοδο του k-οστού νευρώνα του επιπέδου εξόδου. Έτσι το αποτέλεσμα που έχει μια μεταβολή του διανύσματος εξοδού στις τιμές εισόδου των νευρώνων στο επίπεδο εξόδου, υπολογίζεται απο το τύπο (εξίσωση 2.9):

$$\Delta y_k = \frac{\partial E}{\partial x_k} \qquad \text{(εξίσωση 2.9)}$$

όμως ο όρος$\frac{\partial E}{\partial x_k}$ μπορεί να γραφεί ώς,

$$\frac{\partial E}{\partial x_k} = \frac{\partial E}{\partial y_k} \times \frac{dy_k}{dx_k} \qquad \text{(εξίσωση 2.10)}$$

όπου το y_k είναι η συνάρτηση ενεργοποίησης και έχει την εξής μορφή,

$$y_k(x_k) = \frac{1}{1+e^{-x_\kappa}} \qquad \text{(εξίσωση 2.11)}$$

Έτσι συνδιάζοντας την εξίσωση 2.8, 2.10, 2.11 η εξίσωση 2.9 παίρνει την ακόλουθη μορφή,

$$\Delta y_k = \frac{\partial E}{\partial x_k} = (y_k - t_k) \times y_k \times (1 - y_k) \qquad \text{k=1, 2, 3, …, K} \quad \text{(εξίσωση 2.12)}$$

Βήμα 3

Σε αυτή τη φάση, γίνεται η μετάδοση του Δy_kπρος τα πίσω, μέσω των συνδέσεων των εσωτερικών επιπέδων w_{jk} και της συνάρτησης ενεργοποίησης των νευρώνων του κρυμμένου επιπέδου, στην είσοδο του κρυμμένου επιπέδου. Έτσι, προκύπτει (εξίσωση 2.13):

$$\Delta h_j = \sum_{k=1}^{K}\left(w_{jk} \times \Delta y_k\right) \times h_j \times \left(1 - h_j\right) \qquad \text{(εξίσωση 2.13)}$$

όπου το h_j δίνεται από την εξίσωση 2.14:

$$h_j = h_j^{in} = \sum_{i=1}^{I} y_i \times W_{ij} \qquad j=1, 2, 3, \ldots, J \quad \text{(εξίσωση 2.14)}$$

Το Δh_jστην πιο πάνω εξίσωση (εξίσωση 2.15) αντιπροσωπεύει το μέγεθος της μεταβολής των τιμών h_jστην είσοδο των νευρώνων του κρυμμένου επιπέδου.

Βήμα 4

Στο σημείο αυτό θα πρέπει να υπολογιστούν οι μεταβολές στις τιμές των συνοπτικών βαρών w_{ij}, μεταξύ του επιπέδου εισόδου και του κρυμμένου επιπέδου. Αυτό γίνεται χρησιμοποιώντας τον γενικευμένο κανόνα δέλτα και έτσι προκύπτει η ακόλουθη εξίσωση:

$$\Delta^n w_{ij} = \eta \times \Delta h_j + \alpha \times \Delta^{n-1} w_{ij} \qquad \text{(εξίσωση 2.15)}$$

και καταυτόν τον τρόπο έχουμε την μεταβολή στις τιμές w_{ij} :

$$w_{ij} = w_{ij} + \Delta^n w_{ij} \qquad i=1, 2, 3, \ldots, I \qquad \text{(εξίσωση 2.16)}$$

όπου οι σταθερές η και α είναι ο ρυθμός εκμάθησης και η ταχύτητα εκμάθησης, αντίστοιχα.

Βήμα 5

Το τελευταίο βήμα της μεθόδου αυτής είναι η τροποποίηση των τιμών που έχουν τα συναπτικά βάρη w_{jk}, μεταξύ των κρυμμένων επιπέδων και του επιπέδου εξόδου. Κατά αναλογία με την εξίσωση (2.15), προκύπτουν οι εξισώσεις (2.17), (2.18):

$$\Delta^{n} w_{jk} = \eta \times \Delta y_k + \alpha \times \Delta^{n-1} w_{jk} \qquad \text{(εξίσωση 2.17)}$$

και έτσι έχουμε την μεταβολή στις τιμές w_{jk},

$$w_{jk} = w_{jk} + \Delta^{n} w_{jk} \qquad \text{(εξίσωση 2.28)}$$

Για περισσότερες πληροφορίες σχετικά με το πώς καταλήξαμε στη πιο πάνω εξίσωση, μπορείτε να ανατρέξετε στην βιβλιογραφία. (Spentzos, 2005)

Για να αποφευχθεί η σύγχυση οι δείκτες i, k, j αναφέρονται στα επίπεδα εισόδου, στα κρυμμένα επίπεδα και στα επίπεδα εξόδου αντίστοιχα. Ενώ τα I, K, J αναφέρονται στο συνολικό αριθμό νευρώνων των αντίστοιχων επιπέδων i, k, j. Επίσης η μεταβλητή n, είναι μια γενικευμένη μεταβλητή επανάληψης.

Αφού ολοκληρώνεται ο αλγόριθμος αυτός, όλες οι τιμές των συνοπτικών βαρών στο ΤΝΔ έχουν επανεκτιμηθεί σε τιμές που προσφέρουν μικρότερο σφάλμα στην συνάρτηση σφάλματος, Ωστόσο, η διαδικασία αυτή έχει εφαρμοστεί μόνο για ένα μοτίβο και μόνο μια φορά. Οπότε, στη συνέχεια η διαδικασία αυτή επαναλαμβάνεται για όλα τα μοτίβα και επομένως την ολοκλήρωση μιας **εποχής**. Ως μία **εποχή**, μπορεί να οριστεί ένα πλήρες πέρασμα μέσα στο δίκτυο. Τέλος, η διαδικασία επαναλαμβάνεται για τον απαραίτητο αριθμό εποχών για να ικανοποιηθεί με το τρόπο αυτό το κριτήριο ελαχιστοποίησης της συνάρτησης σφάλματος.

Είναι σημαντικό να αναφερθεί ότι η εισαγωγή των μοτίβων σε κάθε εποχή γίνεται με τυχαία σειρά. Αυτό εξασφαλίζει ταχύτερους ρυθμούς εκμάθησης, αποφεύγοντας με τον τρόπο αυτό την απομνημόνευση και αυξάνει την ικανότητα του ΤΝΔ να αντιμετωπίζει προβλήματα στα οποία δεν έχει εκπαιδευτεί.

Στο σημείο αυτό θα ήταν καλό να αναφερθεί ότι σύμφωνα με διάφορα προβλήματα που έχουν επιλυθεί με ΤΝΔ πρόσθιας τροφοδότησης με αλγόριθμο οπίσθιας διάδοσης, υπάρχουν συγκεκριμένες τιμές των σταθερών ταχύτητας εκμάθησης και ρυθμό εκμάθησης που ανταποκρίνονται καλύτερα στη λύση του προβλήματος. Οι τιμές αυτές έχουν προκύψει από διαισθητική ανάλυση και σύμφωνα με τους Kavzoglou and Mather (2003) οι τιμές αυτών είναι (Πίνακας 2.2):

Ρυθμός Εκμάθησης (η)	Ταχύτητα Εκμάθησης (α)	Πηγή
0.01	0.00005	Paola and Schowengerdt (1997)
0.05	-	Lawrence et al. (1996)
0.05	0.5	Partridge and Yates (1996)
0.1	-	Haykin (1999), Gallagher and Downs (1997)
0.1	0.3	Ardö et al. (1997)
0.1	0.9	Foody et al. (1996), Pierce et al. (1994)
0.15 (0.04)	0.075 (0.02)	Eberhart and Dobbins (1990)
0.2	-	Bischof et al. (1992)
0.2	0.6	Gong et al. (1996)
0.25	0.9	Swingler (1996)
0.3	0.6	Gopal and Woodcock (1996)
0.5	0.9	Hara et al. (1994)
0.8	-	Staufer and Fischer (1997)

Πίνακας 2.2: Οι τιμές του ρυθμού εκμάθησης και της ταχύτητας εκμάθησης που προέκυψαν απο διαισθητική ανάλυση .

Γενικά όμως, έχει παρατηρηθεί ότι η χρήση των τιμών 0.2 για τον ρυθμό εκμάθησης και [0.2,0.6] για την ταχύτητα εκμάθησης είναι πολύ αποτελεσματικές για τη ικανοποιητική εκπαίδευση του δικτύου. (*Spentzos, 2005)*

Τέλος, κλείνοντας τη συζήτηση αυτή για τα ΤΝΔ είναι απαραίτητο να αναφερθεί ότι πολύ σημαντικό ρόλο στην εκπαίδευση του δικτύου παίζει και ο αριθμός των μοτίβων που θα χρησιμοποιηθούν για την εκπαίδευση του. Έχοντας επαρκή αριθμό μοτίβων η εκπαίδευση του δικτύου γίνεται πιο αποτελεσματική. Από διαισθητική ανάλυση έχουν προκύψει διάφοροι τύποι με τους οποίους μπορεί κανείς να υπολογίσει τον αριθμό των μοτίβων που είναι απαραίτητοι για την εκπαίδευση του δικτύου. Οι τύποι αυτοί παρουσιάζονται στο πιο κάτω πίνακα (Πίνακας 2.3):

Αριθμός Μοτίβων	Πηγή
$5 \times N_W$	Klimasaukas (1993)
$10 \times N_w$	Baum and Haussler (1989)
$30 \times N_i \times (N_i + 1)$	Hush (1989) (κατώτατο όριο)
$60 \times N_i \times (N_i + 1)$	Hush (1989) (βέλτιστο)
$30 \times N_w$	Garson (1998)

Πίνακας 2.3: Τύποι που προέκυψαν από διαισθητική ανάλυση, για τον υπολογισμό του αριθμού των απαραίτητων μοτίβων για τη βέλτιστη εκπαίδευση του δικτύου.

N_i: αριθμός τιμών εισόδου, N_w: αριθμός συνοπτικών βαρών

2.10.8 Συνοπτική περιγραφή διαδικασίας ανάλυσης δεδομένων μέσω ΤΝΔ

- Γίνεται κατάλληλη επιλογή δεδομένων εισόδου – εξόδου και όλο το πλήθος των δεδομένων κανονικοποιείται στο διάστημα [0, 1]
- Από το συνολικό αριθμό διαθέσιμων δεδομένων επιλέγεται ένας αριθμός μοτίβων για εκπαίδευση του δικτύου (Πίνακας 2.3) και τα υπόλοιπα για επαλήθευση. Ανάλογα με τον αριθμό των δεδομένων εισόδου και εξόδου επιλέγονται ο αριθμός των κρυμμένων επιπέδων και ο αριθμός των κόμβων σε κάθε επίπεδο (Πίνακας 2.1) καθώς και ο αριθμός των διαφόρων παραμέτρων που χρειάζεται το ΤΝΔ (Πίνακας 2.2)
- Εκπαιδεύεται το ΤΝΔ με τα επιλεχθέντα δεδομένα, έως ότου να ικανοποιηθεί κάποιο κριτήριο σφάλματος (συνήθως επιλέγεται το τελικό σφάλμα να είναι κάποιο μικρό ποσοστό του αρχικού, λιγότερο από 1%) ή να συμπληρωθεί ο αριθμός των εποχών για τις οποίες έχει καθοριστεί το δίκτυο ότι πρέπει να τρέξει (συνήθως της τάξης μερικών εκατομμυρίων)
- Αφού τελειώσει η εκπαίδευση του δικτύου, χρησιμοποιούνται τα συναπτικά βάρη που έχουν προβλεφθεί κατά την εκπαίδευση για να διαπιστωθεί η ακρίβεια πρόβλεψης του δικτύου (μέσω της διαδικασίας της επαλήθευσης με δεδομένα για τα οποία δεν έχει εκπαιδευτεί)
- Τέλος, αφού ικανοποιηθούμε με την ακρίβεια του δικτύου, αυτό χρησιμοποιείται για πρόβλεψη σε δεδομένα εισόδου για τα οποία δεν γνωρίζουμε εξαρχής το αποτέλεσμα εξόδου

Ο σειριακός κώδικας σε γλώσσα προγραμματισμού FORTRAN 77 του ΤΝΔ πρόσθιας τροφοδότησης με αλγόριθμο οπίσθιας διάδοσης δίδεται στο Παράρτημα Β. Όλες οι προσομοιώσεις ΤΝΔ στα πλαίσια της παρούσας εργασίας έγιναν στη συστοιχία υπολογιστών METROPOLIS του ερευνητικού κέντρου CCQCN του Τμήματος Φυσικής του Πανεπιστημίου Κρήτης.

ΚΕΦΑΛΑΙΟ 3

ΕΦΑΡΜΟΓΗ ΑΝΑΛΥΣΗΣ ΜΕΓΑΛΩΝ ΔΕΔΟΜΕΝΩΝ ΣΤΟ ΠΡΟΒΛΗΜΑ ΤΩΝ ΑΕΡΟΜΕΤΑΦΟΡΩΝ ΚΑΙ ΧΡΗΣΗ ΤΝΔ ΓΙΑ ΤΗ ΠΡΟΒΛΕΨΗ ΚΑΘΥΣΤΕΡΗΣΕΩΝ

Στο κεφάλαιο αυτό θα ασχοληθούμε με το πρόβλημα των αερομεταφορών και θα χρησιμοποιήσουμε το υπολογιστικό πλαίσιο το οποίο περιγράφηκε στο Κεφάλαιο 2 για την πρόβλεψη χρόνων ομαλοποίησης λειτουργιών αεροδρόμιου μετά από καθυστερήσεις πτήσεων. Αρχικά γίνεται μια γενική εισαγωγή στο θέμα των αερομεταφορών. Ακολούθως, περιγράφεται το στάδιο της ανάκτησης δεδομένων και η διαδικασία επεξεργασίας των ακατέργαστων αυτών δεδομένων, ούτως ώστε να ανακτηθούν τα επιθυμητά δεδομένα για περαιτέρω χρήση. Τέλος, περιγράφεται η διαδικασία της επεξεργασίας και ανάλυσης μέσω ΤΝΔ των επιλεχθέντων δεδομένων.

3.1 Εισαγωγή στο πρόβλημα των αερομεταφορών

Τις τελευταίες δεκαετίες οι αεροπορικές μεταφορές έχουν καταστεί μια από τις βασικότερες μεθόδους μεταφοράς.(Monma and Stoer, 1993, Octavio and Odoni, 1994, Michael, 2011) Με τη μεγάλη αυτή αύξηση της εναέριας κυκλοφορίας σημειώνεται μεγάλη αύξηση της ζήτησης για επέκταση της χωρητικότητας των αερολιμένων. Ωστόσο, η χωρητικότητα του εναέριου χώρου και του αεροδρομίου δεν μπορεί να συνεχίσει να αυξάνεται με το ρυθμό που είναι απαραίτητος για την κάλυψη των αναγκών αυτών. Όταν κατά τη διάρκεια των ωρών αιχμής η ζήτηση για επιπλέον πόρους υπερβαίνει την ικανότητα που μπορεί να έχει ο αερολιμένας, η κατάσταση αυτή είναι γνωστή ως ανισορροπία μεταξύ μεταφορικής ικανότητας και ζήτησης. Η ζήτηση αναφέρεται στον αριθμό των πτήσεων που

προγραμματίζονται να φτάσουν ή να αναχωρήσουν σε ένα αερολιμένα μια δεδομένη χρονική περίοδο. Ως ικανότητα ορίζεται ο μέγιστος αριθμός αφίξεων ή αναχωρήσεων που είναι σε θέση να εξυπηρετήσει ο αερολιμένας σε μια δεδομένη χρονική περίοδο. Ως άμεσο αποτέλεσμα της ανισορροπίας μεταξύ μεταφορικής ικανότητας και ζήτησης έχουμε τη συμφόρηση του αερολιμένα και τη καθυστέρηση πτήσεων (Aisling and Kenneth, 1999).

Οι αεροπορικές εταιρείες είναι οι σημαντικότεροι πελάτες του αεροδρομίου (Ashford and Kenneth, 1999). Η έγκαιρη επίδοση του προγράμματος των αεροπορικών εταιρειών αποτελεί βασικό παράγοντα για τη διατήρηση της ικανοποίησης των πελατών και την προσέλκυση νέων. Το πρόγραμμα πτήσεων του αεροδρομίου είναι το κλειδί για τον καλό προγραμματισμό και την σωστή εκτέλεση των λειτουργιών των αεροπορικών εταιρειών (Wu, 2005). Με κάθε πρόγραμμα, η κάθε αεροπορική εταιρεία καθορίζει τις καθημερινές λειτουργίες της και αναλαμβάνει τους πόρους της για να ικανοποιήσει τις ανάγκες των πελατών της σε όλα τα προγραμματισμένα αεροπορικά ταξίδια.

Κάθε χρόνο περίπου το 20% των αεροπορικών πτήσεων καθυστερούν ή ακυρώνονται, με αποτέλεσμα αυτό να έχει σημαντικό κόστος τόσο στους ταξιδιώτες όσο και στις αεροπορικές εταιρείες. Η καθυστέρηση πτήσης είναι πολύ σύνθετη για να εξηγηθεί, επειδή μια πτήση μπορεί να είναι εκτός προγραμματισμού για διάφορους λόγους, είτε λόγω προβλημάτων στο αεροδρόμιο προέλευσης είτε στο αεροδρόμιο προορισμού ή ακόμη να προκύψουν προβλήματα στο ίδιο το αεροπλάνο κατά τη διάρκεια της πτήσης.

Από το 2003, οι αεροπορικές εταιρίες στις Ηνωμένες Πολιτείες της Αμερικής που αναφέρουν τα δεδομένα σε πραγματικό χρόνο, αναφέρουν επίσης και τα αίτια καθυστερήσεων και ακυρώσεων στο «Bureau of Transportation Statistics». Οι αεροπορικές εταιρίες αναφέρουν τις αιτίες καθυστέρησης με βάση ένα ευρύ φάσμα κατηγοριών που δημιουργήθηκε από τη Συμβουλευτική επιτροπή Αερομεταφορών για την έγκαιρη υποβολή αναφοράς. Οι κατηγορίες αυτές καθορίζονται ως εξής:

I. Αερομεταφορέας

Η αιτία της ακύρωσης ή της καθυστέρησης οφείλεται σε συνθήκες εντός του ελέγχου της αεροπορικής εταιρείας, π.χ. προβλήματα συντήρησης ή πληρώματος, καθαρισμός αεροσκαφών, φόρτωση αποσκευών, τροφοδοσία.

II. Ακραία καιρικά φαινόμενα

Δυσμενείς καιρικές συνθήκες που αποτρέπουν την ομαλή εξέλιξη της πτήσης ή καθιστούν αδύνατη την πραγματοποίησή της με αποτέλεσμα την ακύρωση.

III. Εθνικό Σύστημα Αεροπορίας (NAS)

Καθυστερήσεις και ακυρώσεις που οφείλονται στο εθνικό σύστημα αερομεταφορών, οι οποίες αναφέρονται σε ένα ευρύ φάσμα συνθηκών, όπως οι καιρικές συνθήκες, η λειτουργία των αεροδρομίων, ο μεγάλος όγκος κυκλοφορίας και ο έλεγχος της εναέριας κυκλοφορίας. Οι καθυστερήσεις ή οι ακυρώσεις με κωδικό «NAS» λόγω καιρικών συνθηκών είναι οι καθυστερήσεις που θα μπορούσαν να μειωθούν με την αποκατάσταση των διορθωτικών μέτρων από τα αεροδρόμια ή την Ομοσπονδιακή Διοίκηση Αεροπορίας

IV. Καθυστέρηση Αεροπλάνων

Ορισμένες πτήσεις επηρεάζονται ως συνέπεια της καθυστερημένης άφιξης προηγούμενων πτήσεων, π.χ. μια προηγούμενη πτήση με το ίδιο αεροσκάφος να έφθασε με καθυστέρηση, με αποτέλεσμα να αναχωρήσει αργότερα η τρέχουσα πτήση.

V. Θέματα Ασφάλειας

Οι καθυστερήσεις ή οι ακυρώσεις που προκαλούνται από την εκκένωση τερματικού σταθμού. Ή την προσγείωση αεροσκαφών σε τερματικό διάδρομο που προοριζόταν για χρήση από αεροσκάφος αναχώρησης λόγω εκτάκτου ανάγκης.

Διάφορα μοντέλα έχουν αναπτυχθεί για την επίλυση αυτού του προβλήματος, τα οποία βασίζονται σε πιθανότητες και στατιστικές αναλύσεις. Για παράδειγμα, οι Dou et al. (1999) ανέπτυξαν ένα σύστημα διαχείρισης της εναέριας κυκλοφορίας για δύο προγράμματα της *Εθνικής Υπηρεσίας Αεροναυπηγικής και Διαστήματος* (NASA). Ως εκ τούτου, κρίνεται απαραίτητο ένα μοντέλο πρόβλεψης που να μπορούν να χρησιμοποιούν τα αεροσκάφη και οι αεροπορικές εταιρίες για την πρόβλεψη πιθανών καθυστερήσεων. Από αυτή την άποψη και λαμβάνοντας υπόψη τη δυσκολία της φύσης του προβλήματος αυτού, τα Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ) μπορούν να είναι επωφελή για την ανάπτυξη μιας τέτοιας εφαρμογής, καθώς αυτά αποτελούν ένα πολύ πρακτικό τρόπο επίλυσης μη-γραμμικών προβλημάτων. Επίσης, λόγω της εποπτευόμενης ικανότητας μάθησής τους, μπορούν εύκολα να προσαρμοστούν στη δυναμική της μεταφορικής ικανότητας και της ζήτησης στην εναέρια κυκλοφορία. Έτσι, με την ανάπτυξη μιας τέτοιας εφαρμογής οι αεροπορικές εταιρίες καθώς και οι υπηρεσίες του αεροδρομίου θα μπορούν να προβλέπουν άμεσα πιθανές καθυστερήσεις και να κάνουν τον αντίστοιχο προγραμματισμό τους. Μέσω της εργασίας αυτής, θέλουμε να δούμε κατά πόσο μπορεί να προβλεφθεί σε πραγματικό χρόνο ο μέσος χρόνος σε ημέρες που απαιτείται για να ομαλοποιηθεί η κατάσταση και να επανέλθει ο μέσος χρόνος καθυστέρησης πτήσεων σε χρόνους μικρότερους των 15

λεπτών ανά ημέρα μετά από κάποιο γεγονός που ανέβασε τον μέσο χρόνο καθυστέρησης των πτήσεων ανά ημέρα σε χρόνους άνω του προβλεπόμενου ορίου. Το προβλεπόμενο όριο των 15 λεπτών αναλύεται παρακάτω. Με αυτόν τον τρόπο θα είναι δυνατόν η διεύθυνση του αεροδρομίου να ενεργήσει κατάλληλα για την αντίστοιχη κατανομή πόρων για την απρόσκοπτη λειτουργία του αεροδρομίου. Αρχικά, επιλέχθηκε ένα μόνο αεροδρόμιο για πιλοτική εφαρμογή του υπολογιστικού πλαισίου. Η διαδικασία επιλογής του αεροδρομίου συζητείται στην επόμενη ενότητα.

3.2 Ανάκτηση Δεδομένων και Επεξεργασία

Για να καταστεί δυνατή η μελέτη του συγκεκριμένου προβλήματος των αερομεταφορών απαιτείται ένα μεγάλο σύνολο από συνεχή και συνεπή δεδομένα. Οπότε για την εύρεση αυτών ανατρέξαμε στην ιστοσελίδα του Υπουργείου Μεταφορών των Ηνωμένων Πολιτειών της Αμερικής (http://www.rita.dot.gov). Έτσι είχαμε στην διάθεση μας 22 αρχεία, ένα για κάθε χρονιά από το 1987 έως το 2008, τα οποία περιείχαν πληροφορίες για όλα τα αεροδρόμια και όλες τις πτήσεις των ΗΠΑ. Το κάθε αρχείο, αποτελείτο από 29 στήλες και ο αριθμός των γραμμών ήταν ανάλογος του ετήσιου συνολικού αριθμού πτήσεων (κατά μέσο όρο θα πούμε ότι ήταν 5×10^6). Κάθε μια από τις 29 στήλες περιλάμβανε μια μεταβλητή. Οι μεταβλητές αυτές παρουσιάζονται στον Πίνακα 3.1 που ακολουθεί.

Για να διαχειριστούμε τα δεδομένα αυτά, έγινε χρήση μιας ψευδο-κατανεμημένης συστοιχίας Hadoop και της γλώσσας προγραμματισμού Hive που σχολιάσαμε διεξοδικά στο Κεφάλαιο 2. Διότι τα δεδομένα αυτά ανήκουν στην κατηγορία των «Μεγάλων Δεδομένων» και δεν είναι δυνατόν να επεξεργαστούν και να αποθηκευτούν με τις παραδοσιακές πλατφόρμες, π.χ. Microsoft Excel. Αφού κατεβάσαμε όλα τα αρχεία με τα δεδομένα που χρειαζόμασταν, με την βοήθεια του Hadoop έγινε εισαγωγή των αρχείων ένα προς ένα στο warehouse για να μπορεί να τα «βλέπει» το Hive και να μπορούμε να τα φορτώσουμε σε πίνακες σε αυτό.

Οι εντολές που χρησιμοποιήσαμε για να το κάνουμε αυτό (δίδεται ως παράδειγμα το αρχείο της χρονιάς 2008) είναι:

```
hadoop dfs -put /home/hduser/Downloads/2008.csv /user/hive/warehouse
```

```
CREATE TABLE IF NOT EXISTS atable (year INT, month INT, dayofmonth INT, dayofweek INT, deptime INT, crsdeptime INT, arrtime INT, crsarrtime INT, unqcarrier STRING, flightnum STRING, tailnum STRING, actualelapsedtime INT, crselapsedtime INT, airtime INT, arrdelay INT, depdelay INT, origin STRING, dest STRING, distance INT, taxiln INT, taxiout INT, cancelled INT, cancellationCode STRING, diverted INT, carrierdelay INT, weatherdealy INT, nasdelay INT,
```

```
securitydealy INT, lateaircraftdelay INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY',';
```

```
LOAD DATA INPATH '/user/hive/warehouse/2008.csv' OVERWRITE INTO TABLE atable;
```

Όπου την πρώτη εντολή την χρησιμοποιήσαμε στο Hadoop, ενώ την δεύτερη και την τρίτη στο Hive.

	Όνομα	Περιγραφή
1	Year	1987-2008
2	Month	1-12
3	DayofMonth	1-31
4	DayofWeek	1(Monday)-7(Sunday)
5	DepTime	actual departure time (local, hhmm)
6	CRSDepTime	scheduled departure time (local, hhmm)
7	ArrTime	actual arrival time (local, hhmm)
8	CRSArrTime	scheduled arrival time (local, hhmm)
9	UniqueCarrier	unique carrier code
10	FlightNum	flight number
11	TailNum	plain tail number
12	ActualElapsedTime	in minutes
13	CRSElapsedTime	in minutes
14	AirTime	in minutes
15	ArrDelay	arrival delay, in minutes
16	DepDelay	departure delay, in minutes
17	Origin	origin
18	Dest	destination
19	Distance	in miles
20	TaxiIn	taxi in time, in minutes
21	TaxiOut	taxi out time, in minutes

22	Cancelled	was the flight cancelled?
23	CancellationCode	reason for cancellation (A=carrier, B=weather, C=NAS, D=security)
24	Diverted	1=yes, 0=no
25	CarrierDelay	in minutes
26	WeatherDelay	in minutes
27	NASDelay	in minutes
28	SecuriyDelay	in minutes
29	LateAircraftDelay	in minutes

Πίνακας 3.1: Οι 29 μεταβλητές που περιείχε κάθε αρχείο του Υπουργείου Μεταφορών των Ηνωμένων Πολιτειών της Αμερικής που ανακτήσαμε από την ιστοσελίδα (http://www.rita.dot.gov).

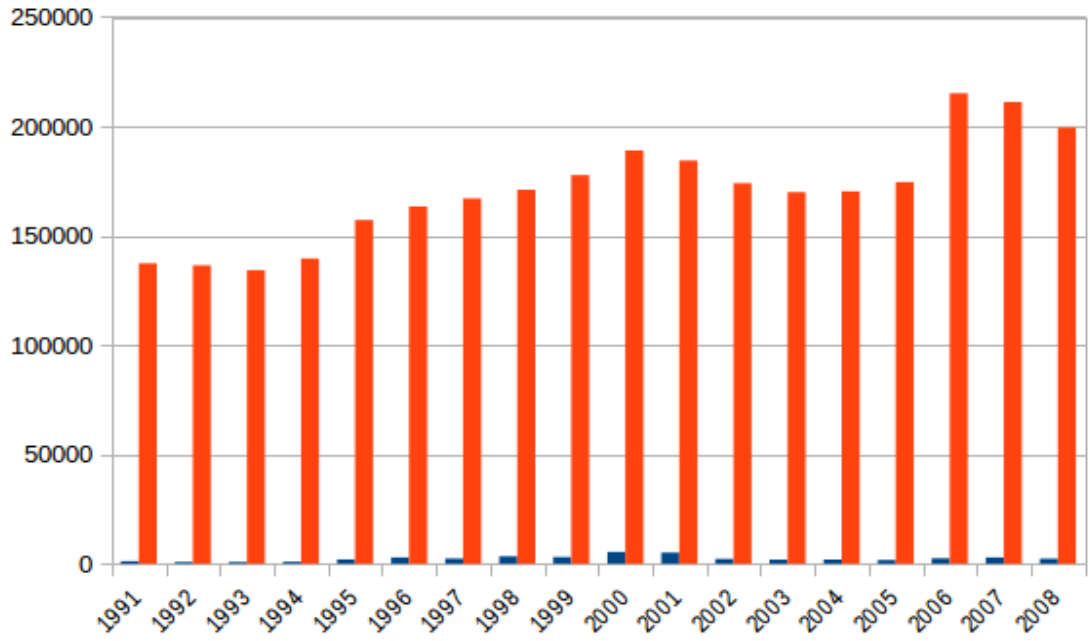
Αφού λοιπόν είχαμε στη διάθεσή μας όλα αυτά τα δεδομένα, αυτό που είχαμε να κάνουμε ακολούθως ήταν να επιλέξουμε τα κατάλληλα δεδομένα από το κατάλληλο αεροδρόμιο, για να τα αναλύσουμε με τη βοήθεια του ΤΝΔ και να αναδειχθούνε με αυτόν τον τρόπο οι δυνατότητες του υπολογιστικού πλαισίου που υλοποιήθηκε στα πλαίσια αυτής της εργασίας. Ως πλέον κατάλληλο αεροδρόμιο επιλέχθηκε (για λόγους που εξηγούνται στη συνέχεια) το Διεθνές Αεροδρόμιο «Phoenix Sky Harbor». Και ως κατάλληλα δεδομένα για ανάλυση επιλέχθηκαν από τον Πίνακα 3.1 τα δεδομένα 15 και 16 καθυστέρησης τόσο κατά την άφιξη όσο και κατά την αναχώρηση πτήσεων .

Το «Phoenix Sky Harbor» είναι ένας πολιτικός-στρατιωτικός αερολιμένας, στην επαρχία Maricopa της Αριζόνας στις Ηνωμένες Πολιτείες της Αμερικής. Είναι το μεγαλύτερο και πιο πολυσύχναστο αεροδρόμιο της Αριζόνας και είναι ανάμεσα στα πιο πολυσύχναστα (με βάση τον συνολικό αριθμό επιβατών που καταχωρήθηκε σύμφωνα με τα στοιχεία που συνέταξε το Διεθνές Συμβούλιο Αεροδρομίων Βορείου Αμερικής κατά τη διάρκεια του 2014) αεροδρόμια των Ηνωμένων Πολιτειών της Αμερικής το 2014, κατέχοντας την $11^{η}$ θέση με αριθμό επιβατών 42.134.662 (Federal Aviation Administration, 2014). Το 2015 ο αερολιμένας εξυπηρέτησε 44.006.205 (Federal Aviation Administration, 2015) επιβάτες, καθιστώντας το το $29^{ο}$ πιο πολυσύχναστο αεροδρόμιο παγκοσμίως, με βάση τον αριθμό των επιβατών που το επισκέφτηκαν.

Οι λόγοι για τους οποίους αποφασίσαμε να επιλέξουμε τον αερολιμένα «Phoenix Sky Harbor» αντί για κάποιο άλλο αερολιμένα της ΗΠΑ αναλύονται πιο κάτω:

- **Μας πρόσφερε επαρκή και συνεχή δεδομένα**

Πλήρης σύνολα δεδομένων, τα οποία παρείχαν επαρκή δεδομένα για όλες τις ημέρες, κάθε χρόνου. Συγκρίνοντας τον συνολικό αριθμό πτήσεων για τις οποίες είχαμε δεδομένα με το συνολικό αριθμό πτήσεων οι οποίες δεν περιείχαν επαρκή δεδομένα, το ποσοστό του δεύτερου ως προς του πρώτου ήταν μικρότερο από 2%. Ποιοτικά, αυτό φαίνεται και στην πιο κάτω γραφική παράσταση (Σχήμα 3.1).

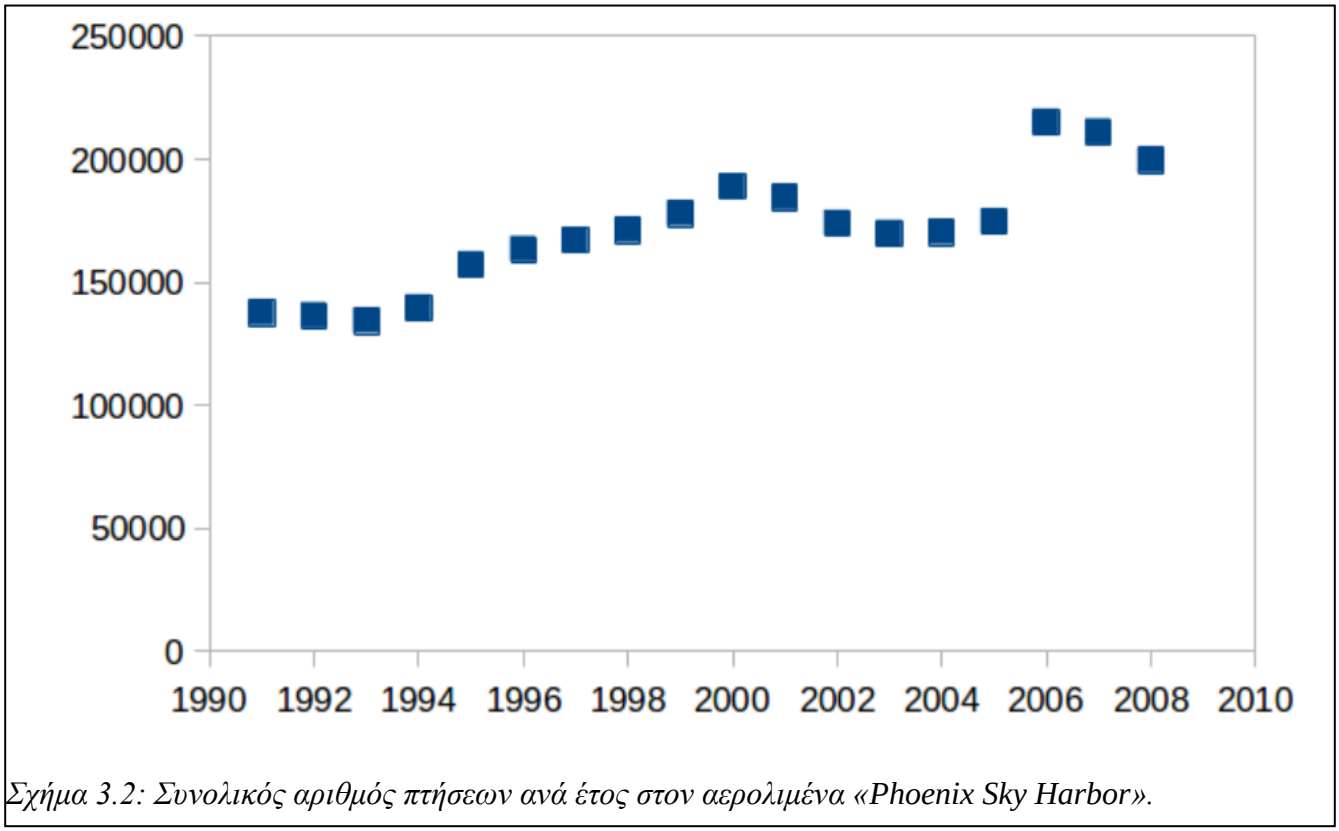


Σχήμα 3.1: Σύγκριση του συνολικού αριθμού πτήσεων ανά έτος για τις οποίες είχαμε πλήρη δεδομένα (πορτοκαλί), με το συνολικό αριθμό πτήσεων για τις οποίες δεν είχαμε επαρκή δεδομένα (μπλε).

- **Σταθερότητα στην δραστηριότητα του αερολιμένα**

Το «Phoenix Sky Harbor» κατασκευάσθηκε το 1928 έχοντας ένα μόνο διάδρομο απογείωσης-προσγείωσης. Κατά το πέρασμα των χρόνων έγιναν πολλές αλλαγές σχετικά με τον αριθμό των πυλών και των διαδρόμων του αερολιμένα αυτού. Το Νοέμβριο του 1990 έγινε η τελευταία προσθήκη διαδρόμου. Έχοντας λοιπόν την τελευταία αυτή προσθήκη το 1990, από τότε και μετά ο αριθμός των πτήσεων δεν έχει μεταβληθεί σημαντικά. Αυτό μπορεί κανείς να το δει από τα δεδομένα που βρίσκονται στην ιστοσελίδα του Υπουργείου Μεταφορών των Ηνωμένων Πολιτειών της Αμερικής (http://www.rita.dot.gov). Στο Σχήμα 3.2 δίδεται μόνο ο συνολικός αριθμός πτήσεων στον αερολιμένα ανά έτος για τα 18 έτη από το 1991 έως το 2008. Στατιστική ανάλυση των δεδομένων αυτών έδειξε ότι ο μέσος όρος των πτήσεων ανά έτος ήταν 170.613 και η τυπική απόκλιση 23.404, κάτω δηλαδή από

15% του μέσου όρου, πράγμα που σημαίνει ότι η τάξη μεγέθους στον αριθμό των πτήσεων ετησίως δεν έχει αλλάξει.. Επίσης, πρέπει να σημειωθεί ότι η δραστηριότητα του αεροδρομίου είναι σταθερή καθ'όλη τη διάρκεια του χρόνου, με το συνολικό αριθμό πτήσεων να μοιράζεται μεταξύ των διαστημάτων «Ιανουάριος-Ιούνιος» και «Ιούλιος-Δεκέμβριος».

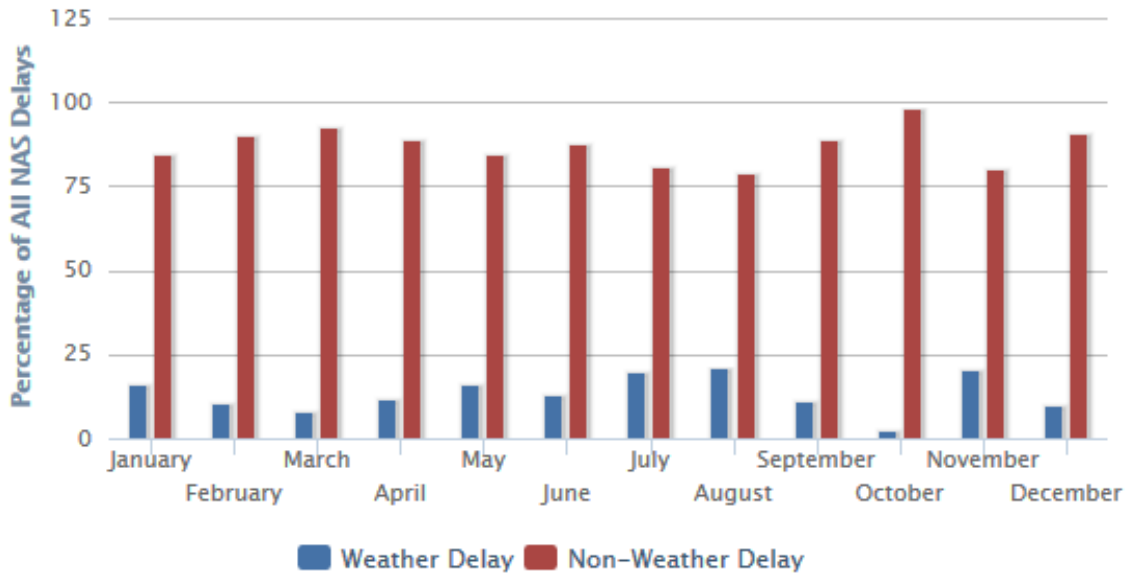


Σχήμα 3.2: Συνολικός αριθμός πτήσεων ανά έτος στον αερολιμένα «Phoenix Sky Harbor».

- **Λιγότερα επηρεαζόμενο από ακραία καιρικά φαινόμενα**

Σε πρώτη φάση θέλουμε να αποκλείσουμε καθυστερήσεις από ακραία καιρικά φαινόμενα, επειδή υπάρχουν πολλών διαφορετικών τύπων φαινόμενα με διαφορετικούς χρόνους ομαλοποίησης μετά το πέρας τους (πχ. μετά από μια σφοδρή καταιγίδα χρειάζεται πολύ λιγότερος χρόνος επανέναρξης λειτουργίας του αεροδρομίου απ'ότι μετά από μια σφοδρή χιονόπτωση). Αυτό σημαίνει ότι η καθυστέρηση λόγω καιρού θα πρέπει να συνοδεύεται και από επιπλέον πληροφορία (δηλαδή τι είδους φαινόμενο ήταν), ώστε να μπορεί το ΤΝΔ να προβλέψει τον σωστό χρόνο ομαλοποίησης. Επειδή λοιπόν θέλουμε να απφύγουμε κάτι τέτοιο, θέλαμε το υπό εξέταση αεροδρόμιο να είναι όσο το δυνατόν λιγότερο εκτεθειμένο σε καιρικά φαινόμενα. Λόγω της γεωγραφικής του θέσης, το

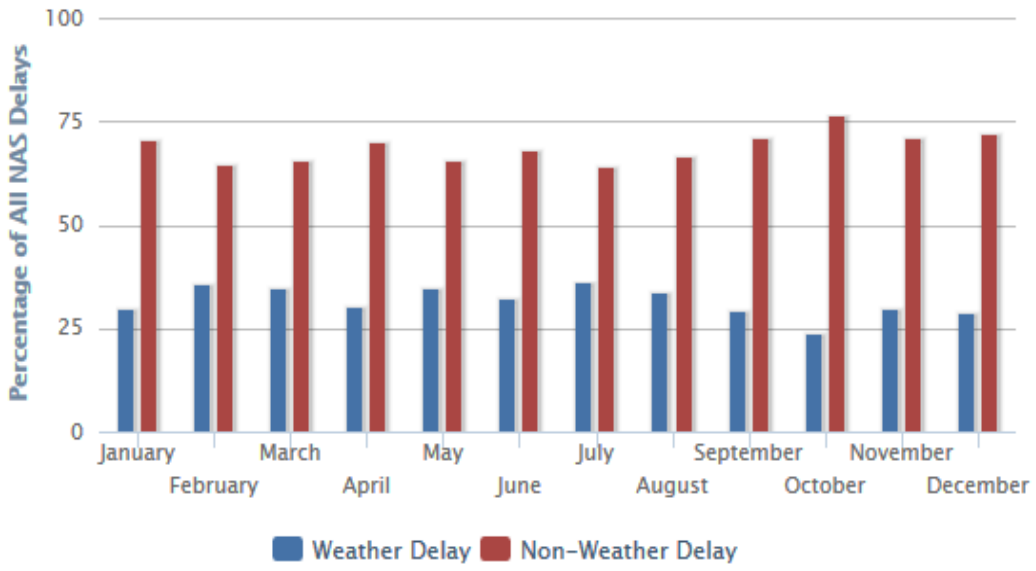
συγκεκριμένο αεροδρόμιο δεν επηρεάζεται συχνά από ακραία καιρικά φαινόμενα. Για τον λόγω αυτό οι αεροπορικές πτήσεις από και προς τον αερολιμένα αυτό δεν έχουν σημαντική καθυστέρηση που να οφείλεται σε κακές καιρικές συνθήκες. Ενδεικτικά αναφέρεται ένα παράδειγμα για την περίοδο Ιανουαρίου-Δεκεμβρίου το 20016, όπου για το αεροδρόμιο «Phoenix Sky Harbor», το ποσοστό καθυστέρησης πτήσεων λόγω καιρικών συνθηκών στο αεροδρόμιο είναι σημαντικά χαμηλότερο από τον αντίστοιχο εθνικό μέσο όρο καθυστέρησης λόγω καιρού (Σχήματα 3.3, 3.4).



Σχήμα 3.3: Ποσοστά καθυστερήσεων λόγω καιρού σε σχέση με άλλες αιτίες για το 2016 στο Διεθνές Αερολιμένα του «Phoenix Sky Harbor».

Πηγή: *https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp?pn=1*

\

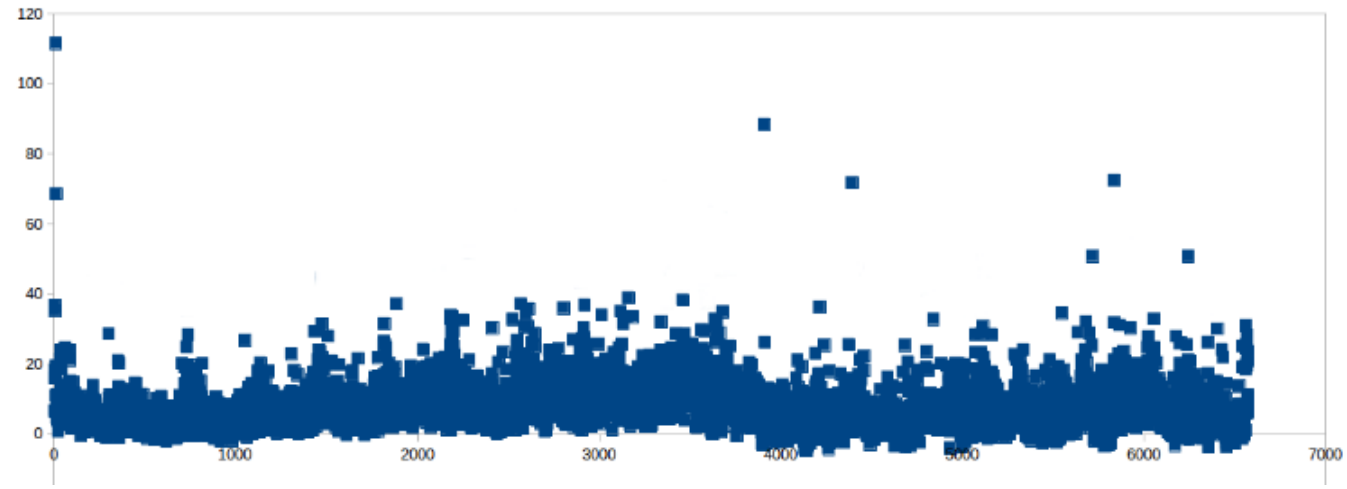


Σχήμα 3.4: Ποσοστά καθυστερήσεων λόγω καιρού σε σχέση με άλλες αιτίες για το 2016 για το σύνολο των αεροδρομίων των ΗΠΑ.

Πηγή: *https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp?pn=1*

Αφού έγινε η επιλογή του κατάλληλου αεροδρομίου, για τους λόγους που είπαμε πιο πάνω και αφού αφαιρέσαμε από τα δεδομένα μας όλες τις πτήσεις για τις οποίες δεν υπήρχαν επαρκή δεδομένα, ακολούθως, επιλέξαμε την περίοδο Ιανουάριος 1991-Δεκέμβριος 2008 να επεξεργαστούμε επειδή, όπως έχουμε προαναφέρει, το Νοέμβριο του 1990 έγινε η τελευταία προσθήκη διάδρομου. Από τότε και μετά ο αριθμός των πτήσεων δεν άλλαξε σημαντικά και αυτό μας πρόσφερε σταθερότητα στα δεδομένα μας. Η σταθερότητα αυτή θα βοηθήσει στην πιο συνεπή ανάλυση των δεδομένων. Οι παράμετροι εισόδου στο ΤΝΔ αποφασίστηκε να σχετίζονται με τα δεδομένα καθυστέρησης των πτήσεων, τα οποία ήταν άμεσα διαθέσιμα, έτσι ώστε να προκύψει ένας μέσος χρόνος καθυστέρησης συνολικά ανά ημέρα. Η διαδικασία εύρεσης αυτού του μέσου χρόνου περιγράφεται πιο κάτω. Στο Σχήμα 3.5 δίδονται οι καθυστερήσεις (σε λεπτά) ανά ημέρα (μέση τιμή) για όλο το υπό εξέταση διάστημα. Από την εξέταση των μέσων αυτών χρόνων είναι προφανές ότι οι χρόνοι καθυστέρησης για κάποιες ημέρες ήταν υπερβολικά μεγάλοι, πράγμα που σημαίνει ότι κάποιο ακραίο γεγονός ίσως να είχε συμβεί εκείνη την περίοδο. Σκοπός της εργασίας αυτής στην παρούσα φάση είναι να αποκλείσει ακραία γεγονότα από την ανάλυση και να τα επεξεργαστεί με διαφορετικό τρόπο στο μέλλον μέσω του ΤΝΔ. Σε αυτό το σημείο είναι σημαντικό να διευκρινίσουμε την έννοια της καθυστέρησης σε μια πτήση σύμφωνα με το Υπουργείου Μεταφορών των Ηνωμένων Πολιτειών της Αμερικής (http://www.rita.dot.gov). Μια πτήση λογίζεται ως «on time», εάν ανταποκρίνεται σε χρονικό διάστημα λιγότερο από 15 λεπτά (θα ονομάσουμε το διάστημα αυτό των 15 λεπτών T_{cr}) μετά την προγραμματισμένη ώρα που εμφανίζεται στα συστήματα ηλεκτρονικών κρατήσεων των αερομεταφορέων. Η απόδοση των πτήσεων κατά την άφιξη βασίζεται στην ώρα άφιξης στην πύλη. Η

απόδοση των πτήσεων κατά την αναχώρηση βασίζεται στην ώρα αναχώρησης από την πύλη. Αναζητώντας στο διαδίκτυο τις συγκεκριμένες ημέρες για τις οποίες οι αεροπορικές πτήσεις είχαν πολύ μεγάλους χρόνους καθυστέρησης, βρήκαμε ότι σχετίζονταν με κάποια ακραία περιστατικά.



Σχήμα 3.5: Ο μέσος χρόνος καθυστέρησης (άξονας y, σε λεπτά) ανά ημέρα (άξονας x), για όλο το υπό εξέταση διάστημα από το 1991 έως το 2008.

Τις ημέρες αυτές λοιπόν αποφασίσαμε να τις αφαιρέσουμε από τα δεδομένα που θα χρησιμοποιούσαμε στη συνέχεια γιατί θα χαλούσαν την συνοχή των δεδομένων μας και θα δυσκόλευαν την διαδικασία της εκπαίδευσης του δικτύου. Οι ημέρες αυτές ήταν οι ακόλουθες:

- Την 9η Ιανουαρίου το 1991, με μέση καθυστέρηση 111 λεπτά και την 10η Ιανουαρίου το 1991, με μέση καθυστέρηση 69 λεπτά Η καθυστέρηση αυτή κατά πάσα πιθανότητα οφείλεται στο γεγονός ότι το «Phoenix Sky Harbor» είναι εκτός από πολιτικός και στρατιωτικός αερολιμένας και αφού εκείνη την περίοδο ήταν η κλιμάκωση του πρώτου πολέμου του Κόλπου, υπάρχει περίπτωση να επηρεάστηκαν οι αεροπορικές μεταφορές από και πρός τον αερολιμένα αυτό. Ο πρώτος πόλεμος του Κόλπου ξεκίνησε στις 2 Αυγούστου 1990 και έληξε στις 28 Φεβρουαρίου 1991 και ήταν μια πολεμική σύρραξη μεταξύ διεθνούς συμμαχίας από τουλάχιστον 31 κράτη υπό την καθοδήγηση των Η.Π.Α.και την εξουσιοδότηση του Ο.Η.Ε. κατά του Ιράκ, για την απελευθέρωση του Κουβέιτ. (*Barzilai G. et al., 1993)*

- Την 11η Σεπτεμβρίου 2001, δεν υπήρξαν καθόλου πτήσεις από και προς το αεροδρόμιο, την 12η όμως Σεπτεμβρίου είχαμε μέση καθυστέρηση 88 λεπτά. Η καθυστέρηση αυτή είναι συνέπεια του πανικού που έσπειρε η τρομοκρατική επίθεση που έγινε την 11η Σεπτεμβρίου στην Νέα Υόρκη. Με αποτέλεσμα, πολλές πτήσεις να ακυρωθούν, ενώ άλλες να πραγματοποιηθούν μετά από σημαντική καθυστέρηση.

- Την 9η Ιανουαρίου το 2003, είχαμε μέση καθυστέρηση πάνω από 70 λεπτά. Η καθυστέρηση αυτή πιστεύουμε ότι οφείλεται στο γεγονός ότι εκείνη την ημέρα υπήρξε το αεροπορικό δυστύχημα της πτήσης US Airways Express 5.481, η οποία πραγματοποιούσε πτήση από το διεθνές αεροδρόμιο Charlotte / Douglas της Charlotte της Βόρειας Καρολίνας στις Ηνωμένες Πολιτείες στο Διεθνές Αεροδρόμιο Greenville-Spartanburg.

- Την 28η Δεκεμβρίου το 2005 και την 25η Αυγούστου το 2006, είχαμε μέση καθυστέρηση πάνω από 50 και 70 λεπτά, αντίστοιχα. Για τις ημέρες αυτές δεν βρήκαμε κάτι συγκεκριμένο, στο οποίο να μπορούμε να αποδώσουμε ευθύνη, οπότε υποθέτουμε ότι οφείλονται στο γεγονός ότι όπως είπαμε ξανά ο αερολιμένας «Phoenix Sky Harbor» είναι και στρατιωτικός αερολιμένας. Τα δεδομένα για στρατιωτικές επιχειρήσεις είναι άκρως απόρρητα, οπότε δεν έχουμε τη δυνατότητα να βρούμε τα απαραίτητα αποδεικτικά στοιχεία για αυτή μας την υπόθεση. Ωστόσο, τα δεδομένα για αυτές τις δύο ημέρες τα αφαιρέσαμε από τα τελικά μας δεδομένα.

- Την 3η Φεβρουαρίου το 2008, είχαμε μέση καθυστέρηση πάνω από 50 λεπτά. Η καθυστέρηση αυτή κατά πάσα πιθανότητα οφείλεται στο ότι εκείνη την ημέρα πραγματοποιούταν το Super Bowl XLII στο Phoenix Stadium στο Glendale της Αριζόνα. Το Super Bowl XLII (το οποίο αποτελεί κορυφαίο αθλητικό γεγονός των ΗΠΑ) είναι ένας αγώνας αμερικάνικου ποδοσφαίρου μεταξύ του πρωταθλητή της Εθνικής Ποδοσφαιρικής Ομοσπονδίας και του πρωταθλητή της Αμερικανικής Ποδοσφαιρικής Ομοσπονδίας για να κρίνει τον πρωταθλητή του Εθνικού ποδοσφαιρικού πρωταθλήματος

Στο σημείο αυτό πρέπει να αναφέρουμε τον τρόπο με τον οποίο εξάγαμε το μέσο χρόνο καθυστέρησης ανά ημέρα.:

I. Για κάθε χρονιά ανακτήσαμε τις εξής πληροφορίες για κάθε ημέρα: «Year, Month, DayofMonth, DayofWeek, ArrDelay, DepDelay, Origin». Στη συνέχεια, η πληροφορία αυτή αποθηκεύτηκε σε ένα αρχείο του οποίου κάθε γραμμή περιείχε τις ακόλουθες πληροφορίες: ημέρα του χρόνου (αριθμημένη από 1 για 1η Ιανουραρίου έως 365 ή 366 για δίσεκτο έτος για 31η Δεκεμβρίου), καθυστέρηση κατά την άφιξη και καθυστέρηση κατά την αναχώρηση σε λεπτά για κάθε αεροσκάφος, για όλες τις πτήσης της αντίστοιχης ημέρας (το αρχείο αυτό για κάθε έτος είχε κατά μέσο όρο 85,000 γραμμές). Όπως παρατηρήσαμε, για ένα αεροσκάφος η τιμή της καθυστέρησης κατά την άφιξη και η τιμή της καθυστέρησης κατά την αναχώρησή του

ήταν περίπου ίδιες. Ο λόγος για τον οποίο συμβαίνει αυτό είναι επειδή οι τιμές αυτές αναφέρονται σε διαδοχικές πτήσεις του ίδιου αεροσκάφους, οπότε όση καθυστέρηση είχε κατά την άφιξη, είναι αναμενόμενο να έχει περίπου την ίδια καθυστέρηση και κατά την αναχώρηση. Εάν ο χρόνος καθυστέρησης ήταν αρνητικός (πράγμα που δηλώνει νωρίτερη άφιξη ή αναχώρηση πτήσης), του αποδίδαμε την τιμή 0, θεωρώντας ότι δεν υπήρχε καθυστέρηση εκείνη την ημέρα. Επίσης, εάν το αεροσκάφος είχε μόνο αναχώρηση ή μόνο άφιξη, το πεδίο της διπλανής καταχώρησης (χρόνος καθυστέρηση σε άφιξη ή σε αναχώρηση, αντίστοιχα) παρέμενε κενό και λογιζόταν μόνο μία πτήση για το συγκεκριμένο αεροσκάφος τη συγκεκριμένη ημέρα)

II. Υπολογίσαμε για κάθε ημέρα τον συνολικό αριθμό πτήσεων, τον συνολικό αριθμό αφίξεων, τον συνολικό αριθμό αναχωρήσεων, το άθροισμα του χρόνου καθυστέρησης κατά την άφιξη και το άθροισμα του χρόνου καθυστέρησης κατά την αναχώρηση.

III. Υπολογίσαμε το μέσο χρόνο καθυστέρησης κατά την άφιξη (άθροισμα του χρόνου καθυστέρησης κατά την άφιξη / το συνολικό αριθμό αφίξεων) και το μέσο χρόνο καθυστέρησης κατά την αναχώρηση (άθροισμα του χρόνου καθυστέρησης κατά την αναχώρηση / το συνολικό αριθμό πτήσεων αναχωρήσεων).

IV. Υπολογίσαμε επίσης το συνολικό μέσο χρόνο καθυστέρησης ανά ημέρα ((άθροισμα του χρόνου καθυστέρησης κατά την αναχώρηση + άθροισμα του χρόνου καθυστέρησης κατά την άφιξη) / το συνολικό αριθμό των πτήσεων εκείνης της ημέρας).

V. Για να μπορέσει να προκύψει το Σχήμα 3.5, ομαδοποιήσαμε σε ένα αρχείο όλες τις πληροφορίες για κάθε ημέρα για αυτά τα 18 υπό εξέταση έτη, το οποίο είχε συνολικά 6,574 γραμμές (όπου η γραμμή 1 αντιστοιχεί στην $1^{η}$ Ιανουαρίου 1991 και η γραμμή 6,574 αντιστοιχεί στην $31^{η}$ Δεκεμβρίου 2008). Κάθε γραμμή περιέχει τη μέση συνολική καθυστέρηση για εκείνη την ημέρα, το μέσο χρόνο καθυστέρησης άφιξης πτήσεων και το μέσο χρόνο καθυστέρησης αναχώρησης πτήσεων. Όλοι οι χρόνοι δίδονται σε λεπτά. Αξίζει να σημειωθεί ότι ο μέσος όρος των δύο χρόνων καθυστέρησης κατά την αναχώρηση και την άφιξη, διαφέρει κατά $1^{ο}$ δεκαδικό ψηφίο από το μέσο χρόνο καθυστέρησης που υπολογίστηκε όπως περιγράφεται στο (III).

VI. Στη συνέχεια, για να εξηγήσουμε τη μορφοποίηση των δεδομένων για την εισαγωγή τους στο ΤΝΔ, ονομάζουμε T_{del} το μέσο χρόνο καθυστέρησης πτήσης ανά ημέρα. Έτσι, καταρχήν ομαδοποιούμε τις διαδοχικές ημέρες με $T_{del} > T_{cr}$. Ενδέχεται να έχουμε και μόνο μια μεμονωμένη ημέρα, εάν την αμέσως επόμενη $T_{del} < T_{cr}$. Θεωρούμε ως ημέρα εμφάνισης καθυστέρησης την πρώτη από τις διαδοχικές αυτές ημέρες και της αποδίδουμε έναν ακέραιο δείκτη I_{del}, ίσο με τον αριθμό των ημερών που απαιτήθηκε ώστε το T_{del} να γίνει μικρότερο του T_{cr}. Για παράδειγμα, εάν για την αμέσως επόμενη ημέρα $T_{del} < T_{cr}$, τότε $I_{del} = 1$, εάν αυτό συμβεί για τη μεθεπόμενη, τότε $I_{del} = 2$, κοκ. Ο δείκτης I_{del} υποδηλώνει τον αριθμό των ημερών

που απαιτήθηκαν για ομαλοποίηση της κατάστασης καθυστέρησης και είναι πάντοτε μεγαλύτερος του μηδενός.

VII. Καταυτόν τον τρόπο δημιουργήθηκε ένα αρχείο το οποίο σε κάθε γραμμή περιείχε για την ημέρα κατά την οποία πρωτοεμφανίζεται η καθυστέρηση, το συνολικό μέσο χρόνο καθυστέρησης ανά πτήση, το μέσο χρόνο καθυστέρησης κατά την άφιξη, το μέσο χρόνο καθυστέρησης κατά την αναχώρηση και το δείκτη I_{del} (αριθμό ημερών για ομαλοποίηση). Από το αρχείο αυτό αφαιρέθηκαν οι καθυστερήσεις που οφείλονταν σε ακραία γεγονότα, όπως περιγράφηκε πιο πάνω. Έτσι, συνολικά απέμειναν 419 γεγονότα καθυστερήσεων από τα οποία έπρεπε να επιλεγούν τα δεδομένα εισόδου στο ΤΝΔ. Σε αυτά τα δεδομένα παρατηρήσαμε ότι ο χρόνος μέσης καθυστέρησης ανά πτήση έφτανε μέχρι 93 λεπτά και ο χρόνος ομαλοποίησης κυμαινόταν από 1 έως 6 ημέρες. Αξίζει να σημειωθεί ότι ο μέγιστος χρόνος ομαλοποίησης των 6 ημερών δεν αντιστοιχούσαν στην μέγιστη καθυστέρηση! Μια στατιστική ανάλυση των δεδομένων για τους μέσους χρόνους καθυστέρησης κατά την αναχώρηση και την άφιξη καθώς και για τον αριθμό των ημερών που χρειάστηκε για ομαλοποίηση, έδειξε ότι:

- Η πλειονότητα των μέσων χρόνων καθυστέρησης ανά πτήση κατά την άφιξη ή την αναχώρηση ήταν κάτω των 30 λεπτών (83% για αφίξεις και 86% για τις αναχωρήσεις), από μισή μέχρι μία ώρα μέσο χρόνο καθυστέρησης είχαν το 15% (αφίξεις) και το 12% (αναχωρήσεις) και μεγαλύτερη της μιας ώρας μέσο χρόνο καθυστέρησης είχαν το 3% (αφίξεις) και 3% (αναχωρήσεις).
- Οι περισσότερες καθυστερήσεις (περίπου το 68%) ομαλοποιήθηκαν εντός της επόμενης ημέρας (267 γεγονότα καθυστέρησης), 81 γεγονότα ομαλοποιήθηκαν εντός 2 ημερών 35 γεγονότα ομαλοποιήθηκαν εντός 3 ημερών, 12 ομαλοποιήθηκαν εντός 4 ημερών, 3 ομαλοποιήθηκαν εντός 5 ημερών και 1 ομαλοποιήθηκε εντός 6 ημερών.

VIII. Τα δεδομένα αυτά κανονικοποιήθηκαν μεταξύ 0 και 1 (επιλέγοντας για κάθε μια στήλη το μέγιστο και το ελάχιστό της) εφαρμόζοντας την εξίσωση (3.1):

$$\frac{(x-x_{min})}{(x_{max}-x_{min})} \qquad \text{(εξίσωση 3.1)}$$

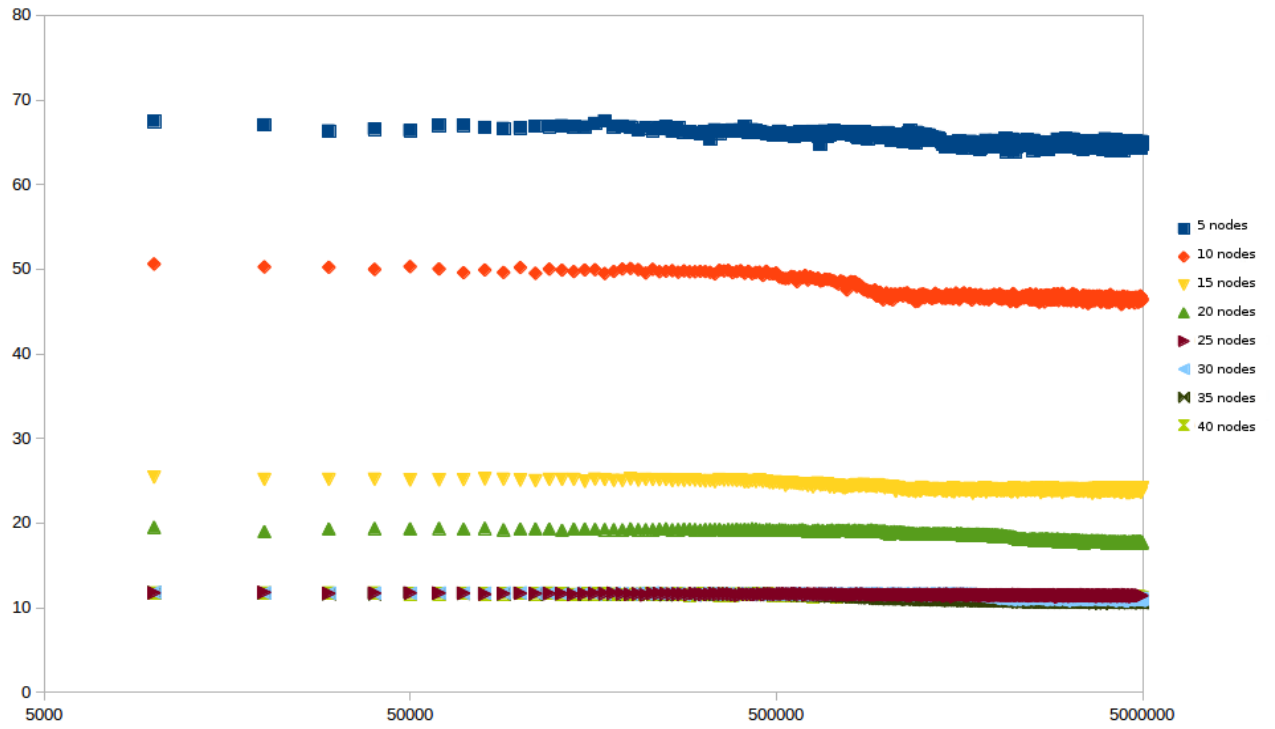
3.3 Εκπαίδευση και ανάλυση αποτελεσμάτων του μοντέλου ΤΝΔ για το πρόβλημα της πρόβλεψης καθυστερήσεων στις αερομεταφορές

Έγιναν πολλές διαφορετικές δοκιμές για δεδομένα εισόδου-εξόδου στο ΤΝΔ. Ενδεικτικά, αναφέρουμε: ένα δεδομένο στην είσοδο – ένα στην έξοδο (πχ. μέσος χρόνος καθυστέρησης - I_{del}), δύο δεδομένα εισόδου – ένα εξόδου (πχ. μέσοι χρόνοι καθυστέρησης άφιξης και αναχώρησης - I_{del}). Ως δεδομένο εξόδου δοκιμάστηκε και η πιθανότητα η καθυστέρηση να επεκταθεί σε περισσότερες της μιας ημέρες. Η διαμόρφωση που φάνηκε να ελαχιστοποιεί γρηγορότερα το σφάλμα ήταν αυτή με δεδομένα εισόδου τους μέσους χρόνους καθυστέρησης κατά την άφιξη και την αναχώρηση και δεδομένο εξόδου το I_{del} και αυτή επιλέχθηκε τελικά.

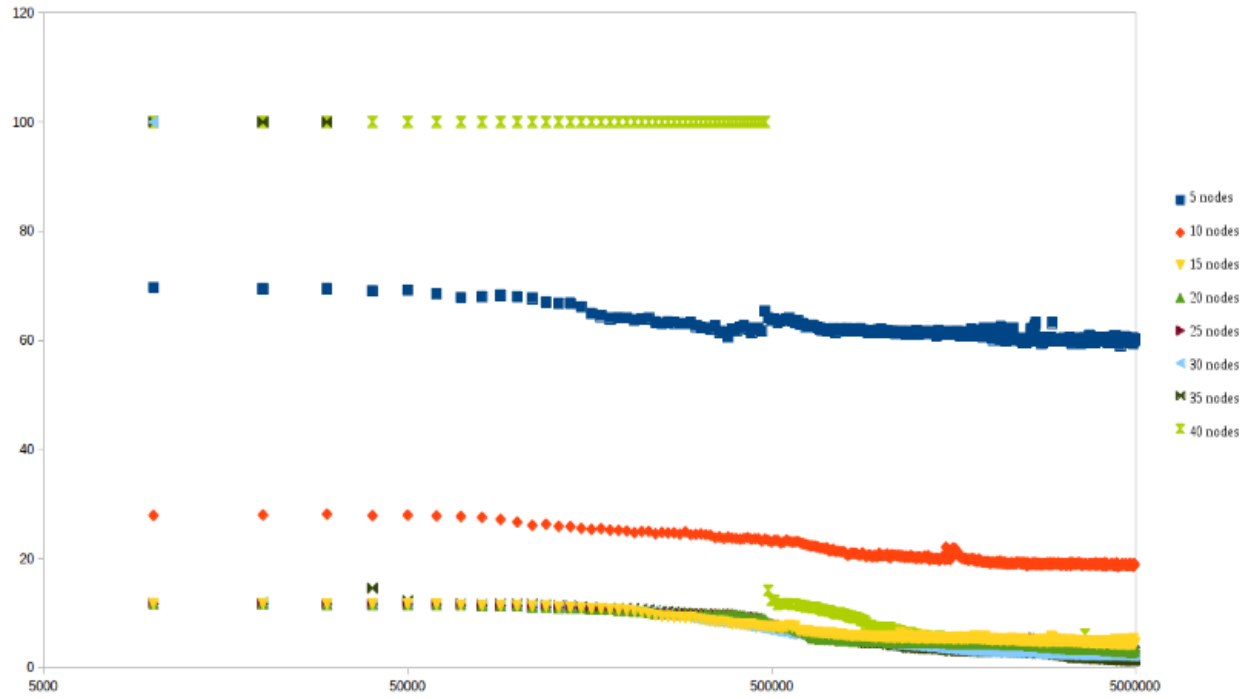
Βάσει των αρχικών δεδομένων, έπρεπε να γίνει επιλογή των πόσων δεδομένων θα κρατηθούν για εκπαίδευση και πόσων για επιβεβαίωση του προβλήματος. Λόγω του περιορισμένου αριθμού των δεδομένων (419), με εφαρμογή της διαισθητικής ανάλυσης κατά Hush (1989) (βλέπε Πίνακα 2.3) ο αριθμός των μοτίβων που απαιτείται για την καλή εκπαίδευση του δικτύου είναι κατ'ελάχιστον 180 και το βέλτιστο 360. Έτσι, αποφασίστηκε να επιλέξουμε το 90% για εκπαίδευση (378 μοτίβα) και το 10% (41 μοτίβα) για επιβεβαίωση. Η επιλογή του 10% των μοτίβων για επιβεβαίωση έγινε με τυχαίο τρόπο (Park and Miller, 1988).

Στο σημείο αυτό, είναι σημαντικό να αναφερθεί ότι στην παρούσα εργασία, βασιζόμενοι σε πρότερη εμπειρία (Christakis et al, 2011), διατηρήσαμε τις τιμές της ταχύτητας εκμάθησης και του ρυθμού εκμάθησης του ΤΝΔ σταθερές και ίσες με 0.2. Επίσης, ως κριτήριο τερματισμού της εκπαίδευσης θέσαμε είτε το σφάλμα κατά την εκπαίδευση να πέσει στο 0.1% του αρχικού ή να συμπληρωθούν 50,000,000 εποχές.

Στη συνέχεια, για να βρεθεί ο βέλτιστος συνδυασμός κρυμμένων επιπέδων-αριθμού νευρώνων σε κάθε κρυμμένο επίπεδο, έγιναν δοκιμές για 5,000,000 εποχές (ικανός αριθμός για να καταλάβουμε την πορεία που θα έχει η διακύμανση του σφάλματος) για 1 και 2 κρυμμένα επίπεδα και 5, 10, 15, 20, 25, 30, 35, 40 νευρώνες σε κάθε κρυμμένο επίπεδο κάθε φορά. Τα αποτελέσματα αυτών των προσομοιώσεων εμφανίζονται στο Σχήμα 3.6.



Σχήμα 3.6 (α) : Διακύμανση σφάλματος (%) σε σχέση με τις εποχές κατά την εκπαίδευση ΤΝΔ με 2 δεδομένα εισόδου και 1 εξόδου, για 1 κρυμμένο επίπεδο και 5, 10, 15, 20, 25, 30, 35, 40 νευρώνες στο κρυμμένο επίπεδο.



Σχήμα 3.6 (β) : Διακύμανση σφάλματος (%) σε σχέση με τις εποχές κατά την εκπαίδευση ΤΝΔ με 2 δεδομένα εισόδου και 1 εξόδου, για 2 κρυμμένα επίπεδα και 5, 10, 15, 20, 25, 30, 35, 40 νευρώνες σε κάθε κρυμμένο επίπεδο.

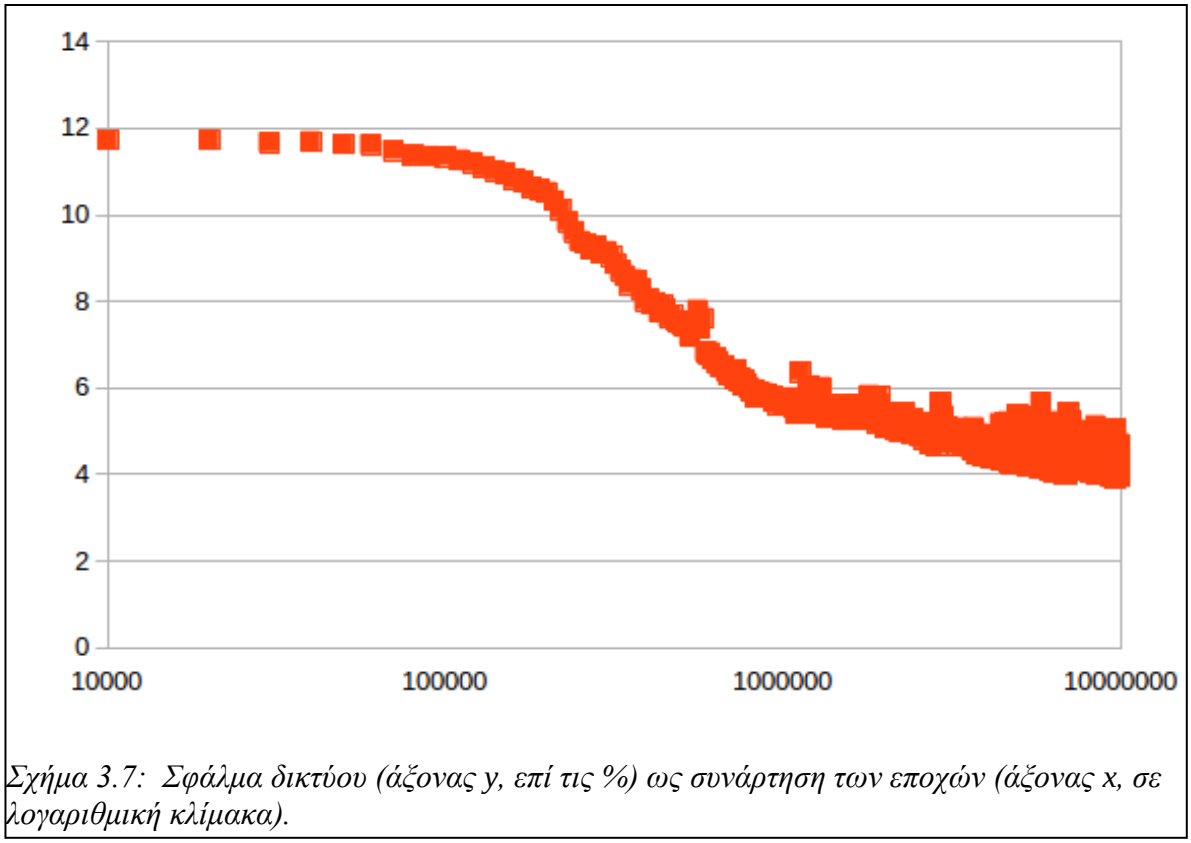
Παρατηρούμε ότι το σφάλμα μετά από 1,000,000 περίπου εποχές σταθεροποιείται (με μικρές ταλαντώσεις) γύρω από μία τιμή. Γίνεται προφανές ότι για 1 κρυμμένο επίπεδο, το σφάλμα δεν πέφτει ποτέ κάτω από 10%, οπότε το 1 κρυμμένο επίπεδο απορρίπτεται. Για 2 κρυμμένα επίπεδα, παρατηρούμε ότι για 15 νευρώνες και πάνω, το σφάλμα πέφτει σημαντικά κάτω από 10% (χωρίς βέβαια ποτέ να προσεγγίσει το 0.1%, που είναι και το κριτήριο σύγκλισης). Για να αποφασιστεί η βέλτιστη διαμόρφωση (μικρότερο δυνατό σφάλμα στο λιγότερο δυνατό χρόνο), ορίσαμε ως δείκτη απόδοσης του ΤΝΔ I_{eff} το γινόμενο σφάλμα στις 5,000,000 εποχές επί το χρόνο ολοκλήρωσής τους (σε ώρες) σε έναν επεξεργαστή. Ως βέλτιστη διαμόρφωση θα επιλέξουμε αυτή που έχει το μικρότερο I_{eff}. Τα αποτελέσματα δίδονται στον Πίνακα 3.2.

Αριθμός νευρώνων ανά επίπεδο	Σφάλμα στις 5,000,000 εποχές	Χρόνος ολοκλήρωσης 5,000,000 σε ένα επεξεργαστή (σε ώρες)	I_{eff}
15	0.044	13.8	0.61
20	0.045	27.6	1.24
25	0.029	41.6	1.20
30	0.022	55.2	1.21
35	0.013	82.8	1.10
40	0.016	96.6	1.57

Πίνακας 3.2: Δείκτης απόδοσης ΤΝΔ με 2 κρυμμένα επίπεδα και για 15, 20, 25, 30, 35, 40 νευρώνες σε κάθε επίπεδο.

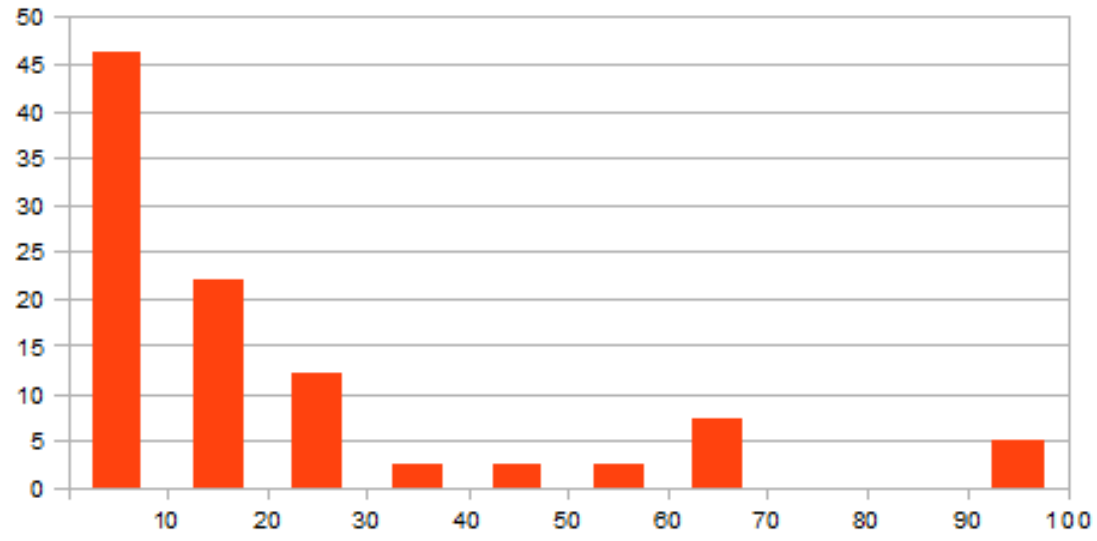
Όπως φαίνεται από τα αποτελέσματα, δείκτης I_{eff} έχει τη μικρότερη τιμή του για 15 νευρώνες σε κάθε επίπεδο κι αυτό γιατί όσο αυξάνεται ο αριθμός των νευρώνων, τόσο αυξάνεται και ο αριθμός των συναπτικών βαρών συνολικά που πρέπει να υπολογιστούν, με αποτέλεσμα τη σημαντική αύξηση στο χρόνο υπολογισμού. Έτσι για παράδειγμα, ενώ για 15 νευρώνες υπολογίζονται συνολικά 270 συναπτικά βάρη, για 20 νευρώνες έχουμε 460 βάρη, για 25 νευρώνες 700 βάρη, για 30 νευρώνες 990 βάρη, για 35 νευρώνες 1,330 βάρη και για 40 νευρώνες 1,720 βάρη.

Η εκπαίδευση του δικτύου πραγματοποιήθηκε λοιπόν με αυτή τη διαμόρφωση του δικτύου (2 κρυμμένα επίπεδα και 15 νευρώνες σε κάθε επίπεδο) και οι 50,000,000 εποχές ολοκληρώθηκαν χωρίς το σφάλμα να πέσει κάτω από 1% ποτέ αλλά μετά από περίπου 1,000,000 να μην μεταβάλλεται σημαντικά και να ταλαντώνεται μεταξύ 3% και 5%. Ο συνολικός χρόνος που πήρε η εκπαίδευση του ΤΝΔ σε έναν επεξεργαστή ήταν της τάξης των 6 ημερών. Στο Σχήμα 3.7 δίδεται η συμπεριφορά του σφάλματος για 10,000,000 εποχές, όπου φαίνεται η ταλάντωση του σφάλματος γύρω από το 4%.



Σχήμα 3.7: Σφάλμα δικτύου (άξονας y, επί τις %) ως συνάρτηση των εποχών (άξονας x, σε λογαριθμική κλίμακα).

Επιστρέφοντας στα δεδομένα που κρατήσαμε για επιβεβαίωση (41 μοτίβα), εισάγαμε στο ΤΝΔ τα συναπτικά βάρη που προέκυψαν από την εκπαίδευση μετά από τις 50,000,000 εποχές και ζητήσαμε την πρόβλεψη του χρόνου ομαλοποίησης αν δώσουμε για δεδομένα εισόδου το μέσο χρόνο (ανά πτήση) καθυστέρησης κατά την άφιξη και κατά την αναχώρηση για τις συγκεκριμένες ημέρες που υπήρχε μέση καθυστέρηση άνω των 15 λεπτών. Προφανώς, το ΤΝΔ δεν είχε εκπαιδευθεί για αυτά τα δεδομένα. Από τις προβλέψεις του ΤΝΔ υπολογίστηκε το σχετικό σφάλμα σε σχέση με τον πραγματικό χρόνο ομαλοποίησης και τα αποτελέσματα συνοψίζονται στο Σχήμα 3.8.



Σχήμα 3.8: Κατανομή δεδομένων (άξονας y, %) σε σχέση με το σχετικό σφάλμα (άξονας x, %) στις προβλέψεις του ΤΝΔ για τον υπολογισμό του χρόνου ομαλοποίησης.

Όπως γίνεται φανερό, η πρόβλεψη για το 46% των δεδομένων είχε σχετικό σφάλμα μικρότερο από 10% και συνολικά για το 80% των δεδομένων το σχετικό σφάλμα ήταν κάτω από 30%. Το ΤΝΔ αστόχησε εντελώς (σφάλμα άνω του 90%) για ένα πολύ μικρό ποσοστό των δεδομένων (της τάξης του 4%) και αυτό είναι κάτι προς διερεύνηση. Πιθανές αιτίες της αστοχίας είναι η μη ύπαρξη γραμμικότητας μεταξύ χρόνου καθυστέρησης και χρόνου ομαλοποίησης, δηλαδή το ότι ένας μέσος χρόνος καθυστέρησης ήταν μεγάλος αυτό δεν σημαίνει αυτόματα ότι ο χρόνος ομαλοποίησης ήταν εξίσου μεγάλος. Αυτό συμβαίνει γιατί οι πιθανές αιτίες των καθυστερήσεων είναι πολλές και διαφορετικές και εμείς στην παρούσα εργασία προσπαθούμε κατά κάποιον τρόπο να τις ομαδοποιήσουμε. Αφαιρέσαμε τον αστάθμητο παράγοντα του καιρού και τα μέχρι στιγμής αποτελέσματα δείχνουν ότι υπάρχει μια ξεκάθαρη σύνδεση μεταξύ καθυστέρησης και αντίστοιχου χρόνου ομαλοποίησης. Κύριος σκοπός της εργασίας αυτής είναι να καταδείξει τη δυνατότητα του υλοποιηθέντος υπολογιστικού πλαισίου να διαχειριστεί προβλήματα Μεγάλων Δεδομένων και πιστεύουμε ότι το παρόν πρόβλημα δείχνει σαφέστατα ότι υπάρχουν δυνατότητες επιτυχούς αντιμετώπισης τέτοιων προβλημάτων. Ελπίζουμε η εργασία αυτή να αποτελέσει εφαλτήριο για περαιτέρω διερεύνηση του προβλήματος της καθυστέρησης στο χώρο των αερομεταφορών.

ΚΕΦΑΛΑΙΟ 4

ΕΦΑΡΜΟΓΗ ΑΝΑΛΥΣΗΣ ΜΕΓΑΛΩΝ ΔΕΔΟΜΕΝΩΝ ΣΕ ΠΡΟΒΛΗΜΑ ΑΣΤΡΟΦΥΣΙΚΗΣ ΚΑΙ ΧΡΗΣΗ ΤΝΔ ΓΙΑ ΤΗ ΠΡΟΒΛΕΨΗ ΤΗΣ ΜΟΡΦΟΛΟΓΙΑΣ ΤΗΣ ΔΟΜΗΣ ΕΞΩΓΑΛΑΞΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Στο κεφάλαιο αυτό θα ασχοληθούμε με ένα πρόβλημα αστροφυσικής και συγκεκριμένα με τη μορφολογία της δομής των εξωγαλαξιακών συστημάτων και με την χρήση ΤΝΔ για τη πρόβλεψη της μορφολογίας των γαλαξιών που έχουν παρατηρηθεί μέσω του «Sloan Digital Sky Survey» (SDSS). Αρχικά γίνεται μια γενική εισαγωγή για τη μορφολογία των γαλαξιών και την ταξινόμηση τους. Ακολούθως, περιγράφεται το στάδιο της επεξεργασίας των δεδομένων και η ανάλυση τους. Στην συνέχεια, περιγράφεται το στάδιο της ανάπτυξης των ΤΝΔ για το συγκεκριμένο πρόβλημα. Και στο τέλος παρουσιάζονται τα αποτελέσματα της διαδικασίας αυτής.

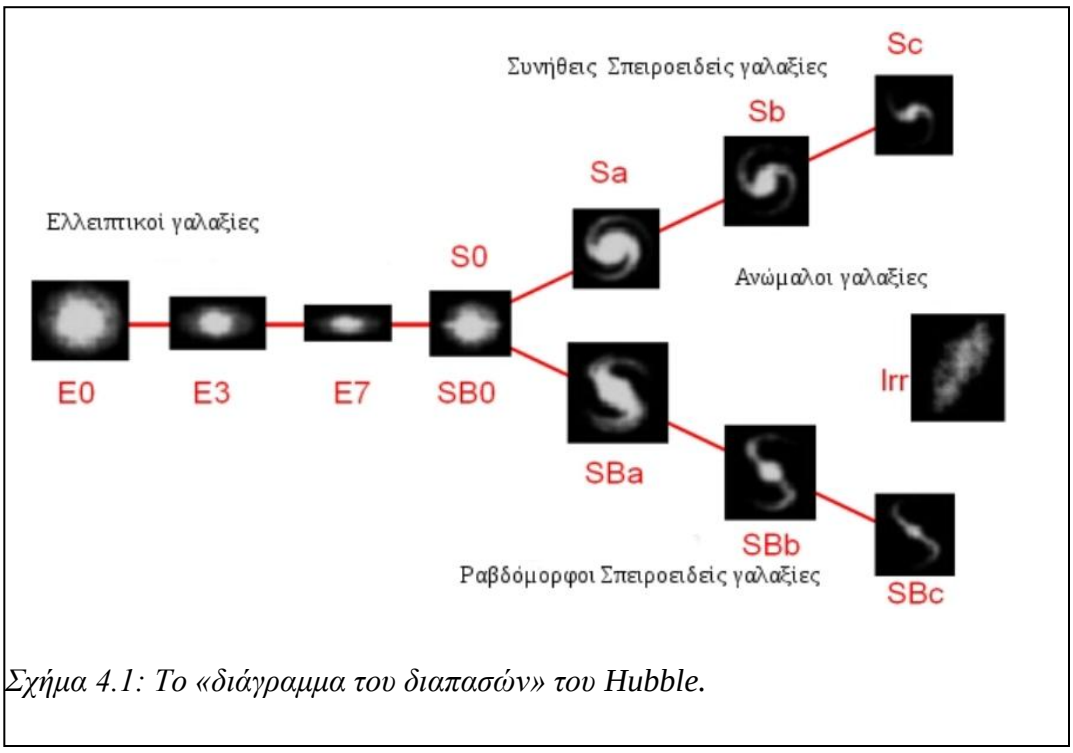
4.1 Εξωγαλαξιακά συτήματα

4.1.1 Μορφολογία γαλαξιών

Εκτός από το Γαλαξία μας, υπάρχει στο σύμπαν ένας τεράστιος αριθμός άλλων γαλαξιών οι οποίοι χαρακτηρίζονται ως εξωγαλαξιακά συστήματα. Από τη στιγμή που έγινε αντιληπτό ότι υπάρχουν και άλλοι γαλαξίες εκτός από το δικό μας, άρχισε η έρευνα για τον καθορισμό των ιδιοτήτων τους. Το πρώτο βήμα στην κατανόηση της φύσης των εξωγαλαξιακών συστημάτων έγινε με τη μορφολογική ταξινόμησή τους. Ο Hubble έπαιξε πολύ σημαντικό ρόλο σε αυτό το βήμα. Στο άρθρο του με τίτλο

«Extra-Galactic Nebulae» (1926) και αργότερα στο βιβλίο του «The Realm of the Nebulae» (1958) πρότεινε την ταξινόμηση των γαλαξιών σε τέσσερεις κατηγορίες με βάση τη μορφολογία τους.

Η ταξινόμηση αυτή, ακόμη και σήμερα ονομάζεται ταξινόμηση του Hubble και χωρίζει τους γαλαξίες σε τέσσερεις κατηγορίες: τους ελλειπτικούς («E»), τους συνήθεις σπειροειδείς («S», «Sa-Sc»), τους ραβδόμορφους σπειροειδείς («Sba-SBc») και τους ανώμαλους («Irr»). Ανάμεσα στους ελλειπτικούς και τους σπειροειδείς, υπάρχει ένας τύπος γαλαξιών που ονομάστηκαν αργότερα φακοειδείς οι οποίοι μπορεί να είναι είτε συνήθεις σπειροειδείς («S0») είτε ραβδόμορφοι («SB0») και δέν συμπεριλαμβάνονται στην αρχική ταξινόμηση. Ο Hubble ταξινόμησε τους μορφολογικούς τύπους που αναφέραμε, όπως φαίνεται στο σχήμα που ακολουθεί (Σχήμα 4.1), το οποίο ονομάζεται διάγραμμα του διαπασών του Hubble.

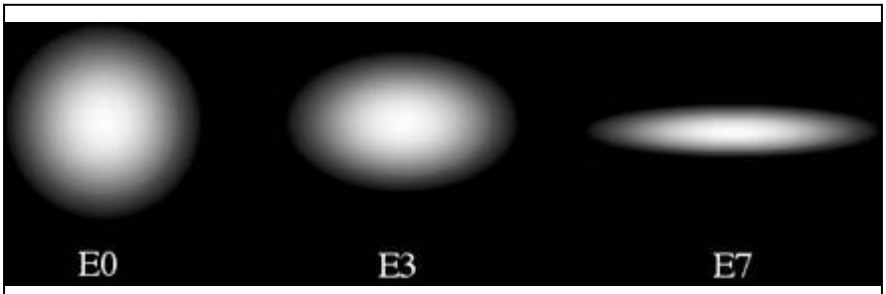


Σχήμα 4.1: Το «διάγραμμα του διαπασών» του Hubble.

Ο Hubble χώρισε την ομάδα των ελλειπτικών γαλαξιών σε υποκατηγορίες ανάλογα με την παρατηρούμενη ελλειπτικότητά τους η οποία ορίζεται ως:

$$\epsilon = 1 - \frac{b}{a} \qquad \text{(εξίσωση 4.1)}$$

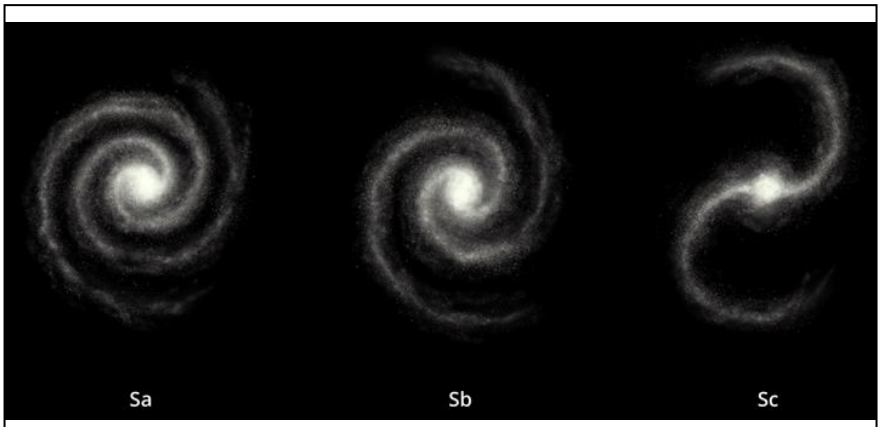
όπου a και b είναι οι φαινόμενοι μεγάλοι και μικροί ημιάξονες των ελλείψεων, αντίστοιχα όπως αυτές προβάλλονται στην ουράνια σφαίρα. Με τον τρόπο αυτό οι ελλειπτικοί γαλαξίες χωρίστηκαν σε υποκατηγορίες, ανάλογα με την ελλειπτικότητά τους, έτσι μετά το γράμμα «E» ακολουθεί ένας αριθμός που είναι η ποσότητα 10ϵ για το γαλαξία. Έτσι ξεκινάμε με όσους φαίνονται να έχουν σφαιρικό σχήμα οι οποίοι αποτελούν τον τύπο «E0» και φτάνουμε τελικά στους γαλαξίες με ϵ=0.7 που αποτελούν τον τύπο «E7» και έχουν πιο ελλειψοειδή μορφή. Ο λόγος που η ακολουθία σταματά στον τύπο «E7» είναι γιατί γαλαξίες με ελλειπτικότητα μεγαλύτερη από 0.7 δεν έχουν παρατηρηθεί (Sloan Digital Sky Survay). Στο Σχήμα 4.2 φαίνονται οι τύποι «E0», «E3» και «E7». Οι ελλειπτικοί γαλαξίες αντιπροσωπεύουν το 20% του συνόλου των γαλαξιών που έχουν παρατηρηθεί.



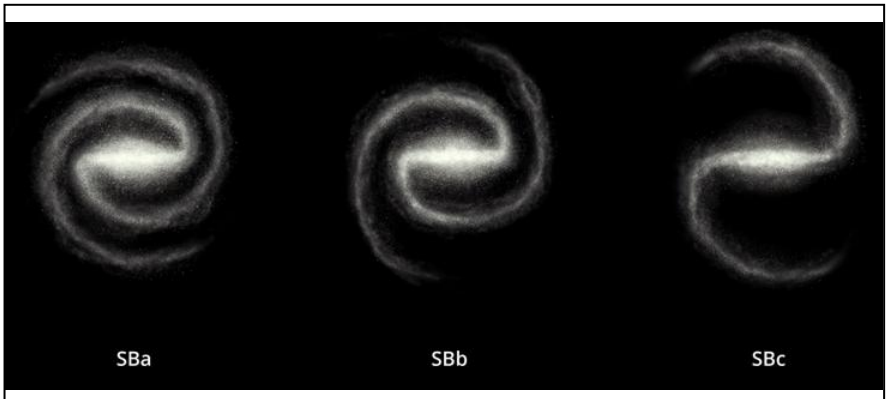
Σχήμα 4.2: Ελλειπτικοί γαλαξίες, τύποι «E0», «E3» και «E7».
Πηγή: http://astronomy.swin.edu.au/cosmos/

Ο Hubble διαίρεσε τους συνήθεις σπειροειδείς γαλαξίες («S») στους τύπους «Sa», «Sb» και «Sc». Οι σπειροειδείς γαλαξίες τύπου «Sa» είναι γαλαξίες που έχουν πολύ σφιχτούς βραχίονες («arms») με πολύ λαμπερή κεντρική περιοχή («bulge»). Οι «Sb» γαλαξίες έχουν πιο χαλαρούς βραχίονες και λιγότερο λαμπερή κεντρική περιοχή και οι τύπου «Sc» έχουν πολύ χαλαρούς βραχίονες και αμυδρή κεντρική περιοχή. Ο βαθμός σύσφιγξης των σπειροειδών βραχιόνων συνδέεται στενά με το σχετικό μέγεθος του κεντρικού εξογκώματος(«bulge»): όσο μεγαλύτερο είναι το κεντρικό εξόγκωμα, τόσο σφιχτότερος είναι ο σπειροειδείς βραχίονας. Οπότε οι γαλαξίες «Sa» τείνουν να έχουν μεγάλα κεντρικά εξογκώματα, ενώ οι «Sc» τείνουν να έχουν μικρά (ή και να μην έχουν καθόλου κεντρικό εξόγκωμα). Στο Σχήμα 4.3 φαίνονται οι τύποι «Sa», «Sb» και «Sc» στους συνήθεις σπειροειδείς γαλαξίες.

Με τον ίδιο τρόπο διαχώρισε και τους ραβδόμορφους σπειροειδείς γαλαξίες στους τύπους «SBa», «SBb», «SBc». Οπότε, οι ραβδόμορφοι γαλαξίες τύπου «SBa» είναι γαλαξίες που έχουν μεγάλα κεντρικά εξογκώματα, πολύ σφιχτούς βραχίονες με πολύ λαμπερή κεντρική περιοχή ενώ οι τύπου «Sbc» έχουν μικρά κεντρικά εξογκώματα (ή και καθόλου), πολύ χαλαρούς βραχίονες και αμυδρή κεντρική περιοχή. Οι συνήθεις και οι ραβδόμορφοι σπειροειδείς αντιστοιχούν στο 77% των συνολικών γαλαξιών που έχουν παρατηρηθεί. Στο Σχήμα 4.4 φαίνονται οι τύποι «SBa», «SBb» και «SBc» στους ραβδόμορφους σπειροειδείς γαλαξίες.



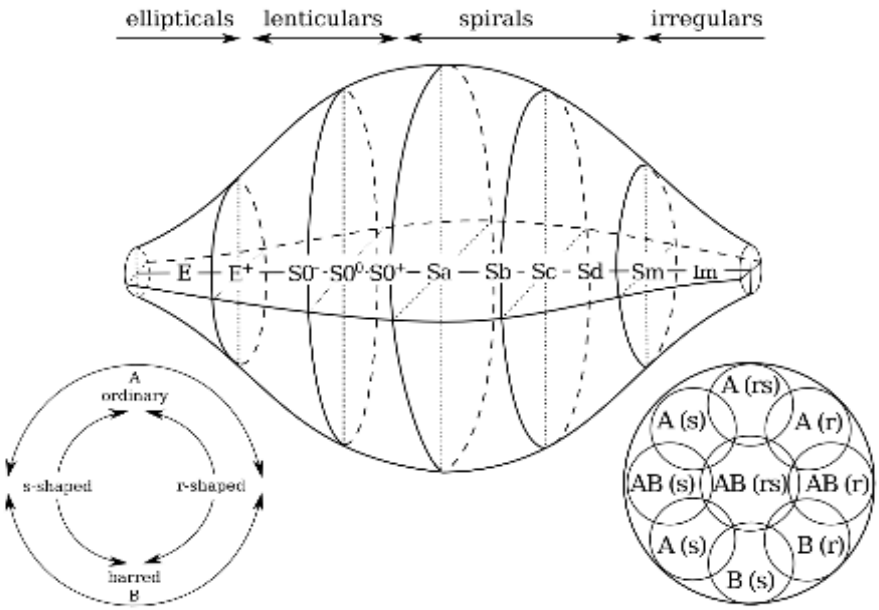
Σχήμα 4.3: Συνήθεις Σπειροειδείς γαλαξίες. Τύπου «Sa», «Sb» και «Sc».



Σχήμα 4.4: Ραβδόμορφοι Σπειροειδείς γαλαξίες. Τύπου «SBa», «SBb» και «SBc».

Την κατηγορία των ανώμαλων γαλαξιών («Irr»), ο Hubble τη διαχώρισε στην κατηγορία των ανώμαλων γαλαξιών τύπου I και τύπου II. Στην τύπου I, ανήκουν γαλαξίες στους οποίους μπορεί να διακριθεί κάποιου είδους οργανωμένη δομή (π.χ. σπείρες) στη μορφολογία των γαλαξιών ενώ στην τύπου II ανήκουν οι γαλαξίες με ακραία ανώμαλη μορφολογία. Αντιπροσωπεύουν μόνο το 3% του συνόλου των γαλαξιών που έχουν παρατηρηθεί.

Από την εποχή της δημοσίευσης της εργασίας του Hubble είχαν ακολουθήσει αρκετές τροποποιήσεις στην αρχική μορφολογική ταξινόμηση των εξωγαλαξιακών συστημάτων. Η πιο γνωστή είναι αυτή που εισήγαγε ο Gérard de Vaucouleurs (1959), ο οποίος πρότεινε μια «τρισδιάστατη» ταξινόμηση των γαλαξιών την οποία ονόμασε αναθεωρημένη ταξινόμηση Hubble. Η οποία σήμερα ονομάζεται ταξινόμηση de Vaucouleurs. Στο Σχήμα 4.5 φαίνεται η ταξινόμηση αυτή.



Σχήμα 4.5: Η ταξινόμηση των εξωγαλαξιακών συστημάτων κατά de Vaucouleurs. Η «διδιάστατη» ταξινόμηση κατά Hubble έχει γίνει «τρισδιάστατη» για να συμπεριληφθούν γαλαξίες με σιγμοειδή και δακτυλιοειδή μορφή.

Πηγή: Combes and Buta, (1996)

Ο de Vaucouleurs τόνισε ότι ο «δισδιάστατος» διαχωρισμός των σπειροειδών γαλαξιών από τον Hubble σε δύο διακριτούς τύπους, τους ραβδόμορφους και τους συνήθεις σπειροειδείς, αποτελεί μια υπεραπλούστευση και θεωρεί ότι υπάρχει μια συνεχής διαβάθμιση ραβδόμορφων, δακτυλιοειδών και σπειροειδών δομών. Υιοθετεί τον συμβολισμό SA και SB αντί για τους τύπους S και SB του Hubble. Επεκτείνει τους δείκτες a, b, c του σχήματος του Hubble προσθέτοντας τους δείκτες d και m (για τα ακανόνιστα νέφη του Μαγγελάνου) για να δηλώσει με τον τρόπο αυτό τη μετάβαση από δομές με καλά σχηματισμένες σπείρες σε δομές με πίο χαοτική μορφή. Επιπλέον, εισάγει τον συμβολισμό με τους δείκτες «(r)» και/ ή «(s)» για να δείξει την παρουσία δακτυλιοειδών και/ ή σπειροειδών χαρακτηριστικών. (Shu, 1982)

Μια άλλη χρήσιμη προσθήκη στο καθαρά μορφολογικό σχήμα του Hubble την έκανε ο Sidney van den Bergh, ο οποίος εισήγαγε την έννοια του τύπου της λαμπρότητας στους σπειροειδείς γαλαξίες. Πρότεινε την ενσωμάτωση ενός ακόμη δείκτη στον τύπο του Hubble, ο οποίος θα καθορίζει την κατηγορία λαμπρότητας του γαλαξία. Οι κατηγορίες λαμπρότητας των γαλαξιών διακρίνονται σε τέσσερις τύπους, από I μέχρι V: η κατηγορία λαμπρότητας I αντιστοιχεί στους ενδογενώς λαμπρότερους σπειροειδείς γαλαξίες, ενώ η κατηγορία V στους ενδογενώς αμυδρότερους.

4.1.2. Χαρακτηριστικά Γαλαξιών

- **Ελλειπτικοί**

Οι Ελλειπτικοί γαλαξίες περιέχουν ελάχιστη σκόνη και αέριο. Η παρουσία σκόνης και αερίου αποτελεί ένδειξη διαδικασιών δημιουργίας αστεριών. Συνεπώς δεν έχουμε δημιουργία νέων αστέρων στους ελλειπτικούς γαλαξίες, οπότε αποτελούνται κατά κύριο λόγο από αστέρια πληθυσμού II, δηλαδή «γέρικα» κόκκινα αστέρια που περιέχουν ελάχιστα μέταλλα. Οι διαστάσεις των ελλειπτικών γαλαξιών ποικίλλουν, από νάνους γαλαξίες π.χ. NGC 185 εώς γιγάντιους γαλαξίες π.χ. M87 (Bingelli et al., 1987). Οι ελλειπτικοί γαλαξίες αντιπροσωπεύουν ένα συνοθύλευμα άστρων που υποστηρίζονται αντιστέκοντας στην αμοιβαία ιδιοβαρυτική τους έλξη από τη τυχαία κατανομή ταχυτήτων των άστρων τους. Στο Σχήμα 4.6 φαίνεται ένα παράδειγμα ελλειπτικού γαλαξία.

- **Σπειροειδείς**

Στους Σπειροειδείς γαλαξίες ανήκει και ο Γαλαξίας μας. Συγκεκριμένα ο Γαλαξίας μας ανήκει στην κατηγορία των ραβδόμορφων σπειροειδών. Το κεντρικό εξόγκωμα ενός σπειροειδούς γαλαξία είναι φτωχό σε αέριο και σκόνη, και περιέχει κυρίως γέρικα αστέρια πληθυσμού II. Το εξόγκωμα περιβάλλεται από ένα δίσκο ο οποίος αποτελείται από αέριο και σκόνη που περιέχουν νεαρά αστέρια πληθυσμού I. Τα αστέρια πληθυσμού I, είναι νεαρά μπλέ αστέρια και είναι πλούσια σε μέταλλα. Όσο μετακινούμαστε προς το δεξιό άκρο του «διαγράμματος του διαπασών του Hubble» τόσο αυξάνεται

και η περιεκτικότητα του γαλαξία σε μέταλλα, δηλαδή ένας Sa γαλαξίας περιέχει λιγότερα μέταλλα απο έναν Sb και αυτός με την σειρά του από έναν Sc. Ένα παράδειγμα σπειροειδούς γαλαξία φαίνεται στο Σχήμα 4.7

- **Φακοειδείς**

Οι Φακοειδείς γαλαξίες είναι μια ενδιάμεση περίπτωση των σπειροειδών και των ελλειπτικών γαλαξιών. Περιέχουν ένα δίσκο και ένα κεντρικό εξόγκωμα αλλά δε περιέχουν σπειροειδείς βραχίονες. Εάν ο δίσκος είναι αμυδρός είναι πιθανόν να υπάρξει σύγχυση με γαλαξία τύπου E0. Οι Φακοειδείς γαλαξίες πολλές φορές ονομάζονται και σπειροειδείς χωρίς βραχίονες. Οι φακοειδείς γαλαξίες έχουν πολύ χαμηλό ρυθμό γέννησης νέων αστεριών (DeGraff et al., 2007). Κατά συνέπεια αποτελούνται κατά κύριο λόγο από αστέρες μεγάλης ηλικίας, όπως συμβαίνει και με τους ελλειπτικούς γαλαξίες. Παρ' όλ' αυτά πολύ συχνά υπάρχουν μεγάλες ποσότητες σκόνης στους δίσκους τους. Στο Σχήμα 4.8 και 4.9 παρουσιάζονται δυο παραδείγματα φακοειδών γαλαξιών.

- **Ανώμαλοι**

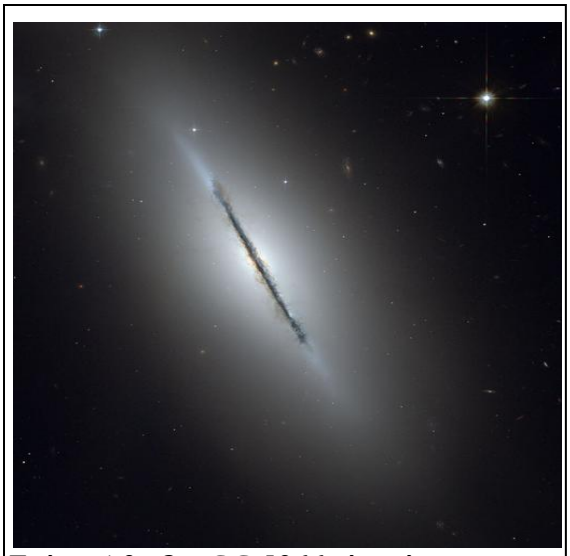
Στους ανώμαλους γαλαξίες ανήκουν οι γαλαξίες που όπως προείπαμε έχουν ακανόνιστο σχήμα και χαρακτηριστικά. Είναι ως επί το πλείστον μικρότεροι σε σύγκριση με τους σπειροειδείς και τους ελλειπτικούς. Στους περισσότερους ανώμαλους γαλαξίες παρατηρείται σχηματισμός αστέρων που οφείλεται στην υψηλή περιεκτικότητα τους σε αέριο. Νεαρά άστρα και λαμπρές περιοχές μεσοαστρικού αερίου κυριαρχούν σε αυτούς τους γαλαξίες. Συνήθως είναι αποτέλεσμα σύγκρουσης δύο γαλαξιών ή βαρυτικής αλληλεπίδρασης γαλαξιών που έτυχε να βρεθούν κοντά. Στο Σχήμα 4.10 υπάρχουν δύο φωτογραφίες από γαλαξίες που ανήκουν στην κατηγορία αυτή.

Σχήμα 4.6: Ο M87, αποτελεί παράδειγμα ελλειπτικού γαλαξία τύπου E1.

Πηγή: http://hubblesite.org

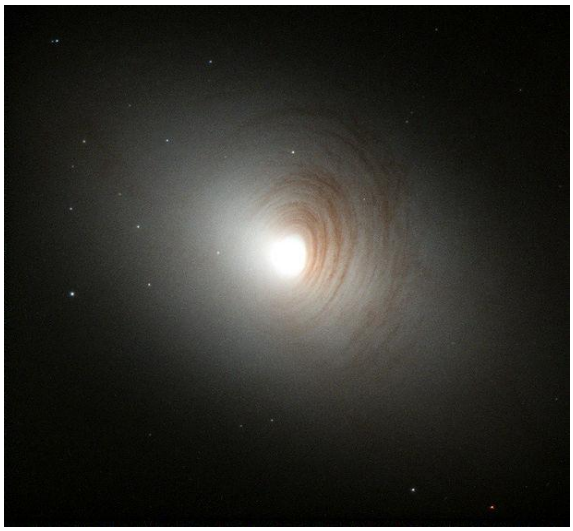
Σχήμα 4.7: Ο NGC1365 αποτελεί παράδειγμα σπειροειδούς γαλαξία τύπου Sbb.

Πηγή: http://hubblesite.org



Σχήμα 4.8: Ο NGC 5866 είναι ένας φακοειδής γαλαξίας στον αστερισμό του Drac. Στην εικόνα αυτή φαίνεται ότι οι φακοειδείς γαλαξίες μπορούν να διατηρούν μια σημαντική ποσότητα σκόνης στο δίσκο τους. Αποτελείται από ελάχιστο έως και καθόλου αέριο.

Πηγή:http://www.spacetelescope.org/images/opo0624a/



Σχήμα 4.9: Ο NGC 2787 είναι ένας φακοειδείς γαλαξίας με ορατή απορρόφηση σκόνης. Ενώ έχει ταξινομηθεί ως ένας γαλαξίας S0, μπορεί κανείς να δει τη δυσκολία διαφοροποίησης μεταξύ σπειροειδών, ελλειπτικών και φακοειδών.

Πηγή:http://www.spacetelescope.org/images/opo0207a/


Σχήμα 4.10: Δύο παραδείγματα ανώμαλων γαλαξιών τύπου Ι. Το Μεγάλο και το Μικρό Νέφος του Μαγγελάνου, αντίστοιχα.

Πηγή: http://hubblesite.org

4.1.3 Χρώμα και Φαινόμενο Μέγεθος γαλαξία

Το χρώμα είναι μια υποκειμενική άποψη. Αυτό που ένα άτομο ονομάζει «μπλε» μπορεί να είναι διαφορετική απόχρωση από το «μπλε» ενός άλλου ατόμου. Έτσι οι αστρονόμοι για το προσδιορισμό των χρωμάτων των αστεριών και των γαλαξιών έχουν δώσει έναν συγκεκριμένο ορισμό του χρώματος τον οποίο ο καθένας μπορεί να συμφωνήσει. Σύμφωνα με τον ορισμό αυτό, το χρώμα είναι η διαφορά του φαινόμενου μεγέθους μεταξύ δύο φίλτρων.

Το φαινόμενο μέγεθος (m) είναι ένας αριθμός που καθορίζει πόσο φωτεινό είναι ένα ουράνιο σώμα π.χ. αστέρας, γαλαξίας, όπως φαίνεται από την Γη. Όσο πιο φωτεινό φαίνεται ένα ουράνιο αντικείμενο, τόσο μικρότερη είναι η αριθμητική τιμή του μεγέθους του. Μια αύξηση του φαινόμενου μεγέθους ενός αριθμού αντιστοιχεί σε μείωση της φωτεινότητας κατά ένα συντελεστή περίπου 2.51. Δηλαδή, ένα αντικείμενο φαινόμενου μεγέθους πέντε είναι 2.51 φορές ασθενέστερο από ένα αντικείμενο που έχει φαινόμενο μέγεθος τέσσερα. Ο ήλιος έχει φαινόμενο μέγεθος -26. Το λαμπρότερο αστέρι στο βόρειο ουρανό, ο Σείριος, έχει φαινόμενο μέγεθος -1.5. Το μικρότερο αντικείμενο που μπορούμε να δούμε με τα μάτια μας έχει φαινόμενο μέγεθος περίπου 6. Το μικρότερο αντικείμενο που μπορούμε να δούμε μέσω SDSS έχει φαινόμενο μέγεθος περίπου 23. Όταν λέμε όμως ότι ένα ουράνιο σώμα έχει ένα ορισμένο φαινόμενο μέγεθος, πρέπει να καθορίσουμε το χρώμα στο οποίο αναφέρεται το μέγεθος.

Το SDSS για την μέτρηση των φαινόμενων μεγεθών των ουράνιων σωμάτων λαμβάνει εικόνες χρησιμοποιώντας σύστημα πέντε φίλτρων, το υπεριώδες «u», το πράσινο «g», το κόκκινο «r» και δύο υπέρυθρα μήκη κύματος το «i» και το «z» οπότε το φαινόμενο μέγεθος του γαλαξία έχει διαφορετική τιμή ανάλογα με το φίλτρο που χρησιμοποιήθηκε για την απεικόνιση του. Οι αστρονόμοι που σχεδίασαν το SDSS επέλεξαν αυτά τα φίλτρα για να δουν ένα ευρύ φάσμα χρωμάτων, εστιάζοντας στα χρώματα των ενδιαφερόντων ουράνιων αντικειμένων. Το χρώμα συμβολίζεται με την αφαίρεση των μεγεθών: m_u- m_g, m_g- m_r, m_r- m_i, m_i- m_z. Ένα ουράνιο σώμα με υψηλό χρώμα m_g- m_r είναι πιο κόκκινο από ένα άλλο με χαμηλό χρώμα m_g- m_r.

Για ένα γαλαξία που παρατηρείται από ένα φίλτρο x, το συνολικό φαινόμενο μέγεθος δίνεται από τον τύπο στην εξίσωση (4.2),

$$m_{total,x} = -2.51 log_{10}\left(\frac{F_x}{F_{x,0}}\right) \qquad \text{(εξίσωση 4.2)}$$

όπου το $F_x = F_{total} = F_{disk} + F_{bulge}$ και αντιστοιχεί στη συνολική ένταση που παρατηρείται από ένα φίλτρο x και $F_{x,0}$ είναι μια σταθερά, η οποία αντιστοιχεί στην ένταση αναφοράς για το συγκεκριμένο φίλτρο. Ως ένταση αναφοράς σε ένα φίλτρο, χρησιμοποιείται ένα αντικείμενο το οποίο έχει μηδενικό φαινόμενο μέγεθος. Συνήθως, για το λόγο αυτό χρησιμοποιείται ο αστέρας Βέγας, ο οποίος έχει μηδενικό φαινόμενο μέγεθος στο UBVRI.

Οι γαλαξίες είναι συλλογές εκατομμυρίων ή δισεκατομμυρίων αστεριών, και περιέχουν αέριο και σκόνη. Εάν τα περισσότερα από τα αστέρια ενός γαλαξία είναι μπλε, ο γαλαξίας θα εμφανιστεί γενικά μπλε. Εάν τα περισσότερα από τα αστέρια είναι κόκκινα, ο γαλαξίας θα εμφανιστεί γενικά κόκκινος. Έτσι, το γενικό χρώμα ενός γαλαξία μπορεί να πει στους αστρονόμους κάτι για το είδος των αστεριών που περιέχει ο γαλαξίας.

Ωστόσο, η ερμηνεία του χρώματος των γαλαξιών περιπλέκεται από δύο παράγοντες. Πρώτον, λόγω της σκόνης στους γαλαξίες η οποία απορροφά το φως σε μικρότερα μήκη κύματος. Επειδή η σκόνη απορροφά το φως μικρού μήκους κύματος από τους γαλαξίες, το φως που φτάνει στη Γη είναι πιο κόκκινο από το φως που εκπέμπουν οι γαλαξίες. Δεύτερον, το χρώμα ενός γαλαξία αλλάζει από το πόσο γρήγορα κινείται. Όταν ένας γαλαξίας μετακινείται μακριά από τη Γη με μεγάλη ταχύτητα, τα μήκη κύματος των φωτεινών κυμάτων που εκπέμπει είναι πιο κόκκινα, λόγω του «redshift». Οι γαλαξίες που απομακρύνονται γρήγορα έχουν μεγαλύτερη «κόκκινη μετατόπιση» και έτσι εμφανίζονται πιο κόκκινοι.

4.1.4 Εξωγαλαξιακά συστήματα και Φάσμα Ηλεκτρομαγνητικής Ακτινοβολίας

Η μορφολογία των γαλαξιών προσφέρει σημαντικές πληροφορίες για τη φύση τους. Οι πληροφορίες αυτές όμως είναι πιο βάσιμες όταν συνδυαστούν με τη μελέτη της ηλεκτρομαγνητικής ακτινοβολίας που εκπέμπουν οι γαλαξίες σε διάφορες φασματικές περιοχές. Όποτε θα κάνουμε μια σύντομη επισκόπηση του φάσματος ενός τυπικού γαλαξία.

Το ορατό φως που εκπέμπει ένας γαλαξίας προέρχεται κυρίως από αστέρια και από θερμά νέφη αερίων. Το φάσμα των αστεριών αυτών αποτελείται από γραμμές απορρόφησης, ενώ το φάσμα στα θερμά νέφη αερίων αποτελείται από γραμμές εκπομπής. Στους σπειροειδείς και τους ανώμαλους γαλαξίες υπάρχει γέννηση νέων αστεριών όπου τα φωτεινότερα αστέρια τους είναι φασματικών τύπων Ο και Β, δηλαδή νεαρά, θερμά αστέρια. Αντίθετα στους ελλειπτικούς γαλαξίες κατά βάση δεν υπάρχει γέννηση νέων αστεριών και τα φωτεινότερα αστέρια τους είναι ερυθροί γίγαντες. Επομένως οι ελλειπτικοί γαλαξίες, των οποίων τα φωτεινά αστέρια είναι σχετικά ψυχρά, είναι πιο κόκκινοι από τους σπειροειδείς και ανώμαλους γαλαξίες των οποίων τα φωτεινά αστέρια είναι πολύ θερμά. Οι ελλειπτικοί γαλαξίες έχουν φάσμα με έντονες γραμμές απορρόφησης και σχεδόν καθόλου γραμμές εκπομπής. Ενώ, οι σπειροειδείς γαλαξίες έχουν έντονες γραμμές απορρόφησης αλλά και μέτριας έντασης γραμμές εκπομπής. Οι ανώμαλοι γαλαξίες γενικά έχουν φάσματα παρόμοια με αυτά των σπειροειδών γαλαξιών.

Στο υπέρυθρο η εκπομπή των γαλαξιών προέρχεται κυρίως από τη σκόνη. Η εκπομπή στο υπέρυθρο είναι πιο έντονη στους σπειροειδείς και τους ανώμαλους γαλαξίες από ότι στους ελλειπτικούς. Στους σπειροειδείς γαλαξίες η υπέρυθρη ακτινοβολία προέρχεται κυρίως από τις σπείρες όπου και η συγκέντρωση σκόνης είναι μεγαλύτερη.

Η ραδιοφωνική εκπομπή συνήθως είναι εντονότερη στους σπειροειδείς γαλαξίες από ότι στους ελλειπτικούς. Τα νέφη αερίου στους σπειροειδείς γαλαξίες είναι περιοχές έντονης ραδιοφωνικής εκπομπής.

Στο υπεριώδες η εκπομπή των γαλαξιών προέρχεται κυρίως από θερμά αστέρια με μικρό χρόνο ζωής. Συνεπώς η υπεριώδης ακτινοβολία ιχνογραφεί τις σπείρες των γαλαξιών όπου και κατά βάση γεννιούνται νέα αστέρια. Στους ελλειπτικούς γαλαξίες η υπεριώδης ακτινοβολία είναι μικρότερη και προέρχεται κυρίως από αστέρια που βρίσκονται στο στάδιο της καύσης του He.

Η εκπομπή στις ακτίνες Χ από τους γαλαξίες προέρχεται κυρίως από αστέρες νετρονίων και μαύρες τρύπες όπως επίσης και από το υπέρθερμο αέριο που βρίσκεται στους γαλαξίες με θερμοκρασίες μεταξύ 10 και 100 εκατομμυρίων βαθμών Κέλβιν.

4.3 Ανάλυση Δεδομένων και χρήση ΤΝΔ για την πρόβλεψη της μορφολογίας των Γαλαξιών

Σκοπός του κεφαλαίου αυτού είναι να ανακτηθούν και να αναλυθούν δεδομένα με τη βοήθεια ΤΝΔ με σκοπό να εξετασθεί κατά πόσον μπορούν να εξαχθούν ασφαλή συμπεράσματα σχετικά με τη μορφολογία γαλαξιών χρησιμοποιώντας πληροφορία για τα χρώματα. Τα δεδομένα που έχουμε χρησιμοποιήσει έχουν ανακτηθεί από ένα κατάλογο που έχουν δημιουργήσει ο Alan Meert και οι συνεργάτες του (2015) από δεδομένα που έχουν πάρει από το Sloan Digital Sky Survey και συγκεκριμένα την βάση δεδομένων DR7 (Data Release 7). Το Sloan Digital Sky Survey έχει χαρτογραφήσει το ¼ του συνόλου του ουρανού και έχει καταγράψει πληροφορίες για περίπου 7×10^5 γαλαξίες. Παρέχει πληροφορίες όπως φάσματα απεικόνισης, φωτογραφίες και «redshifts» για ουράνια σώματα. Οι φωτογραφίες όπως έχουμε προαναφέει λαμβάνονται από ένα φωτομετρικό σύστημα πέντε φίλτρων «u», «g», «r», «i», «z». Οι φωτογραφίες αυτές μπορούν να παρέχουν σημαντικές πληροφορίες για το παρατηρούμενο αντικείμενο, όπως αν είναι σημειακό ή εκτεταμένο (όπως ένας γαλαξίας) ή ακόμη πληροφορίες σχετικά με το πώς εξαρτάται η φωτεινότητα στις CCDs απεικονίσεις με διάφορα αστρονομικά μεγέθη, π.χ. φαινόμενο μέγεθος.

Συγκεκριμένα, είχαμε στη διάθεση μας δεδομένα για 670,722 γαλαξίες σε φωτομετρία σε τρία φίλτρα, το φίλτρο «g» (λ=468.6 nm), το φίλτρο «r» (λ=616.5 nm) και το φίλτρο «i» (λ=748.1 nm), αντίστοιχα. Σε κάθε φίλτρο είχαμε τις εξής πληροφορίες για κάθε γαλαξία:

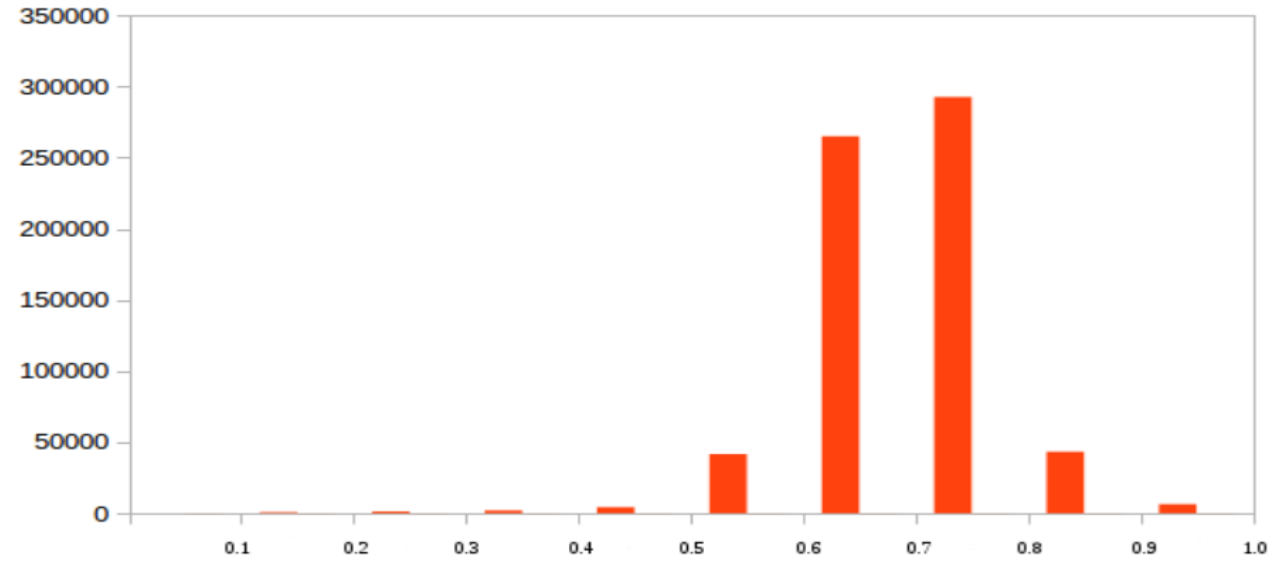
- Το συνολικό φαινόμενο μέγεθος του γαλαξία (**m_{tot}**)
- Το λόγο της έντασης στο σφαιροειδές προς τη συνολική ένταση (**BT – Bulge-to-Total ratio**). Όταν **F_{tot}= F_{bulge}** τότε **BT=1** και σε αυτή τη περίπτωση έχουμε ελλειπτικό γαλαξία
- Τη συνολική ένταση του σφαιροειδούς και το σφάλμα αυτού (**m_{bulge}, $m_{bulgeerr}$**)
- Τη συνολική ένταση του δίσκου και το σφάλμα αυτού (**m_{disk}, $m_{diskerr}$**)

Από το σύνολο των δεδομένων αυτών αφαιρέσαμε τις τιμές για εκείνους τους γαλαξίες για τους οποίους υπήρχαν σφάλματα κατά την μέτρηση (είτε σε ένα ή σε περισσότερα φίλτρα). Επίσης, αφαιρέσαμε τα δεδομένα για τους γαλαξίες τους οποίους τα χρώματα τους, (δηλαδή τα **m_{tot_g}-m_{tot_r}** και **m_{tot_r}-m_{tot_i}**) είχαν τιμή κατ'απόλυτη τιμή μεγαλύτερη από 1.5. Ο λόγος για τον οποίο το κάναμε αυτό είναι επειδή τιμές μεγαλύτερες του 1.5 θεωρούνται μη φυσιολογικές και υποδηλώνουν κάποιο σφάλμα κατά τη μέτρηση των μεγεθών αυτών. Με τον τρόπο αυτό, ο αριθμός των γαλαξιών με

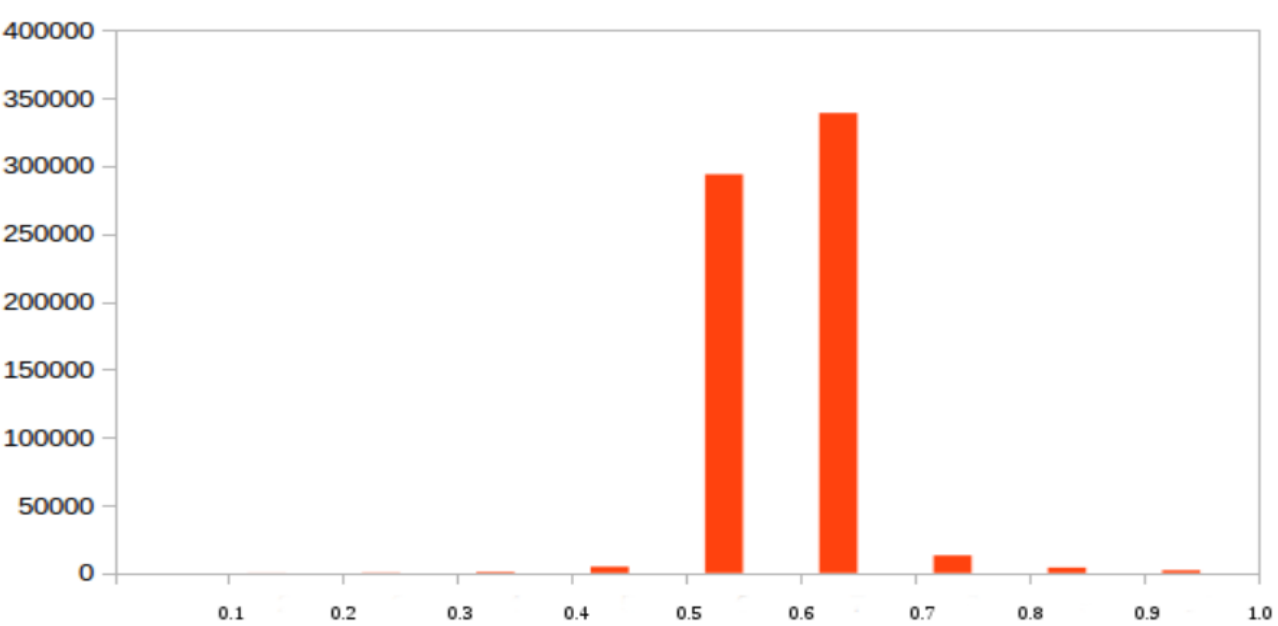
δεδομένα απαλλαγμένα από σφάλματα και με πλήρες σύνολα μετρήσεων έπεσε στους 657.460 (ποσοστό περίπου ίσο με το 98% του αρχικού).
Αποφασίστηκε να εξετασθεί κατά πόσον η μορφολογία των γαλαξιών είναι δυνατόν να προκύψει μόνο από τα χρώματά τους και για το λόγο αυτό δημιουργήθηκε ένα αρχείο του οποίου κάθε του γραμμή περιείχε 3 δεδομένα, δηλαδή τις διαφορές $\mathbf{m_{tot\_g}\text{-}m_{tot\_r}}$ και $\mathbf{m_{tot\_r}\text{-}m_{tot\_i}}$, καθώς και το $\mathbf{BT_r}$. Ο λόγος για τον οποίο χρησιμοποιήσαμε μόνο τις τιμές του $\mathbf{BT_r}$ και όχι του $\mathbf{BT_g}$ ή του **BTi** είναι επειδή στο φίλτρο αυτό μπορούμε πιο εύκολα να ξεχωρίσουμε αν ο γαλαξίας έχει ελλειψοειδή μορφή. Αφού, όπως είναι γνωστό από την θεωρία, στο εξόγκωμα του γαλαξία υπάρχουν γηραιότερα αστέρια, χρώματος κόκκινου και έτσι στην περίπτωση που έχουμε $\mathbf{BT_r=1}$, τότε αυτό σημαίνει ότι ο γαλαξίας έχει ελλειψοειδή μορφή.

Το ΤΝΔ αποφασίστηκε να έχει 2 δεδομένα εισόδου και ένα δεδομένο εξόδου. Τα δεδομένα εισόδου είναι οι διαφορές $\mathbf{m_{tot\_g}\text{-}m_{tot\_r}}$ και $\mathbf{m_{tot\_r}\text{-}m_{tot\_i}}$ και το δεδομένο εξόδου είναι το $\mathbf{BT_r}$. Στην συνέχεια, κανονικοποιήσαμε τα δεδομένα μας, με τον ίδιο τρόπο που περιγράψαμε για την εφαρμογή στο πρόβλημα των αερομεταφορών, χρησιμοποιώντας την εξίσωση (3.1) και το μέγιστο και ελάχιστο κάθε στήλης ξεχωριστά, Αξίζει να σημειωθεί ότι οι τιμές του $\mathbf{BT_r}$ δεν χρειάστηκε να κανονικοποιηθούν αφού ήταν ήδη μεταξύ 0 και 1.

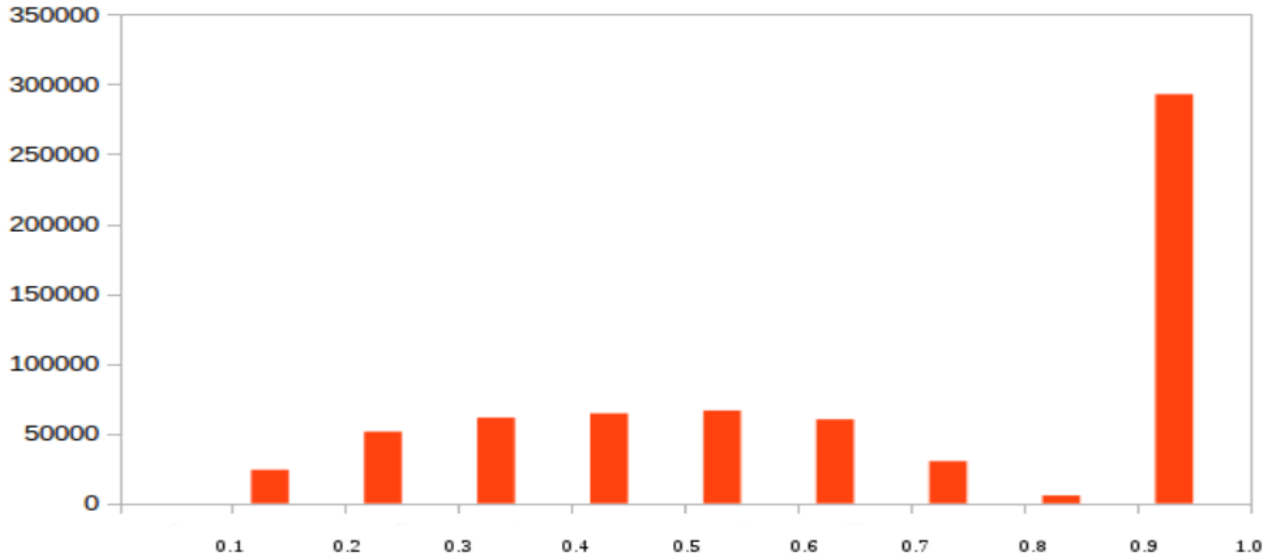
Πριν γίνει επιλογή δεδομένων για εκπαίδευση του ΤΝΔ, πραγματοποιήθηκε στατιστική ανάλυση των κανονικοποιημένων δεδομένων, η οποία παρουσιάζεται στα Σχήματα 4.11, 4.12 και 4.13. Αυτό που γίνεται άμεσα φανερό είναι ότι οι διαφορές των συνολικών μεγεθών (δηλαδή τα χρώματα) ήταν συγκεντρωμένα σε ένα πολύ μικρό εύρος τιμών. Για το κανονικοποιημένο $\mathbf{m_{tot\_g}\text{-}m_{tot\_r}}$ το 80% ήταν μεταξύ των 0.6 και 0.8 (Σχήμα 4.11) και για το κανονικοποιημένο $\mathbf{m_{tot\_r}\text{-}m_{tot\_i}}$ το 90% ήταν στο διάστημα 0.5-0.7 (Σχήμα 4.12). Ταυτόχρονα, για το $\mathbf{BT_r}$ το 40% των τιμών ήταν συγκεντρωμένο στο διάστημα 0.9-1.0 και οι υπόλοιπες τιμές ήταν περίπου ισοκατανεμημένες μεταξύ 0.1 και 0.9 (Σχήμα 4.13). Αξίζει να σημειωθεί ότι από αυτό το 40%, το 99% έχει τιμή $\mathbf{BT_r}$ ακριβώς ίση με 1! Αυτή η σαφέστατα ανισομερής κατανομή μεταξύ δεδομένων εισόδου και εξόδου συνεπάγεται ότι το ΤΝΔ θα έχει δυσκολίες στο να αναγνωρίσει συσχετισμούς μεταξύ των δεδομένων, πράγμα που σημαίνει ότι η εκπαίδευσή του θα είναι προβληματική.



Σχήμα 4.11: Κατανομή κανονικοποιημένων τιμών m_{tot_g}-m_{tot_r}.

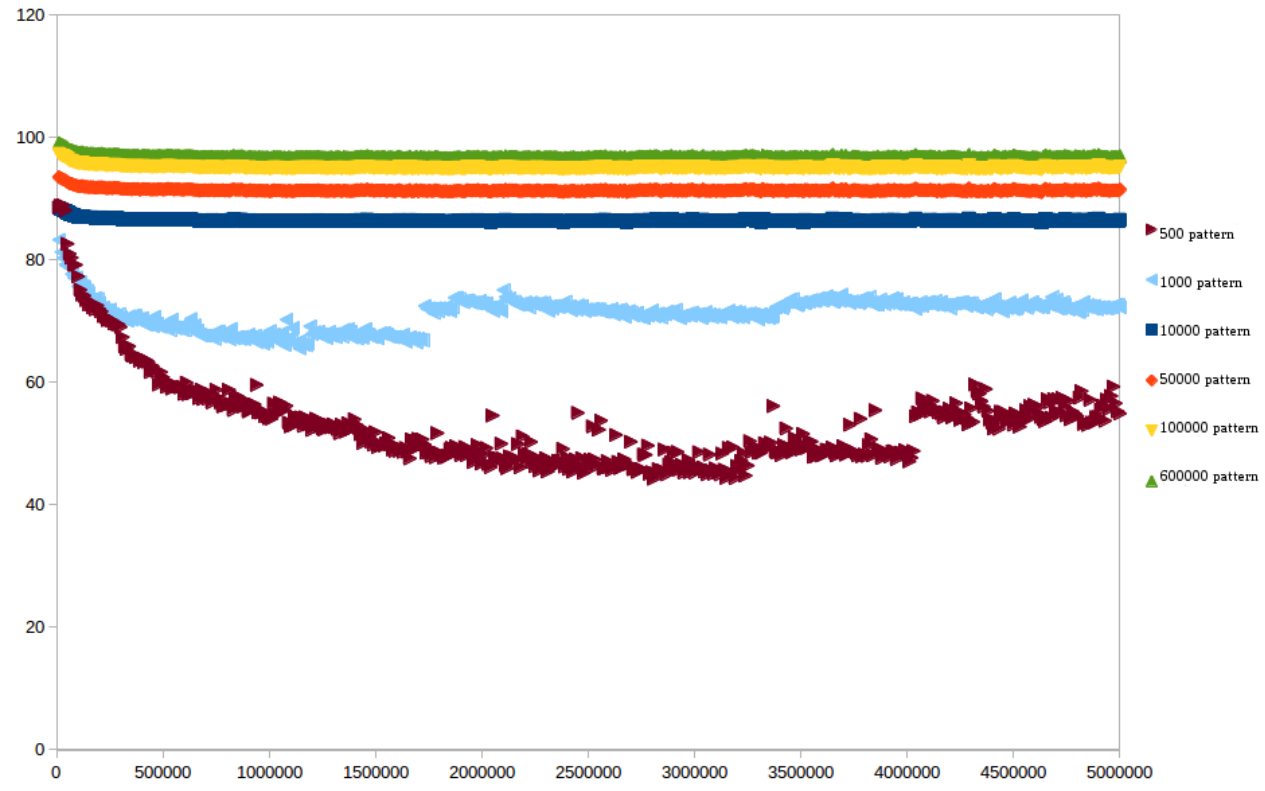


Σχήμα 4.12: Κατανομή κανονικοποιημένων τιμών m_{tot_r}-m_{tot_i}.



Σχήμα 4.13: Κατανομή τιμών BT_r.

Πριν προχωρήσουμε στην εκπαίδευση του ΤΝΔ με 2 δεδομένα εισόδου και 1 εξόδου, αποφασίστηκε να έχουμε 2 κρυμμένα επίπεδα με 15 νευρώνες στο κάθε ένα (και συνολικά 270 συναπτικά βάρη), βασισμένοι στην εμπειρία από τις δοκιμές του προηγούμενου κεφαλαίου. Επίσης, στο πρόβλημα που περιγράφεται σε αυτό το κεφάλαιο, λόγω της πληθώρας των διαθέσιμων μοτίβων (657,460), έγινε μια ανάλυση για το πόσα μοτίβα απαιτούνται για μια καλή εκπαίδευση του δικτύου (βλέπε Πίνακα 2.3). Έτσι, κατά Hush (1989) με 2 δεδομένα εισόδου, ο ελάχιστος αριθμός μοτίβων που απαιτείται είναι 180 και ο βέλτιστος 360. Λαμβάνοντας υπόψη το πλήθος των συναπτικών βαρών, το βέλτιστο πλήθος μοτίβων είναι, κατά Baum and Haussler (1989) 2,700 μοτίβα, κατά Klimasauskas (1993) 1,350 μοτίβα και κατά Garson (1998) 8,100 μοτίβα. Επιλέχθηκε λοιπόν να γίνουν δοκιμές με ένα εύρος διαφορετικών συνδυασμών μοτίβων, για να εξεταστεί εάν υπάρχουν ενδείξεις για την επίτευξη σύγκλισης κατά την εκπαίδευση του δικτύου. Για αυτές τις δοκιμές, επιλέξαμε με τυχαίο τρόπο (κατά Park and Miller, 1989) από το αρχείο με τα 657.460 διαθέσιμα μοτίβα, 500, 1,000, 10,000, 50,000, 100,000 και 600,000 μοτίβα. Αρχικά, η εκπαίδευση έγινε για 5,000,000 εποχές, βασιζόμενοι στην πρότερη εμπειρία ότι μετά από 1,000,000 εποχές το σφάλμα δεν μεταβάλλεται σημαντικά. Η εκπαίδευση του ΤΝΔ σε ένα επεξεργαστή πρέπει να σημειωθεί ότι είναι ιδιαιτέρως χρονοβόρα, λόγω του πλήθους των μοτίβων, και διήρκεσε από 13 περίπου ώρες για τα 500 μοτίβα μέχρι 9 ημέρες για τα 600,000 μοτίβα. Στο σχήμα που ακολουθεί (Σχήμα 4.14) αποτυπώνεται η συμπεριφορά του σφάλματος σε σχέση με τον αριθμό των εποχών κατά τη διάρκεια των διαφόρων τρόπων διαμόρφωσης των μοτίβων εκπαίδευσης του δικτύου.



Σχήμα 4.14: Σφάλμα δικτύου (άξονας y, %) σε σχέση με τον αριθμό των εποχών (άξονας x).

Από τα αποτελέσματα που παρουσιάζονται, γίνεται φανερό ότι το σφάλμα δεν πέφτει ποτέ κάτω από 60%. Αυτό είναι κάτι που το αναμέναμε, όπως εξηγήθηκε νωρίτερα κατά τη συζήτηση της στατιστικής ανάλυσης των κανονικοποιημένων δεδομένων. Επίσης, παρατηρείται ότι το μικρότερο σφάλμα το είχαμε σαφέστατα για τα λιγότερα μοτίβα που χρησιμοποιήθηκαν. Αυτό εξηγείται από το γεγονός ότι όσο περισσότερα μοτίβα χρησιμοποιούνται, τόσο αυξάνει η αβεβαιότητα μέσα στο ΤΝΔ, αφού όλο και πιο πολλές τιμές με πολύ μικρή διασπορά εισάγουμε με τα δεδομένα εισόδου, χωρίς να υπάρχει η αντίστοιχη συμπεριφορά για τα δεδομένα εξόδου. Αυτή η συμπεριφορά φαίνεται να μπερδεύει όλο και περισσότερο κάθε φορά το δίκτυο, με αποτέλεσμα το σφάλμα να αυξάνει διαρκώς με την αύξηση του αριθμού των μοτίβων και μάλιστα για τα 100,000 και 600,000 μοτίβα δεν πέφτει ποτέ κάτω από 90%! , Κρίθηκε ότι σαφέστατα με τη συγκεκριμένη διαμόρφωση των δεδομένων δεν είναι δυνατός ο καθορισμός της μορφολογίας ενός γαλαξία μέσω της πρόβλεψης του $\mathbf{BT_r}$ μιας και το ΤΝΔ αποτυγχάνει να ξεχωρίσει κάποιες τάσεις στα δεδομένα. Για το λόγο αυτό διακόπηκε η εκπαίδευση του ΤΝΔ και προτείνεται να εξετασθεί κάποια διαφορετική διαμόρφωση των δεδομένων με την προσθήκη είτε

παραπάνω είτε διαφορετικών παραμέτρων, έτσι ώστε να καταστεί δυνατή η επιτυχής εκπαίδευση του δικτύου. Αυτό είναι ένα θέμα το οποίο πρέπει να τεθεί προς περαιτέρω διερεύνηση.

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΔΙΕΡΕΥΝΗΣΗ

Στα πλαίσια της εργασίας αυτής ασχοληθήκαμε με το θέμα των Μεγάλων Δεδομένων και τη δημιουργία ενός ολοκληρωμένου υπολογιστικού πλαισίου επεξεργασίας και ανάλυσής τους. Το πλαίσιο αυτό αποτελείται από δύο διακριτά μέρη: ένα κομμάτι, το οποίο θα μπορεί να ανακτά τα δεδομένα και να τα φέρνει σε κάποια μορφή στην οποία να μπορούν να επεξεργαστούν. Και ένα δεύτερο κομμάτι, το οποίο θα είναι σε θέση να μελετά και να αναλύει τα επιλεχθέντα δεδομένα και να εξάγει συμπεράσματα. Έπειτα από εκτενή βιβλιογραφική ανασκόπηση, αποφασίσαμε ότι για την προεπεξεργασία των Μεγάλων Δεδομένων το πλέον κατάλληλο πλαίσιο είναι το Hadoop ενώ για την ανάλυσή τους τα ΤΝΔ. Το Hadoop εγκαταστάθηκε με επιτυχία σε ένα επεξεργαστή με το λειτουργικό σύστημα Unix Ubuntu (για ψευδο-κατανεμημένη λειτουργία) και ο σειριακός κώδικας των ΤΝΔ (σε Fortran77) μπορεί να μεταφερθεί και να τρέξει σε οποιοδήποτε μηχάνημα έχει μεταγλωττιστή της Fortran77. Όλες οι προσομοιώσεις των ΤΝΔ έγιναν στη συστοιχία υπολογιστών Metropolis του Τμήματος Φυσικής του Πανεπιστημίου Κρήτης. Σκοπός της εργασίας αυτής είναι να αναδειχθούν οι δυνατότητες του υπολογιστικού αυτού πλαισίου στην αντιμετώπιση προβλημάτων όπου εμπλέκονται Μεγάλα Δεδομένα. Επιλέχθηκαν δύο προβλήματα, ένα από το χώρο των αερομεταφορών και ένα από το χώρο της αστροφυσικής.

Για το πρόβλημα των αερομεταφορών αξιοποιήθηκε ολόκληρο το υπολογιστικό πλαίσιο, όπου από ένα μεγάλο όγκο δεδομένων από όλα τα αεροδρόμια των ΗΠΑ ανακτήθηκαν δεδομένα για το αεροδρόμιο του Phoenix της Arizona και αυτά μελετήθηκαν και αναλύθηκαν, έτσι ώστε να διερευνηθεί κατά πόσο είναι δυνατόν να προβλεφθεί από τη μέση καθυστέρηση (από 15 έως 93 λεπτά) ανά πτήση για κάποια ημέρα, ο χρόνος ομαλοποίησης (από 1 έως 6 ημέρες) της λειτουργίας ενός αεροδρομίου (όπου ομαλοποίηση σημαίνει ο μέσος χρόνος καθυστέρησης ανά πτήση να είναι μικρότερος των 15 λεπτών). Το πρόβλημα, όπως αποδεικνύεται, είναι σαφώς μη γραμμικό και η επιτυχής αντιμετώπισή του είναι πολύ σημαντική γιατί μπορεί να βοηθήσει στην καλύτερη οργάνωση των πόρων του αεροδρομίου (ανθρώπινο δυναμικό, εξοπλισμός ανάλογα με τις καθυστερήσεις κάθε ημέρας, κλπ.) για τη

βελτιστοποίηση της λειτουργίας του. Το Hadoop μπόρεσε να διαχειριστεί επιτυχώς τα ανακτηθέντα αρχεία και να αναζητήσει ανάμεσα από δισεκατομμύρια δεδομένα τα ζητούμενα στοιχεία σε πολύ σύντομο χρονικό διάστημα. Το ΤΝΔ ανταπεξήλθε επιτυχώς στην διαδικασία εύρεσης συσχετισμών ανάμεσα στους μέσους χρόνους καθυστέρησης και τον απαιτούμενο χρόνο ομαλοποίησης. Ο αριθμός των ανακτηθέντων δεδομένων ήταν σχετικά μικρός (419 μοτίβα συνολικά) και γι αυτό υπήρξε και αστοχία πρόβλεψης της τάξης του 10-20%. Πιστεύουμε ότι με περισσότερα δεδομένα και με την εισαγωγή και άλλων μεταβλητών εισόδου στο δίκτυο (πχ. κάποιος δείκτης που θα συσχετίζει την καθυστέρηση με κάποιο αίτιο), θα βελτιώσει κατά πολύ την ικανότητα του ΤΝΔ για επιτυχή πρόβλεψη.

Για το πρόβλημα της Αστροφυσικής αξιοποιήθηκε μόνο το δεύτερο κομμάτι του υλοποιηθέντος υπολογιστικού πλαισίου (ΤΝΔ) για την ανάλυση των αποτελεσμάτων. Το ερώτημα που τέθηκε είναι κατά πόσον από τα χρώματα γαλαξιών είναι δυνατή η πρόβλεψη της μορφολογίας τους. Εδώ, σε αντίθεση με την προηγούμενη περίπτωση των αερομεταφορών, ο όγκος δεδομένων ήταν πολύ μεγάλος (άνω των 600,000 μοτίβων), αλλά το πρόβλημα που προκύπτει είναι ότι οι περισσότερες τιμές των δεδομένων εισόδου στο ΤΝΔ (της τάξης του 90%) είναι κατανεμημένες σε μια ζώνη τιμών με πάρα πολύ μικρό εύρος, ενώ δεν ισχύει κάτι αντίστοιχο για τα δεδομένα εξόδου. Συνεπώς, η εκπαίδευση του ΤΝΔ καθίσταται προβληματική με συστηματικά μεγάλα σφάλματα (άνω του 60%), χωρίς το δίκτυο να είναι σε θέση να αναγνωρίσει τάσεις ανάμεσα στα δεδομένα. Για το λόγο αυτό, το ΤΝΔ αποτυγχάνει πλήρως στην πρόβλεψη της μορφολογίας ενός γαλαξία μόνο από τα χρώματα και πρέπει να συμπληρωθεί με επιπλέον πληροφορία στην είσοδό του (πληροφορία από άλλα περισσότερα φίλτρα ίσως) για να μπορέσει να έχει καλύτερη απόδοση.

Ένα σημαντικό πρόβλημα είναι το ότι ο κώδικας των ΤΝΔ δεν είναι παράλληλος. Έτσι, οι χρόνοι που απαιτούνται για την εκπαίδευση του δικτύου με μεγαλο αριθμό δεδομένων σε συνδυασμό με μεγάλο αριθμό κρυμμένων επιπέδων με πολλούς κόμβους ανά επίπεδο, είναι πολύ μεγάλοι, πράγμα που σημαίνει ότι περιορίζεται καταυτόν τον τρόπο ο αριθμός δοκιμών που μπορούμε να εκτελέσουμε για να βρούμε τη βέλτιστη μορφοποίηση του δικτύου που να δίνει το μικρότερο σφάλμα (πρβλ. τις δοκιμές στα Κεφάλαια 3 και 4). Συνεπώς, κρίνεται απαραίτητο να παραλληλοποιηθεί άμεσα ο κώδικας των ΤΝΔ και ήδη έχουμε αρχίσει να επεξεργαζόμαστε τα βήματα που θα ακολουθήσουμε προς αυτή την κατεύθυνση.

Παράρτημα Α: Εγκατάσταση Hadoop και Hive

Προτού ξεκινήσει η διαδικασία εγκατάστασης του Hadoop, πρέπει να βεβαιωθούμε ότι υπάρχει ένα περιβάλλον Linux ρυθμισμένο και έτοιμο για χρήση. Η διαδικασία που θα περιγραφεί προυποθέτει ότι έχουμε εγκατεστημένη την έκδοση του Ubuntu 14.04 LTS στο μηχάνημα μας, είτε σε διαμόρφωση με διπλή επιλογή λογισμικού κατά την εκκίνηση, είτε χρησιμοποιώντας μια εικονική μηχανή. Προτιμήθηκε η χρήση του Ubuntu Desktop, αφού είναι πιο ελαφρύ. Η έκδοση του Ubuntu που εργαστήκαμε ήταν η Ubuntu x64 Desktop 14.04 LTS. Αρχικά βεβαιωνόμαστε ότι το σύστημά μας είναι πλήρως ενημερωμένο, χρησιμοποιώντας την ακόλουθη εντολή:

```
sudo apt-get update && sudo apt-get upgrade
```

Προκειμένου να διασφαλίσουμε τις υπηρεσίες του Hadoop, πρέπει να εξασφαλίσουμε ότι το Hadoop θα λειτουργεί σε μια συγκεκριμένη ομάδα σε ένα συγκεκριμένο χρήστη. Αυτός ο χρήστης θα είναι σε θέση να εκκινήσει συνδέσεις SSH (Secure Shell) σε άλλους κόμβους σε ένα σύμπλεγμα, αλλά δεν έχει την ικανότητα για πρόσβαση ούτως ώστε να μπορεί να βλάψει το λειτουργικό σύστημα στο οποίο εκτελείται η υπηρεσία.

Αυτό διασφαλίζει επίσης ότι η εγκατάσταση του Hadoop είναι διαχωρισμένη από άλλες εφαρμογές λογισμικού και άλλους λογαριασμούς χρήστη που τρέχουν στο ίδιο μηχάνημα και αυτό βοηθά στην οργάνωση και στην ασφαλή συντήρηση του μηχανήματος. Για τη δημιουργία της ομάδας του νέου χρήστη Hadoop η εντολή που χρησιμοποιείται είναι:

```
sudo addgroup hadoop
```

Για την προσθήκη του νέου χρήστη η εντολή που χρησιμοποιείται είναι:

```
sudo adduser --ingroup hadoop hduser
```

Στην συνέχεια προσθέτουμε τον χρήστη hduser στην ομάδα sudo ούτως ώστε να μπορεί να εκτελεί sudo εντολές, χρησιμοποιώντας την ακόλουθη εντολή:

```
sudo adduser hduser sudo
```

Ακολούθως, γίνεται επανεκκίνηση του μηχανήματος και σύνδεση στο νέο χρήστη. Στην περιοχή αυτή του νέου χρήστη θα γίνει η εγκατάσταση του Hadoop. Πρωτού όμως γίνει η εγκατάσταση του Hadoop απαιτείται η εγκατάσταση της JAVA, του rsynch και του SSH. Αφού το Hadoop είναι γραμμένο σε

JAVA και χρησιμοποιεί SSH για την απομακρυσμένη διαχείριση των κόμβων στη συστοιχία, ενώ το rsynch δημιουργεί τα αντίγραφα ασφαλείας.

Για την εγκατάσταση της JAVA, η εντολή που χρησιμοποιείται είναι:

```
sudo apt-get install openjdk-8-jdk
```

Για την εγκατάσταση του rsynch, η εντολή που χρησιμοποιείται είναι:

```
sudo apt-get install rsync
```

Στην συνέχεια εγκαθιστούμε το SSH, με την ακόλουθη εντολή:

```
sudo apt-get install ssh
```

Ακολούθως, γίνεται δημιουργία SSH κλειδιού για το χρήστη hduser και αντιγραφή του στο αρχείο authorized_keys, ούτως ώστε να μην χρειάζεται η εισαγωγή του κωδικού ασφαλείας κάθε φορά που το Hadoop αλληλεπιδρά με τους κόμβους της συστοιχίας. Οι εντολές που χρησιμοποιούμαι εδώ είναι:

```
ssh-keygen -t rsa -P ""
```

```
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Αφού έχουν εγκατασταθεί τα προηγούμενα με επιτυχία, μπορεί να γίνει η εγκατάσταση του Hadoop. Η εγκατάσταση του θα γίνει στη τοποθεσία Downloads (αυτό είναι προκαθορισμένο απο το σύστημα, όμως ο χρήστης έχει την δυνατότητα να το αλλάξει και η εγκατάσταση να πραγματοποιηθεί εκεί όπου επιθυμεί φτάνει να είναι συνεπής και να μην δημιουργηθεί σύγχυση).

Η μετακίνηση απο τη τοποθεσία HOME στη τοποθεσία Downloads γίνεται με τη χρήση της ακόλουθης εντολής:

```
cd ~/Downloads
```

Για την εγκατάσταση του Hadoop, αρχικά γίνεται λήψη της έκδοσης Hadoop-2.6.4 απο την επίσημη ιστοσελίδα του Apache Software Foundation. Οι εντολές που χρησιμοποιούνται για την εγκατάσταση του Hadoop είναι:

```
wget http://www-us.apache.org/dist/hadoop/common/hadoop-2.6.4/hadoop-2.6.4.tar.gz
```

```
tar -xvzf hadoop-2.6.4.tar.gz
```

```
sudo mv hadoop-2.6.4 /usr/local/hadoop
```

```
sudo chown -R hduser:hadoop /usr/local/hadoop
```

Με τις εντολές αυτές αρχικά γίνεται η λήψη του αρχείου όπως προείπαμε, στη συνέχεια γίνεται η αποσυμπίεση του αρχείου hadoop-2.6.4.tar.gz, μεταφέρεται στη τοποθεσία όπου θα διατηρήσουμε όλες τις υπηρεσίες Hadoop και στη συνέχεια ορίζονται οι «άδειες πρόσβασης» σε αυτό.

Στην συνέχεια δημιουργούνται στη προσορινή τοποθεσία του hadoop τα αρχεία namenode και datanode και αλλάζουμε τις «άδειες πρόσβασης» της προσωρινής τοποθεσίας του Hadoop, χρησιμοποιώντας τις ακόλουθες εντολές:

```
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
```

```
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
```

```
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
```

Προκειμένου να διασφαλίσουμε ότι όλα θα εκτελεστούν σωστά, θα θέσουμε κάποιες μεταβλητές περιβάλλοντος έτσι ώστε το Hadoop να εκτελεστεί στο σωστό περιβάλλον. Έτσι καταχωρούμε την ακόλουθη εντολή για να ανοίξουμε ένα πρόγραμμα επεξεργασίας κειμένου με το προφίλ του χρήστη hadoop για να αλλάξουμε τις μεταβλητές περιβάλλοντος:

```
gedit /home/hadoop/.bashrc
```

Προσθέτουμε τις ακόλουθες γραμμές στο τέλος του αρχείου αυτού:

```
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
#HADOOP VARIABLES END
```

Και εκτελούμε την ακόλουθη εντολή για να φορτώσουμε ξάνα το αρχείο αυτό:

```
source /home/hadoop/.bashrc
```

Το προτελευταίο βήμα για τη ρύθμιση του Hadoop ως ψευδο-κατανεμημένη συστοιχία είναι η επεξεργασία των αρχείων ρυθμίσεων για το περιβάλλον Hadoop, την τοποθεσία HDFS το YARN και το MapReduce. Αυτό συνεπάγεται ως επί το πλείστον την επεξεργασία των αρχείων διαμόρφωσης.

Η επεξεργασία του αρχείου hadoop-env.sh πραγματοποιείται χρησιμοποιώντας την ακόλουθη εντολή γαι να ανοίξουμε ένα πρόγραμμα επεξεργασίας κειμένου με το προφίλ του χρήστη hadoop για να ορίσουμε το JAVA_HOME στο αρχείο hadoop-env.sh:

```
gedit /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

Αντικαθιστούμε τη γραμμή:

```
export JAVA_HOME=${JAVA_HOME}
```

με την ακόλουθη γραμμή:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Αποθηκεύουμε την αλλαγή αυτή και βγαίνουμε από το αρχείο αυτό.

Στην συνέχεια επεξεργαζόμαστε το αρχείο διαμόρφωσης core-site.xml. Η επεξεργασία του αρχείου core-site.xml πραγματοποιείται χρησιμοποιώντας την ακόλουθη εντολή γαι να ανοίξουμε ένα πρόγραμμα επεξεργασίας κειμένου:

```
gedit /usr/local/hadoop/etc/hadoop/core-site.xml
```

Προσθέτοντας τις ακόλουθες γραμμές μεταξύ των ετικετών <configuration> , <property>:

```
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
```

Ακολούθως διαμορφώνουμε το HDFS, επεξεργαζόμενοι το αρχείο hdfs-site.xml. Η επεξεργασία του αρχείου hdfs-site.xml πραγματοποιείται χρησιμοποιώντας την ακόλουθη εντολή γαι να ανοίξουμε ένα πρόγραμμα επεξεργασίας κειμένου:

```
gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

Προσθέτοντας τις ακόλουθες γραμμές μεταξύ των ετικετών <configuration> , <property>:

```
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
```

Ακολούθως διαμορφώνουμε το YARN, επεξεργαζόμενοι το αρχείο yarn-site.xml. Η επεξεργασία του αρχείου yarn-site.xml πραγματοποιείται χρησιμοποιώντας την ακόλουθη εντολή γαι να ανοίξουμε ένα πρόγραμμα επεξεργασίας κειμένου:

```
gedit /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

Προσθέτοντας τις ακόλουθες γραμμές μεταξύ των ετικετών <configuration> , <property>:

```
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
```

Ακολούθως επεξεργαζόμαστε τη διαμόρφωση του MapReduce. Αρχικά μετονομάζουμε το αρχείο mapred-site.xml.template σε mapred-site.xml και επεξεργάζομαστε το mapred-site.xml.

```
mv /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

Η επεξεργασία του αρχείου mapred-site.xml πραγματοποιείται χρησιμοποιώντας την ακόλουθη εντολή γαι να ανοίξουμε ένα πρόγραμμα επεξεργασίας κειμένου:

```
gedit /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

Προσθέτοντας τις ακόλουθες γραμμές μεταξύ των ετικετών <configuration> , <property>:

```
<name>mapreduce.framework.name</name>
<value>yarn</value>
```

Έχοντας επεξεργαστεί αυτά τα αρχεία, το Hadoop διαμορφώνεται πλήρως ως ψευδο-κατανεμημένο περιβάλλον.

Το τελευταίο βήμα προτού ενεργοποιήσουμε το Hadoop είναι να μορφοποιήσουμε το namenode. Το namenode είναι υπεύθυνο για το HDFS, το κατανεμημένο σύστημα αρχείων. Αυτό γίνεται χρησιμοποιόντας την εντολή:

```
hdfs namenode -format
```

Σε αυτό το σημείο μπορούμε να ξεκινήσουμε και να εκτελέσουμε τα «daemons» του Hadoop, χρησιμοποιώντας τις ακόλουθες εντολές για τα «hdfs daemons» και για τα "mapreduce daemons"

```
start-dfs.sh
start-yarn.sh
```

Για να εκτελέσουμε το Hadoop, η εντολή που χρησιμοποιούμε είναι:

```
start-dfs.sh
ή
start-yarn.sh
```

Για να τερματίσουμε το Hadoop, η εντολή που χρησιμοποιούμε είναι:

```
stop-all.sh
stop-dfs.sh
stop-yarn.sh
```

Για την εγκατάσταση του Hive, αρχικά γίνεται λήψη της έκδοσης Hive-1.2.1 απο την επίσημη ιστοσελίδα του Apache Software Foundation. Η λήψη του αρχείου αυτού γίνεται στην τοποθεσία Downloads. Αν δεν είμαστε ήδη στην τοποθεσία αυτή πρέπει να μετακινηθούμε εκέι. Οι εντολές που χρησιμοποιούνται για την εγκατάσταση του Hive είναι:

```
cd ~/Downloads
```

```
wget http://www-us.apache.org/dist/hive/stable/apache-hive-1.2.1-bin.tar.gz
```

```
tar -xvzf apache-hive-1.2.1-bin.tar.gz
```

```
sudo mv apache-hive-1.2.1-bin /usr/local/hive
```

```
sudo chown -R hduser:hadoop /usr/local/hive
```

Με τις εντολές αυτές αρχικά μεταφερόμαστε στη τοποθεσία Downloads, εκεί γίνεται η λήψη του αρχείου όπως προείπαμε, στη συνέχεια γίνεται η αποσυμπίεση του hive-1.2.1-bin.tar.gz, μεταφέρεται στη τοποθεσία όπου θα διατηρήσουμε όλες τις υπηρεσίες Hive και στη συνέχεια ορίζονται οι «άδειες πρόσβασης» σε αυτό.

Για να εκτελεστούν όλα σωστά, θα επεξεργαστούμε το .bashrc έτσι ώστε το Hive να εκτελεστεί στο σωστό περιβάλλον. Έτσι καταχωρούμε την ακόλουθη εντολή για να ανοίξουμε ένα πρόγραμμα επεξεργασίας κειμένου για να αλλάξουμε τις μεταβλητές περιβάλλοντος:

```
gedit ~/.bashrc
```

Προσθέτουμε τις ακόλουθες γραμμές στο τέλος του αρχείου αυτού:

```
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
export HADOOP_USER_CLASSPATH_FIRST=true
```

Και εκτελούμε την ακόλουθη εντολή για να φορτώσουμε πάλι το αρχείο αυτό:

```
source ~/.bashrc
```

Ακολούθως δημιουργούμε τα ακόλουθα αρχεία tmp, warehouse για το Hive στο HDFS, χρησιμοποιώντας τις πιο κάτω εντολές:

```
hadoop fs -mkdir /tmp
```

```
hadoop fs -mkdir -p /user/hive/warehouse
```

```
hadoop fs -chmod g+w /tmp
```

```
hadoop fs -chmod g+w /user/hive/warehouse
```

Ακολούθως επεξεργαζόμαστε τη διαμόρφωση του hive. Αρχικά μετονομάζουμε το αρχείο hive-env.sh.template σε hive-env.sh και επεξεργάζομαστε το hive-env.sh.

```
mv /usr/local/hive/conf/hive-env.sh.template /usr/local/hive/conf/hive-env.sh
```

Η επεξεργασία του αρχείου hive-env.sh πραγματοποιείται χρησιμοποιώντας την ακόλουθη εντολή γαι να ανοίξουμε ένα πρόγραμμα επεξεργασίας κειμένου:

```
gedit /usr/local/hive/conf/hive-env.sh
```

Προσθέτοντας την ακόλουθη γραμμή στο τέλος του αρχείου:

```
export HADOOP_HOME=/usr/local/hadoop
```

Για να εκτελέσουμε το Hive, η εντολή που χρησιμοποιούμε είναι:

```
hive
```

Για να τερματίσουμε το Hive, η εντολή που χρησιμοποιούμε είναι:

```
quit
ή
exit
```

Σε αυτό το σημείο έχουμε μια πλήρως διαμορφωμένη εγκατάσταση Hadoop, έτοιμη για εκτέλεση σε κατάσταση ψευδο-κατανεμημένης λειτουργίας στο Ubuntu με HDFS, MapReduce, YARN και Hive.

Παράρτημα Β: Κώδικας (εκπαίδευση) σε Fortran77 ΤΝΔ πρόσθιας τροφοδότησης - οπίσθιας διάδοσης

```
      program neural_net
c  Original code by Agis Spentzos 2005
c  Subsequent modifications and optimization by Nicholas Christakis, 2010

      parameter(imax1=100000,imax2=64,ilay=3)
      integer epoch,istart
      integer i,j,k,p,np,list(imax1)
      integer numpattern,numinput,numhidden,numoutput,nml
      real*8 input(0:imax1,0:imax2),target(0:imax1,0:imax2)
      real*8 output(0:imax1,0:imax2)
      real*8 hidden(ilay,imax1,imax2)
      real*8 sum(0:imax1,0:imax2),summ(0:imax1)
      real*8 weightIH(0:imax2,0:imax2)
      real*8 weightHH(ilay,0:imax2,0:imax2)
      real*8 weightHO(0:imax2,0:imax2)
      real*8 deltaO(0:imax1),deltaH(ilay,0:imax1)
      real*8 deltaweightIH(0:imax1,0:imax2)
      real*8 deltaweightHO(0:imax1,0:imax2)
      real*8 deltaweightHH(ilay,0:imax1,0:imax2),inp(imax1,imax2)
      real*8 error,eta,alpha,smallwt,error1,errormax,r
      character*50 patfile
      character*13 wfile

11      format(60(f12.6,2X))
12      format(A50)
13      format(A1)

      smallwt=0.5
      iepoch=1
      error0=0.0

      open(1,file='input1.txt')
      read(1,*)patfile
      read(1,*)istart
      read(1,*)numinput
      read(1,*)nml
      read(1,*)numhidden
      read(1,*)numoutput
      read(1,*)ifresh
      read(1,*)cerror
```

```
      read(1,*)eta
      read(1,*)alpha
      read(1,*)wfile
      close(1)

c      intitialise input_to_hidden weights
      do j=1,numhidden
            do i=1,numinput
                  deltaweightIH(i,j)=0.0
                  weightIH(i,j)=2.0*(rand(0))*smallwt
            enddo
      enddo
c      intitialise hidden_to_output weights
      do k=1,numoutput
            do j=1,numhidden
                  deltaweightHO(j,k)=0.0
                  weightHO(j,k)=2.0*(rand(0))*smallwt
            enddo
      enddo
c      initialise hidden1 to hidden2 weights
      do il=1,nml
      do i=1,numhidden
            do j=1,numhidden
                  deltaweightHH(il,j,i)=0.0
                  weightHH(il,j,i)=2.0*(rand(0))*smallwt
            enddo
      enddo
      enddo

      if (istart.eq.1) then
            write(*,*) '# Hot starting...'
            open(2,file='weights.txt')
            read(2,*)iepoch,error0
            read(2,*)numinput,nml,numhidden,numoutput
            do i=0,numinput
                  do j=0,numhidden
                        read(2,*)weightIH(i,j)
                  enddo
            enddo
            do il=1,nml
            do i=0,numhidden
                  do j=0,numhidden
                        read(2,*)weightHH(il,i,j)
                  enddo
            enddo
            enddo
            do i=0,numhidden
```

```
                do k=0,numoutput
                    read(2,*)weightHO(i,k)
                enddo
            enddo
            write(*,*) '# weight functions read'
            close(2)
        else
            write(*,*) '# Cold starting'
        endif

c       calculate number of weight functions
        norm1=(numinput*numhidden)+(nml-1)*(numhidden**2)+
     &      numhidden*numoutput
        write(*,*) '# ',norm1,' total weight components'
        write(*,*) '# ',(nml-1)*(numhidden*numhidden),
     &      ' weight components in perceptron'
c       the above line compiles but causes the code to output jargon!!!!!


        open(1,file=patfile)
        i=1
10      list(i)=i
c       read(1,*,END=666)input(i,1),input(i,2),input(i,3),target(i,1)
          read(1,*,END=666)(inp(i,j), j=1,numinput+numoutput)
          do j=1,numinput
              input(i,j)=inp(i,j)
          enddo
          do j=1,numoutput
              target(i,j)=inp(i,j+numinput)
          enddo
        i=i+1
        goto 10

666       continue
        numpattern=i-1
        write(*,*) '# ',numpattern,' patterns'
        close(1)
        do epoch=iepoch,50000000
            do i=1,numpattern-1
                number=int(rand(0)*float(2))
                if (number.eq.1) then
                    iii=list(i)
                    list(i)=list(i+1)
                    list(i+1)=iii
                endif
            enddo
            error=0.0
```

```
            errormax=-10.0e+6
            do np=1,numpattern
                p=list(np)

c       propagate forward from input to hidden1
                do j=1,numhidden
                    sum(p,j)=weightIH(0,j)
                    do i=1,numinput
                      sum(p,j)=sum(p,j)+
     &                        input(p,i)*weightIH(i,j)
                    enddo
                    hidden(1,p,j)=1.0/(1.0+exp(-sum(p,j)))
                enddo

c       propagate forward from hidden-1 to hidden-n
                do il=1,nml-1
                do j=1,numhidden
                   sum(p,j)=weightHH(il,0,j)
                   do i=1,numhidden
                     sum(p,j)=sum(p,j)+
     &                 hidden(il,p,i)*weightHH(il,i,j)
                   enddo
                   hidden(il+1,p,j)=1.0/(1.0+exp(-sum(p,j)))
                enddo
                enddo

c       propagate forward from hidden-n to output, calculate error & remedy per output node
                do k=1,numoutput
                    sum(p,k)=weightHO(0,k)
                    do j=1,numhidden
                      sum(p,k)=sum(p,k)+
     &                  hidden(nml,p,j)*weightHO(j,k)
                    enddo
                    output(p,k)=1.0/(1.0+exp(-sum(p,k)))
                    if (epoch.eq.1) error0=error
                                r=target(p,k)-output(p,k)
                                r=abs(r)
c                               r=0.5*r*r
                                error=error+r
                      errormax=dmax1(errormax,r)
                    if (errormax.eq.r) i1=p
                    deltaO(k)=(target(p,k)-output(p,k))*
     &                 (output(p,k)*(1.0-output(p,k)))
                enddo
                if (epoch.eq.1) error0=error
                error1=error/error0
```

```
c       backpropagate from output to hidden-n, use output remedy to calculate remedy per hidden-n node
              do j=1,numhidden
                 summ(j)=0.0
                 do k=1,numoutput
                   summ(j)=summ(j)+
     &               weightHO(j,k)*deltaO(k)
                 enddo
                 deltaH(nml,j)=summ(j)*hidden(nml,p,j)*
     &               (1.0-hidden(nml,p,j))
              enddo

c       backpropagate from hidden-n to hidden1, use hidden-n remedy to calculate remedy per hidden-n-1 node
              do il=nml-1,1,-1
              do j=1,numhidden
                 summ(j)=0.0
                 do jk=1,numhidden
                   summ(j)=summ(j)+
     &               weightHH(il,j,jk)*deltaH(il+1,jk)
                 enddo
                 deltaH(il,j)=summ(j)*
     &             hidden(il,p,j)*(1.0-hidden(il,p,j))
              enddo
              enddo

c       backpropagate from hidden1 to input & use hidden1 remedy to re-calculate i-h1 weights
              do j=1,numhidden
                deltaweightIH(0,j)=eta*deltaH(1,j)+
     &                    alpha*deltaweightIH(0,j)
                weightIH(0,j)=weightIH(0,j)+
     &                  deltaweightIH(0,j)
                do i=1,numinput
                  deltaweightIH(i,j)=
     &              eta*input(p,i)*deltaH(1,j)+alpha*
     &              deltaweightIH(i,j)
                  weightIH(i,j)=weightIH(i,j)+
     &                   deltaweightIH(i,j)
                enddo
              enddo

c       propagate from hidden1 to hidden_n & use hidden2 remedy to recalculate h_n-h_n+1 weights
              do il=1,nml-1
              do j=1,numhidden
                 deltaweightHH(il,0,j)=eta*deltaH(il+1,j)
     &             +alpha*deltaweightHH(il,0,j)
```

```
                    weightHH(il,0,j)=weightHH(il,0,j)+
     &                deltaweightHH(il,0,j)
                    do i=1,numhidden
                      deltaweightHH(il,i,j)=
     &                  eta*hidden(il,p,i)*deltaH(il+1,j)
     &                  +alpha*deltaweightHH(il,i,j)
                      weightHH(il,i,j)=weightHH(il,i,j)+
     &                  deltaweightHH(il,i,j)
                    enddo
                enddo
                enddo

c       propagate from hidden3 to output & use output remedy to re-calculate h3-o weights
                do k=1,numoutput
                    deltaweightHO(0,k)=
     &                eta*deltaO(k)+alpha*deltaweightHO(0,k)
                    weightHO(0,k)=weightHO(0,k)+
     &                deltaweightHO(0,k)
                    do j=1,numhidden
                    deltaweightHO(j,k)=eta*hidden(nml,p,j)*
     &                deltaO(k)+alpha*deltaweightHO(j,k)
                    weightHO(j,k)=weightHO(j,k)+
     &                          deltaweightHO(j,k)
                    enddo
                enddo
            enddo
            if ((epoch/ifresh)*ifresh.eq.epoch) then
                call weights(epoch,error0,numinput,nml,
     &            numhidden,numoutput,
     &                weightIH,weightHH,weightHO,wfile)
                write(*,*) epoch,error1,errormax,i1

            endif
            if (error1.lt.cerror) goto 1000
        enddo

1000        continue
        call weights(epoch,error0,numinput,nml,numhidden,numoutput,
     &          weightIH,weightHH,weightHO,wfile)
        write(*,*) epoch,error1,errormax
        write(*,*) 'The End for authentic'
        stop
        end

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        subroutine weights(epoch,error0,ninput,nl,nhidden,noutput,
     &          wweightIH,wweightHH,wweightHO,wwfile)
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      parameter (imax3=64,ilay=3)
      integer epoch
      real*8 wweightHH(ilay,0:imax3,0:imax3)
      real*8 wweightHO(0:imax3,0:imax3),wweightIH(0:imax3,0:imax3)
      character*13 wwfile
11       format(6(f12.6,2X))

c       weights
      open(1,file=wwfile)
      write(1,*)epoch,error0
      write(1,*)ninput,nl,nhidden,noutput
      do i=0,ninput
            do j=0,nhidden
                  write(1,*)wweightIH(i,j)
            enddo
      enddo
      do il=1,nl
      do i=0,nhidden
            do j=0,nhidden
                  write(1,*)wweightHH(il,i,j)
            enddo
      enddo
      enddo
      do i=0,nhidden
            do k=0,noutput
                  write(1,*)wweightHO(i,k)
            enddo
      enddo
      close(1)
      return
      end
```

Βιβλιογραφία

ACI Airport Key Performance Indicators, 2015, *Year to data Passenger Traffic*, Διαθέσιμο στο διαδίκτυο: http://www.aci.aero/Data-Centre/Monthly-Traffic-Data/Passenger-Summary/Year-to-date *τελευταία ενημέρωση: 23/06/2015*

Aisling R., Kenneth J. B., 1999, An assessment of the capacity and congestion levels at European airports, European Regional Science Association, ERSA conference papers ersa 99, pages 241

Apache, 2008, Welcome to ApacheTM Hadoop!, Διαθέσιμο στον δικτυακό τόπο: https://hadoop.apache.org, τελευταία πρόσβαση στις 28/03/2017

Apache, 2014, Welcome to ApacheTM Hadoop!, Διαθέσιμο στον δικτυακό τόπο: https://hadoop.apache.org, τελευταία πρόσβαση στις 28/03/2017

Apache, 2016, Apache Zookeper, Διαθέσιμο στον δικτυακό τόπο: https://cwiki.apache.org/confluence/display/ZOOKEEPER/Index , *τελευταία ενημέρωση: 26/10/2016*

Apache, 2016, Welcome to Apache Pig!, Διαθέσιμο στον δικτυακό τόπο:https://pig.apache.org/, τελευταία ενημέρωση: 06/08/2016

Apache, 2017, *Apache Hive*, Διαθέσιμο στον δικτυακό τόπο: *https://cwiki.apache.org/confluence/display/Hive/Home*, τελευταία ενημέρωση: 13/01/2017

Apache, 2017, Apache Oozie Workflow Scheduler for Hadoop, Διαθέσιμο στον δικτυακό τόπο: http://oozie.apache.org/, τελευταία ενημέρωση: 20/03/2017

Ardö J., Pilesjö P., Skidmore A., 1997, Neural networks, multitemporal Landsat Thematic Mapper data and topographic data to classify forest damages in the Czech Republic, *Canadian Journal of Remote Sensing, 23, p. 217–229*

Ashford N., Wright P. H., 1992, *Airport Engineering*, John Wiley & Sons Inc.

Barzilai G. et al., 1993, *The Gulf Crisis and Its Global Aftermath*, Routledge, ISBN: 0-415-08002-9

Baum E. B., Haussler D., 1989, What size net gives valid generalization? In Advances in Neural Information Processing Systems I, edited by D. S. Touretzky, San Mateo:Morgan Kaufmann, p. 81–90

Bingelli B. et al., 1987, Studies of the Virgo cluster. VI- Morphological an kinematical structure of the Virgo cluster, *Astronomical Journal, 94, p. 251-277*

Bischof H., Schneider W., Pinz A. J., 1992, Multispectral classification of Landsat-images using neural networks, *IEEE Transactions on Geoscience and Remote Sensing*, 30, p. 482–489

Bishop C. M., 1995, *Neural Networks for pattern recognition,* Oxford University Press

Chen C. L. P. , Zhang C. Y. , 2014, Elsevier Science Publishing Company Inc., *275:314-347*

Christakis N., Barbaris V., A. Spentzos, 2011, A New Approach in Financial Modelling with the Aid of Artificial Neural Networks, *Journal of Algorithms & Computational Technology, 5(3), p.513–530*

Combes F. and Buta R., 1996, Galactic Rings*, Fundamentals of Cosmic Physics, 17, p. 95*

Cox M., Ellsworth D., 1997, Application-controlled demand for out-of-core visualization, *Technical report*

DeGraf et al., 2007, A Galaxy in Transition: Structure, Globular Clusters, and Distance of the Star-Forming S0 Galaxy NGC 1533 in Dorado, *The Astrophysicsl Journal, 671,2, p. 1624-1639*

de Vaucouleurs G., 1959, *Classification and Morphology of External Galaxies*, Handbuch der Physik, 53, p. 275

Devlin B. et al. 2012, Big Data comes of age, *EMA and 9sight Consulting Research Report*

Dou L., Lee D.,Johnson J., Gaier E., Kostiuk P., 1999, *Modeling air traffic management technologies with a queuing network model of the national airspace system, No. NAS 1.26: 208988, National Aeronautics and Space Administration, Langley Research Center*

Doug Laney, *2001, Application Delivery Strategies,META Group,* 949, Stanford

Eberhart R. C., Dobbins R. W., 1990, *Neural Network PC Tools – A Practical Guide*, Academic Press

Federal Aviation Administration, 2014, *Calendar Year 2014 Passenger Boardings at Commercial Service Airports, Διαθέσιμο στο διαδίκτυο: https://www.faa.gov/airports/planning_capacity/passenger_allcargo_stats/passenger/media/cy14-commercial-service-enplanements.pdf,* τελευταία ενημέρωση: 22/09/2015

Federal Aviation Administration, 2015, *Calendar Year 2015 Revenue Enplanements at Commercial Service Airports, Διαθέσιμο στο διαδίκτυο: https://www.faa.gov/airports/planning_capacity/ passenger_allcargo_stats/passenger/*,τελευταία ενημέρωση: 05/07/2015

Fernandéz A., Río S., Ramírez-Gallego S., 2015, *Big Data: Algorithms for Data Preprocessing, Computational Intelligence, and Imbalanced Classes,* Διαθέσιμο στον δικτυακό τόπο: http://sci2s.ugr.es/BigData

Foody G. M., Lucas R. M., Curran P. J., Honzak M., 1996, Estimation of the areal extent of land cover classes that only occur at a sub-pixel level, *Canadian Journal of Remote Sensing, 22, p. 428–432*

Fukushima K., 1975, Cognitron: A self-organizing multilayered neural network, *Biological Cybernetics, 20, 3, p. 120-136*

Gani A. et al., 2016, A survey on indexing techniques for big data: toxonomy and performance evaluation, *Knowledge and Information Systems, 46 (2), p. 241-284*

Garson G. D., 1998, *Neural Networks: An Introductory Guide for Social Scientists*, London: Sage

Gong P., 1996, Integrated analysis of spatial data from multiple sources: using evidential reasoning and artificial neural network techniques for geological mapping, *Photogrammetric Engineering and Remote Sensing, 62, p. 513–523*

Gopal S., and Woodcock C., 1996, Remote sensing of forest change using artificial neural networks, *IEEE Transactions on Geoscience and Remote Sensing, 34, p. 398–403*

Hara Y., Atkins R. G., Yueh S. H., Shin R. T., Kong J. A., 1994, Application of neural networks to radar image classification, *IEEE Transactions on Geoscience and Remote Sensing, 32, p. 100–109*

Haykin S., 1999, *Neural Networks: A Comprehensive Foundation*, Prentice Hall

Hecht- Nielsen R., 1987, Kolmogorov's mapping neural network existence theorem, *Proceedings of the First IEEE International Conference on Neural Networks, San Diego, CA, 21–24 June 1987, edited by M. Caudill and C. Butler, IEEE, p. 11–14*

Hortonworks, 2015, *APACHE HADOOP YARN, The Architectural Center of Enterprise Hadoop*. Διαθέσιμο στον δικτυακό τόπο: http://hortonworks.com/hadoop/yarn/

Hubble E. P., 1926, Extra-Galactic Nebulae, *Astrophysical Journal, 64, p. 321-369*

Hubble E. P., 1958, *The Realm of the Nebulae, Dover Pub., Inc., New York,* SBN: 486-60455-1

Hush D. R., 1989, Classification with neural networks: a performance analysis, *Proceedings*

of the IEEE International Conference on Systems Engineering, Dayton, Ohio, USA, p. 277–280

Kanellopoulos I., Wilkinson G. G., 1997, Strategies and best practice for neural network image classification, *International Journal of Remote Sensing, 18, p. 711–725*

Kavzoglu T., Mather P.M., 2003, The use of backpropagating artificial neural networks in land cover classification, *Int. J. Remote Sensing, 24, 23, p. 4907–4938*

Klimasauskas C. C., 1993, Applying neural networks. *In Neural Networks in Finance and Investing, edited by R. R. Trippi and E. Turban , Cambridge: Probus, p. 47–72*

Lam C., 2010, Hadoop IN ACTION, MANNING

Lawrence S., Giles C. L., Tsoi A. C., 1996, *What size neural network gives optimal generalization? Convergence properties of backpropagation.* UMIA CS-TR-96-22 and CS-TR-3617, Institute for Advanced Computer Studies, University of Maryland, Maryland, USA

Lesk M., 1997, *How much information is there in the world?, Technical report*

Marr B., 2015, *Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance,* Wiley, United Kingdom

Marr B., 2015, *A Brief History of Big Data Everyone Should Read*, Διαθέσιμο στον δικτυακό τόπο: http://www.smartdatacollective.com/bernardmarr/323216/brief-history-big-data-everyone-should-read, τελευταία ενημέρωση: 09/06/2015

Mascetti L. et al., 2015, Disk storage at CERN, *21st International Conference on Computing in High Energy and Nuclear Physics, 664, doi: 10.1088/1742-6596/664/4/042035, Curran Associates Inc., New York*

Meert A., Vikram V., Bernardi M., 2015, A catalogue of 2D photometric decompositions in the SDSS-DR7 spectroscopic main galaxy sample: prederred models and systematic, *MNRAS,* 446, p. 3943-3974

Michael N. S., 2011, *Fundamentals of air traffic control,* Delmar Pub., ISBN:978-1-4354-8272-2

Monma C. L., Stoer M., 1993, *Handbook in operations research and management science,* ELSEVIER

Minsky M. L., Papert S. A., 1969, Perceptrons. Cambridge, MA: MIT Press

Moniruzzaman A.B.M., Hossain S.A., 2013, NoSQL Database: New era of databases for Big Data Analytics-Classification, Characteristics and Comparison, *International Journal of Database Theory and Application, 6 (4), pages: 14*

Murthy A. et al., 2014, *Apache Hadoop YARN, Moving Beyond MapReduce and Batch Processing with Apache Hadoop 2,* Hortonworks Inc.

Octavio R., Odoni A. R., 1994, Dynamic solution to the ground-holding problem in air traffic control, *Transportation Research Part A: Policy and Practice 28,3, p. 167-185*

Paola J. D., 1994, Neural network classification of multispectral imagery, *MSc Thesis, USA*

Paola J. D., Schowengerdt, R. A., 1997, The effect of neural-network structure on a multispectral land-use/land-cover classification, *Photogrammetric Engineering and Remote Sensing, 63, p. 535–544*

Park S. K., Miller K. W., 1988, *Random Number Generators: Good Ones Are Hard To Find, doi: 10.1145/63039.63042*

Partridge D., Yates W. B., 1996, Replicability of neural computing experiments, *Complex Systems, 10, p. 257–281*

Pierce L. E., Sarabandi K., Ulaby F. T., 1994, Application of an artificial neural network in canopy scattering inversion, *International Journal of Remote Sensing, 15, p. 3263–3270*

Plerou A., 2012, Artificial Neyral Networks simulating the Humman brain (in greek), *Open Education-The Journal for Open and Distance Education and Educational Technology, 8(1), p.128-135*

Rahman, M. N.,Esmailpour A., Zhao J., 2016, Machine Learning with Big Data, An Efficient Electricity Generation Forecasting System, *Big Data Research, 5, p. 9-15*

Ripley B. D., 1993, Statistical aspects of neural networks. *In Networks and Chaos – Statistical and Probabilistic Aspects, edited by O. E. Barndorff-Nielsen, J. L. Jensen, and W. S. Kendall, Chapman & Hall, p. 40–123*

Rosenblatt F., 1957, *The Perceptron: A perceiving and recognizing automaton,* Cornell Aeronautical Laboratory, Technical Report 85-460-1

Schmidhuber J., 2015, *Deep Learning in Neural Networks: An Overview, Neural Networks,* Technical Report, 61, p. 85-117

Shu F. H., 1982, *The Physical Universe: An Introduction to Astronomy*, University Science Book

Simon H., 1999, *Neural Networks: A Comprehensive Foundation,* Prentice Hall, ISBN: 978-0-7803-3494-6

Spentzos A., 2005, CFD Analysis of 3D Dynamic Stall, *PhD Thesis, University of Glasgow*

Staufer P., and Fischer M. M., 1997, Spectral pattern recognition by a two-layer perceptron: effects of training set size, *In Neurocomputation in Remote Sensing Data Analysis, edited by I. Kanellopoulos, G. G. Wilkinson, F. Roli, and J. Austin, Springer, p. 105–116*

Strickland J. S., 2016, *Data Analytics Using Open-Source Tools, Lulu*, Inc. ISBN: 978-1-365-27041-3

Sutton R. S., Barto A. G., 1998*, Reinforcement learning: An introduction,* MIT Press

Swingler K., 1996, *Applying Neural Networks – A Practical Guide*, Morgan Kauffman

Tom White, 2012, *Hadoop. The Definitive Guide*, O'Reilly Media, Inc., ISBN: 978-1-449-31152-0

Turner J., 2011*, Hadoop: What it is, how it works, and what it can do, Διαθέσιμο στον δικτυακό τόπο: https://www.oreilly.com/ideas/what-is-hadoop*

Vorhies B., 2013, A Brief History of Big Data Technologies from SQL to NoSQL to Hadoop and Beyond, *Διαθέσιμο στον δικτυακό τόπο:* http://data-magnum.com/a-brief-history-of-big-data-technologies-from-sql-to-nosql-to-hadoop-and-beyond*, τελευταία ενημέρωση: 31/10/2013*

Wang F., 1994, The use of artificial neural networks in a geographical information system *for agricultural land-suitability assessment*, Environment and Planning A, 26, p. 265–284

West D.M., 2012, *Big Data for Education: Data Mining, Data Analytics, and Web Dashboards*, Gov. Stud. Brook. US Reuters

Wu C., 2005, Inherent delays and operational reliability of airline schedules, *Journal of Air Transport Management , 11(4), p. 273-282*

Zikopoulos P. C. et al., 2012, *Understanding Big Data. Analytics for Enterprise Class Hadoop and Streaming Data*, McGraw-Hill Companies, New York

Zikopoulos P. C. et al., 2013, *Harness the Power of Big Data. The IBM Big Data Platform,* McGraw-Hill Companies, New York