Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης Και Υπολογιστων

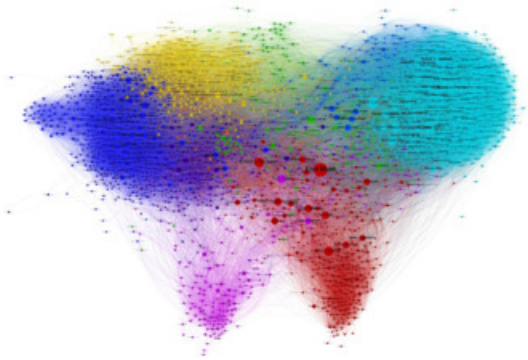
Ανίχνευση Ψευδών Ειδήσεων σε Μέσα Κοινωνικής Δικτύωσης

Θεωρητική θεμελίωση και Υλοποίηση

Διπλωματικη Εργασια

του

ΣΤΑΜΑΤΙΟΥ Ι. ΡΩΜΑ



Επιβλέπων: Δημήτρης Φωτάκης
Καθηγητής ΗΜΜΥ ΕΜΠ
Συνεπιβλέπων: Σπύρος Κοντογιάννης
Καθηγητής Πανεπιστημίου Πατρών

Αθήνα, Ιούλιος 2025

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης Και Υπολογιστων

Ανίχνευση Ψευδών Ειδήσεων σε Μέσα Κοινωνικής Δικτύωσης

Θεωρητική θεμελίωση και Υλοποίηση

Διπλωματικη Εργασια

του

ΣΤΑΜΑΤΙΟΥ Ι. ΡΩΜΑ

Επιβλέπων: Δημήτρης Φωτάκης
Καθηγητής ΗΜΜΥ ΕΜΠ
Συνεπιβλέπων: Σπύρος Κοντογιάννης
Καθηγητής Πανεπιστημίου Πατρών

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11η Ιουλίου 2025.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Δημήτρης Φωτάκης	Σπύρος Κοντογιάννης	Άρης Παγουρτζής
Καθηγητής ΗΜΜΥ ΕΜΠ	Καθηγητής Πανεπιστημίου Πατρών	Καθηγητής ΗΜΜΥ ΕΜΠ

Αθήνα, Ιούλιος 2025

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης Και Υπολογιστων

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....
Σταμάτιος Ρώμας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών

11 Ιουλίου 2025

στην οικογένειά μου , τους φίλους και τους δασκάλους μου

Περίληψη

Η ραγδαία ανάπτυξη των κοινωνικών δικτύων έχει επιτρέψει στους χρήστες να δημοσιεύουν, να αναδημοσιεύουν και να σχολιάζουν ένα ευρύ φάσμα πληροφοριών. Ωστόσο, οι χρήστες ενδέχεται, είτε εσκεμμένα είτε ακούσια να αποτελέσουν παράγοντες παραπληροφόρησης. Καθώς η αλήθεια συνιστά κομβικής σημασίας αξία για τη δημοκρατία, τη δικαιοσύνη και τη κοινωνική συνοχή, καθίσταται αναγκαία η διασφάλισή της στα ψηφιακά δίκτυα, τα οποία εν πολλοίς έχουν αντικαταστήσει τα φυσικά κοινωνικά δίκτυα.

Δεδομένης της σημασίας που έχει η αλήθεια, καθίσταται αναγκαία η ανάπτυξη τεχνικών ανίχνευσης ψευδών ειδήσεων. Οι διαφορετικές προσεγγίσεις εστιάζουν στα χαρακτηριστικά που αξιοποιούνται για να εκτιμήσουν την εγκυρότητα του περιεχομένου: την αξιοπιστία της πηγής, το μοτίβο διάδοσης, το συγγραφικό ύφος και την ακρίβεια του παρεχόμενου υλικού.

Μολονότι δεν υπάρχει ένας παγκοσμίως αποδεκτός ορισμός για τα fake news, κοινό γνώρισμα όλων παραμένει η αναληθής υπόστασή τους. Προηγούμενες μελέτες σε πεδία των Κοινωνικών και Οικονομικών Επιστημών προσφέρουν θεωρητικά πλαίσια κατανόησης της ανθρώπινης συμπεριφοράς και γνωσιακών μηχανισμών, συμβάλλοντας στην ποιοτική και ποσοτική ανάλυση των ψευδών ειδήσεων, καθώς και στη δημιουργία μοντέλων εντοπισμού και διαχείρισής τους. Οι εν λόγω θεωρίες διακρίνονται σε αυτές που σχετίζονται με την πληροφορία (π. χ. συγγραφικό ύφος, ποιότητα έκφρασης, συναισθήματα του κειμένου) και σε εκείνες που αφορούν τα άτομα (π. χ. κακόβουλοι χρήστες που διαδίδουν εν γνώσει τους ψευδείς ειδήσεις, αλλά και συνηθισμένοι χρήστες που εμπλέκονται ακούσια στη μετάδοση αναληθών πληροφοριών).

Στο πλαίσιο της παρούσας μελέτης, αξιοποιούμε το μοτίβο διάδοσης της πληροφορίας. Συγκεκριμένα, παράγονται δέντρα διάδοσης τα οποία, μέσω κατάλληλων μετασχηματισμών (Πυρήνες Γραφημάτων), μετατρέπονται σε διανύσματα χαρακτηριστικών. Στη συνέχεια, ένα μοντέλο επιβλεπόμενης μάθησης εκπαιδεύεται πάνω σε αυτά τα διανύσματα, με τις ετικέτες να υποδηλώνουν αν η αρχική δημοσίευση ήταν αληθής (1) ή ψευδής (0). Στόχος είναι να αποτιμηθεί η συσχέτιση μεταξύ της δομής του δέντρου διάδοσης και της εγκυρότητας της πληροφορίας, θέτοντας τα θεμέλια για πιο αξιόπιστες μεθόδους ανίχνευσης ψευδών ειδήσεων.

Λέξεις Κλειδιά

Ανίχνευση Ψευδών Ειδήσεων, Πυρήνες Γραφημάτων, Διορθωτική Απόσταση Δέντρων, Μηχανική Μάθηση, Δέντρα Ανάπτυξης, Κωδικοποίηση Δέντρων, Ισομορφισμός Δέντρων, Αλγόριθμοι Γραφημάτων, `NetworkX`, Αλγόριθμος `Wasserstein` K-Μέσων, `Twitter`, Ανάλυση Δεδομένων, Πυρήνες Γραφημάτων Weisfeiler-Lehman, Αλγόριθμος K-Μέσων, Βαθμωτή Κάθοδος

Abstract

The rapid development of social networks has enabled users to publish, republish, and comment on a wide range of information. However, users may—either deliberately or unknowingly—become agents of misinformation. Since truth is a core value for democracy, justice, and social cohesion, it is essential to protect its integrity within digital networks, which have largely replaced traditional social structures. Given the significance of truth, the development of techniques for detecting fake news is increasingly important. Different approaches focus on various features to assess the validity of content: the reliability of the source, the pattern of dissemination, the writing style, and the accuracy of the provided material.

Although there is no globally accepted definition for fake news, they share one common trait: their content is fundamentally false. Previous studies in the fields of Social and Economic Sciences provide theoretical frameworks for understanding human behavior and cognitive processes, aiding both the qualitative and quantitative analysis of fake news, as well as the creation of models for their detection and management. These theories can be classified into those related to information (e.g., writing style, expression quality, emotional content) and those related to individuals (e.g., malicious users who knowingly distribute false information, or regular users who unintentionally contribute to the spread of misinformation).

In this study, we utilize the pattern of information dissemination. Specifically, we construct dissemination trees, which are transformed into feature vectors through appropriate methods (Graph Kernels). A supervised learning model is then trained on these vectors, with labels indicating whether the original post was true (1) or false (0). The ultimate objective is to evaluate the correlation between the structure of the dissemination tree and the reliability of the conveyed information, laying the groundwork for more robust methods of fake news detection.

Keywords

Fake News Detection,Graph Kernels,Tree Edit Distance,Machine Learning,Unfolding Trees,Tree Encoding,Tree Isomorphism,Graph Algorithms,`NetworkX`,`Wasserstein` Distance,`Twitter` ,Data Analysis,Weisfeiler-Lehman Graph Kernel,k-Means Algorithm,Gradient Descent

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τους καθηγητές Χρήστο Ζαρολιάγκη και Σπύρο Κοντογιάννη για την επίβλεψη αυτής της διπλωματικής εργασίας και την άψογη συνεργασία που είχαμε. Σε όλη τη διάρκεια της εκπόνησης της ήταν ιδιαίτερα καθοδηγητικοί, βοηθητικοί και μου πέρασαν αξίες και τρόπο σκέψης όσον αφορά την επιστήμη και την έρευνα που μου ήταν και θα μου είναι σημαντικά. Ευχαριστώ τους φίλους μου που στέκονταν δίπλα μου και στα εύκολα και κυρίως στα δύσκολα και με επηρρέασαν σε πολλές πτυχές της προσωπικότητάς μου. Ευχαριστώ την οικογένειά μου για την στήριξη που μου παρείχε όλα αυτά τα χρόνια, τις αξίες, την φροντίδα και την αγάπη καθόλη τη διάρκεια της ζωής μου. Τέλος, θα ήθελα να ευχαριστήσω ιδιαίτερα όλους τους δασκάλους μου που μου έδωσαν τα απαραίτητα ερεθίσματα ώστε να εξελιχθώ. Από τα φοιτητικά μου χρόνια ,ιδιαίτερη τιμή και υποχρέωση οφείλω στους καθηγητές Θεμιστοκλή Ρασσιά, που με την έμπνευση και την ώθηση που μου παρείχε, ανέστησε την νεκρή επιστημονική μου περιέργεια και στους καθηγητές Δημήτρη Φωτάκη και Άρη Παγουρτζή για την διαμόρφωση ενός τρόπου σκέψης και την προσέγγιση επίλυσης προβλημάτων με έξυπνο τρόπο, εργαλεία σημαντικά για έναν μηχανικό.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Πρόλογος

Η εργασία αυτή αποτελεί τη Διπλωματική Εργασία μου εργασία για τη Σχολή των Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Εθνικού Μετσοβίου Πολυτεχνείου. Διενεργήθηκε υπό την επίβλεψη των καθηγητών του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής του Πανεπιστημίου Πατρών Χρήστου Ζαρολιάγκη και Σπύρου Κοντογιάννη, στο Εργαστήριο Intelligent Computing & Engineering εξ αποστάσεως μέσω διαδικτυακών συναντήσεων.

Κεφάλαιο 1

Εισαγωγή

Η ραγδαία ανάπτυξη του διαδικτύου και των κοινωνικών δικτύων έχει μεταμορφώσει ριζικά τον τρόπο με τον οποίο οι πληροφορίες παράγονται, διαμοιράζονται και καταναλώνονται. Τα ψηφιακά μέσα επιτρέπουν τη δημοσίευση και αναδημοσίευση ενός τεράστιου όγκου δεδομένων σε παγκόσμια κλίμακα, δίνοντας στους χρήστες την δυνατότητα τόσο να καταναλώνουν όσο και παράγουν πληροφορίες μέσω των δημοσιεύσεων που γίνονται στα μέσα αυτά. Ως αποτέλεσμα, η διάδοση ψευδών πληροφοριών (fake news) έχει εξελιχθεί σε ένα από τα πλέον ανησυχητικά φαινόμενα της σύγχρονης εποχής. Οι ειδήσεις αυτές ενδέχεται να έχουν πολιτικές, κοινωνικές ή οικονομικές σκοπιμότητες, αλλοιώνοντας δραστικά τη δημόσια σφαίρα και τον τρόπο που οι πολίτες λαμβάνουν αποφάσεις.

Σε αυτό το συνεχώς μεταβαλλόμενο επικοινωνιακό περιβάλλον, η παραπληροφόρηση (misinformation) και η κακόβουλη διασπορά ψευδούς περιεχομένου (disinformation) συχνά αλληλεπικαλύπτονται, συνθέτοντας ένα πολύπλοκο φαινόμενον 'διαταραχής της πληροφορίας' (information disorder) . Πέραν της δυσκολίας διάκρισης των ορίων μεταξύ έγκυρης και αναληθούς πληροφορίας, τα κοινωνικά δίκτυα παρέχουν ένα πολύπλευρο πεδίο, στο οποίο συναντώνται άτομα με διαφορετικές προθέσεις και κίνητρα. Κάποιοι χρήστες μπορεί να είναι 'ανυποψίαστοι', αναδημοσιεύοντας ψευδές περιεχόμενο χωρίς πρόθεση παραπλάνησης, ενώ άλλοι, 'κακόβουλοι χρήστες', επιδιώκουν συνειδητά τη διασπορά τέτοιων ειδήσεων.

1.1 Σημασία του προβλήματος

Οι επιπτώσεις της διάδοσης ψευδών ειδήσεων είναι πολυδιάστατες. Η δημοκρατία, οι κοινωνικές σχέσεις και η εμπιστοσύνη των πολιτών προς τα μέσα ενημέρωσης απειλούνται σοβαρά, καθώς η παραπληροφόρηση υπονομεύει την ποιότητα και την αξιοπιστία της δημόσιας πληροφόρησης. Παράλληλα, τα τελευταία χρόνια έχουν αναπτυχθεί διάφορες προσεγγίσεις για την ανίχνευση και τον περιορισμό των ψευδών ειδήσεων, κάθε μία από τις οποίες εστιάζει σε διαφορετικά στοιχεία της πληροφορίας.

Μια κλασική μέθοδος που βασίζεται στο περιεχόμενο εξετάζει τα γλωσσικά, υφολογικά ή σημασιολογικά χαρακτηριστικά του κειμένου, προσπαθώντας να επισημάνει ανωμαλίες που υποδηλώνουν ψευδή αφήγηση. Επιπλέον, μερικές έρευνες αναλύουν την εγκυρότητα της ίδιας της πηγής, ελέγχοντας προγενέστερες δημοσιεύσεις και συγκρίνοντάς τες με έγκυρα σημεία αναφοράς. Ωστόσο, η διαρκής εξέλιξη της τεχνολογίας και οι όλο και πιο σύνθετοι μηχανισμοί παραπληροφόρησης (π.χ. *bots*, *deepfakes*) καθιστούν δυσχερή την αποτελεσμα-

τική ανίχνευση τους μόνο μέσω της ανάλυσης του περιεχομένου.

Για τον λόγο αυτό, ολοένα και μεγαλύτερη προσοχή δίνεται στις προσεγγίσεις που στηρίζονται στον *τρόπο διάδοσης* (*propagation-based detection*). Η θεμελιώδης ιδέα έγκειται στο γεγονός ότι οι ψευδείς ειδήσεις τείνουν να εμφανίζουν μοναδικά μοτίβα διάχυσης μέσα στις διαδικτυακές πλατφόρμες. Συγκριτικά με τις αληθείς ειδήσεις, παρουσιάζουν μεγαλύτερη ταχύτητα μετάδοσης, συχνά μέσω ομάδων υψηλής διασύνδεσης, κακόβουλων λογαριασμών ή διαδικτυακών ρομπότ (*bots*). Επιπλέον, ακόμη και όταν το περιεχόμενο τροποποιείται ελαφρώς, το «αποτύπωμα» στη δομή της διάδοσης παραμένει σε μεγάλο βαθμό ανιχνεύσιμο, γεγονός που καθιστά αυτή την προσέγγιση ιδιαίτερα ελκυστική.

1.2 Στόχοι της εργασίας

Στην παρούσα εργασία, εστιάζουμε στην ανάπτυξη και αξιολόγηση μεθοδολογιών που βασίζονται στη διερεύνηση και ανάλυση των δικτύων διάδοσης για την ανίχνευση ψευδών ειδήσεων. Ειδικότερα, επιδιώκουμε:

1. **Μελέτη των βασικών χαρακτηριστικών διάδοσης:** Προσδιορισμός των μοτίβων που εντοπίζονται σε ψευδείς ειδήσεις και διαφέρουν από αυτά των αληθινών, ανάλυοντας τα χαρακτηριστικά που παρουσιάζονται στα δέντρα διάδοσης της πληροφορίας τόσο στις αληθείς όσο και στις ψευδείς ειδήσεις και δημιουργώνας έναν ταξινομητή βάσει αυτών των χαρακτηριστικών σαν αρχικό σημείο ώστε να καταδείξουμε ότι η τεχνική που εμείς προτείνουμε είναι πιο εύρωστη.

2. **Ανάπτυξη μοντέλου ανίχνευσης:** Σχεδιάζεται και υλοποιείται ένα σύστημα το οποίο, λαμβάνοντας ως είσοδο τα δέντρα αναδημοσιεύσεων, εφαρμόζει τεχνικές Πυρήνων Υποδέντρων *Weisfeiler-Lehman* (*Weisfeiler-Lehman Subtree Kernels*) για την αναπαράσταση της δομής τους σε διανύσματα χαρακτηριστικών. Με αυτόν τον τρόπο, αποτυπώνονται κρίσιμες τοπολογικές διαφοροποιήσεις που σχετίζονται με ψευδές ή έγκυρο περιεχόμενο, επιτρέποντας την αποτελεσματικότερη διάκριση ανάμεσα σε αξιόπιστες και παραπλανητικές δημοσιεύσεις.

3. **Πειραματική αξιολόγηση:** Υλοποιούμε και δοκιμάζουμε την προσέγγισή μας με υπαρκτά δεδομένα κοινωνικών δικτύων (Twitter15/Twitter16), συγκρίνοντάς τη με μεθόδους στις οποίες τα διανύσματα χαρακτηριστικών εξάγονται από συγκεκριμένες μετρικές πάνω στα δέντρα διάδοσης (π.χ. βάθος, πλάτος, ακολουθία βαθμών).

4. **Διατύπωση προτάσεων για μελλοντική έρευνα:** Προτείνουμε νέες ιδέες για βελτίωση και επέκταση του συστήματος, λαμβάνοντας υπόψη την ταχεία τεχνολογική εξέλιξη και τις συνεχώς μεταβαλλόμενες στρατηγικές παραπληροφόρησης.

Με αυτόν τον τρόπο, επιδιώκουμε να συμβάλουμε στη διαρκώς εντεινόμενη προσπάθεια της επιστημονικής κοινότητας για αποτελεσματικότερη και αυτόματοποιημένη ανίχνευση ψευδών ειδήσεων, ακολουθώντας τις κατευθύνσεις και τα ευρήματα της σύγχρονης βιβλιογραφίας. Η δυνατότητα ανάλυσης του τρόπου διάδοσης δεν προορίζεται να αντικαταστήσει πλήρως τις παραδοσιακές μεθόδους Ωστόσω αναδεικνύεται ως ένα ισχυρό συμπληρωματικό

εργαλείο σε μια εποχή όπου η παραπληροφόρηση εξελίσσεται συνεχώς και γίνεται όλο και πιο πολύπλοκη και συνεπώς όλο και πιο επικίνδυνη.

1.3 Συνεισφορά

1.3.1 Μελέτη

Καθ' όλη τη διάρκεια της παρούσας Διπλωματικής Εργασίας, αντικείμενο ενδελεχούς μελέτης αποτέλεσαν κυρίως αλγόριθμοι γραφημάτων και αλγόριθμοι μηχανικής μάθησης. Επιπρόσθετα, εξετάστηκε το απαραίτητο μαθηματικό υπόβαθρο που απαιτήθηκε τόσο για την υλοποίηση εξειδικευμένων κλάσεων όσο και για τη θεμελίωση της μεθοδολογίας, η οποία στοχεύει στη διερεύνηση της σχέσης μεταξύ της εγκυρότητας μιας δημοσίευσης και του τρόπου διάδοσής της.

1.3.2 Υλοποιηθέντες Αλγόριθμοι

Στο πλαίσιο της Διπλωματικής Εργασίας, σε συνεργασία με τους επιβλέποντες καθηγητές, υλοποιήθηκαν οι ακόλουθοι αλγόριθμοι:

1. Weisfeiler-Lehman Subtree Graph Kernel
2. Relaxed Weisfeiler-Lehman Subtree Kernel
3. i-Unfolding Tree
4. Unfolding Tree Vectors
5. Wasserstein k-Means Algorithm for Unfolding Trees
6. String Tree Encoding
7. Isomorphic Tree Representative

Οι παραπάνω αλγόριθμοι παρουσιάζονται εκτενώς στα επόμενα κεφάλαια, τόσο σε θεωρητικό όσο και σε υλοποιητικό επίπεδο.

1.3.3 Αξιολόγηση Αποτελεσμάτων

Η ανάλυση των πειραματικών αποτελεσμάτων κατέδειξε ότι υφίσταται συσχέτιση μεταξύ της εγκυρότητας της μεταδιδόμενης πληροφορίας και των τρόπων με τους οποίους αυτή διαχέεται. Στην Ενότητα 6.3 παρατίθενται αναλυτικά τα αποτελέσματα που προέκυψαν από τη χρήση διαφορετικών μεθοδολογιών και αλγορίθμων.

1.4 Διάρθρωση της Διπλωματικής Εργασίας

Η εργασία δομείται σε έξι κύρια κεφάλαια · κάθε κεφάλαιο υπηρετεί έναν σαφώς καθορισμένο στόχο, ώστε ο αναγνώστης να ακολουθεί γραμμικά τη μετάβαση από τη θεωρία στην πράξη και, τελικά, στην αξιολόγηση.

Κεφάλαιο 2 – Επισκόπιση Τεχνολογικής Στάθμης Ξεκινά με μία αναλυτική επισκόπηση των τεσσάρων βασικών μεθοδολογιών έρευνας στην ανίχνευση ψευδών ειδήσεων (*γνώση*, *ύφος*, *αξιοπιστία πηγής*, *τρόπο διάδοσης*). Για κάθε τεχνική παρουσιάζονται οι κυριότερες μέθοδοι, τα πλεονεκτήματα και τα μειονεκτήματά τους, καθώς και τα μέχρι σήμερα ανοιχτά ερευνητικά ζητήματα.

Κεφάλαιο 3 – Θεωρητικό Υπόβαθρο Εδραιώνει το μαθηματικό και αλγοριθμικό πλαίσιο που απαιτείται για την κατανόηση της προτεινόμενης μεθόδου. Περιλαμβάνει: (*i*) βασικές έννοιες από μετρικούς χώρους, χώρους Hilbert και τη γεωμετρία Wasserstein, (*ii*) θεμελιώδεις ορισμούς και έννοιες από τη θεωρία γραφημάτων, καθώς και αλγόριθμοι όπως η κωδικοποίηση δέντρων ανάπτυξης σε διανυσματική μορφή καθώς, η ισομορφική κανονικοποίηση δέντρων με επιγραφές στους κόμβους καθώς και οι Πυρήνες Γραφημάτων Weisfeiler-Lehman (*iii*) τις επιλεγμένες τεχνικές μηχανικής μάθησης (Logistic Regression, Gradient Boosted Trees κ.ά.) που αξιοποιούνται στα πειράματα ως ταξινομητές ή ως ενδιάμεσες μεθόδοι (kMeans,kMeans in 1-d, Wasserstein kMeans) καθώς επίσης και οι μετρικές απόδοσεις που χρησιμοποιούνται κατά την πειραματική αξιολόγηση.

Κεφάλαιο 4 – Υλοποίηση Γραφοθεωρητικής Μεθόδου Χρησιμοποιεί τη θεωρία που παρουσιάστηκε στο προηγούμενο κεφάλαιο για να κάνει μία πλήρη περιγραφή της αρχιτεκτονικής του συστήματος. Περιγράφει τη μοντελοποίηση του προβλήματος, την αναπαράσταση των δέντρων διάδοσης ως γράφους, την τεχνική των πυρήνων *Weisfeiler-Lehman* και την γενικευμένη εκδοχή του με χρήση συσταδοποίησης Wasserstein. Παρουσιάζονται και συγκρίνονται τρεις παραλλαγές: βασική, απλού πυρήνα και γενικευμένου πυρήνα.

Κεφάλαιο 5 – Λεπτομέρειες Υλοποίησης Στο κεφάλαιο αυτό πραγματοποιείται η τεκμηρίωση κώδικα. Περιγράφονται αναλυτικά οι δομές δεδομένων,οι εξαρτήσεις λογισμικού, η οργάνωση κλάσεων και η διαχείριση τεχνικών προκλήσεων (μεγάλος όγκος δεδομένων, πολυπλοκότητα αλγορίθμων, ισομορφισμός γραφημάτων).

Κεφάλαιο 6 – Πειραματική Αξιολόγηση Στο κεφάλαιο αυτό παρουσιάζεται αναλυτικά η πειραματική διαδικασία και τα αποτελέσματα στα σύνολα δεδομένων **Twitter15** και **Twitter16**. Αξιολογείται η απόδοση μεθόδων που κυμαίνονται από απλά δομικά χαρακτηριστικά, όπως η ακολουθία βαθμών, έως πιο σύνθετες τεχνικές βασισμένες σε πυρήνες γραφημάτων (Weisfeiler-Lehman). Η σύγκριση πραγματοποιείται με χρήση μετρικών όπως η ακρίβεια και το F_1-σcορε, παρέχοντας μια ολοκληρωμένη ανάλυση της αποτελεσματικότητας και της υπολογιστικής αποδοτικότητας κάθε προσέγγισης.

Κεφάλαιο 7 – Συμπεράσματα και Προοπτικές Στο κεφάλαιο αυτό συνοψίζονται τα βασικά ευρήματα της έρευνας, με κυριότερο ότι η **απλή δομική πληροφορία** (όπως η ακολουθία βαθμών) υπερτερεί τόσο σε απόδοση όσο και σε ταχύτητα έναντι πιο σύνθετων μεθόδων. Επιπλέον, στο κεφάλαιο προτείνονται ιδέες για μελλοντική έρευνα, όπως ο συνδυασμός της δομικής ανάλυσης με **χαρακτηριστικά περιεχομένου και χρηστών**, η ανάπτυξη **μοντέλων**

πραγματικού χρόνου καθώς και η χρήση στατιστικών μεθόδων και ταχύτερες μέθοδοι ομαδοποίησης δέντρων ανάπτυξης με σκοπό την μείωση της πολυπλοκότητας του μοντέλου.

Μέρος I

Επισκόπιση Τεχνολογικής Στάθμης για Ανίχνευση Ψευδών Ειδήσεων

Κεφάλαιο 2

Μέθοδοι Ανίχνευσης Ψευδών Ειδήσεων: Επισκόπηση

Η ταχεία και σε μεγάλο βαθμό ανεξέλεγκτη διάχυση πληροφοριών στα κοινωνικά δίκτυα έχει αναγάγει την *Ανίχνευση Ψευδών Ειδήσεων* σε κρίσιμο ερευνητικό πεδίο. Η διεθνής βιβλιογραφία, και ιδίως η συστηματική ανασκόπηση των Zhou & Zafarani [1], διαχωρίζει τις μεθόδους που χρησιμοποιούνται για την ανίχνευση ψευδών ειδήσεων στις εξής **τέσσερις** κύριες κατηγορίες:

1. **Μέθοδοι βασισμένες στη Γνώση** (*knowledge-based*). Επαληθεύουν την ακρίβεια των δηλώσεων αντιπαραβάλλοντας το περιεχόμενο των δημοσιεύσεων με αξιόπιστες βάσεις γνώσης ή με αυτόματα συμπεράσματα γεγονότων.[6, 7]

2. **Μέθοδοι βασισμένες στο Ύφος** (*style-based*). Εκμεταλλεύονται ρητορικά, λεξιλογικά και ψυχογλωσσικά χαρακτηριστικά κειμένου και εικόνας για να ανιχνεύσουν παραπλανητικά μοτίβα γραφής ή ασυμφωνία κειμένου–εικόνας.[8, 9]

3. **Μέθοδοι βασισμένες στον τρόπο Διάδοσης** (*propagation-based*). Μοντελοποιούν την εξάπλωση των ειδήσεων ως γραφήματα διάδοσης και αξιοποιούν δομικά ή χρονικά χαρακτηριστικά (π.χ. βάθος, ταχύτητα, πυρήνες γραφημάτων) για να διακρίνουν τις ψευδείς από τις αληθείς ειδήσεις.[10, 11]

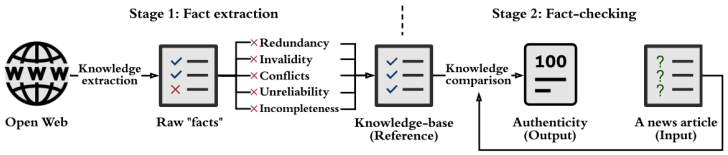
4. **Μέθοδοι βασισμένες στην Πηγή** (*source-based*). Αξιολογούν την αξιοπιστία του συγγραφέα, του εκδότη ή των χρηστών που διαδίδουν την είδηση, αξιοποιώντας μεταδεδομένα, ιστορικό δημοσιεύσεων και δομικά πρότυπα στον κοινωνικό γράφο.[12, 13]

Στα επόμενα τμήματα εξετάζουμε αναλυτικά κάθε κατηγορία, συνοψίζοντας τις κυριότερες τεχνικές, τα πλεονεκτήματα και τους περιορισμούς τους, καθώς και την αλληλεπίδρασή τους με εφαρμογές έγκαιρης ανίχνευσης και μείωσης παραπληροφόρησης.

2.1 Ανίχνευση μέσω Σύγκρισης με Γνώση

Η προσέγγιση που βασίζεται στη *Γνώση* (knowledge-based) επιχειρεί να ελέγξει την αυθεντικότητα μιας είδησης συγκρίνοντας τα γεγονότα που δηλώνονται σε αυτή με γεγονότα που είναι τεκμηριωμένα. Η διαδικασία αυτή είναι γνωστή στο δημοσιογραφικό πεδίο ως **Επαλήθευση Γεγονότων** (fact-checking) και αποτελείται από δύο βασικά στάδια: (*i*) *εξαγωγή*

γνώσης από το κείμενο και (*ii*) *επαλήθευση* της γνώσης σε σχέση με μια βάση/γράφημα γνώσης (knowledge-base/knowledge-graph) [1].Το συνοπτικό σχήμα της αυτόματης επαλήθευσης γεγονότος παρουσιάζεται στο Σχήμα 2.1 :



Σχήμα 2.1: *Διαδιακασία επαλήθευσης γεγονότος. [1]*

2.1.1 Χειροκίνητος Έλεγχος Γεγονότων

Με ειδικούς (*expert-based*). Ο έλεγχος διενεργείται από μικρές ομάδες υψηλής αξιοπιστίας· ιστότοποι όπως *PolitiFact* και *GossipCop* δημοσιεύουν λεπτομερείς αναλύσεις, οι οποίες χρησιμεύουν ως επαληθευμένα δεδομένα (ground-truth) για την εκπαίδευση αλγορίθμων [14, 15]. Πλεονέκτημα αποτελεί η υψηλή ακρίβεια. Παρόλαυτα τα βασικά μειονεκτήματα είναι τα ακόλουθα:

- **Περιορισμένη κλιμακωσιμότητα** λόγω της ανάγκης για έμπειρους επαληθευτές και χρονοβόρες διαδικασίες.
- **Υψηλό λειτουργικό κόστος** που προκύπτει από τους ειδικούς ανθρώπινους πόρους και τις απαιτητικές μεθοδολογίες.
- **Αδυναμία ταχείας επεξεργασίας** μεγάλων όγκων δεδομένων σε πραγματικό χρόνο.

Με τη συνεισφορά των χρηστών (*crowd-sourced*). Πλατφόρμες τύπου *Mechanical Turk* ή *Fiskkit* βασίζονται στη συλλογική νοημοσύνη (collective intelligence). Παρότι επιτυγχάνουν καλύτερη κλιμάκωση, πάσχουν από *ετερογένεια αξιοπιστίας* και χρήζουν επίλυσης αντικρουόμενων απαντήσεων [16]. Ωστόσο, επιτρέπουν λεπτομερή σχολιασμό (στάση, συναίσθημα) που μπορεί να αξιοποιηθεί στην ανάλυση της πρόθεσης.

2.1.2 Αυτόματος Έλεγχος Γεγονότων

Η χειροκίνητη διαδικασία δεν κλιμακώνεται στον ρυθμό παραγωγής περιεχομένου στα κοινωνικά δίκτυα. Τα αυτόματα συστήματα συνδυάζουν Τεχνικές Ανάκτησης Πληροφοριών, Επεξεργασίας Φυσικής Γλώσσας και Μηχανικής Μάθησης [1].

Αναπαράσταση γνώσης. Κάθε ισχυρισμός μετατρέπεται σε τριπλέτα (Subject, Predicate, Object) (SPO triple). Ένα σύνολο επιβεβαιωμένων γεγονότων αποτελεί την *βάση γνώσης* (knowledge-base ή για συντομογραφία KB). Η βάση γνώσης απεικονίζεται ως *γράφος γνώσης* (KG), όπου κόμβοι είναι οι οντότητες και ακμές αποτελούν τις μεταξύ τους σχέσεις [17].

Στάδιο Α: Εξαγωγή Γεγονότων. Η εξαγωγή πραγματοποιείται είτε από *μοναδική* πηγή (π.χ. Wikipedia, DBpedia), είτε από *πολλαπλές* ανοιχτές πηγές (open-source extraction). Κατά τη δημιουργία του τελικού γράφου-γνώσης πρέπει να αντιμετωπιστούν πολλά προβλήματα όπως οι *πλεονασμοί* (entity resolution), η *χρονική ακρίβεια*, τα *αντικρουόμενα γεγονότα*, οι *αναξιόπιστες πηγές* και οι *ελλείψεις*. Οι τεχνικές συμπλήρωσης γράφων γνώσης βασίζονται κυρίως σε τρεις κατηγορίες μεθόδων:

- **Μοντέλα λανθάνουσων χαρακτηριστικών** (latent feature models), όπως τα TransE και DistMult,
- **Μεθόδοι βασισμένες σε γραφο-χαρακτηριστικά** (graph feature-based), όπως ο αλγόριθμος Path Ranking,
- **Πιθανοτικά μοντέλα** (probabilistic graphical models), όπως τα Μαρκοβιανά Τυχαία Πεδία (Markov Random Fields ή MRFs) [17].

.

Στάδιο Β: Επαλήθευση Γεγονότων. Για κάθε τριπλέτα (s, p, o) της είδησης, πραγματοποιούνται τα ακόλουθα βήματα:

1. *Εντοπισμός οντοτήτων*: τα (s, o) αντιστοιχίζονται σε κόμβους της βάσης γνώσης.
2. *Έλεγχος σχέσης*: αν υπάρχει ακμή (s, p, o) στη βάση γνώσης, ο ισχυρισμός θεωρείται αληθής.
3. *Μη ύπαρξη ακμής*: Σε περίπτωση που η τριπλέτα (s, p, o) δεν αντιστοιχεί σε καμία υπάρχουσα ακμή της βάσης γνώσης, η εγκυρότητα του ισχυρισμού εξαρτάται από την παραδοχή που υιοθετείται:
   - **Υπόθεση κλειστού κόσμου (Closed-world assumption)**: η μη ύπαρξη της σχέσης υποδηλώνει ότι η τριπλέτα είναι *ψευδής*.
   - **Υπόθεση ανοικτού κόσμου (Open-world assumption)**: η μη ύπαρξη της σχέσης υποδηλώνει ότι η τριπλέτα είναι *άγνωστη* και ενδεχομένως αληθής ή ψευδής.
   - **Υπόθεση τοπικά κλειστού κόσμου (Local closed-world assumption)**: η τριπλέτα θεωρείται *ψευδής* αν υπάρχουν άλλες καταγεγραμμένες σχέσεις (s, p, o') για το ίδιο υποκείμενο και κατηγόρημα ($T(s, p) > 0$), αλλιώς χαρακτηρίζεται *άγνωστη* [18].

   Σε περιπτώσεις αβεβαιότητας, εφαρμόζονται τεχνικές *πρόβλεψης συνδέσμων* (link prediction), όπως *σημασιολογική εγγύτητα* (semantic proximity) ή LinkNBed, για εκτίμηση πιθανών σχέσεων μεταξύ των οντοτήτων.

Ανοικτά ζητήματα. Παρά τη σημαντική πρόοδο στον τομέα της αυτόματης επαλήθευσης ειδήσεων με χρήση γνώσης, εξακολουθούν να υφίστανται κρίσιμες προκλήσεις που περιορίζουν την εφαρμογή της σε πραγματικά περιβάλλοντα. Ειδικότερα:

- **Δυναμική ενημέρωση γράφων γνώσης (dynamic KG)**: Οι ειδήσεις χαρακτηρίζονται από έντονη χρονική ευαισθησία, καθώς επικεντρώνονται σε πρόσφατα και μη εγκυκλοπαιδικά γεγονότα. Ως εκ τούτου, η στατική γνώση που παρέχουν παραδοσιακές πηγές, όπως η Wikipedia, συχνά αποδεικνύεται ανεπαρκής. Η δυνατότητα αυτόματης προσθήκης νέων γεγονότων και αφαίρεσης παρωχημένης ή εσφαλμένης πληροφορίας είναι θεμελιώδους σημασίας για την αξιοπιστία και τη χρησιμότητα ενός σύγχρονου γράφου γνώσης [1].

- **Επιλογή και αποθήκευση «πολύτιμης» γνώσης**: Αν και οι σύγχρονες προσεγγίσεις επικεντρώνονται στη συλλογή όσο το δυνατόν περισσότερων γεγονότων, η ανάγκη για έγκαιρο και αποδοτικό έλεγχο απαιτεί τη χρήση μικρότερων αλλά πιο στοχευμένων γράφων γνώσης. Οι γράφοι αυτοί θα πρέπει να περιλαμβάνουν αποκλειστικά υψηλής ποιότητας και σημασίας γεγονότα, επιτρέποντας ταχύτερη και ακριβέστερη σύγκριση με τα προς επαλήθευση γεγονότα [1].

- **Εντοπισμός ελέγξιμων ισχυρισμών (check-worthy claims)**: Η αυτόματη αναγνώριση των αποσπασμάτων ενός κειμένου που χρήζουν επαλήθευσης συνιστά ένα ξεχωριστό και εν μέρει άλυτο πρόβλημα. Η ακριβής και αποδοτική ανίχνευση τέτοιων ισχυρισμών αποτελεί απαραίτητη προϋπόθεση για την αποτελεσματική εφαρμογή μεθόδων αυτόματου ελέγχου γεγονότων [1].

Συνοψίζοντας, η ανίχνευση ψευδών ειδήσεων με βάση τη γνώση συνδυάζει την ακρίβεια της χειροκίνητης δημοσιογραφικής επαλήθευσης με την κλιμάκωση που προσφέρουν οι αυτοματοποιημένες τεχνικές. Ωστόσο, η επιτυχής υλοποίησή της απαιτεί την αντιμετώπιση πολυδιάστατων ερευνητικών προκλήσεων, μεταξύ των οποίων συγκαταλέγονται η δυναμική κατασκευή γράφων γνώσης και η ταχύτητα επεξεργασίας, θέματα τα οποία παραμένουν αντικείμενο ενεργούς επιστημονικής μελέτης [1].

2.2 Ανίχνευση με Βάση το Ύφος και το Περιεχόμενο του Κειμένου

Η **ανίχνευση ψευδών ειδήσεων βάσει του ύφους** (*style-based fake-news detection*) επικεντρώνεται στην εξαγωγή πληροφοριών που αφορούν τις *προθέσεις* του συντάκτη, ανιχνεύοντας χαρακτηριστικά ύφους ή ρητορικά μοτίβα που συνοδεύουν παραπλανητικά κείμενα, σε αντίθεση με τις μεθόδους που *βασίζονται στη Γνώση* (knowledge-based) οι οποίες ελέγχουν τη *γνησιότητα* των δηλωθέντων γεγονότων [1].

Ορισμός 2.1 (Ύφος ψευδών ειδήσεων [1])**.** *Ως ύφος ψευδών ειδήσεων ορίζεται το σύνολο ποσοτικοποιήσιμων χαρακτηριστικών* $\mathbf{f} \in \mathbb{R}^k$ *που αναπαριστούν επαρκώς την πληροφορία μιας ψευδούς είδησης και την διαφοροποιούν από τις ειδήσεις με αληθές περιεχόμενο.*

Το πρόβλημα διατυπώνεται ως **δυαδική ταξινόμηση**. Σε κάθε είδηση N αντιστοιχούμε ένα διάνυσμα χαρακτηριστικών $\mathbf{f} \in \mathbb{R}^k$. Μέσω του συνόλου εκπαίδευσης:

$$\mathcal{T}_{\text{train}} = \{(\mathbf{f}^{(l)}, y^{(l)}) \mid \mathbf{f}^{(l)} \in \mathbb{R}^k,\ y^{(l)} \in \{0,1\},\ l = 1,\ldots,n\},$$

σκοπός μας είναι η εκπαίδευση ενός μοντέλου $S_\partial : \mathbb{R}^k \to \{0,1\}$ που για κάθε διάνυσμα χαρακτηριστικών του ύφους του κειμένου αντιστοιχεί την ετικέτα $\hat{y}$ (με 0 =true, 1 =fake):

$$\hat{y} = S_\partial(\mathbf{f}).$$

Οι παράμετροι του μοντέλου προκύπτουν από την ελαχιστοποίηση κατάλληλης συνάρτησης κόστους (π.χ. της *binary cross-entropy* $\mathcal{L}(S_\partial; \mathcal{T}_{\text{train}})$).

Η επιτυχία του μοντέλου εξαρτάται εξαρτάται από (α) από την ποιότητα των χαρακτηριστικών $\mathbf{f}$ που κωδικοποιούν το στυλ της δημοσίευσης και (β) από τη ρυθμιστική ικανότητα του μοντέλου S_∂.

2.2.1 Αναπαράσταση Ύφους (*Style Representation*)

Η αναπαράσταση του ύφους ενός ειδησεογραφικού άρθρου επιτυγχάνεται μέσω ενός συνόλου ποσοτικών χαρακτηριστικών. Αυτά τα χαρακτηριστικά μπορούν να κατηγοριοποιηθούν σε δύο κύριες κατηγορίες: *κειμενικά* και *οπτικά* χαρακτηριστικά.

Κειμενικά Χαρακτηριστικά

Τα κειμενικά χαρακτηριστικά χωρίζονται σε (α) **γενικά** και (β) **λανθάνοντα** χαρακτηριστικά.

Γενικά χαρακτηριστικά. Αυτά αφορούν μετρήσιμες στατιστικές ιδιότητες του κειμένου και διακρίνονται σε τέσσερις βασικές ομάδες:[8, 9, 19, 20]:

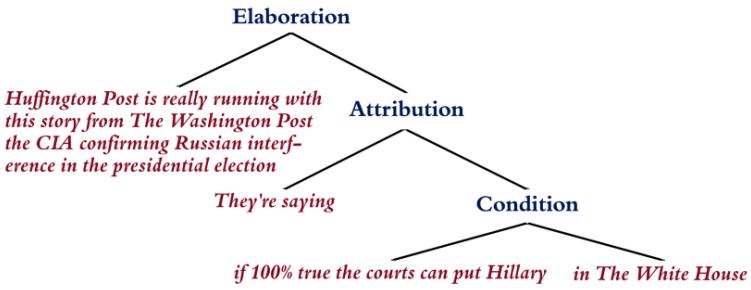
1. **Λεξικά χαρακτηριστικά:** Στατιστικές συχνοτήτων λέξεων ή n-γραμμάτων που υπολογίζονται με τεχνικές όπως Bag-of-Words (BoW) και TF–IDF.

2. **Συντακτικά χαρακτηριστικά:** Περιλαμβάνουν τόσο ρηχά συντακτικά γνωρίσματα, όπως στατιστικές κατανομές μερών του λόγου (POS tags), όσο και βαθύτερα συντακτικά γνωρίσματα μέσω κανόνων Πιθανοτικών Γραμματικών Ελευθέρων Συμφραζομένων (Probabilistic Context-Free Grammars - PCFG), καθώς και διάφορες παραλλαγές αυτών.

3. **Χαρακτηριστικά ρητορικής δομής:** Χρησιμοποιούν τη Θεωρία Ρητορικής Δομής (Rhetorical Structure Theory - RST) για την ανάλυση των σχέσεων μεταξύ προτάσεων εντός του κειμένου, καταγράφοντας τις συχνότητες των διαφορετικών εννοιολογικών σχέσεων. (Σχήμα 2.2)

4. **Σημασιολογικά και ψυχογλωσσικά χαρακτηριστικά:** Αντλούνται από εργαλεία ανάλυσης, όπως το LIWC, και άλλες ψυχογλωσσικές κατηγορίες που περιλαμβάνουν χα-

ρακτηριστικά όπως συναισθηματικό φορτίο, υποκειμενικότητα, πολυπλοκότητα, ποικιλομορφία, και αναγνωσιμότητα.



Σχήμα 2.2: *Παράδειγμα ανάλυσης πρότασης με χρήση Θεωρίας Ρητορικής Δομής.*

Η συχνότητα μπορεί να οριστεί και να υπολογιστεί με τρεις τρόπους. Έστω ένα σώμα κειμένων (corpus) C που περιέχει p άρθρα ειδήσεων, $C = \{A_1, A_2, \ldots, A_p\}$, και ένα σύνολο από q στοιχεία $W = \{w_1, w_2, \ldots, w_q\}$ (π.χ. POS tags, rewrite rules, rhetorical relationships κ.λπ.). Αν το x_j^i δηλώνει τον αριθμό εμφανίσεων του στοιχείου w_j μέσα στο άρθρο A_i, τότε η «συχνότητα» του w_j για το άρθρο A_i μπορεί να είναι:

- **Απόλυτη συχνότητα** f_a: ορίζεται ως ο απόλυτος αριθμός εμφανίσεων του χαρακτηριστικού w_j στο άρθρο A_i:
$$f_a = x_j^i$$
- **Κανονικοποιημένη συχνότητα** f_s: κανονικοποιεί την απόλυτη συχνότητα ως προς το συνολικό μήκος του άρθρου, ώστε να εξαλειφθεί η επίδραση του μήκους του κειμένου:
$$f_s = \frac{x_j^i}{\sum_j x_j^i}$$
- **Σχετική συχνότητα (TF–IDF)** f_r: υπολογίζει τη συχνότητα του χαρακτηριστικού w_j συγκριτικά με τη συχνότητα εμφάνισής του σε ολόκληρο το σύνολο των άρθρων:
$$f_r = \frac{x_j^i}{\sum_j x_j^i} \cdot \ln \frac{p}{\sum_{i=1}^{p} x_j^i}$$

Λανθάνοντα χαρακτηριστικά. Τα λανθάνοντα χαρακτηριστικά προκύπτουν από ενσωματώσεις (embeddings) λέξεων ή προτάσεων (π.χ. word2vec, GloVe, Doc2Vec). Τέτοιες ενσωματώσεις συνήθως τροφοδοτούν μοντέλα βαθιάς μάθησης, όπως CNN, LSTM/GRU και Transformer, τα οποία παράγουν αυτόματα υψηλού επιπέδου αναπαραστάσεις που αξιοποιούνται για την ταξινόμηση.

Οπτικά Χαρακτηριστικά

Όπως και στο κείμενο, τα γνωρίσματα εικόνας χωρίζονται σε (α) **χειροκίνητα** και (β) **λανθάνοντα** [1].

Χειροκίνητα (hand-crafted) γνωρίσματα. Περιλαμβάνουν μετρικές που υπολογίζονται απευθείας από τα εικονοστοιχεία, π.χ. *οξύτητα*, *χρωματική συνοχή*, *ποικιλία* και *πυκνότητα συστάδων*. Τα στατιστικά χαρακτηριστικά που προτείνονται [1] έχουν αποδειχθεί ιδιαίτερα αποτελεσματικά στην ανίχνευση:

- Αλλοιωμένου οπτικού περιεχομένου (tampered visual content)
- Παραπλανητικών πολυμεσικών στοιχείων (deceptive multimedia elements)

Λανθάνοντα Γνωρίσματα (Latent Features). Κάθε ψηφιακή εικόνα αναπαρίσταται μαθηματικά ως τανυστής τρίτης τάξης:

$$\mathbf{I} \in \mathbb{R}^{W \times H \times C}$$

όπου:

- W, H: οι χωρικές διαστάσεις (εύρος/ύψος σε pixels)
- C: ο αριθμός καναλιών χρωματικής αναπαράστασης
  - $C = 1$: Ασπρόμαυρη ή Γκρίζα Κλίμακα (Grayscale)
  - $C = 3$: Χρωματικός χώρος RGB (Κόκκινο-Πράσινο-Μπλε)

Ο τανυστής εισόδου διέρχεται από προ-εκπαιδευμένα συνελικτικά δίκτυα, όπως *VGG-16/19* ή *ResNet*. Τα δίκτυα αυτά

- *εξάγουν ιεραρχικές απεικονίσεις* μέσω διαδοχικών στρωμάτων συνελίξεων και συγκέντρωσης (*convolution / pooling*),
- *παράγουν συμπαγή διανύσματα χαρακτηριστικών* $\mathbf{e} \in \mathbb{R}^d$ που κωδικοποιούν το οπτικό περιεχόμενο, και
- *συλλαμβάνουν αφηρημένα σημασιολογικά μοτίβα* σχήματος, υφής και χρωματικής σύνθεσης.

Οι προκύπτουσες *ενσωματώσεις* (embeddings) μπορούν στη συνέχεια να συγχωνευθούν με κειμενικά γνωρίσματα σε πολυτροπικές αρχιτεκτονικές, ενισχύοντας την ακρίβεια των ταξινομητών στην ανίχνευση παραπλανητικού περιεχομένου που συνοδεύεται από εικόνες.

Συνολικά, η αποτελεσματικότητα της ανίχνευσης ψευδών ειδήσεων μέσω ανάλυσης ύφους εξαρτάται άμεσα από την επιλογή των χαρακτηριστικών αυτών και την αποτελεσματική ενσωμάτωσή τους στα μοντέλα ταξινόμησης που θα παρουσιαστούν στη συνέχεια.

2.2.2 Ταξινόμηση Ύφους (*Style Classification*)

Η ανίχνευση στηριζόμενη στο ύφος υλοποιείται είτε με **παραδοσιακές** τεχνικές Μηχανικής Μάθησης είτε με **βαθιά** νευρωνικά δίκτυα. Οι μέθοδοι διαφοροποιούνται κυρίως ως προς (i) τον τρόπο αναπαράστασης των χαρακτηριστικών και (ii) τον αλγόριθμο εκπαίδευσης.

Κλασικές Μεθόδοι Μηχανικής Μάθησης

Σύμφωνα με την κλασική μεθοδολογία, το άρθρο κωδικοποιείται με χειροκίνητα επιλεγμένα γνωρίσματα— λανθάνοντα ή ρητά— π.χ. λεξιλογικές συχνότητες, POS / PCFG κανόνες, ρητορικές σχέσεις και ψυχογλωσσικούς δείκτες. Επομένως, το πρόβλημα ανάγεται σε *δυαδική ταξινόμηση*: μοντέλα όπως SVM, Random Forest και XGBoost έχουν αποδειχθεί ιδιαίτερα αποτελεσματικά, ιδίως όταν *μη λανθάνοντα* (ρητά) γνωρίσματα συνδυάζονται από πολλαπλά γλωσσικά επίπεδα [8, 9, 20]. Τα αποτελέσματα της μελέτης του [20] υποδεικνύουν ότι (α) τα ρητά χαρακτηριστικά υπερέχουν των λανθανόντων, (β) η διαστρωμάτωση γνωρισμάτων βελτιώνει την επίδοση έναντι μονοεπιπέδων, και (γ) οι κανονικοποιημένες συχνότητες λεξιλογίων και κανόνων PCFG αποτελούν τον πλέον διακριτικό περιγραφέα ύφους, αν και είναι υπολογιστικά κοστοβόρο.

Μέθοδοι Βαθιάς Μάθησης

Στις προσεγγίσεις που γίνεται χρήση νευρωνικών δικτύων, το κείμενο αντιστοιχίζεται αρχικά σε ενσωματώσεις λέξεων ή προτάσεων (word2vec, GloVe κ.ά.), ενώ οι εικόνες σε τανυστές εικονοστοιχείων. Οι ενσωματώσεις διέρχονται από δίκτυα CNN (π.. Text-CNN, VGG-19), RNN (LSTM, GRU), ή Transformer, παράγοντας λανθάνοντα οπτικο-κειμενικά διανύσματα τα οποία συνενώνονται και ταξινομούνται μέσω μιας συνάρτησης softmax. Αυτή η γενική διαδικασία μπορεί να επεκταθεί με πιο σύνθετες αρχιτεκτονικές που εισάγουν εξειδικευμένους μηχανισμούς για τη βελτίωση της απόδοσης και της ερμηνευσιμότητας, όπως:

- **EANN (Event Adversarial Neural Network)** [21]: Μια αρχιτεκτονική που ενισχύει την ποιότητα των χαρακτηριστικών, εξάγοντας αναπαραστάσεις που είναι **ανεξάρτητες από το εκάστοτε γεγονός** (event-invariant). Αποτελείται από τρία μέρη: (1) έναν πολυτροπικό εξαγωγέα χαρακτηριστικών (multi-modal feature extractor) για κείμενο και εικόνα, (2) έναν διαχωριστή γεγονότων (event discriminator) που μέσω ανταγωνιστικής μάθησης (adversarial learning) απομονώνει τα ανεξάρτητα από το γεγονός χαρακτηριστικά, και (3) έναν τελικό ανιχνευτή ψευδών ειδήσεων (fake news detector).
- **SAFE (Similarity-Aware Multi-Modal Fake News Detection)** [22]: Μια πολυτροπική μέθοδος που εστιάζει στην ανίχνευση της **ασυμφωνίας** (disagreement) μεταξύ του κειμένου και της εικόνας ενός άρθρου. Βασίζεται στην παρατήρηση ότι στις ψευδείς ειδήσεις εμφανίζεται συχνά ένα σημασιολογικό χάσμα μεταξύ του κειμενικού και του οπτικού περιεχομένου, για παράδειγμα, με τη χρήση ελκυστικών αλλά θεματικά άσχετων εικόνων.

2.2.3 Εντοπισμένα Μοτίβα Ύφους

Εμπειρικές μελέτες σε πολιτικά σύνολα δεδομένων [20, 23] έχουν καταδείξει ότι τα ψευδή άρθρα:

- χρησιμοποιούν *ανεπίσημη* γλώσσα (υψηλή συχνότητα απρεπών λέξεων),
- επιδεικνύουν μεγαλύτερη *ποικιλία* ρημάτων και εικόνων αναφοράς,
- είναι περισσότερο *υποκειμενικά/συναισθηματικά*,
- ενώ οι συνοδευτικές εικόνες παρουσιάζουν υψηλότερη *ευκρίνεια* και *συνοχή*, αλλά χαμηλότερη *ποικιλία*.

Η συστηματική ενσωμάτωση των παραπάνω μοτίβων σε υβριδικά συστήματα αναμένεται να βελτιώσει τόσο την ακρίβεια όσο και την ερμηνευσιμότητα των μεθόδων ανίχνευσης ψευδών ειδήσεων.

2.2.4 Συζήτηση – Ανοικτές Προκλήσεις

Η ισχυρή εξάρτηση των μεθόδων αυτών από το περιεχόμενο προσφέρει το πλεονέκτημα της *έγκαιρης ανίχνευσης* (early detection), δηλαδή πριν μια είδηση διαδοθεί ευρέως. Ωστόσο, αυτή ακριβώς η εξάρτηση αποτελεί και τη μεγαλύτερή τους αδυναμία. Οι δημιουργοί ψευδών ειδήσεων μπορούν να διαμορφώσουν το ύφος γραφής τους με τέτοιο τρόπο ώστε να παρακάμψουν τους ανιχνευτές, οδηγώντας σε ένα διαρκές «παιχνίδι γάτας-ποντικού» (cat-and-mouse game) [1]. Κάθε επιτυχία στην ανίχνευση μπορεί να δημιουργήσει νέες τεχνικές παράκαμψης.

Επιπλέον, τα υπάρχοντα ευρήματα για τα υφολογικά μοτίβα είναι συχνά περιορισμένα σε συγκεκριμένες θεματικές περιοχές (π.χ. πολιτική) ή γλώσσες. Για τους λόγους αυτούς, η μελλοντική έρευνα καλείται να διεξάγει πιο ολοκληρωμένες αναλύσεις του ύφους σε διαφορετικούς τομείς, γλώσσες και χρονικές περιόδους. Για την ενίσχυση της ανθεκτικότητας των συστημάτων, κρίνεται αναγκαίος ο συνδυασμός της ανάλυσης ύφους με πληροφορίες από το κοινωνικό πλαίσιο.

2.3 Ανίχνευση με Βάση την Αξιοπιστία της Πηγής

Η προσέγγιση που βασίζεται στην πηγή (source-based) αποτελεί μια έμμεση μέθοδο ανίχνευσης, καθώς δεν εξετάζει το περιεχόμενο κάθε είδησης ξεχωριστά, αλλά αξιολογεί την αξιοπιστία του φορέα της πληροφορίας. Ως «πηγή» ορίζεται ο συγγραφέας, ο εκδότης (π.χ. ειδησεογραφικός ιστοτόπος) ή ο χρήστης κοινωνικών δικτύων που διαδίδει την είδηση [12]. Η λογική είναι ότι μια είδηση που προέρχεται από αναξιόπιστη πηγή είναι πιθανότερο να είναι ψευδής.

Αν και αυτή η μέθοδος ενέχει τον κίνδυνο να χαρακτηρίσει εσφαλμένα μια αληθή είδηση ως ψευδή, είναι υπολογιστικά πολύ αποδοτική (efficient), καθώς ο αριθμός των πηγών είναι κατά πολύ μικρότερος από τον αριθμό των ειδήσεων που παράγονται καθημερινά [24].

2.3.1 Αξιολόγηση Συγγραφέων και Εκδοτών

Μοτίβα Ομοιογένειας σε Δίκτυα. Έρευνες έχουν δείξει ότι συγγραφείς και εκδότες εμφανίζουν έντονη ομοιογένεια (homogeneity) στα δίκτυά τους. Για παράδειγμα, σε δίκτυα συνεργασίας συγγραφέων, οι δημιουργοί ψευδών ειδήσεων τείνουν να συνεργάζονται πολύ πιο πυκνά μεταξύ τους παρά με συγγραφείς αληθών ειδήσεων. Παρόμοια, σε δίκτυα διαμοιρασμού περιεχομένου (content-sharing networks) μεταξύ ειδησεογραφικών ιστοτόπων, παρατηρούνται ευδιάκριτες κοινότητες βάσει πολιτικής ιδεολογίας και αξιοπιστίας (π.χ. κυρίαρχα μέσα, συνωμοσιολογικά δίκτυα, κ.λπ.), καθιστώντας τη θέση ενός κόμβου στο γράφημα ισχυρή ένδειξη για την αξιοπιστία του [13].

Αυτοματοποιημένη Ανίχνευση Ανεπιθύμητου Περιεχομένου στον Ιστό (Web Spam). Η αξιοπιστία ενός εκδότη είναι στενά συνδεδεμένη με την ποιότητα του ιστοτόπου του. Επομένως, τεχνικές ανίχνευσης Web spam μπορούν να χρησιμοποιηθούν για τον εντοπισμό αναξιόπιστων πηγών. Οι τεχνικές αυτές διακρίνονται σε μεθόδους που βασίζονται:

- **στο περιεχόμενο** (π.χ. ανάλυση συχνοτήτων λέξεων),
- **στους υπερσυνδέσμους** (π.χ. αλγόριθμοι όπως οι PageRank και HITS, ανάλυση ανωμαλιών στη δομή του γράφου),
- **στη συμπεριφορά των χρηστών** (π.χ. ανάλυση clickstreams).

Εξωτερικοί Κατάλογοι Αξιοπιστίας. Πλατφόρμες όπως οι *Media Bias/Fact-Check*, *NewsGuard* και συστήματα όπως το *MediaRank* [25] παρέχουν αξιολογήσεις για την αξιοπιστία και την πολιτική μεροληψία χιλιάδων ιστοτόπων. Αυτές οι αξιολογήσεις μπορούν να χρησιμοποιηθούν ως δεδομένα αναφοράς (ground truth) για την εκπαίδευση ή την επικύρωση αλγορίθμων.

2.3.2 Αξιολόγηση Χρηστών Κοινωνικών Δικτύων

Κακόβουλοι Χρήστες και Κοινωνικά Bots. Οι αυτόματοι λογαριασμοί, γνωστοί ως social bots, παίζουν δυσανάλογα μεγάλο ρόλο στη διάδοση παραπληροφόρησης, έχοντας χρησιμοποιηθεί ακόμα και για την προσπάθεια επηρεασμού εκλογικών αναμετρήσεων [26]. Εκτιμάται ότι το 9-15% των ενεργών λογαριασμών στο Twitter είναι bots [27]. Συστήματα όπως το *Botometer* ταυτοποιούν αυτούς τους λογαριασμούς με υψηλή ακρίβεια (AUC 0.95), αναλύοντας χαρακτηριστικά από το δίκτυο, το προφίλ, το περιεχόμενο και τον χρονισμό των αναρτήσεων. Έχει αποδειχθεί ότι τα bots διαδίδουν ψευδείς ειδήσεις πολύ νωρίς στον κύκλο ζωής τους, αυξάνοντας την πιθανότητα να γίνουν δημοφιλείς (viral) [28].

Ευάλωτοι Κανονικοί Χρήστες. Πέρα από τους κακόβουλους χρήστες, ορισμένοι κανονικοί χρήστες είναι πιο επιρρεπείς (vulnerable) στην ακούσια αναμετάδοση ψευδών ειδήσεων. Αυτή η ευαλωτότητα επηρεάζεται από παράγοντες όπως: (ι) η **κοινωνική επιρροή** (π.χ. πόσοι άλλοι διαδίδουν την είδηση) και (ιι) η **αυτο-επιρροή** (π.χ. κατά πόσο η είδηση επιβεβαιώνει τις προϋπάρχουσες πεποιθήσεις του χρήστη - confirmation bias). Η ακριβής

ποσοτικοποίηση αυτών των παραγόντων παραμένει μια σημαντική ανοικτή πρόκληση στην έρευνα.

2.3.3 Συζήτηση

Η προσέγγιση που βασίζεται στην πηγή είναι εξαιρετικά κλιμακώσιμη και αποτελεσματική, ειδικά για τον χειρισμό τεράστιου όγκου ειδήσεων. Το κύριο μειονέκτημά της είναι ο κίνδυνος εσφαλμένης ταξινόμησης, δηλαδή η απόρριψη μιας αληθούς είδησης απλώς και μόνο επειδή προέρχεται από μια πηγή που έχει χαρακτηριστεί ιστορικά ως αναξιόπιστη. Για τον λόγο αυτό, η μέθοδος αυτή αποδίδει καλύτερα όταν συνδυάζεται με μεθόδους που βασίζονται στο περιεχόμενο (style-based) και τα γεγονότα (knowledge-based).

2.4 Ανίχνευση με Βάση τον Τρόπο Διάδοσης της Πληροφορίας

Η **διάδοση** (propagation) μιας είδησης στα κοινωνικά δίκτυα αποτελεί πλούσια πηγή σήματος για την ανίχνευση ψευδών ειδήσεων, καθώς το «αποτύπωμα» της πληροφορίας στο κοινωνικό γράφημα ενσωματώνει τη συλλογική συμπεριφορά των χρηστών. Σε αντιδιαστολή με τις μεθόδους που εξετάζουν το περιεχόμενο, οι τεχνικές που βασίζονται στον τρόπο διάδοσης (propagation-based) εστιάζουν στις *δομές διάδοσης* και όχι στο ίδιο το κείμενο. Οι προσεγγίσεις αυτές λαμβάνουν ως είσοδο είτε (α) **δέντρα διάδοσης** (news cascades), που αποτελούν την άμεση αναπαράσταση της εξάπλωσης της πληροφορίας, είτε (β) **ιδιοκατασκευασμένους γράφους** (self-defined graphs), που αποτυπώνουν έμμεσα και άλλες σχέσεις.

2.4.1 Ανίχνευση μέσω Δέντρων Διάδοσης

Ένα *δέντρο διάδοσης* (cascade) είναι μια δενδρική δομή $T = (V, E)$ που αποτυπώνει την εξάπλωση μιας είδησης, με ρίζα τον αρχικό χρήστη και ακμές που δηλώνουν τις αναμεταδόσεις [12]. Τα δέντρα αυτά μπορούν να αναλυθούν βάσει **βημάτων** (hop-based), επιτρέποντας τον υπολογισμό μετρικών όπως το βάθος, το πλάτος και το μέγεθος, ή βάσει **χρόνου** (time-based), με μετρικές όπως η διάρκεια ζωής και η ένταση (heat) της διάδοσης (Σχήμα 2.3).

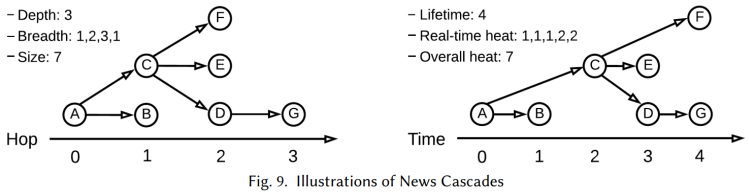


Fig. 9. Illustrations of News Cascades

Σχήμα 2.3: *Δέντρα διάδοσης με Βάσει Βημάτων vs Βάσει Χρόνου.*

Παραδοσιακά Μοντέλα Μηχανικής Μάθησης. Στο κλασικό πλαίσιο, από κάθε δέντρο εξάγεται ένα διάνυσμα χαρακτηριστικών (π.χ. μέγεθος, βάθος, δομική ιογένεια - structural virality [29]), το οποίο χρησιμοποιείται ως είσοδο σε ταξινομητές όπως SVM ή Random

Forests. Εμπειρικές μελέτες [30] έχουν αποδείξει ότι οι ψευδείς ειδήσεις διαδίδονται σημαντικά **γρηγορότερα, βαθύτερα και ευρύτερα**, με αποτέλεσμα τα δέντρα διάδοσής τους να είναι μεγαλύτερα και πιο πολύπλοκα από αυτά των αληθών ειδήσεων.

Προσεγγίσεις Βαθιάς Μάθησης. Βάσει αυτής της προσέγγισης, το δέντρο διάδοσης τροφοδοτείται απευθείας σε ένα νευρωνικό δίκτυο με δενδρική δομή, όπως τα *Αναδρομικά Νευρωνικά Δίκτυα* (Recursive Neural Networks - RvNN) με κόμβους τύπου GRU. Για παράδειγμα, το μοντέλο των Ma et al. [31] μαθαίνει αναδρομικά μια αναπαράσταση για κάθε κόμβο του δέντρου και στη συνέχεια χρησιμοποιεί ένα επίπεδο συγκέντρωσης μεγίστου (max-pooling) για να παράγει μια συνολική αναπαράσταση για ολόκληρο το δέντρο, επιτυγχάνοντας υψηλή ακρίβεια ταξινόμησης.

2.4.2 Ανίχνευση μέσω Ιδιοκατασκευασμένων Γράφων

Όταν το πραγματικό δέντρο διάδοσης δεν είναι διαθέσιμο ή θέλουμε να ενσωματώσουμε πρόσθετες πληροφορίες, κατασκευάζουμε γράφους που μπορεί να είναι ομοιογενείς, ετερογενείς ή ιεραρχικοί.

Ομοιογενείς Γράφοι. Περιέχουν έναν τύπο κόμβων και ακμών. Παραδείγματα αποτελούν τα **δίκτυα διαδοτών** (spreader networks), όπου οι κόμβοι είναι χρήστες και οι ακμές δηλώνουν σχέσεις ¨ακολούθησης¨ (follow), και τα **δίκτυα στάσης** (stance networks), όπου οι κόμβοι είναι αναρτήσεις και οι ακμές δηλώνουν υποστήριξη ή αντίθεση. Η ανίχνευση ψευδών ειδήσεων σε αυτά τα δίκτυα ανάγεται συνήθως σε πρόβλημα ταξινόμησης γράφων ή σε πρόβλημα βελτιστοποίησης [32].

Ετερογενείς Γράφοι. Περιέχουν πολλαπλούς τύπους κόμβων (π.χ. χρήστες, άρθρα, εκδότες) και ακμών. Αυτό επιτρέπει τον συνδυασμό πληροφοριών από διαφορετικές πηγές, όπως το περιεχόμενο ενός άρθρου, η μεροληψία του εκδότη και τα χαρακτηριστικά των χρηστών που το διαδίδουν. [1]

Ιεραρχικά Δίκτυα. Αποτελούν επεκτάσεις των δέντρων διάδοσης, όπου οι κόμβοι και οι ακμές οργανώνονται σε ιεραρχικά επίπεδα (π.χ. είδηση → tweets → retweets → απαντήσεις). Επιτρέπουν μια πιο πολυεπίπεδη ανάλυση της διάδοσης, συχνά μέσω μεθόδων βελτιστοποίησης σε γράφους [1].

2.4.3 Συζήτηση

Οι μέθοδοι που βασίζονται στη διάδοση είναι **ανθεκτικές** στην επιτήδευση του ύφους γραφής, καθώς αξιοποιούν τη συλλογική συμπεριφορά. Το κύριο μειονέκτημά τους, ωστόσο, είναι ότι απαιτούν να έχει ήδη πραγματοποιηθεί ένα σημαντικό μέρος της διάδοσης, καθιστώντας τες αναποτελεσματικές για την **έγκαιρη ανίχνευση** (early detection).

Μια άλλη κρίσιμη πρόκληση είναι η **εξάρτηση από τα δεδομένα εκπαίδευσης**. Πολλά μοντέλα είναι επιβλεπόμενα (supervised) και απαιτούν ετικέτες, όμως τα περισσότερα σύνολα

δεδομένων δεν παρέχουν πληροφορίες για την «πρόθεση παραπλάνησης», την οποία ακριβώς υποτίθεται ότι αποκαλύπτουν τα μοτίβα διάδοσης.

Τέλος, η έρευνα έχει δείξει ότι οι ψευδείς ειδήσεις με πολιτικό περιεχόμενο διαδίδονται διαφορετικά από αυτές σε άλλους τομείς (π.χ. επιστήμη, τρομοκρατία) [30]. Η ανακάλυψη περισσότερων τέτοιων διαφορετικών προτύπων και ο συνδυασμός μεθόδων διάδοσης, πηγής και περιεχομένου αποτελούν βασικές κατευθύνσεις για τη δημιουργία πιο εύρωστων και ερμηνεύσιμων συστημάτων.

2.5 Σύνοψη & Μελλοντική Έρευνα

Η πρόσφατη βιβλιογραφία επιβεβαιώνει ότι *καμία* μεμονωμένη προσέγγιση (γνώση, ύφος, τρόπος διάδοσης ή αξιοπιστία πηγής) δεν επαρκεί για την αποτελεσματική αντιμετώπιση του φαινομένου· Οι πλέον αποτελεσματικές λύσεις προκύπτουν από **συνδυαστικές προσεγγίσεις** οι οποίες συνδυάζουν τεκμηριωμένα χαρακτηριστικά από πολλαπλά επίπεδα [33, 34, 35]. Παρά τα ουσιαστικά βήματα προόδου, παραμένουν κρίσιμες ερευνητικές προκλήσεις όπως:

1. **Μη-παραδοσιακές μορφές.** Εντοπισμός παρωχημένων ή *μερικώς* ψευδών ειδήσεων απαιτεί δυναμικά γράφηματα γνώσης και πολυεπιγραφική μοντελοποίηση [21].

2. **Έγκαιρη ανίχνευση.** Η αρχική φάση διάδοσης διαθέτει ελάχιστο κοινωνικό πλαίσιο· απαιτούνται συμβατές διατομεακές/διγλωσσικές διαστάσεις ύφους και αποτελεσματικοί αλγόριθμοι σε «αραιά» δεδομένα [36, 37].

3. **Εντοπισμός check-worthy περιεχομένου.** Ιεράρχηση φράσεων ή θεμάτων με υψηλή ειδησεογραφική αξία και ιστορική ροπή σε παραπληροφόρηση [38].

4. **Διατομεακή / διγλωσσική γενίκευση.** Συστηματική μελέτη μοτίβων διάδοσης σε πολιτικές και μη θεματικές περιοχές, πολλαπλές γλώσσες και πλατφόρμες [30].

5. **Εξηγησιμότητα.** Ενσωμάτωση ψυχοκοινωνικών θεωριών και μηχανισμών προσοχής για διαφανείς προβλέψεις [39, 40].

6. **Στοχευμένες παρεμβάσεις.** Συνδυασμός δομικής αποκοπής κρίσιμων ακμών με διαφοροποιημένη μεταχείριση κακόβουλων / ευάλωτων χρηστών [41].

Η διερεύνηση των παραπάνω κατευθυντήριων αναμένεται να ενισχύσει ταυτόχρονα *αξιοπιστία*, *ερμηνευσιμότητα* και *διαλειτουργικότητα* των συστημάτων ανίχνευσης σε δυναμικά, πολυγλωσσικά περιβάλλοντα υψηλής πληροφοριακής ροής.

Μέρος II

Θεωρητικό Μέρος

Κεφάλαιο 3

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό παρουσιάζονται αναλυτικά όλα τα θεωρητικά εργαλεία που έχουν χρησιμοποιηθεί για την ανάπτυξη της υλοποίησης. Είναι χωρισμένο σε τρεις ενότητες:

α) Μαθηματικό Υπόβαθρο

β) Ορισμοί και Αλγόριθμοι Γραφημάτων

γ) Εργαλεία Μηχανικής Μάθησης

Το πρώτο υποκεφάλαιο περιλαμβάνει όλα τα μαθηματικά εργαλεία που είναι απαραίτητα για την κατανόηση της υλοποίησης μας. Στο δεύτερο υποκεφάλαιο παρουσιάζονται όσοι ορισμοί και αλγόριθμοι είναι απαιραίτητοι ως θεωρητικό υπόβαθρο στην εφαρμογή μας. Στο τρίτο υποκεφάλαιο περιλαμβάνονται ορισμοί και μεθοδολογίες από τον κλάδο της Μηχανικής Μάθησης και που έχουν χρησιμοποιηθεί στην υλοποίησης της Διπλωματικής Εργασίας καθώς και το πως τα εφαρμόζουμε στην δική μας υλοποίηση. Στόχος του κεφαλαίου αυτού είναι μια, όσο το δυνατόν, πιο πλήρης θεωρητική θεμελίωση των εννοιών και των εργαλείων που χρησιμοποιούνται στο Πρακτικό Μέρος.

3.1 Μαθηματικό Υπόβαθρο

3.1.1 Μετρικοί Χώροι

¨Μέτρησε ό,τι είναι μετρήσιμο και κάνε μετρήσιμο ό,τι δεν είναι.¨
Galileo Galilei

Μετρικός χώρος είναι ένας χώρος που αποτελείται από ένα μη κενό σύνολο $\mathbb{K}$ μαζί με μία συνάρτηση $d : \mathbb{K}\ x\ \mathbb{K} \mapsto \mathbb{R}$ που ορίζει την απόσταση μεταξύ των στοιχείων του συνόλου $\mathbb{K}$. Η συνάρτηση $d(\cdot,\cdot)$ για κάθε $x, y, z \in \mathbb{K}$ ικανοποιεί τις ακόλουθες ιδιότητες:

$$\begin{aligned} &1.\ d(x,x) = 0 \\ &2.\ d(x,y) \neq 0 \text{ για } x \neq y \\ &3.\ d(x,y) = d(y,x) \\ &4.\ d(x,z) \leq d(x,y) + d(y,z) \end{aligned} \tag{3.1}$$

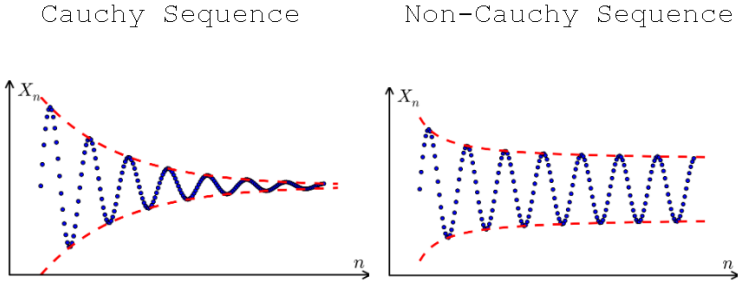
3.1.2 Χώροι Hilbert

"Wir müssen wissen.Wir werden wissen."
David Hilbert

Μία ακολουθία $x_1, x_2, x_3, \ldots$ σε έναν μετρικό χώρο (X, d) λέγεται **ακολουθία Cauchy** αν για κάθε πραγματικό αριθμό $r > 0$ υπάρχει φυσικός αριθμός n_0 τέτοιος ώστε για κάθε $n, m > n_0$:

$$d(x_n, x_m) < r \tag{3.2}$$

Όπως βλέπουμε και στο Σχήμα 3.1, η πρώτη εικόνα απεικονίζει μία ακολουθία Cauchy η οποία συγκλίνει σε ένα τελικό σημείο, ενώ στην δεύτερη εικόνα, η ακολουθία που εικονίζεται δεν είναι ακολουθία Cauchy.



Σχήμα 3.1: *Cauchy and Non-Cauchy Sequence.*

Ένας μετρικός χώρος (X,d) λέγεται **πλήρης** αν ισχύει ότι το όριο κάθε ακολουθίας Cauchy του X είναι ένα σημείο που ανήκει στο X.

Έστω ένας χώρος $E = \mathbb{K}^n$ εφοδιασμένος με την πράξη του εσωτερικού γινομένου. Αν ο χώρος αυτός είναι πλήρης ως προς την μετρική που ορίζει το εσωτερικό γινόμενο,τότε, ο χώρος $(E, \langle \cdot, \cdot \rangle)$ λέγεται χώρος Hilbert.

3.1.3 Νόρμα Frobenius,Απόσταση Wasserstein και Βαρύκεντρα

Η **νόρμα Frobenius** για έναν πίνακα $A_{n\times n}$ ορίζεται ως ακολούθως:

$$\|A\|_F = \sqrt{\langle A, A \rangle} = \sqrt{\text{trace}(A^T A)} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{i,j}|^2} \tag{3.3}$$

Το **εσωτερικό γινόμενο Frobenius** μεταξύ δύο πινάκων $\mathbf{A}_{n\times n}$, $\mathbf{B}_{n\times n} \in \mathbf{C}^{n\times n}$ ορίζεται ως:

$$\langle \mathbf{A}, \mathbf{B} \rangle_{\text{F}} = \sum_{i,j} \overline{a_{ij}} b_{ij} = \text{Tr}\,(\mathbf{A}^* \mathbf{B}) \tag{3.4}$$

Για δύο διανύσματα, $\vec{x_1} \in \mathbb{R}^{n_1}$, $\vec{x_2} \in \mathbb{R}^{n_2}$, ένας **πίνακας μεταφοράς** είναι ένας πίνακας $T \subseteq R^{n_1 x n_2}$ τέτοιος ώστε:

$$1) \quad T \cdot \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{n_2 \times 1} = \vec{x_1} \tag{3.5}$$

$$2) \quad \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}_{1 \times n_1} \cdot T = \vec{x_2}^T$$

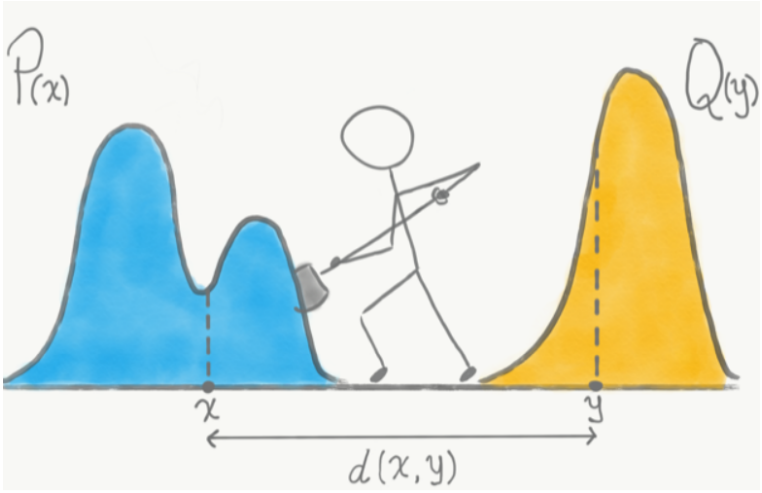
Για δύο διανύσματα $\vec{x_1} \in \mathbb{R}^{n_1}$, $\vec{x_2} \in \mathbb{R}^{n_2}$ και έναν πίνακα κόστους $C \in \mathbb{R}^{n_1 \times n_2}$ που ορίζει το κόστος μετατροπής μιας μονάδας "μάζας" μιας συνιστώσας του $\vec{x_1}$ σε μονάδα "μάζας" μιας συνιστώσας του $\vec{x_2}$, η **απόσταση Wasserstein** ορίζεται ως εξής [3]:

$$W^C(\vec{x_1}, \vec{x_1}) = \min_T \langle \mathbf{C}, \mathbf{T} \rangle_{\text{F}} \tag{3.6}$$

Στο Σχήμα 3.2 βλέπουμε διαισθητικά την φυσική ερμηνεία του μεγέθους αυτού. Για δύο κατανομές $P(x)$, $Q(y)$ για τις οποίες ισχύει:

$$\int P(x)\, dx = \int Q(y)\, dy \tag{3.7}$$

και για μία συνάρτηση $d(x, y)$ η οποία είναι μια συνάρτηση κόστους μετατροπής (στο Σχήμα μας μεταφοράς) μιας μονάδας από το $P(x) \mapsto Q(y)$. Έχουμε μία κατανομή $P(x)$ και θέλουμε να την μετατρέψουμε σε μία κατανομή $Q(y)$. Γνωρίζοντας τη συνάρτηση $d(x, y)$ η οποία είναι μια συνάρτηση κόστους μετατροπής (στο Σχήμα μας μεταφοράς) μιας μονάδας από το $P(x) \mapsto Q(y)$, η απόσταση *Wasserstein* ορίζεται ως το ελάχιστο συνολικό κόστος μετατροπής της κατανομής $P(x)$ στη $Q(y)$.



Σχήμα 3.2: *Φυσική ερμηνεία απόστασης Wasserstein.*

Ο **βέλτιστος πίνακας μεταφοράς** είναι ο πίνακας μεταφοράς από τον οποίο προκύπτει η απόσταση *Wasserstein* και ορίζεται από την Σχέση 3.6:

$$T_{\text{opt}} = \underset{T}{\text{argmin}} \langle T \ , \ C \rangle_{\text{F}} \tag{3.8}$$

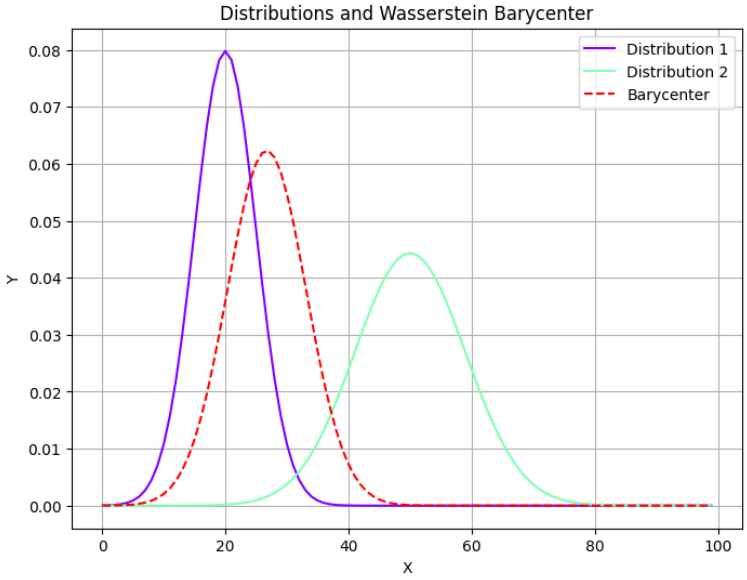
Μία βασική προϋπόθεση για να μετρήσουμε απόσταση *Wasserstein* μεταξύ των $\vec{x_1}$,$\vec{x_2}$ είναι να ισχύει ότι θα έχουν ίδια L1 - νόρμα, δηλαδή να ισχύει η ακόλουθη σχέση:

$$\|\vec{x_1}\|_1 = \|\vec{x_2}\|_1 \tag{3.9}$$

Για ένα σύνολο στοιχείων $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_k} \in \mathbb{R}^n$ και έναν πίνακα κόστους $C^{n \times n}$, το βαρύκεντρο ορίζεται ως [3]:

$$barycenter = \arg\min_{c} \sum_{i=1}^{n} W^C(x_i, c) \tag{3.10}$$

Για παράδειγμα, στο Σχήμα 3.3 εικονίζονται οι κατανομές $X_1, \sim \mathcal{N}(20, 5^2)$ και η $X_2 \sim \mathcal{N}(50, 9^2)$ και το βαρύκεντρο. Το βαρύκεντρο είναι μία κατανομή που ελαχιστοποιεί το άρθοισμα της απόστασης *Wasserstein* των κατανομών X_1, X_2, δεδομένου ενός πίνακα κόστους.



Σχήμα 3.3: *Κατανομές X_1, X_2 και Βαρύκεντρο.*

3.2 Ορισμοί και Αλγόριθμοι Γραφημάτων

3.2.1 Βασικοί Ορισμοί

¨Αρχή σοφίας η των ονομάτων επίσκεψις¨
Σωκράτης

Για ένα γράφημα $G = (V, E)$, ορίζουμε ως $V(G)$ το σύνολο των κορυφών του και ως $E(G)$ το σύνολο των ακμών του.

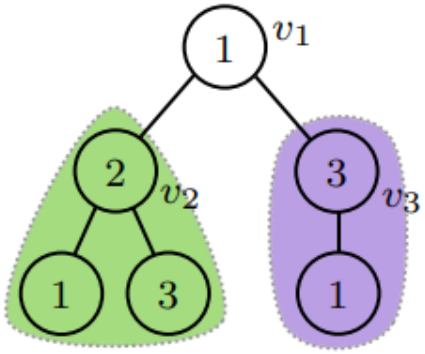
Έστω ένα αλφάβητο από n επιγραφές $\Sigma = \{l_1, l_2, ...l_n\}$. **Γράφημα με Επιγραφές**, λέγεται ένα γράφημα $G = (V(G), E(G), l)$ όπου $l(\cdot)$ είναι μία συνάρτηση $l : V(G) \mapsto \Sigma$ [2]. Έτσι, η συνάρτηση αυτή αντιστοιχίζει κάθε κόμβο του γραφήματος G σε μία επιγραφή του αλφάβητου Σ. Μέσω αυτής της κωδικοποίησης έχουμε μία πιο πλούσια αναπαράσταση του γραφήματος καθώς ενσωματώνουμε στο γράφημα πληροφορία που σχετίζεται με τους κόμβους. Οι κόμβοι του γραφήματος κατηγοριοποιούνται με βάση τις επιγραφές τους, όπου κάθε διαφορετική επιγραφή ορίζει μια ξεχωριστή κλάση κόμβων του γραφήματος..

Για ένα κατευθυνόμενο δέντρο T με μία συνεκτική συνιστώσα και επομένως μία ρίζα, ορίζουμε [3] :

1. Την ρίζα του T ως r(T)
2. Το σύνολο των υποδέντρων τα οποία έχουν ως ρίζα τα παιδιά της ρίζας του δέντρου T ως F(r(T))

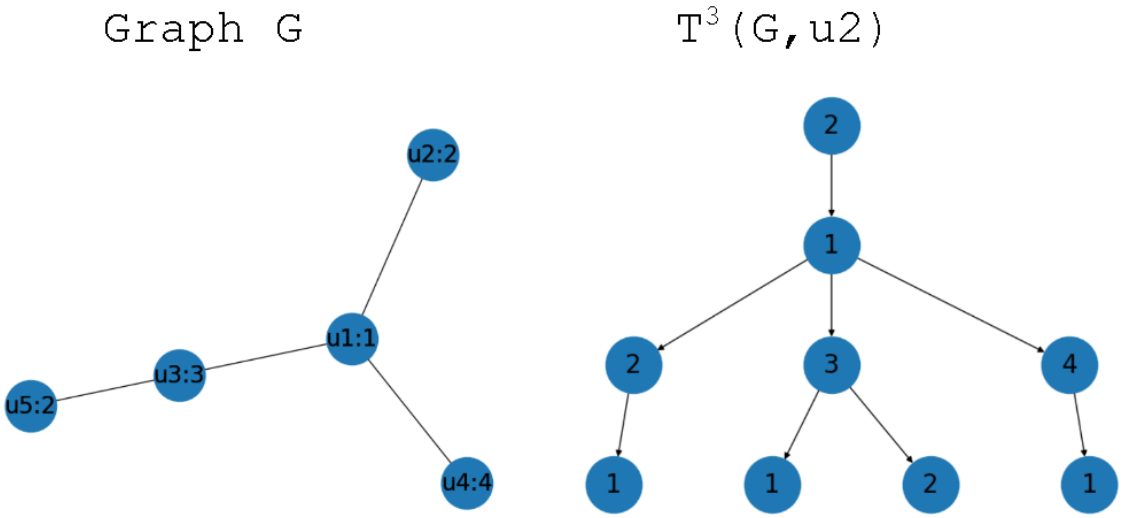
Για παράδειγμα, για το δέντρο T του Σχήματος 3.4 το r(T) είναι ο κόμβος u_1. Τα παιδιά του r(T) = u_1 είναι οι κόμβοι u_2, u_3. Το F(r(T)) για το συγκεκριμένο δέντρο είναι το σύνολο

που αποτελείται από τα δύο υποδέντρα, το πράσινο και το μωβ, όπου έχουν ως ρίζες τους κόμβους u_2, u_3. Στο συγκεκριμένο δέντρο, το αλφάβητο είναι το $\Sigma = \{1, 2, 3\}$ και οι επιγραφές των κόμβων u_1, u_2, u_3 είναι $l(u_1) = 1$, $l(u_2) = 2$, $l(u_3) = 3$.



Σχήμα 3.4: *Ρίζα r(T) = u_1 i ό F(r(T)) [2]*

Για παράδειγμα, για το δέντρο T του Σχήματος 3.4 το r(T) είναι ο κόμβος u_1. Τα παιδιά του r(T) = u_1 είναι οι κόμβοι u_2, u_3. Το F(r(T)) για το συγκεκριμένο δέντρο είναι το σύνολο που αποτελείται από τα δύο υποδέντρα, το πράσινο και το μωβ, όπου έχουν ως ρίζες τους κόμβους u_2, u_3. Στο συγκεκριμένο δέντρο, το αλφάβητο είναι το $\Sigma = \{1, 2, 3\}$ και οι επιγραφές των κόμβων u_1, u_2, u_3 είναι $l(u_1) = 1$, $l(u_2) = 2$, $l(u_3) = 3$.



Σχήμα 3.5: *Γράφημα G και ένα i-Δέντρο Ανάπτυξης.*

Ο τυπικός ορισμός του Δέντρου Ανάπτυξης δίνεται από την ακόλουθη αναδρομική σχέση [42]:

$$T^i(G,u) = \begin{cases} Tree(u) & , \ i = 0 \\ Tree(u, T^{i-1}(G, N(u)) & : \ i > 0 \end{cases} \tag{3.11}$$

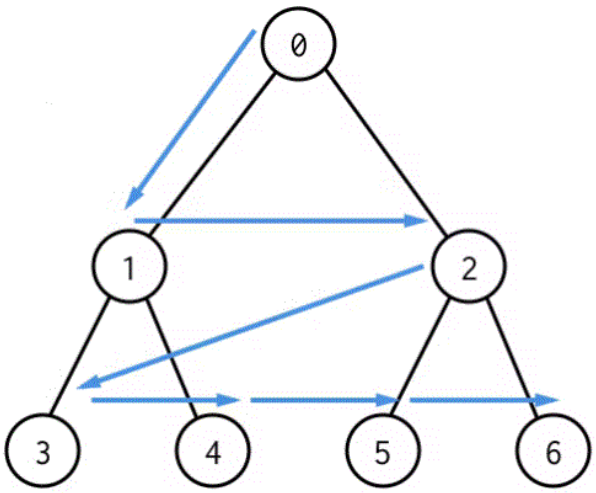
όπου $Tree(u)$ είναι ένα δέντρο που περιλαμβάνει μόνο τον κόμβο u με την επιγραφή του, και $Tree(u, T^{i-1}(G, N(u)))$ είναι ένα δέντρο με ρίζα τον κόμβο u, στο οποίο συνδέονται τα

υποδέντρα $Tree(v, T^{i-1}(G, v))$, με $v \in Neighbour(u)$, και περιλαμβάνουν τις επιγραφές που έχουν οι κόμβοι στο γράφημα G.

Τα δέντρα ανάπτυξης στον αλγόριθμο Weisfeiler Lehman έχουν ουσιώδη ρόλο καθώς όπως θα δούμε αναλυτικότερα στο Κεφάλαιο 3.2.4, κατά τη διαδικασία μετατροπής ενός γραφήματος σε μία διανυσματική αναπαράσταση, κάθε συνιστώσα του διανύσματος χαρακτηριστικών που προκύπτει αποτελεί ένα συγκεκριμένο δέντρο ανάπτυξης [3].

3.2.2 Κωδικοποίηση Δέντρων με χρήση Συμβολοσειρών

Δεδομένης μιας συλλογής δέντρων με επιγραφές στους κόμβους, επιδιώκουμε την κατασκευή ενός συνόλου δέντρων με επιγραφές τέτοιου ώστε κάθε ζεύγος στοιχείων σε αυτό το σύνολο να είναι ισομορφικά. Η βασική ιδέα είναι να μετασχηματίσουμε τα δέντρα σε μια ενιαία αναπαράσταση, ώστε να μπορούμε να ελέγξουμε γρήγορα αν δύο δέντρα είναι ισομορφικά. Ένας βασικός ενδιάμεσος αλγόριθμος είναι ο **Αλγόριθμος Κωδικοποίσης Δέντρων με επιγραφές με χρήση Συμβολοσειρών**,ο οποίος λαμβάνει ένα γράφημα με επιγραφές και το αντιστοίχιζει σε κάποια συμβολοσειρά.



Σχήμα 3.6: *BFS Διάσχιση Δέντρου*

Βασική Ιδέα του Αλγορίθμου

1. **Ενοποίηση Ισομορφικών Δέντρων:** Ξεκινάμε από ένα σύνολο δέντρων $\mathcal{T}$ με επιγραφές και επιδιώκουμε να αναγνωρίσουμε ποια είναι ισομορφικά μεταξύ τους. Κάθε ισομορφικό σύνολο δέντρων μπορεί να αναπαρασταθεί από έναν κοινό αντιπρόσωπο (ένα «πρότυπο» δέντρο). Έτσι, διαχωρίζουμε το σύνολο $\mathcal{T}$ σε ανεξάρτητα υποσύνολα $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$, όπου κάθε δύο στοιχεία από κάποιο από τα σύνολα $\mathcal{T}_i$ είναι ισομορφικά.

2. **Μετατροπή σε Μοναδικό Αντιπρόσωπο:** Δύο ισομορφικά δέντρα μετατρέπονται σε έναν κοινό αντιπρόσωπο· έτσι, ο αντιπρόσωπος αυτός αντιστοιχεί σε μια κλάση ισομορφισμού. Τον αλγόριθμο για το πως λειτουργεί αυτό το βήμα θα το δούμε αναλυτικότερα στο Κεφάλαιο 3.2.3 που αφορά τον Ισομορφισμό Δέντρων με επιγραφές στους κόμβους.

3. **Κωδικοποίηση μέσω BFS σε Συμβολοσειρά:** Αφού επιλέξουμε τον αντιπρόσωπο της κλάσης Ισομορφισμού, μετατρέπουμε τη δομή του σε συμβολοσειρά. Η κωδικοποίηση πραγματοποιείται διασχίζοντας το δέντρο κατά επίπεδα (BFS traversal), αριστερά προς τα δεξιά, όπως στο Σχήμα 3.6. Με αυτό τον τρόπο έχουμε σε μία συμβολοσειρά αποθηκευμένη την τοπολογία του δέντρου μαζί με την πληροφορία που φέρουν οι κόμβοι.

4. **Έλεγχος Ισομορφίας μέσω Συμβολοσειρών:** Δύο δέντρα είναι ισομορφικά αν, μετά τη μετατροπή τους στις αντίστοιχες συμβολοσειρές οι συμβολοσειρές αυτές ταυτίζονται.

Τυπική Διατύπωση:

Δεδομένου ενός συνόλου Δέντρων με επιγραφές $\mathcal{T}$, ορίζουμε:

- $f(\cdot) : \mathcal{T} \to \mathcal{T}$ μια συνάρτηση που αντιστοιχεί κάθε δέντρο στον «αντιπρόσωπο» της κλάσης ισομορφισμού του.
- $s(\cdot) : \mathcal{T} \to \Sigma^*$ μια συνάρτηση που αντιστοιχεί κάθε δέντρο με επιγραφές στους κόμβους του σε μια συμβολοσειρά.

Βάσει αυτών των ορισμών, δύο δέντρα T,T' είναι ισομορφικά αν:

$$s(f(T)) = s(f(T')) \tag{3.12}$$

Η απαραίτητη πληροφορία όσον αφορά την δομή του δέντρου είναι :

1. Η δομή του δέντρου, δηλαδή η αρίθμηση των κόμβων και πως οι κόμβοι συνδέονται μεταξύ τους.
2. Οι επιγραφές που φέρουν οι κόμβοι.

Έτσι, για να κωδικοποιήσουμε όλη τη απαιτούμενη πληροφορία την οποία περιλαμβάνει ένα δέντρο, ακολουθούμε την μεθοδολογία που αναφέρουμε στον Αλγόριθμο 3.1.

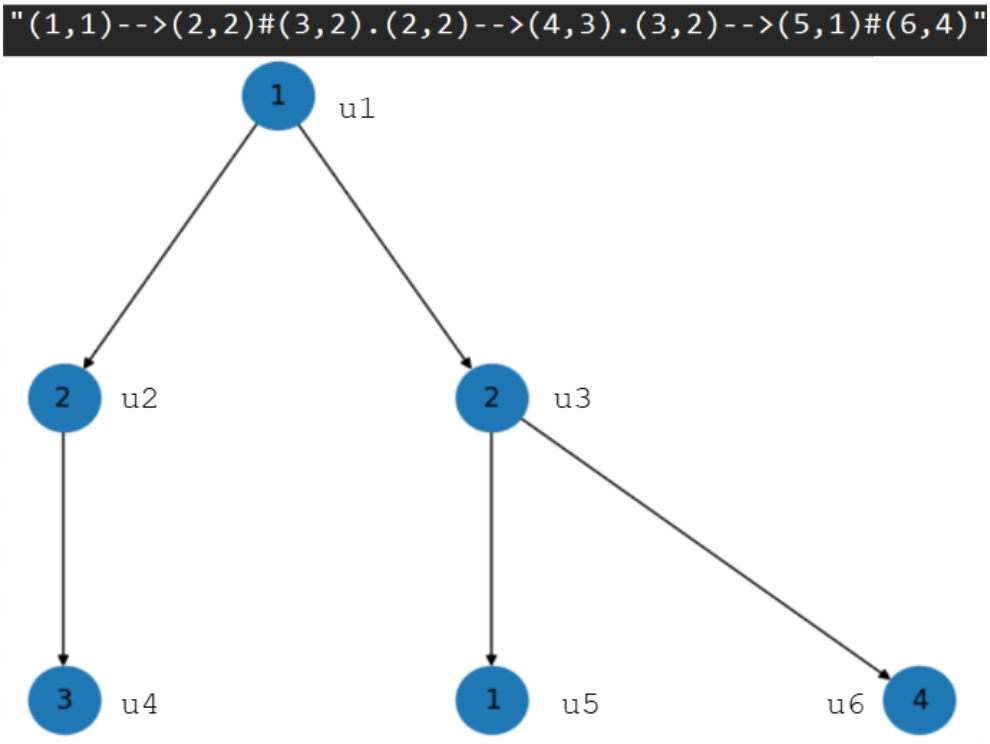
Για να αντιστοιχίσουμε δύο ισομορφικά δέντρα στην ίδια συμβολοσειρά, είναι σημαντικό οι κόμβοι να είναι αριθμημένοι από αριστερά προς τα δεξία και από τη ρίζα προς τα φύλλα. Έτσι, κάθε συμβολοσειρά αντιπροσωπεύει ένα σύνολο δέντρων που είναι ισομορφικά μεταξύ τους.

Στο Σχήμα 3.7 βλέπουμε την τιμή της συμβολοσειράς της κωδικοποίησης για ένα συγκεκριμένο δέντρο με επιγραφές T.

Αλγοριθμος 3.1: *Κωδικοποίηση Δέντρου με επιγραφές στους Κόμβους με χρήση Συμβολοσειρών*

Είσοδος: Δέντρο T με επιγραφές πάνω στους κόμβους $\in \{1, 2, 3, \ldots, n\}$
Έξοδος: Συμβολοσειρά στην οποία περιλαμβάνεται όλη η πληροφορία για την δομή του δέντρου T

```
treeStrEnc = ""
for u ∈ V(T) do
    if outDegree(u) ≠ 0 then
        nodeChildren = {v ∈ V(T) | (u, v) ∈ E(T)}
        nodeChldStr = ""
        for nodeChild ∈ nodeChildren do
            nodeChldStr = nodeChldStr + (nodeChild, label(nodeChild)) + "#"
        end for
        Διάγραψε το τελευταίο χαρακτήρα από το nodeChldStr
        nodeChldStr = (u, label(u)) + "-->" + nodeChldStr
        treeStrEnc = treeStrEnc + nodeChldStr + "."
    end if
end for
Διάγραψε το τελευταίο χαρακτήρα από το treeStrEnc
Επέστρεψε τη μεταβλητή treeStrEnc
```



Σχήμα 3.7: *Κωδικοποίηση Δέντρου T με χρήση Συμβολοσειρών.*

Ο αλγόριθμος **Κωδικοποίησης Δέντρου με επιγραφές στους Κόμβους με χρήση Συμβολοσειρών** έχει ως στόχο να μετατρέψει τη δομή ενός δέντρου T σε μια μοναδική συμβολοσειρά που περιλαμβάνει όλες τις πληροφορίες για τη δομή και τις επιγραφές των κόμβων του. Ακολουθούν αναλυτικά τα βήματα του:

1. **Αρχικοποίηση:** Δημιουργείται μια κενή συμβολοσειρά *treeStrEnc* η οποία θα περιέχει

το αποτέλεσμα της κωδικοποίησης.

2. **Διάσχιση Κόμβων:** Ο αλγόριθμος επαναλαμβάνει τα παρακάτω για κάθε κόμβο u του δέντρου:

   (α') Ελέγχεται αν ο κόμβος u έχει παιδιά, δηλαδή αν το $outDegree(u) \neq 0$.

   (β') Αν ο κόμβος u έχει παιδιά:

   i. Δημιουργείται ένα σύνολο *nodeChildren* που περιλαμβάνει όλους τους κόμβους v που είναι παιδιά του u (δηλαδή για τους οποίους ισχύει $(u, v) \in E(T)$).

   ii. Δημιουργείται μια κενή συμβολοσειρά *nodeChldStr* για την προσωρινή αποθήκευση των πληροφοριών των παιδιών του κόμβου u.

   iii. Για κάθε παιδί *nodeChild* του u, προστίθεται στη συμβολοσειρά *nodeChldStr* το ζεύγος $(nodeChild, label(nodeChild))$, ακολουθούμενο από το χαρακτήρα ¨#¨.

   iv. Διαγράφεται ο τελευταίος χαρακτήρας ¨#¨ από τη συμβολοσειρά *nodeChldStr*.

   v. Στη συνέχεια, προστίθεται η πληροφορία του κόμβου u (δηλαδή το ζεύγος $(u, label(u))$) στην αρχή της *nodeChldStr*.

   vi. Τέλος, προστίθεται η *nodeChldStr* στη συνολική συμβολοσειρά *treeStrEnc*, διαχωριζόμενη με τον χαρακτήρα '.'.

3. **Τελική Επεξεργασία:** Μετά την ολοκλήρωση της διάσχισης όλων των κόμβων, διαγράφεται ο τελευταίος χαρακτήρας '.' από τη συμβολοσειρά *treeStrEnc*.

4. **Επιστροφή Αποτελέσματος:** Η τελική συμβολοσειρά *treeStrEnc* επιστρέφεται ως αποτέλεσμα.

Με τη διαδικασία αυτή, έχουμε αποθηκεύσει όλη την απαιτούμενη πληροφορία σε μία συμβολοσειρά, κάνοντας έτσι ταχύτερο τον αλγόριθμο δημιουργίας ενός συνόλου δέντρων τέτοιων ώστε δύο οποιαδήποτε στοιχεία του να μην είναι ισομορφικά μεταξύ τους.

3.2.3 Ισομορφισμός Δέντρων

Στην εφαρμογή μας προκύπτει το εξής πρόβλημα. Έχουμε μια συλλογή δέντρων με επιγραφές και θέλουμε να βρούμε ένα θεμελιώδες σύνολο τέτοιο ώστε δύο οποιαδήποτε στοιχεία του να είναι μη ισομορφικά μεταξύ τους. Γι' αυτό είναι σημαντικό να βρούμε μία διαδικασία μέσω της οποίας να εξετάζουμε αν δύο Δέντρα με επιγραφές είναι ισομορφικά. Αυτό το επιτυγχάνουμε με τον αλγόριθμο που παρουσιάζουμε σε αυτό το κεφάλαιο.

Τυπική Διατύπωση Προβλήματος

Έστω δέντρο T με επιγραφές στους κόμβους που ανήκουν στο αλφάβητο $\Sigma = \{1, 2, \ldots, n\}$. Ο στόχος είναι να τροποποιηθεί το δέντρο αυτό και να μετασχηματιστεί σε μία μοναδική

μορφή T'.Έτσι, βάσει αυτής της μεθόδου για δύο ισομορφικά δέντρα T_1 και T_2 μετά την εφαρμογή της αυτής της διαδικασίας να ισχύει:

$$T_1' = T_2'$$

Η διαδικασία περιλαμβάνει την εξίσωση των βαθών, την ταξινόμηση κόμβων και την κωδικοποίηση μέσω συμβολοσειρών, με σκοπό τη δημιουργία ενός μοναδικού αντιπροσώπου T' για κάθε κλάση ισομορφισμού.

Τυπικός Ορισμός Ισομορφισμού

Δύο γραφήματα G_1, G_2 είναι **ισομορφικά** αν ισχύουν τα ακόλουθα:

1. $|V(G_1)| = |V(G_2)|$
2. $|E(G_1)| = |E(G_2)|$
3. Υπάρχει συνάρτηση $f : V(G_1) \to V(G_2)$, τέτοια ώστε:
   $\forall e = (u_i, u_j) \in E(G_1) \Rightarrow (f(u_i), f(u_j)) \in E(G_2)$

Ο ορισμός που δώσαμε παραπάνω είναι ο γενικός ορισμός ισομορφισμού δύο γραφημάτων. Ωστόσω, σε περίπτωση που έχουμε Γραφήματα με επιγραφές υπάρχει σαν επιπλέον προυπόθεση οι δύο κόμβοι $u_i, f(u_i)$ των γραφημάτων G_1, G_2 να έχουν την ίδια επιγραφή, δηλαδή να ισχύει:

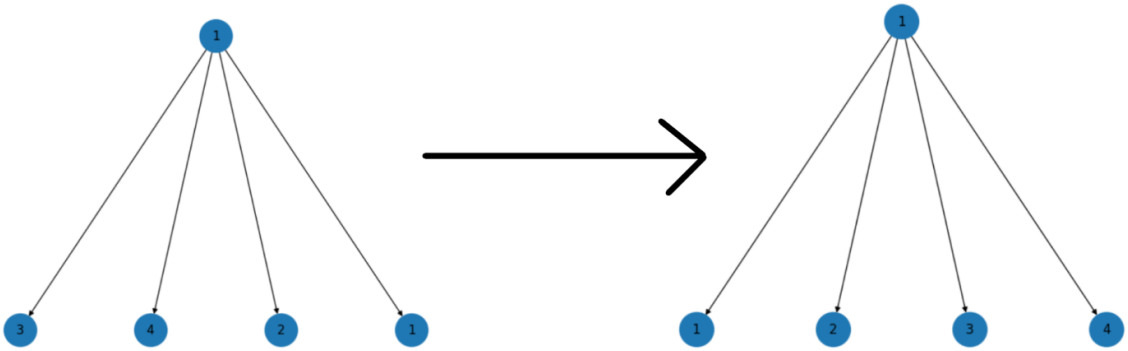
$$l(u_i) = l(f(u_i))$$

Βασική Ιδέα του Αλγορίθμου

Ο αλγόριθμος υπολογισμού του αντιπροσώπου μιας κλάσης ισομορφικών δέντρων με επιγραφές μετατρέπει ένα δέντρο T με επιγραφές στους κόμβους σε μια μοναδική κανονικοποιημένη μορφή T'. Η κανονικοποιημένη μορφή διασφαλίζει ότι οποιαδήποτε δύο ισομορφικά δέντρα T_1, T_2 αντιστοιχίζονται στο ίδιο δέντρο T'. Τα βασικά βήματα του αλγορίθμου είναι τα εξής:

1. **Εξίσωση των Βαθών**: Τα φύλλα με διαφορετικά βάθη αποκτούν το ίδιο βάθος μέσω προσθήκης κόμβων με επιγραφή 0 στα φύλλα που δεν έχουν μέγιστο βάθος στο γράφημα. (Σχήμα 3.9)
2. **Ταξινόμηση Κόμβων**: Οι κόμβοι με κοινό γονέα ταξινομούνται με βάση τις επιγραφές τους (Σχήμα 3.8).
3. **Κωδικοποίηση Υποδέντρων**: Τα υποδέντρα στα δύο τελευταία επίπεδα μετατρέπονται σε συμβολοσειρές. (Σχήμα 3.12)
4. **Σύμπτυξη Υποδέντρων**: Τα δύο τελευταία επίπεδα του δέντρου αντικαθίστανται από έναν κόμβο, του οποίου η επιγραφή προκύπτει από την κωδικοποίηση. (Σχήματα 3.13 ,3.14)

5. **Αφαίρεση Βοηθητικών Κόμβων**: Οι κόμβοι με επιγραφή 0 αφαιρούνται, αφήνοντας μόνο τη δομή και τις επιγραφές που περιγράφουν το δέντρο.



Σχήμα 3.8: *Ταξινόμηση κόμβων με κοινό γονέα βάσει των επιγραφών*

Αλγοριθμος 3.2: *Υπολογισμός Αντιπρόσωπου Κλάσης Ισομορφικών Δέντρων*

Είσοδος: Tree T with labels $\in \{1, 2, 3, \ldots, n\}$
Έξοδος: Isomorphic Tree Representative T'
-MaxTreeDepth = Maximum distance between the root and the leaves
-Add to leaves nodes whose distance is leafDist < MaxTreeDepth path with nodes with label 0 and size = MaxTreeDepth-leafDist
while *treeDepth* ≥ 2 **do**
-Sort leaves of T which have the same parent according to their labels.
-SubtreeStrs = Last Two Level Subtree Strings.
-MaxTreeLabel = Maximum Label of the tree T.
-Sort SubtreeStrs and create a physical enumaration of these strings begining the enumaration with beggining value MaxTreeLabel + 1.
-Replace each subtree of the last two levels with one node and label that is defined from the previous step.
end while
-Sort all Leaves of the tree T.
while *treeDepth* $\neq$ *MaxTreeDepth* **do**
-Replace each node whose label didn't exist initial with its equivalent subtree.
end while
-Delete nodes with zero labels.
-Return the isomorphic tree representative T.

Ανάλυση Βημάτων του Αλγορίθμου

Βήμα 1: Υπολογισμός Μέγιστου Βάθους Το μέγιστο βάθος MaxTreeDepth ορίζεται ως η μεγαλύτερη απόσταση από τη ρίζα μέχρι οποιοδήποτε φύλλο. Αυτή η πληροφορία χρησιμοποιείται για την εξίσωση των βαθών των φύλλων.

Βήμα 2: Εξίσωση Βαθών Φύλλων Προστίθενται κόμβοι με επιγραφή 0 στα φύλλα που έχουν βάθος μικρότερο από MaxTreeDepth, ώστε όλα τα φύλλα να αποκτήσουν το ίδιο βάθος όπως φαίνεται στο Σχήμα 3.9. Το βήμα αυτό είναι απαραίτητο γιατί μας μειώνει σημαντικά τον απαιτούμενο χρόνο για την ανάλυση όλων των επιμέρους περιπτώσεων.

Βήμα 3: Ταξινόμηση Κόμβων Οι κόμβοι με κοινό γονέα ταξινομούνται με βάση τις επιγραφές τους, διασφαλίζοντας σταθερή σειρά ανεξάρτητα από την αρχική δομή όπως βλέπουμε στο Σχήμα 3.9.

Βήμα 4: Κωδικοποίηση Υποδέντρων Τα υποδέντρα στα δύο τελευταία επίπεδα μετατρέπονται σε συμβολοσειρές, οι οποίες περιγράφουν τη δομή και τις επιγραφές των κόμβων. Από τα Σχήματα 3.12 και 3.13 βλέπουμε πως κωδικοποιούνται τα υποδέντρα σε συμβολοσειρές.

Βήμα 5: Λεξικογραφική Ταξινόμηση Συμβολοσειρών Οι συμβολοσειρές ταξινομούνται λεξικογραφικά, και σε κάθε συμβολοσειρά αντιστοιχίζεται μια νέα επιγραφή, ξεκινώντας από MaxTreeLabel + 1. Στο Σχήμα 3.13 βλέπουμε πως έχει γίνει η Λεξικογραφική Ταξινόμηση των συμβολοσειρών που προέκυψαν από το προηγούμενο βήμα.

Βήμα 6: Σύμπτυξη Υποδέντρων Τα υποδέντρα αντικαθίστανται από έναν κόμβο με την αντίστοιχη νέα επιγραφή, μειώνοντας το βάθος του δέντρου. Στο Σχήμα 3.14 βλέπουμε σε αντιστοιχεί ο νέος κόμβος και πως ένα υποδέντρο κωδικοποιήθηκε στον αριθμό της νέας επιγραφής.

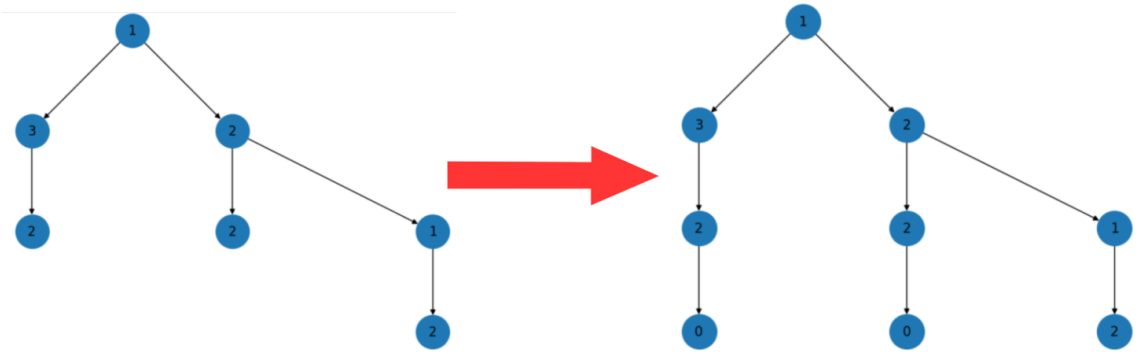
Βήμα 7: Επαναληπτική Κανονικοποίηση Η διαδικασία επαναλαμβάνεται έως ότου το δέντρο αποκτήσει βάθος 1. Όταν το δέντρο έχει μόνο δύο επίπεδα, ταξινομούμε τα παιδιά του r(T). Στη συνέχεια κάνουμε "ξεδίπλωση" του δέντρου, αντικαθιστώντας κάθε κόμβο που έχει κάποια επιγραφή l_i με το υποδέντρο στο οποία αντιστοιχεί αυτή η επιγραφή.

Βήμα 8: Αφαίρεση Βοηθητικών Κόμβων Οι κόμβοι με επιγραφή 0 που προστέθηκαν κατά την εξίσωση των βαθών αφαιρούνται, αφήνοντας μόνο το αρχικό τμήμα του δέντρου.

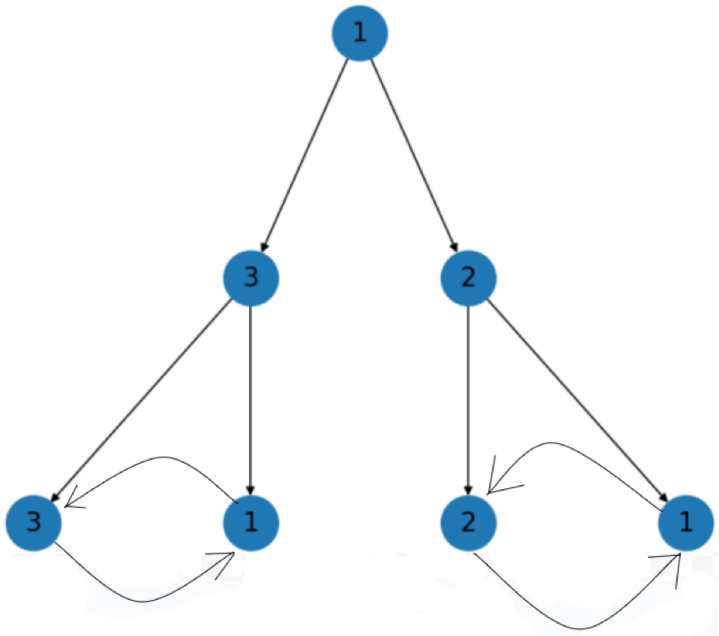
Βήμα 9: Επιστροφή Αποτελέσματος Το αποτέλεσμα είναι η κανονικοποιημένη μορφή T', η οποία είναι μοναδική για κάθε κλάση ισομορφικών δέντρων.

Μέσω αυτού του αλγορίθμου κάνουμε το πρώτο βήμα που είναι απαραίτητο για απάντηση στο ερώτημα αν δύο δέντρα T_1, T_2 είναι ισομορφικά βάσει των επιγραφών του. Στη συνέχεια,

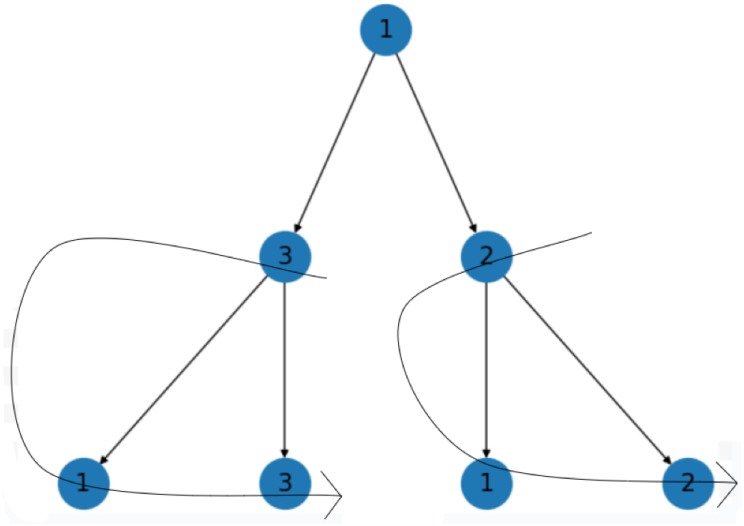
με χρήση της διαδικασίας που δείξαμε στο Κεφάλαιο 3.2.2 , αντιστοιχίζουμε τα μετασχηματισμένα δέντρα σε συμβολοσειρές και τις συγκρίνουμε. Αν είναι ίδιες, τα αρχικά δέντρα T_1, T_2 είναι ισομορφικά, διαφορετικά δεν είναι.



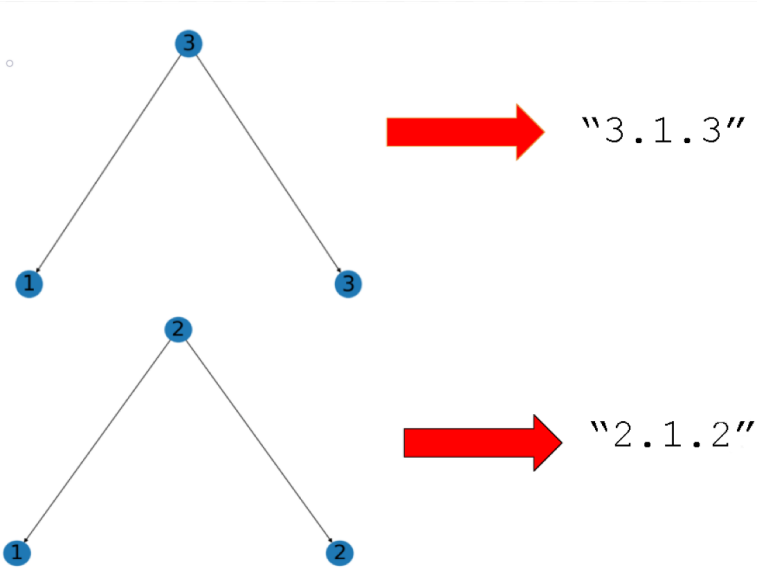
Σχήμα 3.9: *Ισοϋψής μετατροπή δέντρου*



Σχήμα 3.10: *Ταξινόμηση φύλλων βάσει επιγραφής.*



Σχήμα 3.11: *Διάτρεξη κόμβων υποδέντρων για συμπίεση.*

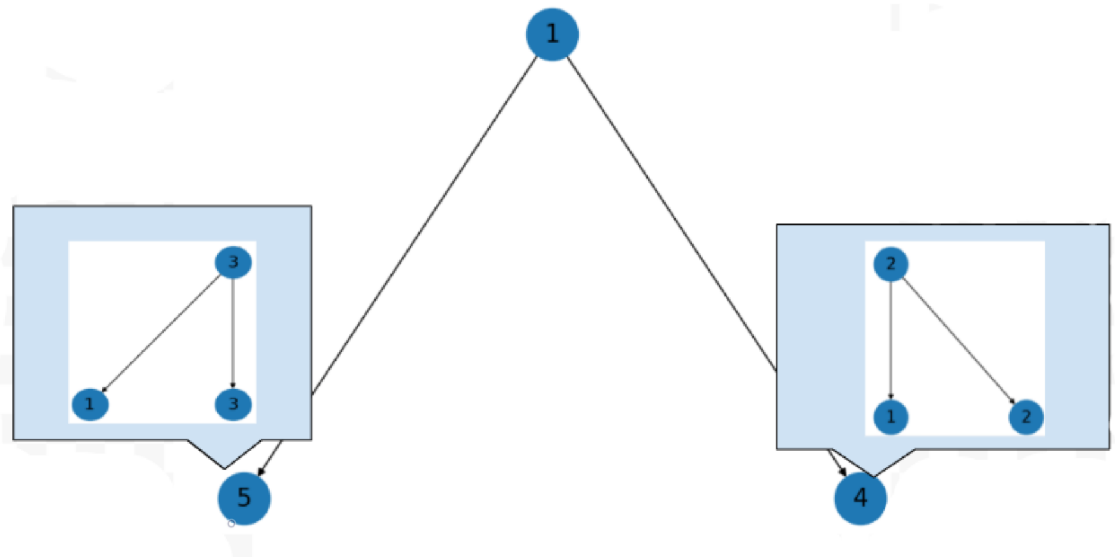


Σχήμα 3.12: *Αντιστοίχιση υποδέντρων σε συμβολοσειρές.*

```
3.1.3}  sort()  2.1.2}   enumaration()   2.1.2 --> 4
    }   ===>       }     ===>
2.1.2}          3.1.3}   maxLabel = 3    3.1.3 --> 5
```

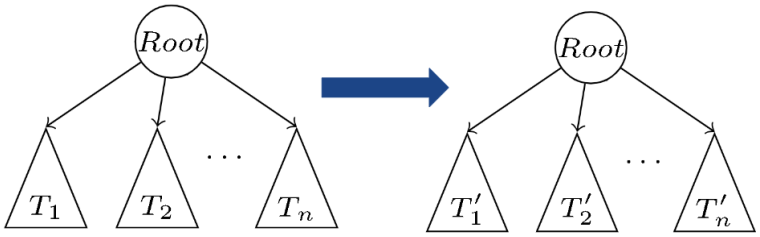
Σχήμα 3.13: *Συμπίεση υποδέντρων.*

Απόδειξη ορθότητας : Στο τελευταίο βήμα του αλγορίθμου πριν ξεκινήσει η διαδικασία της ξεδίπλωσης του δέντρου,όπου προκύπτει ένα δέντρο με δύο επίπεδα και τα φύλλα είναι ταξινομημένα βάσει της επιγραφής τους, κάθε κόμβος αντιστοιχεί σε ένα υποδέντρο. Οι κόμβοι είναι ταξινομημένοι λεξικογραφικά. Έτσι, για ένα δέντρο T με $F(r(T)) = \{T_1, T_2, ..., T_n\}$ υπάρχει συνάρτηση $f(\cdot) : \mathcal{T} \mapsto \mathcal{N}$ η οποία παίρνει ένα δέντρο με επιγραφές και το αντιστοιχίζει σε έναν φυσικό αριθμό μέσω της οποίας ορίζεται διάταξη μεταξύ των στοιχείων του συνόλου $F(r(T))$. Η τιμή της συνάρτησης $f(T_i)$ αντιστοιχεί στη τιμή της επιγραφής του κόμβου που αναπαραστά το υποδέντρο T_i. Έτσι, για δύο δέντρα T_i, Tj με $i < j$, $f(T_i) \leq f(T_j)$.



Σχήμα 3.14: *Αντικατάσταση υποδέντρου με έναν κόμβο.*

Όταν ισχύει ότι $f(T_i) = f(T_j)$ αυτό σημαίνει πως τα υποδέντρα T_i, T_j είναι ισομορφικά, και επομένως η ανταλλαγή των θέσεων τους δεν επηρεάζει την τελική μορφή του προκύπτοντος δέντρου T′. Διαφορετικά, το δέντρο T_i είναι λεξικογραφικά μικρότερο από το T_j (Σχήμα 3.15).



Σχήμα 3.15: *Αναδιάταξη υποδέντρων με χρήση Λεξικογραφικής Ταξινόμησης επιγραφών.*

3.2.4 Πυρήνες Γραφημάτων

"Whether you can observe a thing or not depends on the theory which you use."
Albert Einstein

Κατά την υλοποίηση του βασικού μας αλγορίθμου, συχνά απαιτείται η συγκριτική αξιολόγηση και ταξινόμηση γραφημάτων που αποτελούν τα δεδομένα εισόδου. Για να πετύχουμε αυτή την ταξινόμηση, πρώτα μετατρέπουμε τα γραφήματα σε διανυσματικές αναπαραστάσεις, ώστε να μπορούν να χρησιμοποιηθούν από μοντέλα Μηχανικής Μάθησης [2]. Σε αυτό το στάδιο, εισάγουμε τους **Πυρήνες Γραφημάτων** (*Graph Kernels*), οι οποίοι χρησιμοποιούνται ως αλγόριθμος στην υλοποίησή μας για την αντιστοίχιση των γραφήματα σε διανύσματα χαρακτηριστικών.

Ορισμός

Οι Πυρήνες Γραφημάτων είναι συναρτήσεις που υπολογίζουν το εσωτερικό γινόμενο των διανυσματικών αναπαραστάσεων δύο γραφημάτων. Με αυτόν τον τρόπο, μπορούμε να μετρήσουμε τον βαθμό ομοιότητάς τους. Έστω ένα σύνολο γραφημάτων με επιγραφές στους κόμβους:

$$\Omega = \{G \mid G = (V, E, l(\cdot))\},$$

και δύο γραφήματα $G_1, G_2 \in \Omega$. Έστω επίσης ένας χώρος Χίλμπερτ $\mathcal{H}$ και μία συνάρτηση $\varphi : \Omega \to \mathcal{H}$ που απεικονίζει κάθε γράφημα σε ένα σημείο της $\mathcal{H}$. Τότε ένας Πυρήνας Γραφημάτων $k : \Omega \times \Omega \to \mathbb{R}$ ορίζεται ως:

$$k(G_1, G_2) = \langle \varphi(G_1), \varphi(G_2) \rangle_{\mathcal{H}},$$

όπου η τιμή του $k(G_1, G_2)$ είναι ανάλογη της ομοιότητα των δύο γραφημάτων [43].

Κατηγορίες Πυρήνων Γραφημάτων

Η επιλογή του κατάλληλου Πυρήνα Γραφημάτων εξαρτάται από τη δομή και την εφαρμογή του προβλήματος [43]. Υπάρχουν τρεις θεμελιώδεις κατηγορίες [2]:

1. **Πυρήνες Υποδέντρων (Subtree Kernels)**.
2. **Πυρήνες Τυχαίων Περιπάτων (Random Walk Kernels)**.
3. **Πυρήνες Συντομότερων Μονοπατιών (Shortest Path Kernels)**.

Στη δική μας περίπτωση, μας ενδιαφέρει η πληροφορία των υποδέντρων ενός γράφηματος και για αυτό κατάλληλοι είναι οι Πυρήνες Υποδέντρων, και συγκεκριμένα ο *Weisfeiler-Lehman* Πυρήνας Υποδέντρων.

Αλγόριθμος

Στο Σχήμα 3.16 παρουσιάζεται ο ψευδοκώδικας για μία επανάληψη του αλγορίθμου *Weisfeiler-Lehman*. Παρακάτω δίνουμε επεξήγηση των βημάτων του αλγορίθμου.

Algorithm 1 One iteration of the 1-dim. Weisfeiler-Lehman test of graph isomorphism

1: Multiset-label determination
- For $i = 0$, set $M_i(v) := l_0(v) = \ell(v)$. [2]
- For $i > 0$, assign a multiset-label $M_i(v)$ to each node v in G and G' which consists of the multiset $\{l_{i-1}(u) | u \in \mathcal{N}(v)\}$.

2: Sorting each multiset
- Sort elements in $M_i(v)$ in ascending order and concatenate them into a string $s_i(v)$.
- Add $l_{i-1}(v)$ as a prefix to $s_i(v)$ and call the resulting string $s_i(v)$.

3: Label compression
- Sort all of the strings $s_i(v)$ for all v from G and G' in ascending order.
- Map each string $s_i(v)$ to a new compressed label, using a function $f : \Sigma^* \to \Sigma$ such that $f(s_i(v)) = f(s_i(w))$ if and only if $s_i(v) = s_i(w)$.

4: Relabeling
- Set $l_i(v) := f(s_i(v))$ for all nodes in G and G'.

Σχήμα 3.16: *Μία επανάληψη του αλγορίθμου Weisfeiler-Lehman*

Βήματα του Αλγορίθμου Weisfeiler-Lehman Πυρήνα Γραφημάτων

Σε κάθε επανάληψη του Weisfeiler-Lehman, πραγματοποιούνται τα εξής βήματα:

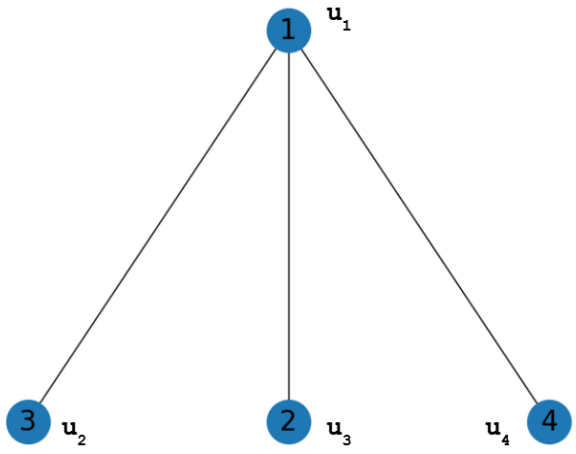
1. **Υπολογισμός επιγραφών Πολυσυνόλων**: Για κάθε κόμβο υπολογίζουμε μια συμβολοσειρά που περιέχει την επιγραφή του κόμβου και τις επιγραφές των γειτόνων του, ταξινομημένες σε αύξουσα σειρά.Για παράδειγμα, η επιγραφή πολυσυνόλου που προκύπτει για το δέντρο T του Σχήματος 3.17 για τον κόμβο u_1 είναι ¨1.2.3.4¨ καθώς $l(u_1)$ = 1 και οι επιγραφές των γειτονικών κόμβων του u_1 είναι σε ταξινομημένη σειρά

$$[1, 2, 3]$$

.

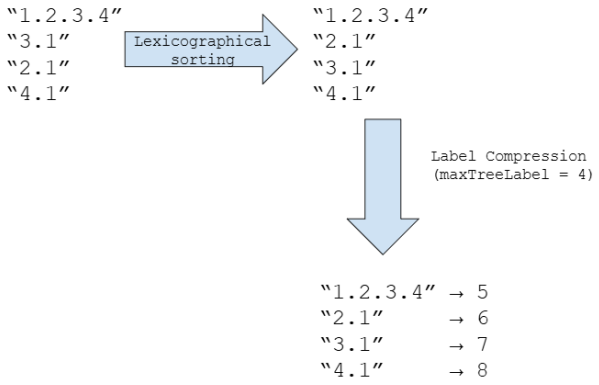
2. **Δημιουργία Αρίθμησης Νέων επιγραφών**: Οι συμβολοσειρές ταξινομούνται λεξικογραφικά και αντιστοιχίζονται σε νέες ακέραιες τιμές, ξεκινώντας την αρίθμηση από μία τιμή μεγαλύτερη της μέγιστης επιγραφής του αρχικού δέντρου. Στο παράδειγμα του Σχήματος 3.17, προκύπτουν οι επιγραφές Πολυσυνόλου για όλους τους κόμβους του δέντρου T που φαίνεται στο Σχήμα 3.18 . Στη συνέχεια, τις ταξινομούμε λεξικογραφικά και αντιστοιχίζουμε κάθε επιγραφή πολυσυνόλου σε έναν διαδοχικό φυσικό αριθμό. Η μεγαλύτερη τιμή επιγραφής στο δέντρο T είναι 4, και για αυτό ξεκινάει η αντιστοίχιση των επιγραφών Πολυσυνόλου από την τιμή 5.

3. **Επανατοποθέτηση επιγραφών**: Οι κόμβοι όλων των γραφημάτων ενημερώνονται με τις νέες επιγραφές. Η διαδικασία αυτή μπορεί να επαναληφθεί πολλές φορές, καθεμιά εξετάζοντας πιο σύνθετα υποδέντρα. Στο Σχήμα 3.19 βλέπουμε τις νέες επιγραφές των κόμβων του δέντρου T του Σχήματος 3.17 χρησιμοποιώντας την αντιστοίχιση που προκύπτει από τη διαδικασία που φαίνεται στο Σχήμα 3.16.

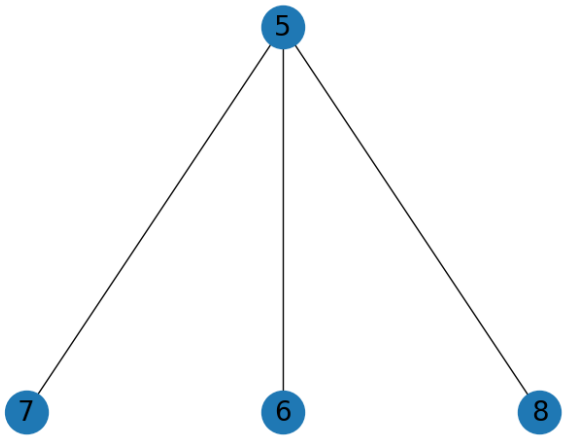


Σχήμα 3.17: *Δέντρο T με επιγραφές στους κόμβους*

Μετά το πέρας των επαναλήψεων του αλγορίθμου, υπολογίζεται η συχνότητα εμφάνισης κάθε επιγραφής για κάθε γράφημα και έτσι δημιουργείται το τελικό διάνυσμα χαρακτηριστικών για κάθε γράφημα. Αυτό το διάνυσμα ενσωματώνεται στο βασικό μας αλγόριθμο ως είσοδος σε έναν ταξινομητή, επιτρέποντας την πρόβλεψη της κλάσης ενός νέου γραφήματος.



Σχήμα 3.18: *Συμπίεση επιγραφών*



Σχήμα 3.19: *Δέντρο T μετά την ανανέωση των τιμών των επιγραφών*

Σχέση επιγραφών και Δέντρων Ανάπτυξης

Οι επιγραφές που προκύπτουν από την διαδικασία Weisfeiler-Lehman έχουν άμεση συσχέτιση με τα υποδέντρα όλων των γραφημάτων. Κάθε συνιστώσα του διανύσματος χαρακτηριστικών αντιστοιχεί στον αριθμό εμφάνισης ενός συγκεκριμένου δέντρου ανάπτυξης [3]. Στην εφαρμογή μας, οι κόμβοι μπορεί να αντιπροσωπεύουν χρήστες σε δέντρα διάδοσης μιας δημοσίευσης. Στον κλάδο της χημείας, οι χημικές ενώσεις θα μπορούσαν να αναπαρασταθούν με γραφήματα με επιγραφές όπου ο κάθε κόμβος έχει επιγραφή ανάλογα με το χημικό στοιχείο που αναπαραστά [2].

Έτσι, εάν η δομή σχετίζεται με κάποια ιδιότητα (π.χ. τοξικότητα μιας χημικής ένωσης), η διανυσματική αναπαράσταση επιτρέπει την εκπαίδευση ενός μοντέλου ταξινόμησης που, για μια νέα ένωση, θα εκτιμήσει αν διαθέτει αυτήν την ιδιότητα, αξιοποιώντας αποκλειστικά την τοπολογική πληροφορία που ενσωματώθηκε στο διάνυσμα[2].

Με αυτό τον τρόπο, οι Πυρήνες Γραφημάτων παρέχουν ένα αποτελεσματικό πλαίσιο για την ενσωμάτωση της δομής και των επιγραφών του γραφήματος σε αλγορίθμους ταξινόμησης και άλλες αναλυτικές τεχνικές, ενσωματώνοντας πλήρως την τοπολογική πληροφορία στον βασικό αλγόριθμο.

Στο Σχήμα 3.21 βλέπουμε και τυπικά τον αλγόριθμο για μία επανάληψη της διαδικασίας στη περίπτωση που έχουμε N γραφήματα. Το πλήθος των επαναλήψεων h εξετάζει όλο και μεγαλύτερα σε βάθος Δέντρα Ανάπτυξης.

Algorithm 2 One iteration of the Weisfeiler-Lehman subtree kernel computation on N graphs

1: Multiset-label determination
- Assign a multiset-label $M_i(v)$ to each node v in G which consists of the multiset $\{l_{i-1}(u)|u \in \mathcal{N}(v)\}$.

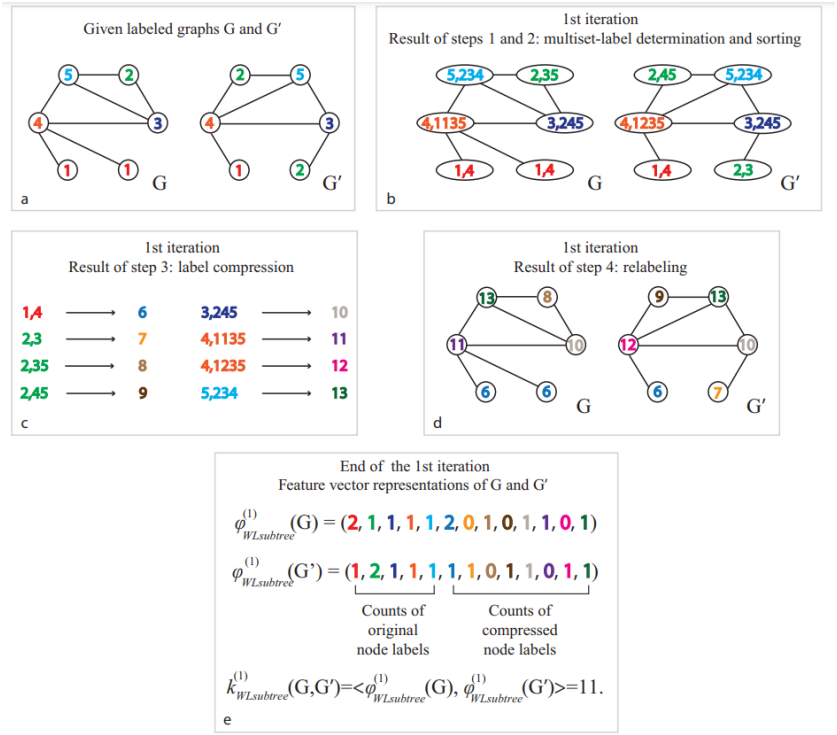
2: Sorting each multiset
- Sort elements in $M_i(v)$ in ascending order and concatenate them into a string $s_i(v)$.
- Add $l_{i-1}(v)$ as a prefix to $s_i(v)$.

3: Label compression
- Map each string $s_i(v)$ to a compressed label using a hash function $f : \Sigma^* \to \Sigma$ such that $f(s_i(v)) = f(s_i(w))$ if and only if $s_i(v) = s_i(w)$.

4: Relabeling
- Set $l_i(v) := f(s_i(v))$ for all nodes in G.

Σχήμα 3.20: *Αλγόριθμος WL για 1 επανάληψη σε N γραφήματα*

Στο Σχήμα 3.21 βλέπουμε συγκεντρωμένα ένα πρακτικό παράδειγμα του πως λειτουργεί ο αλγόριθμος για δύο γραφήματα με επιγραφές G, G' για δύο επαναλήψεις και πως μέσω αυτού λαμβάνουμε τα διανύσματα χαρακτηριστικών.

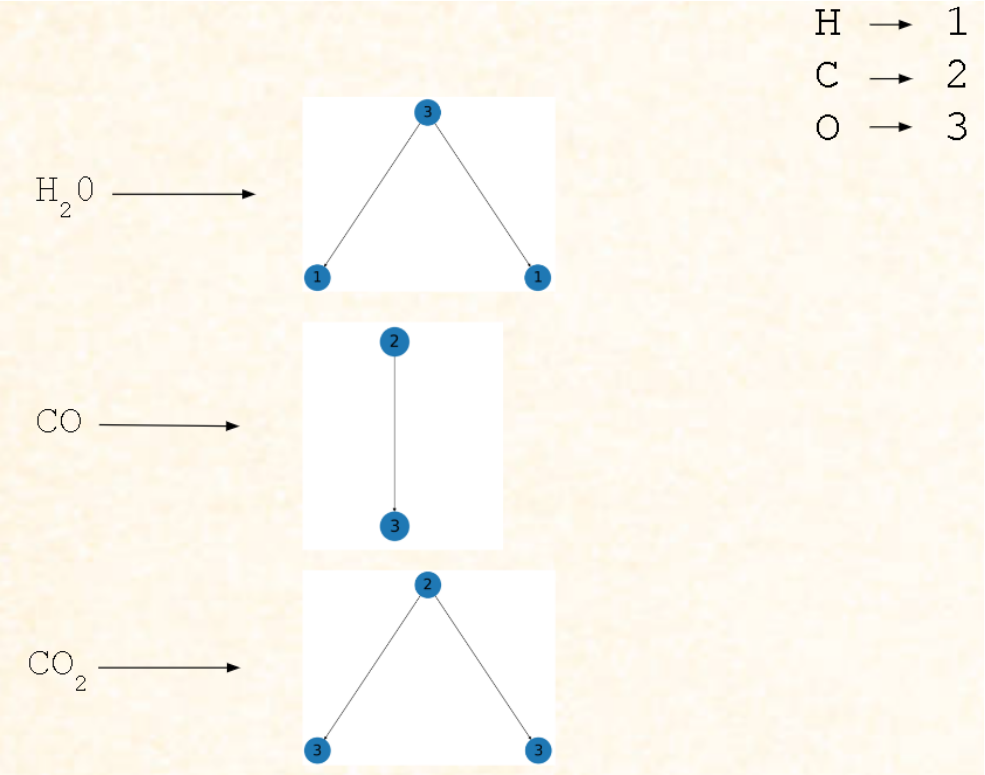


Σχήμα 3.21: *Υπολογισμός διανυσμάτων χαρακτηριστικών μέσω του αλγορίθμου Ωεισφειλερ-Λεημαν*

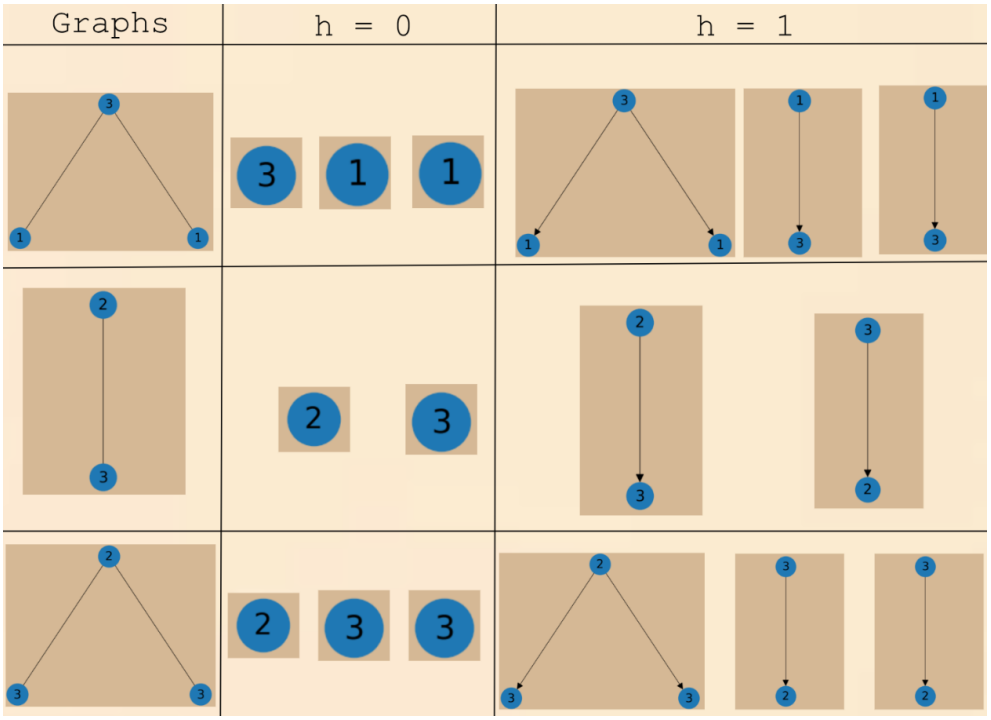
Παράδειγμα

Οι επιγραφές στους κόμβους των γραφημάτων παρέχουν πρόσθετη πληροφορία σχετικά με τους κόμβους. Για παράδειγμα, στο Σχήμα 3.22, με τη χρήση αντιστοιχίσεων χημικών στοιχείων στους διαδοχικούς φυσικούς αριθμούς του αλφαβήτου Σ = {1,2,3}, αναπαριστούμε χημικές ενώσεις ως Γραφήματα με επιγραφές. Αυτή η αναπαράσταση συντελεί στην εξαγωγή

διανυσματικών χαρακτηριστικών μέσω του αλγορίθμου *Weisfeiler-Lehman*, που εφαρμόζεται για έναν καθορισμένο αριθμό επαναλήψεων ($h = 1$, με $h = 0, 1$). Όπως παρουσιάζεται στο Σχήμα 3.23, για τα γραφήματα του Σχήματος 3.22, υπολογίζουμε όλα τα i-Δέντρα Ανάπτυξης για $i = 0, 1$ για τα γραφήματα αυτά.



Σχήμα 3.22: *Αναπαράσταση χημικών ενώσεων με χρήση Γραφημάτων με επιγραφές*



Σχήμα 3.23: *0,1 - Δέντρα Ανάπτυξης στα γραφήματα του Σχήματος 3.22*

Στη συνέχεια, δημιουργούμε μια νέα κωδικοποίηση για όλα τα Δέντρα Ανάπτυξης ύψους $i = 1$. Ταξινομούμε αυτά τα δέντρα και τους αποδίδουμε νέες επιγραφές, όπως απεικονίζεται στο Σχήμα 3.24.

Σχήμα 3.24: *Αντιστοίχιση υποδέντρων σε επιγραφές*

Υπολογισμός Διανυσματικών Χαρακτηριστικών

Μετά την κωδικοποίηση των υποδέντρων, υπολογίζουμε τη συχνότητα εμφάνισης κάθε επιγραφής στις δύο επαναλήψεις του αλγορίθμου Weisfeiler-Lehman. Τα διανύσματα χαρακτηριστικών που αντιστοιχίζεται κάθε χημική ένωση υπολογίζονται ως η συχνότητα εμφάνισης καθεμίας επιγραφής στις δύο επαναλήψεις του αλγορίθμου και ισούνται με:

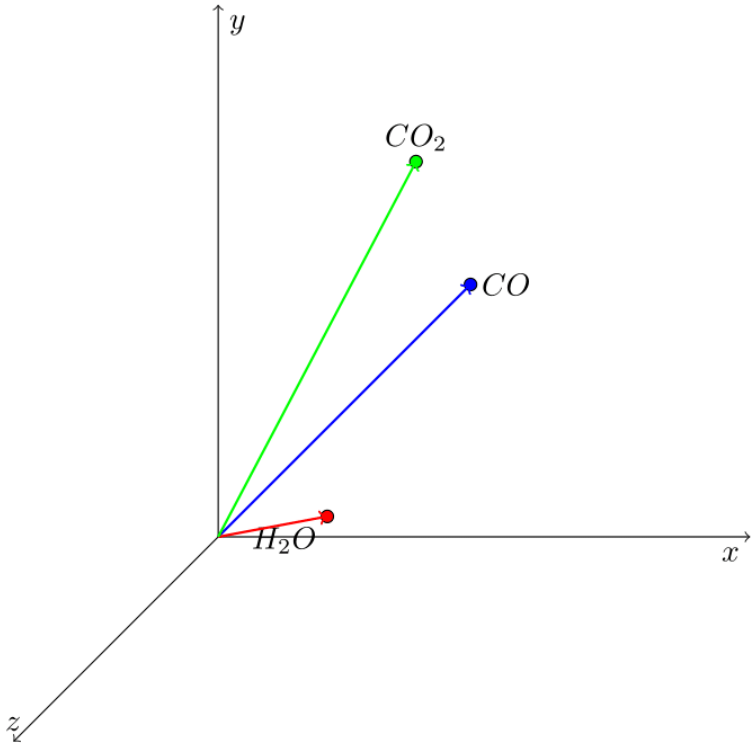
$$\varphi(H_2O) = [2, 0, 1, 2, 0, 0, 1, 0]$$
$$\varphi(CO) = [0, 1, 1, 0, 1, 0, 0, 1]$$
$$\varphi(C_2O) = [0, 1, 2, 0, 0, 1, 0, 2]$$

Ανάλυση Ομοιότητας Χημικών Ουσιών

Η ομοιότητα μεταξύ των δομών των χημικών ενώσεων αξιολογείται μέσω των αποστάσεων μεταξύ των διανυσμάτων των χαρακτηριστικών τους. Συγκεκριμένα:

$$dist(\varphi(H_2O), \varphi(CO)) > dist(\varphi(CO), \varphi(CO_2))$$
$$dist(\varphi(H_2O), \varphi(CO_2)) > dist(\varphi(CO), \varphi(CO_2))$$

Αυτές οι σχέσεις υποδεικνύουν ότι οι δομές των ουσιών CO και CO_2 είναι πιο όμοιες μεταξύ τους σε σύγκριση με την H_2O όπως φαίνεται και στο Σχήμα 3.25.



Σχήμα 3.25: *Διανυσματική αναπαράσταση των χημικών στοιχείων*

3.2.5 Διορθωτική Απόσταση Δέντρων

Η διορθωτική απόσταση δέντρων (*Tree Edit Distance*) είναι μια μετρική που καθορίζει πόσο όμοια είναι δύο δέντρα με επιγραφές στους κόμβους. Βασίζεται στις ακόλουθες επιτρεπτές λειτουργίες που εφαρμόζονται σε ένα δέντρο με επιγραφές T για την μετατροπή του σε ένα δέντρο T′:

- **Μετονομασία κόμβου:** Αλλαγή της επιγραφής ενός κόμβου.
- **Εισαγωγή κόμβου:** Προσθήκη ενός νέου κόμβου στο δέντρο με μία συγκεκριμένη επιγραφή.
- **Διαγραφή κόμβου:** Αφαίρεση ενός κόμβου από το δέντρο με μία συγκεκριμένη επιγραφή.

Η απόσταση ορίζεται ως το ελάχιστο συνολικό κόστος που απαιτείται για τη μετατροπή ενός δέντρου T σε ένα άλλο δέντρο T', χρησιμοποιώντας τις παραπάνω λειτουργίες.

Τυπική Διατύπωση

Δοσμένων δύο δέντρων με επιγραφές T_1, T_2, και μιας συνάρτησης κόστους $\gamma : \Sigma^{\perp} \times \Sigma^{\perp} \to \mathbb{R}$, η οποία ορίζει το κόστος για:

- τη μετατροπή ενός κόμβου με επιγραφή l_1 σε μια άλλη με επιγραφή l_2
- την εισαγωγή ενός κόμβου με επιγραφή l_2 (όταν $l_1 = \perp$),
- τη διαγραφή ενός κόμβου με επιγραφή l_1 (όταν $l_2 = \perp$)

η $\gamma(l_1, l_2)$ ορίζεται ως:

$$\gamma(l_1, l_2) = \begin{cases} \text{κόστος μετατροπής } l_1 \to l_2, & \text{αν } l_1, l_2 \neq \perp, \\ \text{κόστος εισαγωγής } l_2, & \text{αν } l_1 = \perp, \\ \text{κόστος διαγραφής } l_1, & \text{αν } l_2 = \perp\ . \end{cases}$$

Η διορθωτική απόσταση υπολογίζεται ως το ελάχιστο κόστος όλων των πιθανών ακολουθιών αυτών των λειτουργιών για την μετατροπή του T_1 στο T_2.

Στο Σχήμα 3.26 βλέπουμε τον ψευδοκώδικα για τον αλγόριθμο Structure & Depth Preserving Tree Edit Distance. Ο αλγόριθμος λαμβάνει ως είσοδο δύο δέντρα ξεδίπλωσης T, T′ ίδιου ύψους με επιγραφές και τον πίνακα κόστους όλων των λειτουργιών επεξεργασίας. Τα βήματα του αλγορίθμου περιγράφονται ακολούθως.

Algorithm 1 COMPUTE SDTED

input: Trees T, T', cost function $\gamma : \Sigma^{\perp} \times \Sigma^{\perp} \to \mathbb{R}$
output: Structure and depth preserving tree edit distance between T and T'

SDTED(T, T'):
1: $F := F(r(T))$, $F' := F(r(T'))$
2: Pad F and F' with empty trees $T_{\perp}$ such that $|F| = |F'| = deg(r(T)) + deg(r(T'))$
3: **for all** $T_i \in F$, $T'_j \in F'$ **do**

$$\delta_{ij} = \begin{cases} \text{SDTED}(T_i, T'_j) & \text{if } T_i \in F(r(T)) \text{ and } T'_j \in F(r(T')) \\ \sum_{v \in V(T_i)} \gamma(\ell(v), \perp) & \text{if } T_i \in F(r(T)) \text{ and } T'_j \notin F(r(T')) \\ \sum_{v' \in V(T'_j)} \gamma(\ell(v'), \perp) & \text{if } T_i \notin F(r(T)) \text{ and } T'_j \in F(r(T')) \\ 0 & \text{o/w .} \end{cases}$$

4: Let $S \subseteq F \times F'$ be a minimum cost perfect bipartite matching w.r.t. distances δ
5: **return** $\gamma(\ell(r(T)), \ell(r(T'))) + \sum_{(T_i, T'_j) \in S} \delta_{ij}$

Σχήμα 3.26: *Αλγόριθμος SD Tree Edit Distance [3]*

Αλγόριθμος Υπολογισμού

Ο αλγόριθμος υπολογισμού της διορθωτικής απόστασης δέντρων αποτελείται από τα εξής βήματα:

1. **Υπολογισμός των συνόλων υποδέντρων:** Για κάθε δέντρο T και T', υπολογίζονται τα σύνολα F και F', τα οποία περιέχουν τα υποδέντρα που έχουν ως ρίζα τους τα παιδιά της ρίζας των T και T' αντίστοιχα. Στο Σχήμα 3.27 βλέπουμε πάνω στο σχήμα ένα δέντρο T και πως από αυτό προκύπτει το F(r(T)).

2. **Εξίσωση πλήθους υποδέντρων:** Εισάγονται κενά δέντρα στα σύνολα F και F' έτσι ώστε να ισχύει $|F| = |F'| = deg(r(T)) + deg(r(T'))$. Στο Σχήμα 3.28 έχουμε ότι $deg(r(T)) = 2$ και $deg(r(T')) = 1$. Επειδή θέλουμε να ισχύει η σχέση $|F| = |F'| = deg(r(T)) + deg(r(T')) = 3$, προσθέτουμε ένα κενό δέντρο στο σύνολο $F(r(T))$ (επειδή περιέχει ήδη 2 στοιχεία) και δύο κενά δέντρα στο $F(r(T'))$

3. **Υπολογισμός κόστους μετατροπής υποδέντρων:** Το κόστος μετατροπής κάθε υποδέντρου $f \in F$ σε κάθε υποδέντρο $f' \in F'$ υπολογίζεται βάσει της συνάρτησης γ. Οι περιπτώσεις είναι:

   (α΄) Αν και τα δύο υποδέντρα είναι κενά, το κόστος είναι 0.

   (β΄) Αν το δεύτερο υποδέντρο είναι κενό, το κόστος είναι το συνολικό κόστος διαγραφής όλων των κόμβων του πρώτου υποδέντρου. Στο παράδειγμα του Σχήματος 3.29, το συνολικό κόστος μετατροπής του δέντρου T στο κενό δέντρο είναι:
   $\gamma(2, \perp) + \gamma(2, \perp) + \gamma(3, \perp) + \gamma(1, \perp)$.

   (γ΄) Αν το πρώτο υποδέντρο είναι κενό, το κόστος είναι το συνολικό κόστος εισαγωγής όλων των κόμβων του δεύτερου υποδέντρου. Στο Σχήμα 3.30 βλέπουμε μία τέτοια περίπτωση. Το συνολικό κόστος μετατροπής του κενού δέντρου στο τελικό δέντρο T ισούται με:
   $\gamma(1, \perp) + \gamma(2, \perp) + \gamma(3, \perp)$.

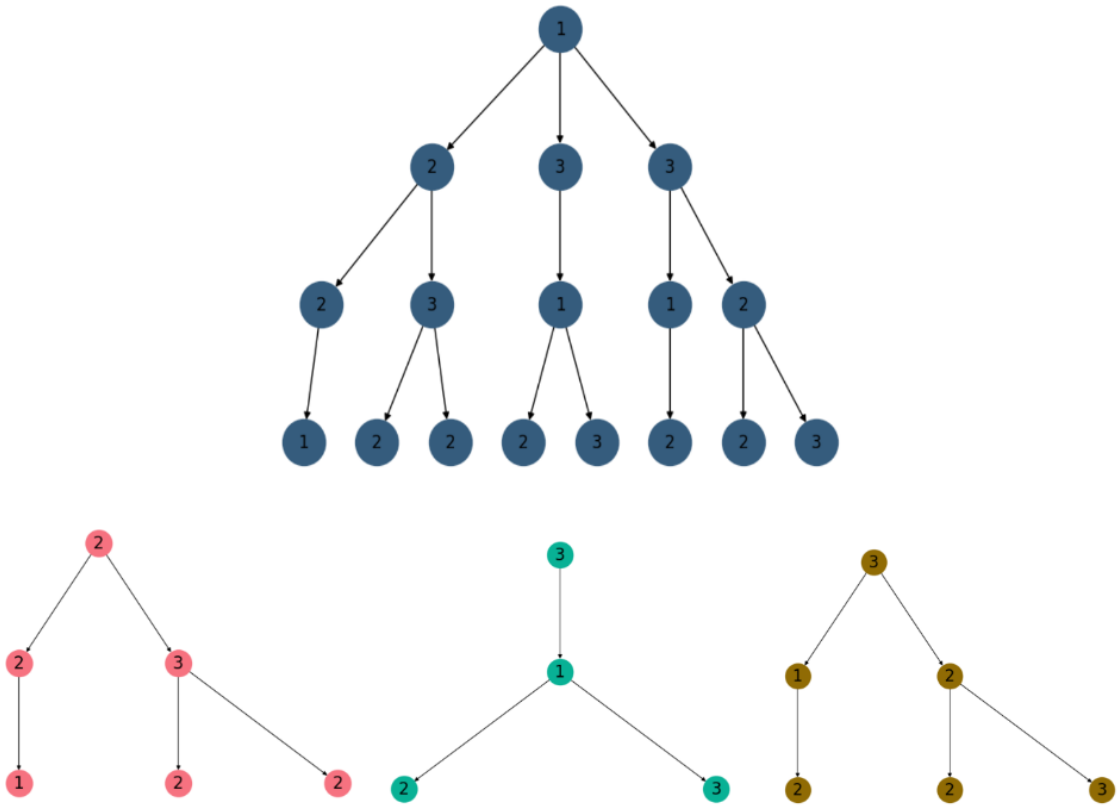
(δ') Αν κανένα υποδέντρο δεν είναι κενό επαναλαμβάνεται αναδρομικά η ίδια διαδικασία για τα δύο υποδέντρα.

4. **Εύρεση ελάχιστου κόστους διμερούς ταιριάσματος:** Με βάση τα υπολογισμένα κόστη $\delta_{i,j}$ για κάθε ζεύγος υποδέντρων $f_i \in F$ και $f'_j \in F'$, κατασκευάζεται ένα διμερές γράφημα όπως βλέπουμε στο Σχήμα 3.28. Οι κορυφές αναπαριστούν τα υποδέντρα και οι ακμές το κόστος μετατροπής. Η εύρεση του ελάχιστου κόστους διμερούς ταιριάσματος δίνει το βέλτιστο τρόπο μετατροπής.

5. **Υπολογισμός συνολικού κόστους:** Το συνολικό κόστος αποτελεί το άθροισμα των ακόλουθων κόστων:

   (α') Το κόστος μετατροπής της επιγραφής της ρίζας του T σε αυτή της ρίζας του T'.

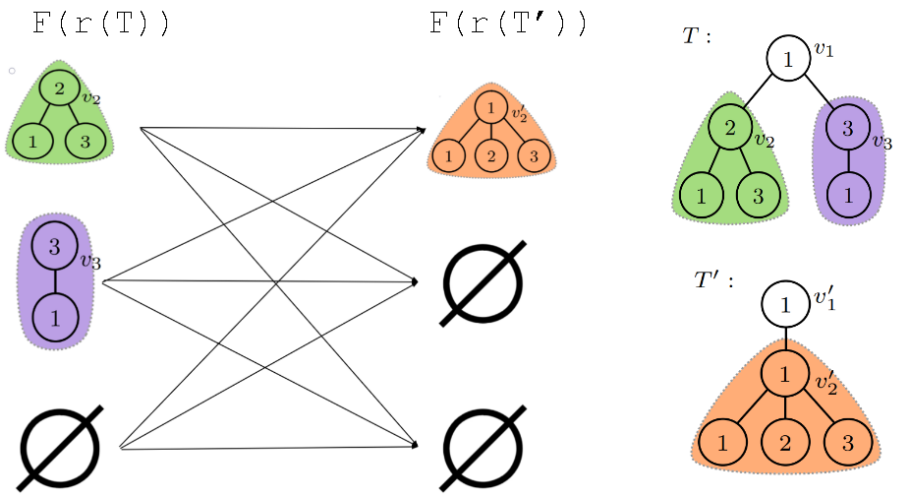
   (β') Το συνολικό κόστος του ελάχιστου κόστους διμερούς ταιριάσματος των υποδέντρων.



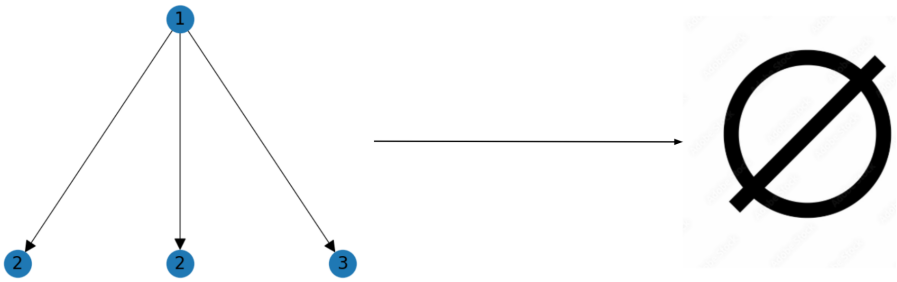
Σχήμα 3.27: *Δέντρο T (πάνω) και F(r(T)) (κάτω)*

Με βάση το παράδειγμα του Σχήματος 3.31, για να μετατρέψουμε το δέντρο T με επιγραφές στο δέντρο με επιγραφές T',εξετάζουμε αν συμφέρει να διαγράψουμε το πράσινο υποδέντρο και να μετατρέψουμε το μωβ υποδέντρο στο πορτοκαλί ή αν συμφέρει να διαγράψουμε το μωβ υποδέντρο και να μετατρέψουμε το πράσινο υποδέντρο στο πορτοκαλί. Στην πρώτη περίπτωση, το κόστος υπολογίζεται ως εξής:

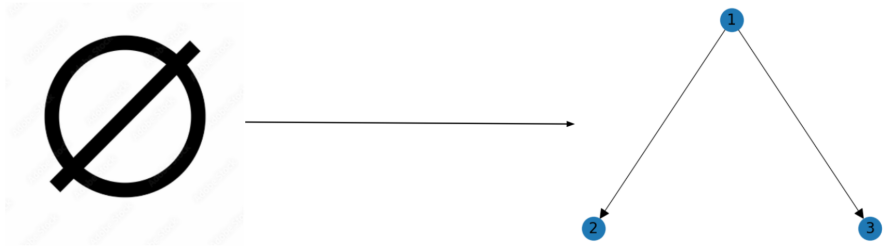
$$\text{del(green)} + \text{Transform(purple} \rightarrow \text{orange)} = 3 + 3 = 6$$



Σχήμα 3.28: *Διμερές γράφημα μετατροπής υποδέντρων $F(r(T)) \mapsto F(r(T'))$ [3]*



Σχήμα 3.29: *Μετατροπή ενός δέντρου T στο κενό δέντρο.*

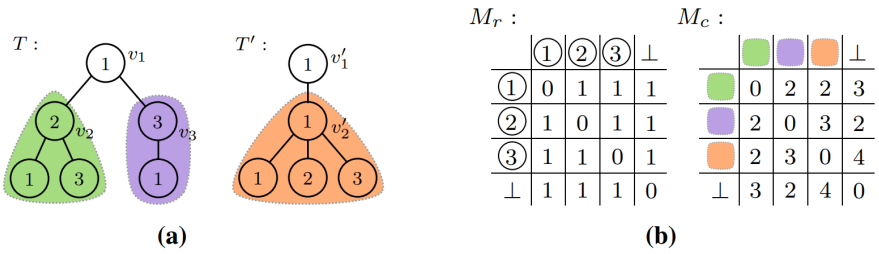


Σχήμα 3.30: *Μετατροπή ενός κενού δέντρου σε ένα δέντρο T*

Αντίστοιχα, στη δεύτερη περίπτωση, το κόστος είναι:

$$\text{del(purple)} + \text{Transform(green} \rightarrow \text{orange)} = 2 + 2 = 4$$

Έτσι, εφόσων οι τιμές των επιγραφών των ριζών είναι ίδιες, προκύπτει πως η Διορθωτική Απόσταση των Δέντρων T,T' είναι 4.



Σχήμα 3.31: *Δέντρα με επιγραφές T, T' και πίνακες κόστους επιγραφών και υποδέντρων M_r, M_c αντιστοίχως. [3]*

Εφαρμογές της Διορθωτικής Απόστασης Δέντρων

Η διορθωτική απόσταση δέντρων βρίσκει εφαρμογή σε πολλά επιστημονικά πεδία, όπως:

- **Βιοπληροφορική/Υπολογιστική Βιολογία:** Σύγκριση φυλογενετικών δέντρων.
- **Επεξεργασία Φυσικής Γλώσσας:** Σύγκριση συντακτικών ή σημασιολογικών δέντρων προτάσεων (Σχήμα 3.32)
- **Επεξεργασία Εικόνας:** Σύγκριση ιεραρχικών δομών εικόνων.
- **Λογισμικό:** Μέτρηση ομοιότητας μεταξύ πηγαίων κειμένων ή δομών δεδομένων. Για παράδειγμα, η αναπαράσταση κειμένων με δέντρα των οποίων οι κόμβοι περιλαμβάνουν λέξεις μπορεί να δείξει ομοιότητα μεταξύ δύο κειμένων μέσω της απόστασης.

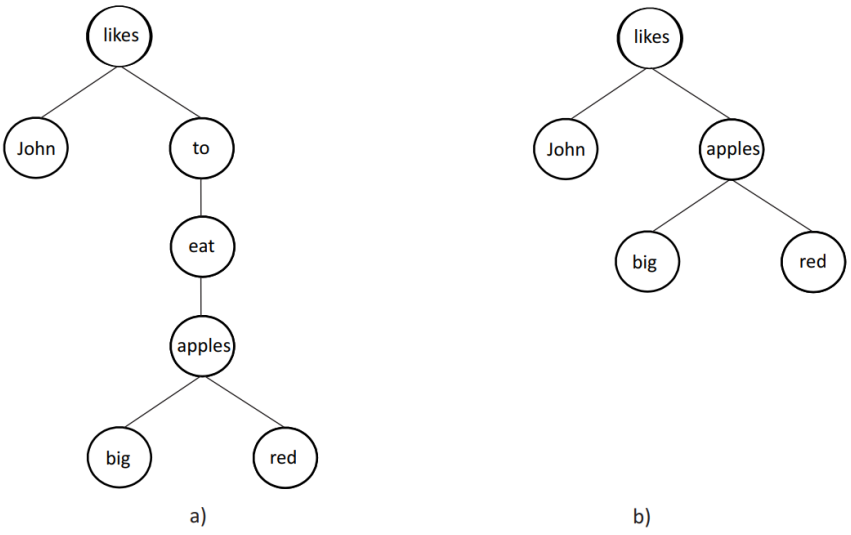


Fig. 1. Syntactic trees of the sentences (a) *John likes to eat big red apples.* and (b) *John likes big red apples.*

Σχήμα 3.32: *Αναπαράσταση δύο προτάσεων με χρήση δέντρων [4]*

3.2.6 Κωδικοποίση Δέντρων Ανάπτυξης με Χρήση Διανυσμάτων

Η κωδικοποίηση Δέντρων Ανάπτυξης με χρήση διανυσμάτων αποσκοπεί στην αναπαράσταση ενός δέντρου ανάπτυξης με επιγραφές T με χρήση ενός διανύσματος χαρακτηριστικών $V(T)$ [3]. Το διάνυσμα αυτό αποτελείται από δύο συνιστώσες:

- Το $V_r(T)$, το οποίο αναπαριστά την επιγραφή της ρίζας του δέντρου.
- Το $V_c(T)$, το οποίο κωδικοποιεί τη δομή και την πληροφορία των υποδέντρων.

Η διαδικασία αυτή είναι χρήσιμη για την εξαγωγή χαρακτηριστικών, τη σύγκριση και την ομαδοποίηση δέντρων, καθώς παρέχει μια αριθμητική αναπαράσταση κατάλληλη για αλγορίθμους μηχανικής μάθησης.

Τυπική Διατύπωση

Για ένα σύνολο $\{T_1, T_2, \ldots, T_k\}$ από i-Δέντρα Ανάπτυξης με επιγραφές στους κόμβους $l \in \{l_1, l_2, \ldots, l_n\}$, η διανυσματική αναπαράσταση ενός δέντρου T ορίζεται ως:

$$V(T) = (V_r(T), V_c(T)),$$

όπου:

1. $V_r(T) = (x_1, x_2, \ldots, x_n, x_\perp)$, με:

$$x_i = \begin{cases} 1, & \text{αν } l(r(T)) = l_i, \\ 0, & \text{αλλιώς.} \end{cases}$$

2. $V_c(T) = (x_1, x_2, \ldots, x_q, x_{q+1})$, με:

$$x_i = \begin{cases} |\{T \in F(r(T)) \mid T \equiv T_i^{(h-1)}\}|, & \text{αν } i \in [q], \\ 2d - \deg(r(T)), & \text{αν } i = q+1. \end{cases}$$

με

Το $V_r(T)$ αποτελεί μια μοναδιαία κωδικοποίηση (*one − hot encoding*) της επιγραφής της ρίζας του δέντρου T ενώ το $V_c(T)$ περιλαμβάνει την πληροφορία των υποδέντρων του T. Το d είναι ο μέγιστος βαθμός της ρίζας όλων των δέντρων T_i ($d = \max(\deg(r(T)))$).

Αλγόριθμος Κωδικοποίησης Δέντρων Ανάπτυξης

Αλγοριθμος 3.3: *Κωδικοποίηση Δέντρων Ανάπτυξης με Χρήση Διανυσμάτων*

Είσοδος: Σύνολο από i-Δέντρα Ανάπτυξης $\{T_1, T_2, \ldots, T_k\}$,με επιγραφές $l \in \{l_1, l_2, \ldots, l_n\}$
Έξοδος: Διάνυσμα $V(T) = (V_r(T), V_c(T))$ για κάθε δέντρο T

1: **for** κάθε δέντρο $T \in \{T_1, T_2, \ldots, T_k\}$ **do**
2: **Υπολογισμός** $V_r(T)$:
3: Δημιουργούμε το $V_r(T)$, ένα διάνυσμα μήκους $n+1$, με:

$$x_i = \begin{cases} 1, & \text{αν } l(r(T)) = l_i \\ 0, & \text{αλλιώς} \end{cases}$$

4: **Υπολογισμός** $V_c(T)$:
5: Υπολογίζουμε όλα τα h-Δέντρα Ανάπτυξης του T.
6: Δημιουργούμε φυσική απαρίθμηση των μη ισομορφικών υποδέντρων $\{T_1^{(h-1)}, \ldots, T_q^{(h-1)}\}$.
7: Για κάθε συνιστώσα x_i του $V_c(T)$, θέτουμε:

$$x_i = \begin{cases} |\{T \in F(r(T)) \mid T \equiv T_i^{(h-1)}\}|, & \text{αν } i \in [q] \\ 2d - \deg(r(T)), & \text{αν } i = q+1 \end{cases}$$

με $d = \max(\{\deg(r(T)) \mid T \in \mathcal{G}\})$.
8: Επιστρέφουμε το $V(T) = (V_r(T), V_c(T))$.
9: **end for**

Βήματα Υπολογισμού της Κωδικοποίησης

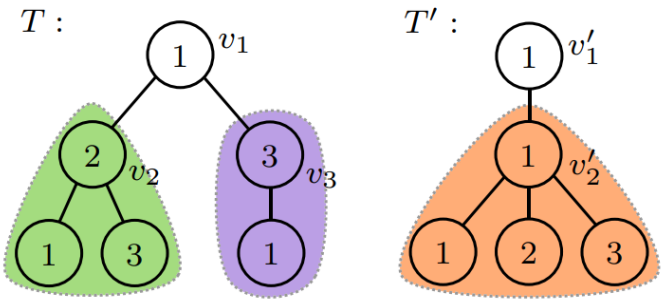
1. **Υπολογισμός επιγραφής Ρίζας**: Δημιουργούμε το διάνυσμα $V_r(T)$, με μήκος $n+1$. Κάθε συνιστώσα x_i δείχνει αν η επιγραφή της ρίζας $l(r(T))$ ισούται με l_i.

2. **Συλλογή Υποδέντρων**: Για όλα τα δέντρα στο σύνολο $\mathbf{T} = \{T_1, T_2, \ldots, T_k\}$ υπολογίζουμε το σύνολο $\mathbf{U} = \bigcup_{i=1}^{k} F(r(T_i))$.

3. **Φυσική Απαρίθμηση Υποδέντρων**: Δημιουργούμε μια απαρίθμηση των μη ισομορφικών υποδέντρων στο σύνολο $\mathbf{U}$ αντιστοιχώντας κάθε υποδέντρο σε έναν φυσικό αριθμό.

4. **Υπολογισμός** $V_c(T)$: Για κάθε δέντρο T, το $V_c(T)$ περιλαμβάνει τη συχνότητα εμφάνισης κάθε υποδέντρου από το σύνολο $\mathbf{U}$ καθώς και μια επιπλέον συνιστώσα x_{q+1}, που εξασφαλίζει την ίδια L_1-νόρμα για όλα τα διανύσματα $V_c(T_i)$.

5. **Επιστροφή αποτελέσματος**: Επιστρέφουμε για κάθε Δέντρο Ανάπτυξης T_i την διανυσματική του αναπαράσταση $V(T) = (V_r(T), V_c(T))$.



Σχήμα 3.33: *Δύο Δέντρα Ανάπτυξης T και T' [3].*

Παράδειγμα Εκτέλεσης

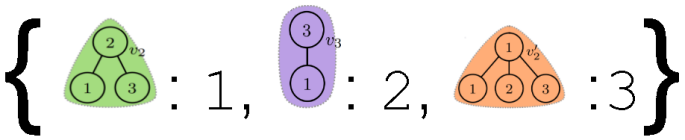
Για τα δέντρα T και T' που παρουσιάζονται στο Σχήμα 3.33:

$$V_r(T) = [1,0,0,0], \quad V_r(T') = [1,0,0,0].$$

Τα διανύσματα $V_r(T)$ και $V_r(T')$ έχουν μήκος 4, καθώς το πλήθος των διαφορετικών επιγραφών στα δέντρα T και T' είναι 3. Κάθε συνιστώσα του διανύσματος παίρνει την τιμή 1 στη θέση που αντιστοιχεί στην επιγραφή της ρίζας του δέντρου και 0 στις υπόλοιπες συνιστώσες. Συνεπώς, προκύπτουν τα διανύσματα:

$$V_r(T) = [1,0,0,0], \quad V_r(T') = [1,0,0,0],$$

καθώς ισχύει ότι $l(r(T)) = l(r(T')) = 1$.



Σχήμα 3.34: *Φυσική Απαρίθμηση Υποδέντρων των T και T'.*

Σύμφωνα με τη φυσική απαρίθμηση που φαίνεται στο Σχήμα 3.34, τα διανύσματα $V_c(T), V_c(T')$ των δέντρων T και T' ορίζονται ως εξής:

$$V_c(T) = [1,1,0,1], \quad V_c(T') = [0,0,1,2].$$

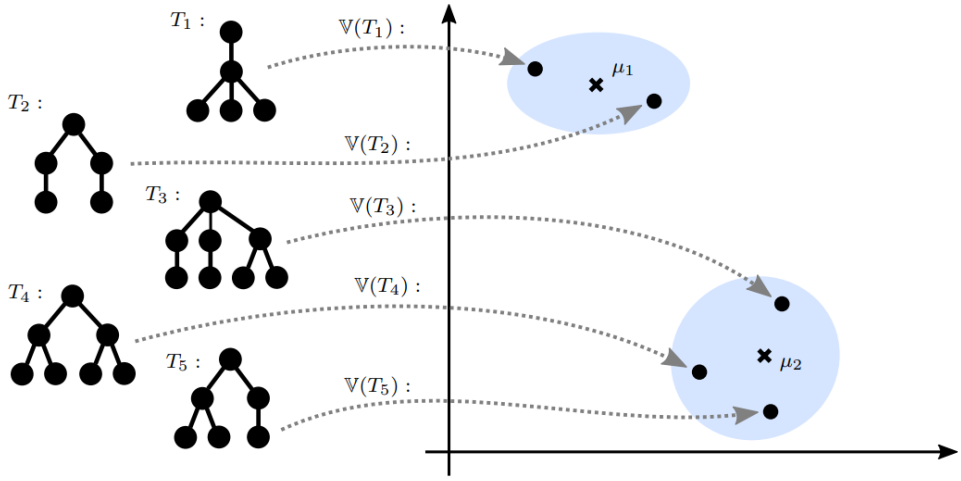
Η τιμή κάθε συνιστώσας στα διανύσματα $V_c(T)$ και $V_c(T')$ αντιστοιχεί στον πλήθος των υποδέντρων που βρίσκονται σε κάθενα από τα δέντρα T,T'. Συγκεκριμένα, το δέντρο T περιέχει ένα πράσινο υποδέντρο και ένα μωβ υποδέντρο (αύξων αριθμός υπόδεντρου 1,2), ενώ το δέντρο T' περιέχει μόνο το πορτοκαλί υποδέντρο (αύξων αριθμός υπόδεντρου 3).

Η τελευταία συνιστώσα κάθε διανύσματος χρησιμοποιείται ως υπολειπόμενη τιμή, ώστε να εξασφαλίζεται η κανονικοποίηση των διανυσμάτων $V_c(T_i)$. Με αυτόν τον τρόπο, διασφαλίζεται ότι όλα τα διανύσματα έχουν την ίδια L_1-νόρμα.

Σημασία και Εφαρμογές

Η κωδικοποίηση ενός Δέντρου Ανάπτυξης T με τη χρήση του διανύσματος $V(T)$ έχει σημαντικές εφαρμογές όπως:

- Επιτρέπει τη σύγκριση και τη συσταδοποίηση δέντρων ανάπτυξης.
- Χρησιμοποιείται σε αλγορίθμους όπως ο *Generalized Weisfeiler-Lehman Subtree Kernel* για τη πιο «χαλαρή» αντιμετώπιση υποδέντρων που δεν είναι ισομορφικά αλλά έχουν παρόμοια δομή.
- Μέσω της απόστασης *Wasserstein* και του αλγορίθμου *Wasserstein K-Μέσων*, μπορούμε να ομαδοποιήσουμε δέντρα με μικρές διαφορές. (Σχήμα 3.35)



Σχήμα 3.35: *Χρήση του Αλγορίθμου Wasserstein K-Μέσων σε Δέντρα Ανάπτυξης.* [3]

3.3 Εργαλεία Μηχανικής Μάθησης

3.3.1 Logistic Regression

"Data is the new oil."
Clive Humby

Η **Λογιστική Παλινδρόμηση** είναι ένας Ταξινομητής Μηχανικής Μάθησης με Επίβλεψη.Ο ταξινομητής λειτουργεί ως εξής · έχουμε ένα σύνολο κλάσεων και μέσω ενός Συνόλου Δεδομένων Εκπαίδευσης ρυθμίζονται κατάλληλα τα βάρη του μοντέλου. Έπειτα κάθε αντικείμενο τίθεται σε μία από τις υπάρχουσες κλάσεις ανάλογα με την τιμή εξόδου του μοντέλου.

Για το πρόβλημα της Δυαδικής Ταξινόμησης με χρήση Λογιστικής Παλινδρόμησης, αν οι παράμετροι του μοντέλου είναι $\vec{w} = [w_1, w_2, \ldots, w_n]^T, b$ και το διάνυσμα χαρακτηριστικών ενός στοιχείου $\vec{x} \in \mathbb{R}^n$ που θέλουμε να ταξινομήσουμε σε μία κλάση $c \in \{0, 1\}$ είναι $\vec{x} = [x_1, x_2, \ldots, x_n]$, υπολογίζουμε αρχικά τον γραμμικό συνδυασμό που δίνεται από τη Σχέση 2.17:

$$z = \sum_{i=1}^{n} w_i \cdot x_i + b = \vec{w} \cdot \vec{x} + b. \tag{3.13}$$

Στη συνέχεια, χρησιμοποιούμε την **σιγμοειδή συνάρτηση** $\sigma(\cdot) : \mathbb{R} \mapsto (0, 1)$, η οποία ορίζεται ως εξής:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.14}$$

και θέτοντας την τιμή **z** της Σχέσης 3.13 σε αυτή, και ανάλογα με το αν η τιμή $\sigma(z)$ είναι μικρότερη ή μεγαλύτερη του 0.5, εκτιμούμε ότι το στοιχείο $\vec{x}$ ανήκει στην Κλάση 0 ή στην Κλάση 1 αντιστοίχως, όπως φαίνεται και από τη Σχέση 3.15.

$$class(\vec{x}) = \begin{cases} 0 & : \sigma(z) < 0.5 \\ 1 & : \sigma(z) \geq 0.5 \end{cases} \tag{3.15}$$

Το σ(z) ορίζεται ως εξής:

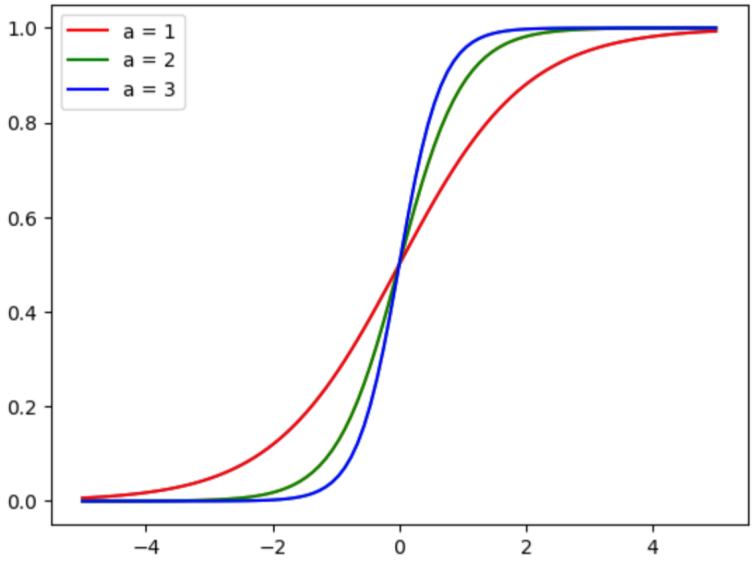
$$\sigma(z) = P(c = 1|\vec{x}) = \frac{1}{1 + e^{-(\vec{x} \cdot \vec{w} + b)}} \tag{3.16}$$

Στο Σχήμα 3.36 βλέπουμε για τις τιμές $a \in \{1, 2, 3\}$ την μορφή της καμπύλης $\sigma(a \cdot x)$. Όσο μεγαλύτερη είναι η τιμή του a τόσο πιο απότομη είναι η κλίση της καμπύλης γύρω από τον άξονα x = 0 και επομένως, για συγκεκριμένη τιμή του z δίδεται μεγαλύτερη πιθανότητα να ανήκει σε μία από τις δύο κλάσεις.

Αυτό αποδεικνύεται ως εξής: Αν θεωρήσουμε την $\sigma(x, a) = \frac{1}{1 + e^{-ax}}$, βρίσκουμε την κλίση της ευθείας στο σημείο x = 0, και βρίσκουμε ότι:

$$\left.\frac{d\sigma(x, a)}{da}\right|_{x=0} = \left.\frac{a \cdot e^{-ax}}{(1 + e^{-ax})^2}\right|_{x=0} = \frac{a}{4}$$

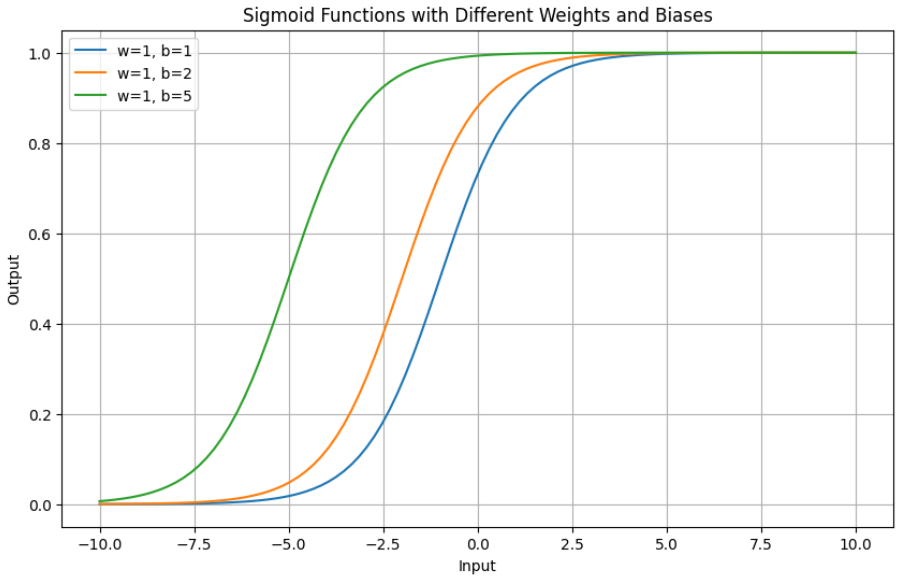
δηλαδή η κλίση της ευθείας στο σημείο x = 0 είναι ανάλογη της παραμέτρου a.



Σχήμα 3.36: *Σιγμοειδής Καμπύλες για διαφορερτικές τιμές Παραμέτρου a.*

Με τη μεταβολή της παραμέτρου a επηρεάζεται αποκλειστικά η κλίση της σιγμοειδούς συνάρτησης. Ωστόσο, για την οριζόντια μετατόπιση της καμπύλης απαιτείται η εισαγωγή μιας παραμέτρου, της **παραμέτρου μεροληψίας** (bias b). Η παράμετρος b επιτρέπει την οριζόντια μετακίνηση της καμπύλης, όπως παρουσιάζεται στο Σχήμα 3.37.

Για παράδειγμα, σε εφαρμογές ανίχνευσης απάτης, αν διαπιστώσουμε ότι η αρχική τιμή της συνάρτησης δεν εξυπηρετεί βέλτιστα τη διάκριση μεταξύ των περιπτώσεων απάτης και μη-απάτης, η παράμετρος b μπορεί να προσαρμοστεί ώστε το όριο απόφασης να μεταφερθεί σε σημείο που βελτιστοποιεί την ακρίβεια των προβλέψεων. Έτσι, η χρήση της b καθίσταται κρίσιμη για την επίτευξη βελτιωμένων αποτελεσμάτων σε πραγματικά προβλήματα ταξινόμησης.



Σχήμα 3.37: *Σιγμοειδής Καμπύλες για διαφορερτικές τιμές Παραμέτρων Μεροληψίας.*

Η συνάρτηση κόστους είναι το άθροισμα της **Διασταυρωμένης Εντροπίας** (*cross-entropy*) όλων των δεδομένων εκπαίδευσης. Για ένα στοιχείο $(\vec{x}^i, y^i)$ του συνόλου εκπαίδευσης, η τιμή της συνάρτησης **διασταυρωμένης εντροπίας** (cross-entropy function) ορίζεται ως:

$$L_{CE}(\hat{y}, y) = -[y \cdot log(\sigma(\vec{w} \cdot \vec{x} + b)) + (1 - y) \cdot log(1 - \sigma(\vec{w} \cdot \vec{x} + b))] \tag{3.17}$$

Έτσι, για ένα Σύνολο Δεδομένων $\left\{(\vec{x}^1, y^1), (\vec{x}^2, y^2), \ldots, (\vec{x}^m, y^m)\right\}$, η συνάρτηση κόστους ορίζεται ως:

$$Cost(\hat{y}, y) = \frac{1}{m} \cdot \sum_{i=1}^{m} L_{CE}(\hat{y}^i, y^i) \tag{3.18}$$

Στο πλαίσιο του βασικού μας αλγορίθμου, σκοπό έχουμε να ταξινομήσουμε δημοσιεύσεις σε δύο κατηγορίες (*true/false*) χρησιμοποιώντας τα χαρακτηριστικά που έχουν εξαχθεί από τα αντίστοιχα δέντρα διάδοσής τους. Αφού έχουμε υπολογίσει μια διανυσματική αναπαράσταση κάθε δέντρου διάδοσης μέσω του Πυρήνα Γραφημάτων *Weisfeiler-Lehman*, το τελευταίο βήμα του βασικού αλγορίθμου μας είναι να εφαρμόσουμε έναν ταξινομητή. Για το σκοπό αυτό, έναν από τους ταξινομητές που επιλέγουμε είναι η Λογιστική Παλινδρόμηση, ώστε να εκτιμήσουμε την πιθανότητα μιας δημοσίευσης να είναι ψευδής ή αληθής.

Το συνολικό κόστος δίδεται από τη Σχέση 3.18. Έτσι, χρησιμοποιώντας τον **Αλγόριθμο Βαθμωτής Καθόδου** για την εκπαίδευση του μοντέλου (που θα παρουσιάσουμε στο κεφάλαιο 3.3.4), υπολογίζονται οι παράμετροι για τους οποίους ελαχιστοποιείται το συνολικό σφάλμα και επομένως μεγιστοποιείται η συνολική ακρίβεια του μοντέλου.

3.3.2 Αλγόριθμος K-Means

"Όμοιος ομοίω αεί πελάζει."
Πλάτων

Κατά την υλοποίηση του βασικού μας αλγορίθμου, προκύπτει συχνά η ανάγκη να ομαδοποιήσουμε δεδομένα σε συστάδες, βάσει της εγγύτητας των διανυσματικών αναπαραστάσεων. Σε αυτό το σημείο εντάσσεται η χρήση του **Αλγορίθμου Κ-Μέσων (K-Means)**, ο οποίος είναι ένας μη επιβλεπόμενος αλγόριθμος συσταδοποίησης. Λαμβάνει ως είσοδο ένα σύνολο δεδομένων και έναν αριθμό συστάδων K, και παράγει ως έξοδο τα κέντρα των συστάδων, καθώς και την ανάθεση κάθε στοιχείου σε κάποια από αυτές.

Στο πλαίσιο του βασικού αλγορίθμου, ο αλγόριθμος Κ-Μέσων χρησιμοποιείται στο στάδιο όπου έχουμε ήδη εξαγάγει διανυσματικές αναπαραστάσεις από τα δεδομένα (π.χ. μέσω χαρακτηριστικών που έχουν υπολογιστεί σε προηγούμενα βήματα). Με βάση αυτές τις αναπαραστάσεις, ο αλγόριθμος αυτός αναλαμβάνει να συσταδοποιήσει τα δεδομένα, ώστε παρόμοια αντικείμενα να καταλήγουν στην ίδια συστάδα, βελτιώνοντας έτσι την οργάνωση και την κατανόηση του συνόλου δεδομένων.

=

Αλγοριθμος 3.4: *Αλγόριθμος K-Μέσων*

Είσοδος: DataSet $X = [\overrightarrow{x_1}, \overrightarrow{x_2}, \ldots, \overrightarrow{x_n}]$, Number of clusters K
Έξοδος: Cluster centroids $C = \{c_1, c_2, \ldots, c_K\}$, Cluster assignments $A = \{a_1, a_2, \ldots, a_n\}$

1: **Βασικά Βήματα του Αλγορίθμου:**
2: **Βήμα 1: Αρχικοποίηση** Επιλέγουμε τυχαία K σημεία από το X ως αρχικά κέντρα $C = \{c_1, c_2, \ldots, c_K\}$.
3: Δημιουργούμε έναν πίνακα ανάθεσης A με κενές τιμές, όπου a_i δείχνει σε ποια συστάδα ανήκει το στοιχείο $\overrightarrow{x_i}$.
4: **Βήμα 2: Ανάθεση Στοιχείων σε Συστάδες** Για κάθε $\overrightarrow{x_i}$, υπολογίζουμε την απόσταση $d(x_i, c_j)$ από κάθε κέντρο c_j. Αναθέτουμε το $\overrightarrow{x_i}$ στη συστάδα του εγγύτερου κέντρου:

$$a_i = \arg\min_j d(\overrightarrow{x_i}, c_j).$$

5: **Βήμα 3: Ενημέρωση Κέντρων** Για κάθε συστάδα j, υπολογίζουμε το νέο κέντρο c_j ως το μέσο όλων των στοιχείων που ανήκουν στη συστάδα αυτή.
6: **Βήμα 4: Έλεγχος για Σύγκλιση** Αν τα κέντρα δεν αλλάζουν σημαντικά ή έχει συμπληρωθεί το μέγιστο πλήθος επαναλήψεων, ο αλγόριθμος σταματά.

Σχήμα 3.38: *Ψευδοκώδικας Αλγορίθμου K-Μέσων*

Ψευδοκώδικας

Ανάλυση των Βημάτων

1. **Αρχικοποίηση:** Ο K-Means είναι ένας αλγόριθμος συσταδοποίησης που χρησιμοποιείται όταν θέλουμε να ομαδοποιήσουμε δεδομένα σε ομοιογενείς ομάδες (σύσταδες). Αρχικά, επιλέγουμε τυχαία K στοιχεία ως κέντρα. Αυτή η τυχαία επιλογή εξασφαλίζει διαφορετικές αρχικές διαμερίσεις, επηρεάζοντας την τελική δομή των συστάδων.

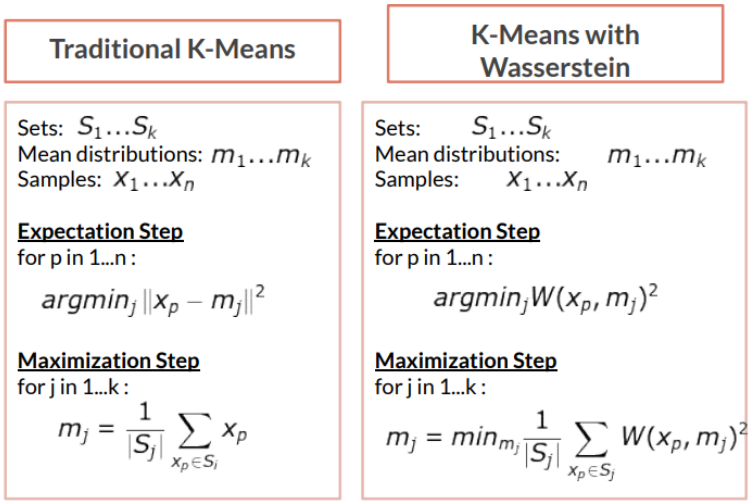
2. **Ανάθεση Στοιχείων:** Χρησιμοποιώντας τη μετρική απόστασης (π.χ. Ευκλείδεια απόσταση, απόσταση Wasserstein), αναθέτουμε κάθε στοιχείο στην πλησιέστερη συστάδα. Αυτό αποτελεί το κυρίως αλγοριθμικό βήμα όπου οι διανυσματικές αναπαραστάσεις αξιοποιούνται απευθείας.

3. **Ενημέρωση Κέντρων:** Μετά την ανάθεση, υπολογίζουμε το μέσο των σημείων σε κάθε συστάδα για να ενημερώσουμε το κέντρο της. Αυτό το βήμα επαναλαμβάνεται σε κάθε επανάληψη, οδηγώντας σταδιακά σε πιο σταθερές συστάδες.

4. **Έλεγχος Σύγκλισης:** Ο αλγόριθμος σταματά όταν τα κέντρα δεν μεταβάλλονται σημαντικά ή όταν συμπληρωθεί ένας μέγιστος αριθμός επαναλήψεων. Στο βασικό αλγόριθμο, αυτό σημαίνει ότι έχουμε επιτύχει μια σταθερή συσταδοποίηση των δεδομένων.

Στον Αλγόριθμο *Wasserstein* K-Μέσων το βαρύκεντρο χρησιμοποιείται όπως χρησιμοποιείται το κέντρο μάζας για ένα σύνολο στοιχείων που ανήκουν σε μία συγκεκριμένη συστάδα. Η διαφορά των δύο αλγορίθμων έγκειται στο πως υπολογίζουμε αποστάσεις για την εύρεση της συστάδας στην οποία ανήκει το κάθε στοιχείο του συνόλου καθώς και στο πως

ανανεώνουμε τα καινούργια κέντρα της κάθε συστάδας. Στο Σχήμα 3.39 βλέπουμε τις διαφορές των αλγορίθμων **K-Μέσων** και *Wassertein* **K-Μέσων**..



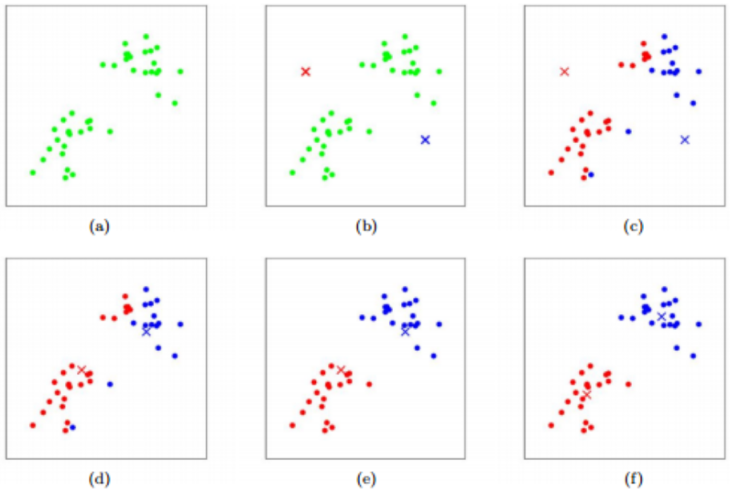
Σχήμα 3.39: *k-Means vs Wasserstein k-Means*

Παραδείγματα Εκτέλεσης

Παράδειγμα σε 2 Διαστάσεις: Στο Σχήμα 3.40 παρουσιάζεται η λειτουργία του αλγορίθμου Κ-Μέσων Κ-Μεανς για $K = 2$ συστάδες και 3 επαναλήψεις. Αρχικά, επιλέγονται τυχαία δύο σημεία ως κέντρα. Σε κάθε επανάληψη:

- Αναθέτουμε κάθε σημείο στη συστάδα εκείνη της οποίας το κέντρο είναι εγγύτερα.
- Υπολογίζουμε τα νέα κέντρα κάθε συστάδας ως το βαρύκεντρο των σημείων που ανήκουν στην συγκεκριμένη συστάδα.

Όπως φαίνεται και στο το παρακάτω σχήμαα, μετά από 3 επαναλήψεις, οι συστάδες σταθεροποιούνται.



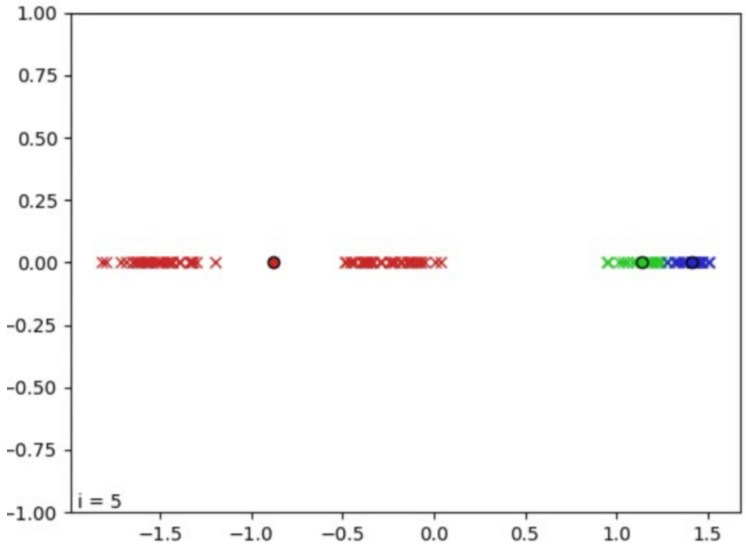
Σχήμα 3.40: *Αλγόριθμος Κ-Μέσων για 3 επαναλήψεις και* $K = 2$.

Παράδειγμα σε 1 Διάσταση: Σε 1-Διάσταση, ο αλγόριθμος K-Μέσων (K-Means) μπορεί να δώσει μια πιο ¨δίκαιη¨ κατανομή τιμών σε συστάδες. Έστω ο πίνακας τιμών: $[1, 5, 10, 55, 453, 784, 964, 1043, 4382, 7382, 9013]$. Χωρίς k-Means, μπορεί να προκύψει μια ομαδοποίηση που δεν αντικατοπτρίζει πραγματικά την εγγύτητα των τιμών. Αν εφαρμόσουμε τον K-Means με $K = 4$, προκύπτει η εξής διαμέριση:

$$\{\{1, 5, 10, 55, 453, 784, 964, 1043\}, \{4382\}, \{7382\}, \{9013\}\}.$$

Τα κέντρα των συστάδων είναι:

$$\{414.375, 4382.0, 7382.0, 9013.0\}.$$



Σχήμα 3.41: *Αναπαράσταση αποτελέσματος Αλγορίθμου K-Μέσων στη 1-Διάσταση μετά από 5 επαναλήψεις και $K = 3$.*

Σε αυτό το παράδειγμα, το K-Means σε μία διάσταση προσφέρει μια πιο ρεαλιστική ομαδοποίηση των τιμών, αναδεικνύοντας τη σημασία της εφαρμογής του αλγορίθμου σε διαφορετικούς χώρους χαρακτηριστικών για τη βέλτιστη οργάνωση των δεδομένων, συγκριτικά με άλλες τεχνικές όπως την ομαδοποίηση τιμών σε λογαριθμικά διαστήματα (log-bin bucketing [43]).

Ανανέωση των Κέντρων

Μετά την αρχική ανάθεση των σημείων στις συστάδες , ακολουθεί η ανανέωση των κέντρων τους. Αυτή η ανανέωση είναι καθοριστική για την προοδευτική βελτίωση της ποιότητας των συστάδων, καθώς σε κάθε επανάληψη τα κέντρα προσαρμόζονται ώστε να αντιπροσωπεύουν καλύτερα τα δεδομένα που τους έχουν ανατεθεί.

Βήματα στον Βασικό Αλγόριθμο:

1. **Ανάθεση στοιχείων σε συστάδες:** Αφού υπολογίσουμε την απόσταση κάθε σημείου από τα υπάρχοντα κέντρα, αναθέτουμε κάθε σημείο στην κοντινότερη συστάδα.

2. **Ανανέωση κέντρων:** Με βάση την ανάθεση, επανυπολογίζουμε τα κέντρα των συστάδων. Η διαδικασία αυτή εγγυάται ότι οι συστάδες θα είναι πιο ομοιογενείς σε κάθε επόμενη επανάληψη.

Έστω ότι σε μια συγκεκριμένη επανάληψη του αλγορίθμου *K-Μέσων* έχουμε υπολογίσει τον πίνακα ανάθεσης, δηλαδή γνωρίζουμε σε ποια συστάδα ανήκει κάθε στοιχείο. Ορίζουμε:

$$\omega_j = \{\vec{x} \in X \mid dist(\vec{x}, \vec{c_j}) < dist(\vec{x}, \vec{c_i}), j \neq i \wedge i,j \in [K]\},$$

όπου $\vec{c_i}, \vec{c_j}$ είναι τα κέντρα των συστάδων i,j αντίστοιχα, ω_j το σύνολο των σημείων που ανήκουν στη συστάδα j, και $dist(u, v)$ η Ευκλείδεια απόσταση μεταξύ των σημείων u και v.

Ο υπολογισμός του νέου κέντρου $\vec{c_j}$ βασίζεται στην ελαχιστοποίηση μίας συνάρτησης κόστους L_j, η οποία ορίζεται ως το άθροισμα των τετραγώνων των αποστάσεων όλων των σημείων της συστάδας ω_j από το κέντρο $\vec{c_j}$:

$$L_j = \sum_{\vec{x} \in \omega_j} |\vec{x} - \vec{c_j}|^2.$$

Αν $\vec{x} = [x_1, x_2, \ldots, x_n]$ και $\vec{c_j} = [c_1, c_2, \ldots, c_n]$, τότε:

$$|\vec{x} - \vec{c_j}|^2 = (x_1 - c_1)^2 + (x_2 - c_2)^2 + \cdots + (x_n - c_n)^2.$$

Θεωρούμε το σύνολο $\omega_j = \{\vec{x_1}, \vec{x_2}, \ldots, \vec{x_{n_j}}\}$ που περιλαμβάνει τα στοιχεία που έχουν ανατεθεί στην κλάση j. Θέλουμε να βρούμε το σημείο $\vec{c_j}$ που ελαχιστοποιεί το L_j.

Αναπτύσσοντας τη συνάρτηση κόστους:

$$L_j = \sum_{i=1}^{n_j} |\vec{x_i} - \vec{c_j}|^2 = \sum_{i=1}^{n_j} \sum_{k=1}^{n} (x_{ik} - c_{jk})^2.$$

Για να βρούμε το σημείο $\vec{c_j} = [c_1, c_2, \ldots, c_n]$ που ελαχιστοποιεί το L_j, εξετάζουμε τις μερικές παραγώγους της L_j ως προς c_k:

$$\frac{\partial L_j}{\partial c_k} = \sum_{i=1}^{n_j} 2(x_{ik} - c_k)(-1) = -2 \sum_{i=1}^{n_j} (x_{ik} - c_k).$$

Θέτοντας $\frac{\partial L_j}{\partial c_k} = 0$, καταλήγουμε:

$$\sum_{i=1}^{n_j} (x_{ik} - c_k) = 0 \implies n_j c_k = \sum_{i=1}^{n_j} x_{ik}.$$

Άρα:

$$c_k = \frac{x_{1k} + x_{2k} + \cdots + x_{n_j k}}{n_j}, \quad k \in [n].$$

Συνεπώς, το κέντρο μιας συστάδας ισούται με το σημείο των οποίων οι συνιστώσες είναι οι μέσες τιμές των αντίστοιχων συνιστωσών των στοιχείων που ανήκουν στη συστάδα. Η εξαγωγή αυτή είναι θεμελιώδης και χρησιμοποιείται σε κάθε επανάληψη του αλγορίθμου *K-Μέσων* κατά την ανανέωση των κέντρων, οδηγώντας τελικά σε σύγκλιση του αλγορίθμου σε μια σταθερή διαμέριση των δεδομένων.

3.3.3 Gradient Boosted Trees

¨Η ισχύς ευ τη ευώσει.¨
Αίσωπος

Η τεχνική των **Gradient Boosted Trees** ανήκει στη κατηγορία αλγορίθμων ενίσχυσης (*Boosting*), επιτρέποντας τη δημιουργία ενός ισχυρού προβλεπτικού μοντέλου από ένα σύνολο απλών και σχετικά αδύναμων μοντέλων. Στόχος είναι, ακόμα κι αν κάθε επιμέρους μοντέλο ($M_1, M_2, \ldots, M_k$) παρουσιάζει χαμηλή προβλεπτική ικανότητα, ο συνδυασμός τους να οδηγεί σε ένα τελικό μοντέλο με υψηλή ακρίβεια. Στον βασικό μας αλγόριθμο, η μέθοδος των Gradient Boosted Trees χρησιμοποιείται στο στάδιο όπου επιδιώκουμε τη βέλτιστη αξιοποίηση μιας συλλογής απλών μοντέλων δέντρων, αναβαθμίζοντας διαδοχικά την απόδοσή τους σε κάθε επανάληψη. Παρακάτω βλέπουμε τον αλγόριθμο για την τεχνική αυτή και στη συνέχεια δίνουμε εξηγείται η χρήση καθενός βήματος πως πραγματοποιείται.

Αλγοριθμος 3.5: *Αλγόριθμος Gradient Boosted Trees*

Είσοδος: Σύνολο δεδομένων (X, Y), όπου X είναι τα χαρακτηριστικά εισόδου και Y οι επιθυμητές επιγραφές εξόδου, καθώς και αριθμός ασθενών μοντέλων k.
Έξοδος: Τελικό μοντέλο $\hat{M}$ που συνδυάζει τα μοντέλα $\{M_1, M_2, \ldots, M_k\}$ για υψηλή ακρίβεια.

1: **Βήμα 1: Αρχικοποίηση** Ξεκινάμε με ένα απλό μοντέλο M_0.
2: **Βήμα 2: Υπολογισμός Υπολοίπων** Για κάθε δείγμα, υπολογίζουμε την απόκλιση μεταξύ της προβλεπόμενης τιμής του τρέχοντος μοντέλου και της πραγματικής τιμής, δημιουργώντας ένα διάνυσμα υπολοίπων.
3: **Βήμα 3: Εκπαίδευση Νέου Δέντρου** Εκπαιδεύουμε ένα νέο ασθενές μοντέλο δέντρου M_i πάνω στα τρέχοντα υπόλοιπα, επιχειρώντας να βελτιώσουμε τις προβλέψεις εκεί όπου το τρέχον μοντέλο υστερεί.
4: **Βήμα 4: Ενημέρωση του Συνδυαστικού Μοντέλου** Προσθέτουμε το νέο δέντρο M_i στο συνδυαστικό μοντέλο, αναπροσαρμόζοντας τις προβλέψεις ώστε να μειώνεται το σφάλμα.
5: **Βήμα 5: Επανάληψη** Επαναλαμβάνουμε τα βήματα 2-4 για k φορές ή μέχρι να επιτευχθεί σύγκλιση. Το τελικό μοντέλο είναι το άθροισμα όλων των ασθενών μοντέλων που έχουν προστεθεί διαδοχικά.

Επεξήγηση των βημάτων του Αλγορίθμου

1. **Αρχικοποίηση**: Η διαδικασία ξεκινά από ένα απλό μοντέλο. Στον βασικό αλγόριθμό μας, αυτό σημαίνει ότι πριν χρησιμοποιηθούν τα Γραδιεντ Βοοστεδ Τρεες, έχουμε ήδη κατασκευάσει ένα αρχικό σημείο εκκίνησης.

2. **Υπολογισμός Υπολοίπων**: Τα υπόλοιπα εκφράζουν πόσο μακριά βρίσκονται οι τρέχουσες προβλέψεις του μοντέλου από τις πραγματικές τιμές. Αποτελούν τον οδηγό για το πώς θα βελτιώσουμε το μοντέλο σε επόμενα βήματα.

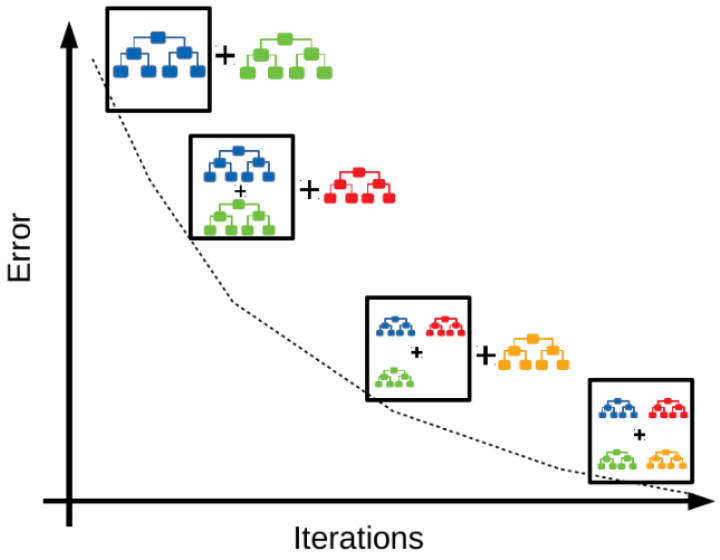
3. **Εκπαίδευση Νέου Δέντρου**: Σε κάθε επανάληψη, εκπαιδεύουμε ένα νέο, απλό δέντρο (weak learner) πάνω στα υπόλοιπα. Αυτό το νέο δέντρο εστιάζει στα σημεία όπου το μοντέλο αποτυγχάνει, βελτιώνοντας σταδιακά την απόδοση.

4. **Ενημέρωση του Συνδυαστικού Μοντέλου**: Προσθέτοντας το νέο δέντρο στο μοντέλο, μειώνουμε το συνολικό σφάλμα. Σε αυτό το στάδιο του βασικού αλγορίθμου, τα Γραδιεντ Βοοστεδ Τρεες αξιοποιούν τα απλά δέντρα ώστε να διορθώνουν συστηματικά τα λάθη των προηγούμενων βημάτων.

5. **Επανάληψη έως Σύγκλιση**: Η διαδικασία επαναλαμβάνεται μέχρι η βελτίωση να πάψει να είναι σημαντική. Τελικά, το συνδυαστικό μοντέλο, που αποτελείται από την αθροιστική συνεισφορά όλων των απλών δέντρων, παρουσιάζει υψηλή προβλεπτική ικανότητα.

Σχηματική αναπαράσταση της Βελτίωσης του τελικού μοντέλου

Στο Σχήμα 3.42 παρουσιάζεται η λογική της μεθόδου *Gradient Boosting*. Σε κάθε επανάληψη, προσθέτουμε ένα μοντέλο που συνεισφέρει στη μείωση του σφάλματος. Με αυτόν τον τρόπο, η τελική πρόβλεψη προσεγγίζει ολοένα και περισσότερο τις πραγματικές τιμές.



Σχήμα 3.42: *Ελαχιστοποίηση Σφάλματος με χρήση Gradient Boosted Trees.*

Συνδυασμός Μοντέλων σε περίπτωση Ταξινόμησης

Για προβλήματα ταξινόμησης με κλάσεις $\{C_1, C_2, \ldots, C_n\}$, αν τα $\{M_1, M_2, \ldots, M_k\}$ είναι τα μοντέλα που έχουν προστεθεί διαδοχικά, και αν $y_k^{(i)}(x)$ είναι η έξοδος του k-οστού μοντέλου για την κλάση i, τότε, αν θεωρήσουμε όλα τα μοντέλα ισοδύναμα, το τελικό συνδυαστικό

μοντέλο αποφασίζει την κλάση μέσω ψηφοφορίας:

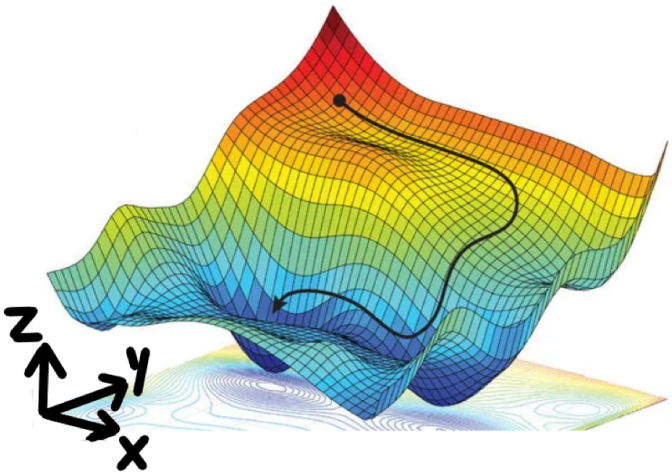
$$\hat{y}(x) = \arg\max_{c} \sum_{i=1}^{M} y_i^{(c)}(x).$$

Με αυτόν τον τρόπο, τα Gradient Boosted Trees αξιοποιούν σταδιακά τα ασθενή μοντέλα για να συγκλίνουν σε ένα τελικό, ισχυρό μοντέλο υψηλής ακρίβειας.

3.3.4 Αλγόριθμος Βαθμωτής Καθόδου

¨Η επιτυχία είναι το άθροισμα των
μικρών προσπαθειών, που
επαναλαμβάνονται μέρα με τη μέρα.¨
Robert Collier

Ο **Αλγόριθμος Βαθμωτής Καθόδου** είναι ένας αλγόριθμος που χρησιμοποιείται για τον υπολογισμό των παραμέτρων που ελαχιστοποιούν μια συνάρτηση κόστους. Γεωμετρικά η διαδικασία στηρίζεται στην ακόλουθη ιδέα · ξεκινάμε από ένα αρχικό σημείο και προσπαθούμε να βρούμε μετά από ένα πλήθος επαναλήψεων ένα τελικό σημείο στο οποίο να ελαχιστοποιείται η συνάρτηση κόστους. Για παράδειγμα στο Σχήμα 3.43 ,οι άξονες x,y αναπαριστούν τις μεταβλητές που θέλουμε να ελαχιστοποιήσουμε και ο άξονας z αναπαριστούν την τιμή της συνάρτησης κόστους για τις συγκεκριμένες τιμές x,y. Βλέπουμε πώς ξεκινώντας από ένα αρχικό σημείο, καταλήγουμε σε ένα σημείο στο οποίο η συνάρτηση ελαχιστοποιείται.



Σχήμα 3.43: *Εύρεση ελαχίστου με τον Αλγόριθμο Βαθμωτής Καθόδου.*

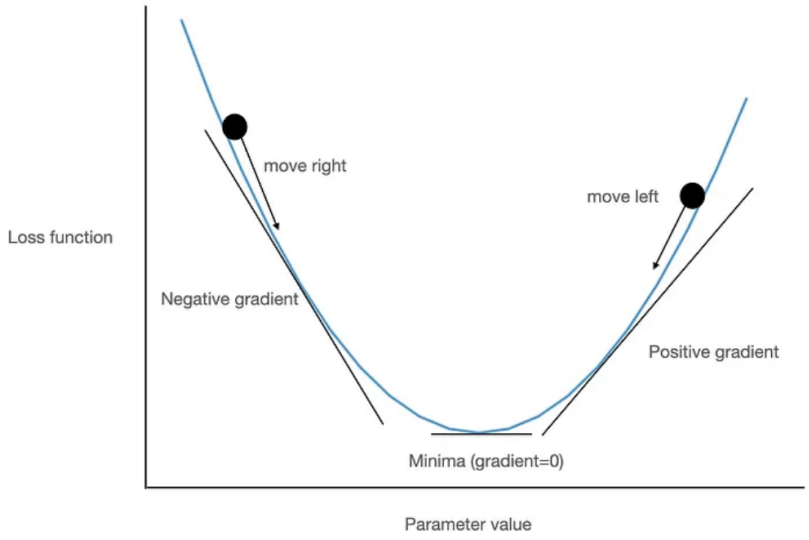
Πιο τυπικά, έστω ότι έχουμε έναν εκτιμητή $\hat{y}$ = f(x;w) ο οποίος κάνει πρόβλεψη για ένα στοιχείο x χρησιμιποιώντας την παράμετρο w. Αν ορίσουμε ως L(y,$\hat{y}$) μία μία συνάρτηση κόστους, τότε σκοπός μας είναι να ελαχιστοποιήσουμε την ποσότητα:

$$L(\hat{y}, y) = L(f(x; w), y) \tag{3.19}$$

Αν χρησιμοποιήσουμε μία παράμετρο, τον **ρυθμό μάθησης η**, μία σταθερά που ορίζει το μήκος του βήματος προς την βέλτιστη λύση, η λύση του παραπάνω προβλήματος γίνεται με χρήση της αναδρομικής σχέσης:

$$w^{t+1} = w^t - \eta \left. \frac{dL(f(x;w),y)}{dw} \right|_{w=w^t}, \text{ με } w^0 \text{ τυχαία επιλεγμένο} \tag{3.20}$$

Στο Σχήμα 3.44 θέλουμε να προσδιορίσουμε το ελάχιστο μιας συνάρτησης κόστους μιας μεταβλητής. Αν η κλίση είναι θετική, ανανεώνουμε το παράμετρο μειώνοντας την τιμή της. Αν η κλίση είναι αρνητική, ανανεώνουμε την παράμετρο αυξάνοντας την τιμή της παραμέτρου. Με αυτόν τον τρόπο η ανανέωση των παραμέτρων έχουν κατεύθυνση αντίθετη της κλίσης της ευθείας στο συγκεκριμένο σημείο.



Σχήμα 3.44: *Εύρεση ελαχίστου μίας παραμέτρου.*

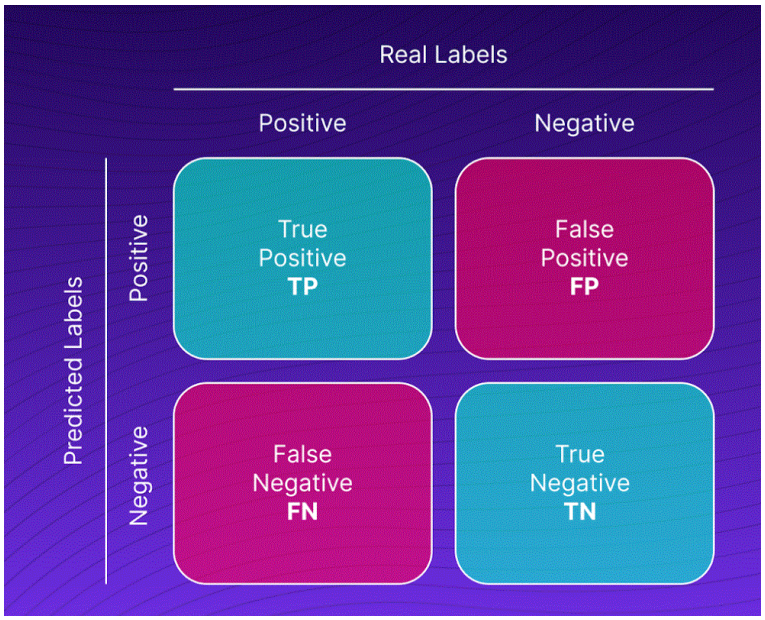
3.3.5 Μετρικές Απόδοσης των Μοντέλων

¨Το να είσαι γρήγορος είναι καλό,
αλλά η ακρίβεια είναι το παν.¨
Wyatt Earp

Για την εκτίμηση της απόδοσης των μοντέλων και της προβλεπτικής τους ικανότητας, είναι σημαντικό να ορίσουμε μετρικές με τις οποίες να υπολογίζουμε την ακρίβεια του μοντέλου.

Στο Σχήμα 3.45 , βλέπουμε τον **Πίνακα Σύγχυσης** στον οποίο ορίζονται τα εξής μεγέθη:

- TP = Τα στοιχεία που προβλέφθηκα ως True και ανήκουν στην κλάση True.
- TN = Τα στοιχεία που προβλέφθηκαν ως False και ανήκουν στην κλάση False.
- FP = Τα στοιχεία που προβλέφθηκαν ως True αλλά ανήκουν στην κλάση False.
- FN = Τα στοιχεία που προβλέφθηκαν ως False αλλά ανήκουν στην κλάση False.



Σχήμα 3.45: *Confusion Matrix*

Η πιο απλοϊκή μετρική είναι η Ακρίβεια (Accuracy), η οποία ορίζεται όπως φαίνεται ακολούθως:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Σε περίπτωση που τα δεδομένα μας είναι ¨ανισσόροπα¨, δηλαδή το σύνολο στοιχείων της κλάσης True διαφέρει σημαντικά από το σύνολο στοιχείων της κλάσης False, αυτή η μετρική δεν μας δίνει ακριβή εικόνα της προβλεπτικής ικανότητας του μοντέλου. Για παράδειγμα, στην περίπτωση που είχαμε 9900 στοιχεία της κλάσης True και 100 στοιχεία της κλάσης False, και από τα 100 στοιχεία της κλάσης False προβλέψουμε σωστά το 1 και τα 9899 από τα True τελικά θα έχουμε accuracy = 0.99. Ωστόσο, στα *False* η προβλεπτική ικανότητα τοου μοντέλου είναι εξαιρετικά χαμηλή.

Γι ′ αυτό ορίζουμε τις ακόλουθες μετρικές:

- Precision = Το πόσοστό των στοιχείων που προβλέφθηκαν ότι ανήκουν στην κλάση True και ανήκουν στην κλάση αυτή προς το σύνολο των στοιχείων που προβλέφθηκαν ότι ανήκουν στην κλάση True.

$$Precision = \frac{TP}{TP + FP}$$

- Recall = Το πόσοστό των στοιχείων που προβλέφθηκαν ότι ανήκουν στην κλάση True και ανήκουν στην κλάση αυτή προ προς το σύνολο των στοιχείων της κλάσης True.

$$Recall = \frac{TP}{TP + FN}$$

Στην υλοποίηση μας, θα χρησιμοποιήσουμε τη μετρική F1-score , η οποία ορίζεται μέσω των μεγεθών Precision και Recall όπως φαίνεται ακολούθως:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Υπάρχουν επίσης οι μετρικές Sensitivity και Specificity που ορίζονται ως:

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

και ορίζονται ως:

- Sensitivity = Το πόσοστό των στοιχείων που προβλέφθηκαν ότι ανήκουν στην κλάση True και ανήκουν στην κλάση αυτή προς το σύνολο των στοιχείων που ανήκουν στην κλάση True.
- Specitivity = Το πόσοστό των στοιχείων που προβλέφθηκαν ότι ανήκουν στην κλάση False και ανήκουν στην κλάση αυτή προς το σύνολο των στοιχείων της κλάσης False.

Μέρος III

Πρακτικό Μέρος

Κεφάλαιο 4

Υλοποίηση Γραφοθεωρητικής Μεθόδου Πρόγνωσης

4.1 Εισαγωγή

Στο κεφάλαιο που ακολουθεί, θα εξετάσουμε το σχεδιασμό και την αρχιτεκτονική του συστήματος που υλοποιήθηκε για την μελέτη της συσχέτισης του τρόπου διάδοσης με την εγκυρότητα της διαδιδούμενης πληροφορίας. Αρχικά, θα αναλύσουμε τον τρόπο με τον οποίο έχει μοντελοποιηθεί το πρόβλημα και τον σχεδιασμό της προσέγγισής μας. Στη συνέχεια, θα παρουσιάσουμε τις προτεινόμενες προσεγγίσεις μοντελοποίησης και την αρχιτεκτονική του συστήματος που αναπτύχθηκε. Τέλος, θα γίνει μια σύντομη σύνοψη των κυριότερων σημείων του κεφαλαίου.

4.2 Μοντελοποίηση του Προβλήματος

4.2.1 Προβληματική και Σχεδιασμός Προσέγγισης

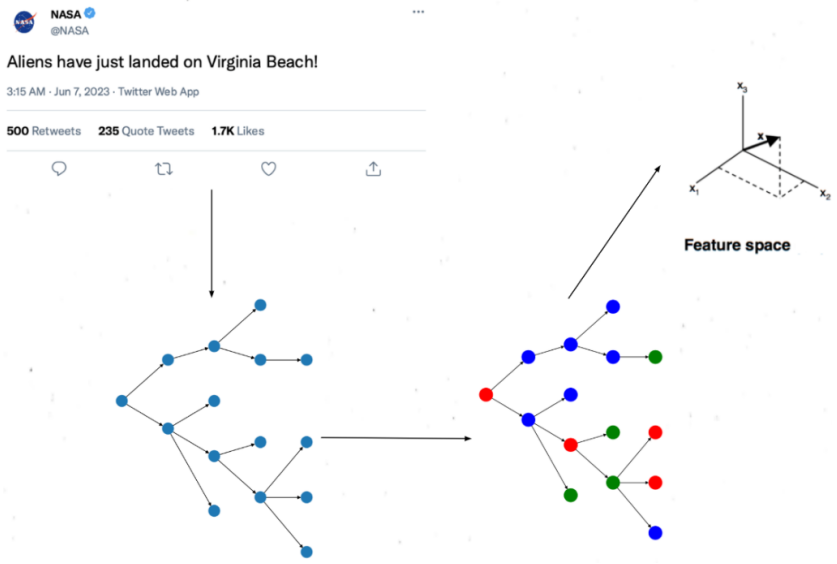
"Το όλον είναι μεγαλύτερο από το άθροισμα των μερών του."
Αριστοτέλης

Όπως επισημάνθηκε στην Εισαγωγή, το αντικείμενο της εργασίας επικεντρώνεται στην ανίχνευση ψευδών ειδήσεων σε μέσα κοινωνικής δικτύωσης, αξιοποιώντας αποκλειστικά δέντρα διάδοσης πληροφορίας.

Η παραπάνω διατύπωση οδηγεί σε δύο επιμέρους υποπροβλήματα:

1. **Κωδικοποίηση ενός δέντρου διάδοσης με ετικέτες στους κόμβους σε μια διανυσματική αναπαράσταση.**
2. **Αξιοποίηση ενός δυαδικού ταξινομητή για διάκριση μεταξύ αληθών και ψευδών ειδήσεων.**

Στο Σχήμα 4.1 παρουσιάζεται η διαδικασία που ακολουθείται για το πρώτο βήμα. Από μια δεδομένη δημοσίευση, για την οποία διαθέτουμε το δέντρο διάδοσης, δημιουργούμε ένα δέντρο με επιγραφές στους κόμβους (στο Σχήμα αποδίδονται ως χρώματα κόμβων). Κατόπιν, μέσω Πυρήνων Γραφημάτων (Graph Kernels), προκύπτουν τα διανύσματα χαρακτηριστικών για κάθε δέντρο.



Σχήμα 4.1: *Διαδικασία μετατροπής δέντρων διάδοσης σε διανύσματα χαρακτηριστικών.*

4.2.2 Γραφοθεωρητική Αναπαράσταση

Το σύνολο εισόδου αποτελείται από γραφήματα (δέντρα διάδοσης) με ετικέτες κλάσεων, οι οποίες στην υπό μελέτη περίπτωση αντιστοιχούν στις κατηγορίες Άληθής' και Ψευδής'. Κάθε κόμβος του γραφήματος διάδοσης αντιστοιχεί σε έναν μοναδικό χρήστη, ενώ οι κατευθυνόμενες ακμές αναπαριστούν τη ροή της πληροφορίας μέσω της διαδικασίας της αναδημοσίευσης μιας είδησης από έναν χρήστη προς τους ακολούθους του. Το σύνολο δεδομένων περιλαμβάνει δημοσιεύσεις με επισήμανση εγκυρότητας (binary class labels: «αληθής» / «ψευδής»), καθώς και πληροφορία σχετική με τη δομή και τη δυναμική της διάδοσής τους.

Κύριος στόχος της έρευνας είναι η διερεύνηση της συσχέτισης ανάμεσα στην τοπολογική δομή των γραφημάτων διάδοσης και την εγκυρότητα της διαδιδόμενης πληροφορίας. Επιπλέον, για την ενίσχυση της ανάλυσης, κάθε κόμβος εμπλουτίζεται με μεταδεδομένα που αντικατοπτρίζουν χαρακτηριστικά του αντίστοιχου χρήστη προκειμένου να εξεταστεί ο πιθανός ρόλος τους στη διαδικασία διάδοσης.

Οι επιγραφές που αποδίδονται στους κόμβους περικλείουν πληροφορίες σχετικές με τους χρήστες, π. χ. αν διαθέτουμε πρόσθετα μεταδεδομένα (για παράδειγμα στατιστικά του κάθε χρήστη ή πρόσθετες πληροφορίες σχετικά με αυτούς). Στην περίπτωση που τα εν λόγω δεδομένα δεν είναι διαθέσιμα, μπορούμε να χρησιμοποιήσουμε διαφόρων τύπων «τοπικά» χαρακτηριστικά των κόμβων, όπως τον εξωτερικό βαθμό (out-degree). Ανάλογα με την πηγή αυτών των χαρακτηριστικών, διακρίνουμε:

1. **Επισήμανση βάσει καταρράκτη (Local Labeling):** Σε αυτήν την προσέγγιση, η ετικέτα κάθε κόμβου καθορίζεται αποκλειστικά από τοπικές ιδιότητες και δομικά χαρακτηριστικά που απορρέουν από τη θέση του εντός του συγκεκριμένου δέντρου διάδοσης. Ενδεικτικά, τέτοια γνωρίσματα μπορεί να είναι το βάθος από τη ρίζα ή ο εξωτερικός βαθμός του κάθε κόμβου [43].

2. **Επισήμανση βάσει δικτύου (Interconnected Labeling):** Εδώ, η ετικέτα κάθε κόμβου βασίζεται σε συνολικές πληροφορίες που αφορούν τον χρήστη που τον αντιπρο-

σωπεύει, όπως αυτές προκύπτουν από τη συλλογική του δραστηριότητα σε πολλά και διαφορετικά δίκτυα διάδοσης. Η μέθοδος αυτή εισάγει χαρακτηριστικά που αντανακλούν παγκόσμια μοτίβα συμπεριφοράς, όπως η συνολική συμμετοχή, ο μέσος βαθμός επιρροής ή δείκτες αξιοπιστίας του χρήστη σε διαχρονικό ορίζοντα.

4.2.3 Χρήση Weisfeiler-Lehman Πυρήνων Γραφημάτων

Η επιλογή κατάλληλου πυρήνα γραφημάτων εξαρτάται από τη μορφή και τις ιδιότητες των προς ανάλυση δομών. Στο πρόβλημα της ανίχνευσης παραπληροφόρησης, όπου τα δεδομένα έχουν τη μορφή δέντρων διάδοσης (cascades) με σχετικά μικρό βάθος, οι πυρήνες που βασίζονται σε τυχαίους περιπάτους ή μονοπάτια δεν αποδίδουν επαρκώς, καθώς αδυνατούν να εκμεταλλευτούν αποτελεσματικά τη χωρική και ιεραρχική πληροφορία τέτοιων δομών.

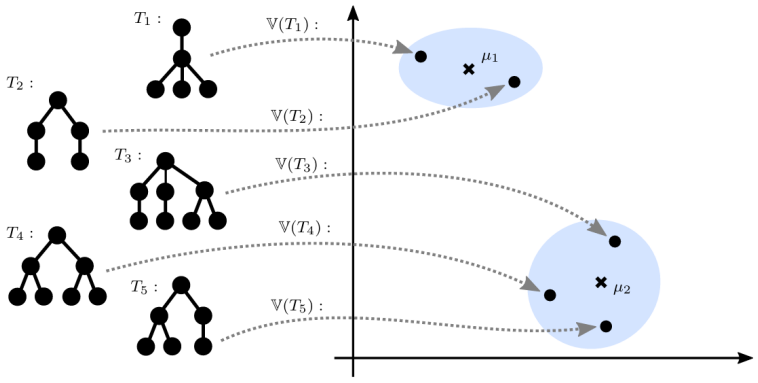
Αντίθετα, οι πυρήνες υποδέντρων και ειδικότερα ο Πυρήνας Γραφημάτων Weisfeiler-Lehman είναι ιδιαίτερα κατάλληλοι για την αποτύπωση των τοπικών και ιεραρχικών υποδομών που χαρακτηρίζουν τη διάδοση φημών ή ψευδών ειδήσεων σε κοινωνικά δίκτυα. Ο συγκεκριμένος πυρήνας βασίζεται στην επαναληπτική αναρίθμηση των επιγραφών των κόμβων (μέσω χρωματισμού γειτονιάς), προσφέροντας τη δυνατότητα για σύγκριση μεταξύ δέντρων με βάση την κατανομή και τη συχνότητα εμφάνισης ισομορφικών υποδέντρων. Με αυτόν τον τρόπο, οι τοπικές πληροφορίες ενοποιούνται με τη συνολική μορφολογία κάθε δέντρου διάδοσης, οδηγώντας σε διανυσματικές αναπαραστάσεις των γραφημάτων οι οποίες μετά χρησιμοποιούνται σε ταξινομητικά μοντέλα για την εκτίμηση της εγκυρότητας της διαδιδούμενης πληροφορίας.

Συμπερασματικά, η χρήση του Weisfeiler-Lehman Subtree Kernel αποτελεί θεμελιώδη επιλογή για την ανάλυση και σύγκριση δέντρων διάδοσης, καθώς επιτρέπει την εκμετάλλευση της δομικής πληροφορίας που είναι κρίσιμη για τον διαχωρισμό φημών και έγκυρων ειδήσεων.

4.2.4 Χρήση γενικευμένου πυρήνα Weisfeiler-Lehman με χρήση Wasserstein k-Means

Για να αντιμετωπίσουμε περιπτώσεις όπου δύο μοτίβα υποδέντρων είναι «σχεδόν ισομορφικά», αλλά όχι απόλυτα [3], υλοποιούμε ένα πρόσθετο βήμα κωδικοποίησης και ομαδοποίησης. Ειδικότερα, κωδικοποιούμε τα *δέντρα ανάπτυξης* (unfolding trees)—δηλαδή, την τοπική δομή που προκύπτει αν εξετάσουμε όλες τις διαδρομές από έναν κόμβο έως ένα συγκεκριμένο βάθος—σε διανυσματικούς χώρους. Στη συνέχεια, τα δέντρα ανάπτυξης κωδικοποιούνται σε διανυσματικές αναπαραστάσεις και, με εφαρμογή του αλγορίθμου K-Means, πραγματοποιείται συσταδοποίηση, έτσι ώστε σχεδόν ισομορφικά μοτίβα να αντιστοιχούν στην ίδια συστάδα.

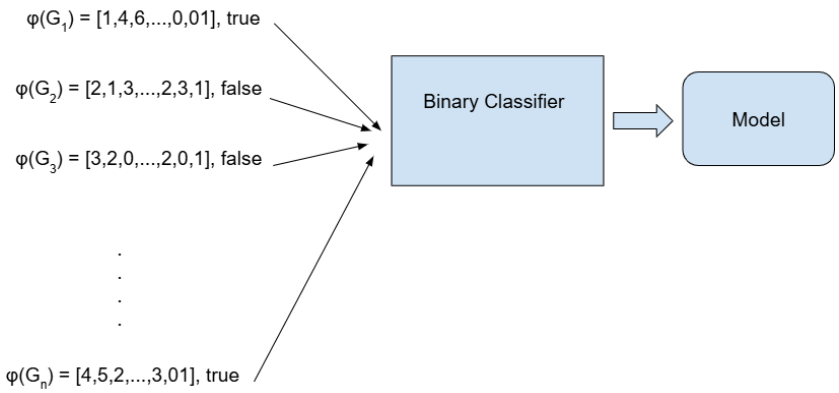
Ως μετρική απόστασης χρησιμοποιείται η **απόσταση Wasserstein** (Wasserstein Distance), η οποία ενσωματώνεται στον αλγόριθμο Wasserstein K-Means. Παρά την υψηλή υπολογιστική της πολυπλοκότητα, η απόσταση Wasserstein παρέχει υψηλή διαχωριστική ικανότητα, αναγνωρίζοντας λεπτές διαφορές στις κατανομές που παραλείπονται από συμβατικές μετρικές.



Σχήμα 4.2: *Αντιστοίχιση δέντρων ανάπτυξης σε συστάδες (clusters) με χρήση k-Means [3]*

4.2.5 Δυαδική Ταξινόμηση για Ανίχνευση Ψευδών Ειδήσεων

Όσον αφορά το δεύτερο βήμα, μετά τη μετατροπή των δέντρων διάδοσης σε διανύσματα χαρακτηριστικών, εκπαιδεύουμε γραμμικά και μη γραμμικά μοντέλα (*classifiers*) για τη διάκριση μεταξύ αληθών και ψευδών ειδήσεων. Πρακτικά, τα μοντέλα λαμβάνουν ως είσοδο τα διανύσματα χαρακτηριστικών κάθε δέντρου διάδοσης, σε συνδυασμό με την ετικέτα κάθε στοιχείου του συνόλου δεδομένων (Αληθής/Ψευδής), και παράγουν έναν ταξινομητή ικανό να προβλέπει την εγκυρότητα οποιασδήποτε νέας είδησης, για την οποία υπάρχει διαθέσιμο το γράφημα διάδοσής της. Στο Σχήμα 4.3 παρουσιάζεται η διαδικασία εκπαίδευσης για τη δυαδική ταξινόμηση, η οποία αποτελεί τον πυρήνα της ανίχνευσης ψευδών ειδήσεων.



Σχήμα 4.3: *Διαδικασία εκπαίδευσης δυαδικού ταξινομητή για την ανίχνευση ψευδών ειδήσεων.*

4.3 Προτεινόμενες Προσεγγίσεις Μοντελοποίησης

Παρουσιάζονται οι τρεις διαφορετικές γραφοθεωρητικές μέθοδοι που αναπτύχθηκαν για την ανίχνευση ψευδών ειδήσεων.Η πρώτη είναι η βασική μέθοδος στην οποία χρησιμοποιούμε βασικές μετρικές στα γραφήματα διάδοσης για να αντιστοιχήσουμε τα γραφήματα σε διανύσματα χαρακτηριστικών. Η δεύτερη είναι ο απλός Πυρήνας Γραφημάτων Weisfeiler-Lehman που παρουσιάσαμε στο κεφάλαιο 3.2.4. Η τρίτη προσέγγιση είναι ο Γενικευμένος

Πυρήνας Γραφημάτων Weisfeiler-Lehman που σαν σκοπό έχει να μειώσει την διαστατικότητα των διανυσμάτων χαρακτηριστικών που κατασκευάζονται από τον απλό πυρήνα γραφημάτων Weisfeiler-Lehman.

4.3.1 Βασική Προσέγγιση

Στην ενότητα αυτή παρουσιάζεται η διαδικασία μετατροπής των γραφημάτων διάδοσης σε διανυσμάτα χαρακτηριστικών με χρήση κάποιων χαρακτηριστικών των γραφημάτων.

Ακολουθία Βαθμών Κόμβων

Στην παρούσα υλοποίηση υιοθετήθηκε μια ελαφριά, αμιγώς δομική αναπαράσταση κάθε γραφήματος διάδοσης, βασισμένη στην **ακολουθία βαθμών** (*degree sequence*). Η διαδικασία που ακολουθήθηκε έχει τα εξής βήματα:

1. **Μετατροπή δέντρων διάδοσης σε απλούς γράφους.** Κάθε κατευθυνόμενο γράφημα διάδοσης μετατρέπεται σε μη κατευθυνόμενο γράφημα ώστε να αποτυπώνεται μόνον η τοπολογία και όχι η φορά των ακμών.

2. **Εξαγωγή βαθμών κόμβων.** Για κάθε γράφημα $G = (V, E)$ υπολογίζονται οι βαθμοί $d(v)$ όλων των κόμβων $v \in V$ και ταξινομούνται σε φθίνουσα σειρά:

$$\mathbf{d} = [d_1 \geq d_2 \geq \ldots \geq d_{|V|}].$$

3. **Δημιουργία διανύσματος χαρακτηριστικών.**
Όσον αφορά την δημιουργία του διανύσματος που προκύπτει από την ακολουθία βαθμών έχουμε δύο προσεγγίσεις:

   - *Στατική περικοπή:* κρατούμε τους πρώτους $n_{\max} = 50$ βαθμούς και συμπληρώνουμε με μηδενικά όπου χρειάζεται.
   - *Δυναμική σάρωση:* παράγουμε πολλαπλά διανύσματα, μεταβάλλοντας το μήκος n από 1 έως 50 μελετώντας την ακρίβεια του ταξινομητή ανάλογα με το τελικό διάνυσμα που προκύπτει.

   Για κανονικοποίηση, κάθε βαθμός d_i διαιρέθηκε με $(|V| - 1)$—το μέγιστο θεωρητικό βαθμό σε δέντρο—ώστε $d_i \in [0, 1]$.

4. **Τυποποίηση και εκπαίδευση ταξινομητών.** Οι πίνακες χαρακτηριστικών $\mathbf{X} \in \mathbb{R}^{m \times n}$ τυποποιήθηκαν με `StandardScaler` και χρησιμοποιήθηκαν στα μοντέλα της Λογιστική Παλινδρόμηση και των Ενισχυόμεων Δέντρων Απόφασης (`GradientBoostingClassifier`). Κάθε μοντέλο αξιολογήθηκε με δεκαπλάσια *διασταυρούμενη επικύρωση*.

Δομικά Μετρικά Διάδοσης

Στη δεύτερη προσέγγιση μετατροπής γραφημάτων σε διανύσματα χαρακτηριστικών χρησιμοποιήθηκαν **δομικές μετρικές διάδοσης** που ποσοτικοποιούν τον τρόπο εξάπλωσης μιας είδησης μέσα στο κοινωνικό δίκτυο. Για κάθε γράφο $G = (V, E)$ υπολογίστηκαν οι επτά

μετρικές του Πίνακα 4.1· η i-οστή μετρική αντιστοιχεί στην i-οστή συνιστώσα του διανύσματος

$$\mathbf{x}(G) = [x_1, x_2, x_3] \in \mathbb{R}^3.$$

Πίνακας 4.1: *Ορισμός δομικών μετρικών διάδοσης.*

x_1	Κανονικοποιημένο Μέγιστο Βάθος
x_2	Κανονικοποιημένο Μέγιστο Πλάτος
x_3	Δομική Ικότητα (structural virality [29])

Για τα δύο πρώτα μεγέθη έχουμε · το μέγιστο βάθος αντιστοιχεί στη μεγαλύτερη απόσταση από τον κόμβο-ρίζα έως οποιονδήποτε άλλο κόμβο του δέντρου, ενώ το μέγιστο πλάτος ορίζεται ως ο μεγαλύτερος αριθμός κόμβων που βρίσκονται στην ίδια απόσταση από τη ρίζα. Για κανονικοποίηση, τα μεγέθη αυτά διαιρούνται με τα αντίστοιχα καθολικά μέγιστα: το μέγιστο βάθος του δέντρου διαιρείται με το μέγιστο βάθος που παρατηρείται σε όλα τα γραφήματα του συνόλου δεδομένων, και το μέγιστο πλάτος του δέντρου διαιρείται με το αντίστοιχο μέγιστο πλάτος όλων των γραφημάτων. Αυτή η διαδικασία επιτρέπει τη σύγκριση των μεγεθών σε μια αναλογική κλίμακα

4.3.2 Προσέγγιση με Απλό Πυρήνα Γραφημάτων Weisfeiler-Lehman

Η δεύτερη γραφοθεωρητική μέθοδος βασίζεται στον **απλό πυρήνα Weisfeiler–Lehman (WL)**, ο οποίος εφαρμόζει επαναληπτική αναρίθμηση ετικετών ώστε να παράγει ένα κανονικοποιημένο σύνολο ετικετών για κάθε γράφημα, όπου κάθε ετικέτα αντιστοιχεί σε ένα μοναδικό Δέντρο-Ξεδίπλωσης (unfolding-tree) σύμφωνα με την διαδικασία Weisfeiler-Lehman.

1. Αρχική τοποθέτηση επιγραφών στους κόμβους. Τα δέντρα διάδοσης φορτώνονται (`readDataSetFromFullPath`) και τίθενται επιγραφές στους κόμβους τους ανάλογα με την διαδικασία που έχουμε επιλέξει για την τοποθέτηση επιγραφών πάνω στους κόμβους των γραφημάτων. (κλάση nodeLabelingOfDataSet).

2. Αλγόριθμος επαναληπτικής αναρίθμησης. Με την κλάση `WLGraphListKernel` εφαρμόζονται h επαναλήψεις του αλγορίθμου όπου οι ετικέτες κάθε κόμβου σε κάθε γράφημα διάδοσης ανανεώνεται βάσει της ακόλουθης αναδρομικής σχέσης:

$$\ell_{t+1}(v) = \text{hash}\big(\ell_t(v),\ \text{sort}\{\ell_t(u) : (v,u) \in E\}\big), \quad t = 0, 1, \ldots.$$

Οι ακέραιες επιγραφές κάθε γραφήματος αποθηκεύονται σε μία δομή πίνακα, από την οποία εξάγεται το διάνυσμα χαρακτηριστικών μέσω της κατανομής συχνοτήτων κάθε επιγραφής.

3. Μετατροπή σε διάνυσμα χαρακτηριστικών. Για κάθε γράφημα παράγεται ιστόγραμμα συχνοτήτων ετικετών $\mathbf{x} \in \mathbb{R}^d$, όπου :

$$d = \max_{G \in \mathcal{G}} \left(\max_{u \in V(G)} l(u) \right)$$

.

4.3.3 Προσέγγιση με Γενικευμένο Πυρήνα Γραφημάτων *Weisfeiler-Lehman*

Η τρίτη και περισσότερο σύνθετη προσέγγιση τροποποιεί τον κλασικό Πυρήνα Γραφημάτων *Weisfeiler-Lehman (WL)* με μία **ιεραρχική ομαδοποίηση** των Δέντρων Ξεδίπλωσης (*unfolding trees*) μέσω της Διορθωτικής Απόστασης Δέντρων (*Tree Edit Distance, TED*). Το αποτέλεσμα είναι ένας πυρήνας που:

1. συμπεριλαμβάνει λεπτές δομικές ομοιότητες,
2. συμπιέζει δραστικά τη διάσταση των διανυσμάτων χαρακτηριστικών που προκύπτουν από τον απλό Πυρήνα Γραφημάτων Weisfeiler-Lehman,
3. είναι εύρωστος σε τοπικές παραλλαγές της διάδοσης.

Επισκόπηση Βημάτων

1. **Χαρακτηρισμός κόμβων γραφήματος** Τα δέντρα διάδοσης φορτώνονται και οι κόμβοι επισημαίνονται με ακέραιες επιγραφές.
2. **Παραγωγή και Κανονικοποίηση Δέντρων Ξεδίπλωσης** Για κάθε κόμβο v σε κάθε γράφημα G δημιουργούνται τα δέντρα ανάπτυξης του γραφήματος G για τον κόμβο u:

   $$T^{(1)}(G, v), \quad T^{(2)}(G, v)$$

   βάθους 1 και 2 αντίστοιχα. Κάθε T^i μετασχηματίζεται σε κανονική μορφή. Όλα τα ισομορφικά δέντρα μετασχηματίζονται στην ίδια αναπαράσταση μέσω αυτής της διαδικασίας.
3. **Διανυσματική περιγραφή υποδέντρων** Κάθε $T^{(d)}$ κωδικοποιείται σε διάνυσμα όπως παρουσιάστηκε στο Κεφάλαιο 3.2.6.
4. **Συσταδοποίηση υποδέντρων** Μέσω του αλγορίθμου Wasserstein K-Means πραγματοποιείται η συσταδοποίηση των διανυσματικών αναπαραστάσεων των $T^{(i)}$. Μετά από αυτή τη διαδικασία, κάθε διάνυσμα αντιστοιχίζεται σε μία συστάδα και μπορούμε έτσι να ομαδοποιήσουμε τα δέντρα ξεδίπλωσης σε συστάδες.
5. **Λεξικά Αντιστοίχισης (*Bucket Mapping*)** Για την αποφυγή διαρροής πληροφοριών από το σύνολο ελέγχου στο σύνολο εκπαίδευσης, καθώς και για τη συνεπή κωδικοποίηση των διανυσμάτων χαρακτηριστικών που προκύπτουν στο σύνολο ελέγχου, ορίζουμε δύο αμετάβλητα λεξικά:

   $$\mathcal{M}_1 : \{T^{(1)}\} \longrightarrow \{0, \ldots, k_1 - 1\}, \qquad \mathcal{M}_2 : \{T^{(2)}\} \longrightarrow \{0, \ldots, k_2 - 1\},$$

   όπου $\{T^{(d)}\}$ είναι το σύνολο όλων των μοναδικών δέντρων ξεδίπλωσης βάθους d που εντοπίστηκαν στο εκπαιδευτικό σύνολο και k_1, k_2 αποτελούν το πλήθος των συστάδων για την ομαδοποίηση των δέντρων ξεδίπλωσης ύψους 1 και 2 αντιστοίχως.

Λειτουργία:

(α') Κατά την εκπαίδευση, τα λεξικά $\mathcal{M}_1, \mathcal{M}_2$ γεμίζουν μόνο με τα υποδέντρα που προκύπτουν από τα γραφήματα του $\mathcal{T}_{\text{train}}$.

(β') Κατά τη διαδικασία υπολογισμού των διανυσμάτων χαρακτηριστικών για τα γραφήματα του $\mathcal{T}_{\text{test}}$, εάν εμφανιστεί ένα υποδέντρο $T^{(i)} \notin \text{dom}(\mathcal{M}_i)$, εφαρμόζουμε *Διορθωτική Απόσταση Δέντρων*:

$$\mathcal{M}_i(T^{(i)}) = \mathcal{M}_i(\arg \min_{S \in \text{dom}(\mathcal{M}_i)} \text{SDTED}(T^{(i)}, S)).$$

Σε περίπτωση πολλαπλών ελάχιστων γειτόνων, η βαρύτητα της ανάθεσης κατανέμεται ισομερώς μεταξύ τους.

Με τον τρόπο αυτό εξασφαλίζεται ότι:

- Η φάση εκπαίδευσης παραμένει ανεπηρέαστη από νέα δομικά μοτίβα του ελέγχου.
- Κάθε νέο υποδέντρο αντιστοιχεί σε κλάση που έχει ήδη εκπαιδευτεί, ή σε μία κοντινότερη σε αυτή με βάση τη διορθωτική απόσταση.
- Διατηρείται σταθερό το μέγεθος των διανυσμάτων χαρακτηριστικών (k_1 και k_2 διαστάσεις αντίστοιχα) για όλα τα παραγόμενα γράφηματα.

6. **Σύνθεση Τελικού Διανύσματος** Για κάθε γράφημα G το διάνυσμα χαρακτηριστικών που προκύπτει τελικά έχει τη μορφή:

$$\mathbf{x}(G) = \underbrace{\mathbf{h}_{\text{WL}}}_{\ell_{\max}} \parallel \underbrace{\mathbf{c}_1}_{k_1} \parallel \underbrace{\mathbf{c}_2}_{k_2},$$

όπου $\mathbf{h}_{\text{WL}}$ είναι το ιστογράφημα ετικετών WL και $\mathbf{c}_i[j]$ το πλήθος υποδέντρων που ανήκουν στην j-οστή συστάδα του $\mathcal{M}_i$.

4.4 Αρχιτεκτονική του Συστήματος

4.4.1 Ανάλυση - περιγραφή αρχιτεκτονικής

Στην ενότητα αυτή παρουσιάζεται η ανάλυση του συστήματος και ο χωρισμός του σε υποσυστήματα όσον αφορά την αρχιτεκτονική.

Διαχωρισμός υποσυστημάτων

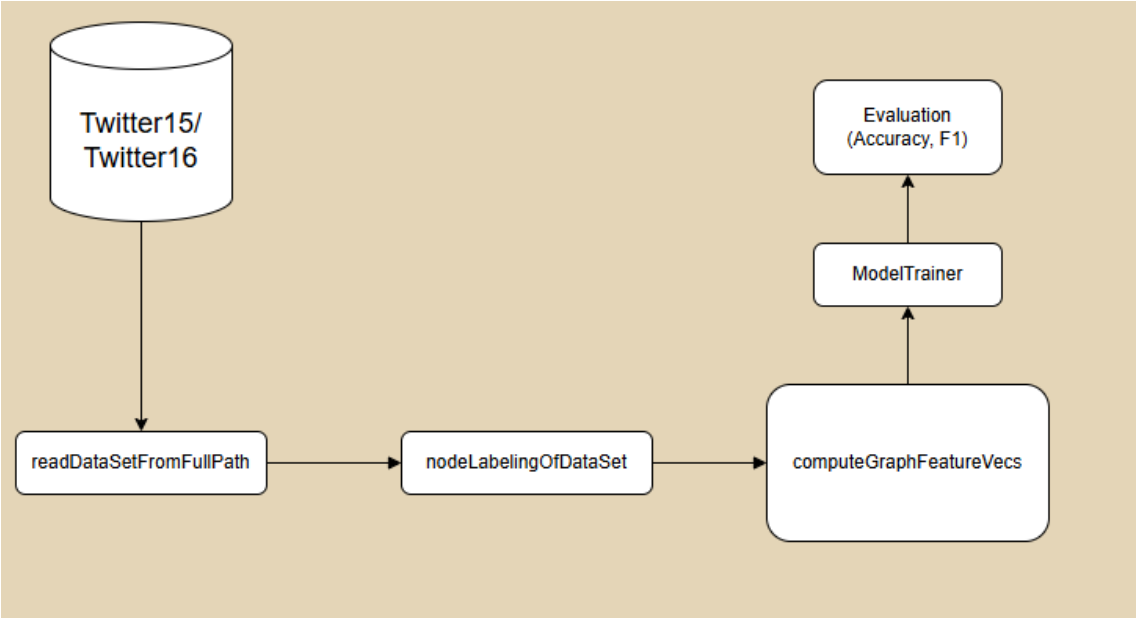
Το σύστημα αποτελείται από επιμέρους κλάσεις οι οποίες συνδυαζόμενες καθιστούν εφικτή την μελέτη της συσχέτισης μεταξύ της εγκυρότητας της διαδιδόμενες πληροφορίας και του μοτίβου που αυτή διαδίδεται. Το σύστημα αποτελείται από τα εξής υποσυστήματα:

- Υποσύστημα ανάγνωσης δεδομένων από το Σύνολο Δεδομένων Twitter-15/Twitter-16.

- Υποσύστημα τοποθέτησης ετικετών πάνω στους κόμβους των δέντρων διάδοσης της πληροφορίας.

- Υποσύστημα δημιουργίας των διανυσμάτων χαρακτηριστικών καθενός δέντρου διάδοσης σύμφωνα με καθεμία από τις τρεις διαφορετικές προσεγγίσεις που αναφέραμε στο Κεφάλαιο 4.3.

- Υποσύστημα υπολογισμού της ακρίβειας του μοντέλου.



Σχήμα 4.4: *Διαδικασία Πειραματικής Μελέτης.*

Στο Σχήμα 4.4 απεικονίζεται η σειρά με την οποία χρησιμοποιήσαμε τα διάφορα υποσυστήματα για την πειραματική μας μελέτη. Στην αρχή έχουμε τα Σύνολα Δεδομένων Twitter-15/Twitter-16, μία κλάση που τα διαβάζει και τα αποθηκεύει στην κατάλληλη προς επεξεργασία μορφή (class readDataSetFullPath), στη συνέχεια μία κλάση που θέτει επιγραφές πάνω στους κόμβους βάσει μιας μεθόδου, που εμείς ορίζουμε (class nodeLabelingClass) και έπειτα μία κλάση μετασχηματίζει κάθε δέντρο διάδοσης σε ένα διάνυσμα χαρακτηριστικών ανάλογα με την τεχνική που εφαρμόζουμε. Στο τελευταίο βήμα, μέσω της κλάσης ModelTrainer, χρησιμοποιούμε τα σύνολα εκπαίδευσης και ελέγχου (trainingSet, testSet) που υπολογίστηκαν από την προηγούμενη κλάση για να υπολογίσουμε την ακρίβεια του μοντέλου.

4.4.2 Περιγραφή υποσυστημάτων

Παρακάτω δίνεται λεπτομερής περιγραφή για καθένα από τα υποσυστήματα που αναφέραμε παραπάνω. Η περιγραφή αυτή γίνεται με βάση τα διαγράμματα ροής δεδομένων και με την σειρά που εικονίζονται στο Σχήμα 4.4.

Υποσύστημα Ανάγνωσης Δεδομένων

Το υποσύστημα αυτό (class readDataSetFullPath) λαμβάνει την πλήρη διαδρομή από το Σύνολο Δεδομένων Twitter 15 και τα αποθηκεύει σε μία λίστα. Η δομή του Twitter15 περιγράφεται στο Κεφάλαιο 6.2. Διαβάζει το αρχείο που περιλαμβάνει την κατάσταση εγκυρότητας καθεμίας δημοσίευσης, και δημιουργεί ένα λεξικό με την εξής μορφή:

$$\text{tweetIdVeracityMap} = \{tweetId_1 : verStat_1, tweetId_2 : verStat_2, \ldots, tweetId_n : verStat_n\}$$

όπου:

- $tweetId_i$: Αναγνωριστικός Κωδικός κάθε Δημοσίευσης
- $verStat_i$: : 'true' ή 'false'

Στη συνέχεια, για όλες εκείνες τις δημοσιεύσεις οι οποίες είναι αποθηκευμένες ως κλειδιά στο λεξικό της εγκυρότητας κάθε δημοσίευσης, κατασκευάζουμε το δέντρο διάδοσης,μέσω του αρχείου που υπάρχει στον φάκελο tree του Συνόλου Δεδομένων Twitter15 (στο αρχείο tweetId.txt).

Τέλος, δημιουργούμε μία λίστα, την dataSet η οποίο είναι της μορφής:

$$\text{dataSet} = [(tr_1, v_1, twId_1), (tr_2, v_2, twId_2), \ldots (tr_n, v_n, twId_n)]$$

όπου:

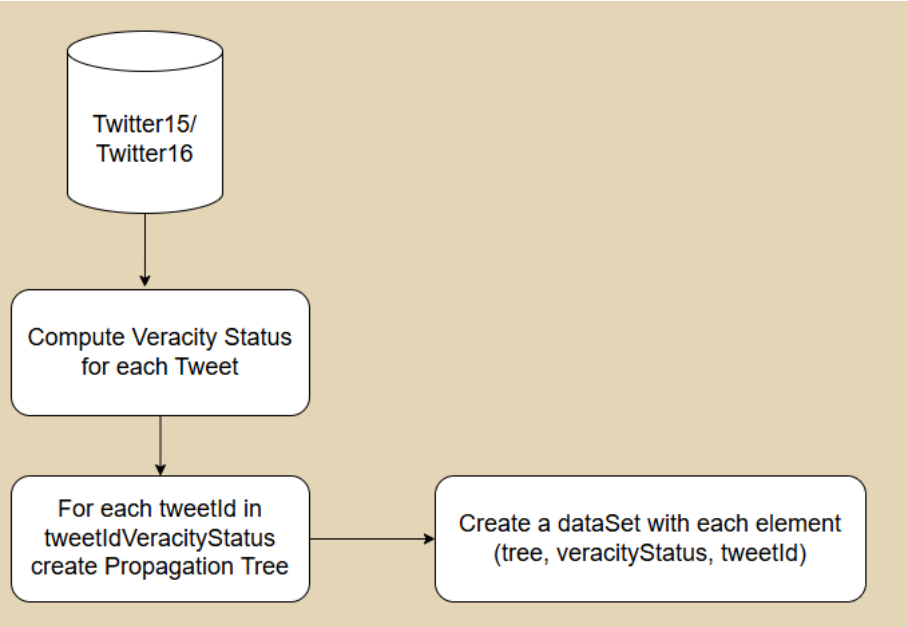
- tr_i: : Δέντρο διάδοσης πληροφορίας
- v_i: : 'true' ή 'false'
- $twId_i$: Αναγνωριστικός Κωδικός κάθε Δημοσίευσης

Υποσύστημα Τοποθέτησης Ετικετών στους Κόμβους

Το υποσύστημα αυτό (class nodeLabelingClass), λαμβάνει το Σύνολο Δεδομένων στην μορφή που παρουσιάσαμε στο προηγούμενο υποσύστημα (class readDataSetFullPath) και θέτει ετικέτες πάνω στους κόμβους ανάλογα με τη μέθοδο που επιλέγουμε. Παρόλο που έχουμε υλοποιήσει αρκετές διαδικασίες που το πραγματοποιούν αυτό, όλες μπορούν να διαχωριστούν στις ακόλουθες κατηγορίες:

- Τοποθέτηση ετικετών βάσει του Δέντρο Διάδοσης.
- Τοποθέτηση ετικετών βάσει επιπλέον πληροφοριών που εξάγονται από το Σύνολο Δεδομένων.

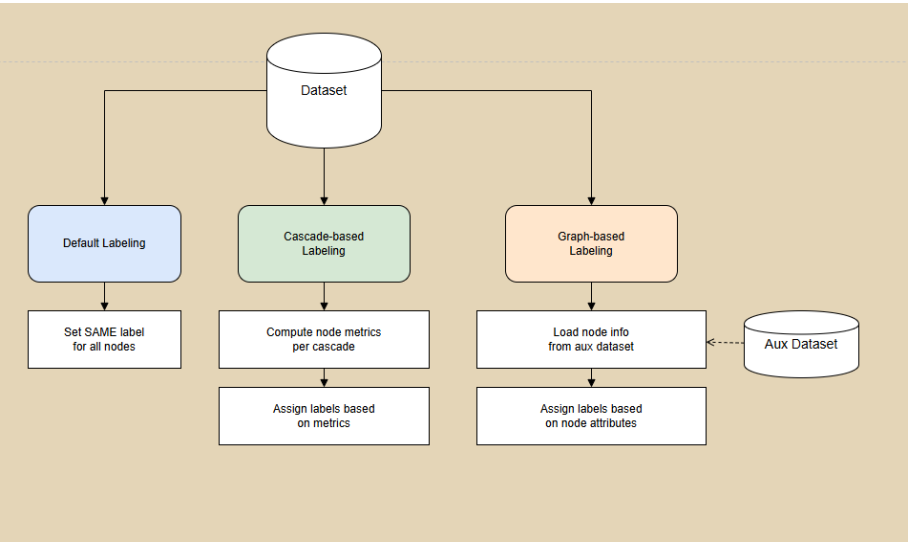
Όπως βλέπουμε και στο Σχήμα 4.6 , τοποθετούμε ετικέτες πάνω στους κόμβους των Δέντρων Διάδοσης είτε βασιζόμενοι στο ίδιο το δέντρο, είτε σε διάφορες πληροφορίες τις οποίες εξάγουμε από τα συμφραζόμενα από το Σύνολο Δεδομένων.



Σχήμα 4.5: *Διαδικασία Δημιουργίας Συνόλου Δεδομένων.*

Παραδείγματα τοποθέτησης ετικετών που να βασίζονται στο Δέντρο Διάδοσης μπορεί να απότελούν:

- Ο λογάριθμος του εξωτερικού βαθμού κάθε κόμβου.(out_degree log-bin).
- Η συστάδα στην οποία ανήκει ο εξωτερικός βαθμός του κάθε κόμβου, για Κ ορισμένες συστάδες (kMeans in 1-d)
- Κοινή ετικέτα για όλους τους κόμβους (σε περίπτωση απουσίας δεδομένων χρηστών.)



Σχήμα 4.6: *Διαδιακασία υποσυστήματος τοποθέτησης επιγραφών.*

Υποσύστημα Μετασχηματισμού Δέντρων Διάδοσης σε Διανύσματα Χαρακτηριστικών

Το παρόν υποσύστημα υλοποιεί τη μετατροπή δέντρων διάδοσης σε διανυσματικές αναπαραστάσεις μέσω τριών διαφορετικών προσεγγίσεων. Στην *βασική μέθοδο*, κάθε δέντρο αντιστοιχείται σε ένα n-διάστατο διάνυσμα χαρακτηριστικών $\mathbf{v} \in \mathbb{R}^n$, του οποίου οι συντεταγμένες προκύπτουν από τον υπολογισμό συγκεκριμένων μετρικών του γραφήματος.

Στην περίπτωση του Απλού Πυρήνα Γραφημάτων Weisfeiler-Lehman (Σχήμα 4.7) η μέθοδος εκτελεί h επαναλήψεις της ακόλουθης διαδικασίας:

1. Υπολογισμός επιγραφών πολυσυνόλου (multiset labels) για όλους τους κόμβους
2. Εφαρμογή αλγορίθμου συμπίεσης επιγραφών (label compression)
3. Ανανεωση των επιγραφών όλων των κόμβων σε όλα τα γραφήματα με βάση τη νέα αρίθμηση

Το τελικό χαρακτηριστικό διάνυσμα $\mathbf{v}_G \in \mathbb{R}^d$ για κάθε δέντρο G του συνόλου εκπαίδευσης προκύπτει από την ακόλουθη σχέση:

$$\mathbf{v}_G[i] = \text{συχνότητα εμφάνισης της ετικέτας } \ell_i \tag{4.1}$$

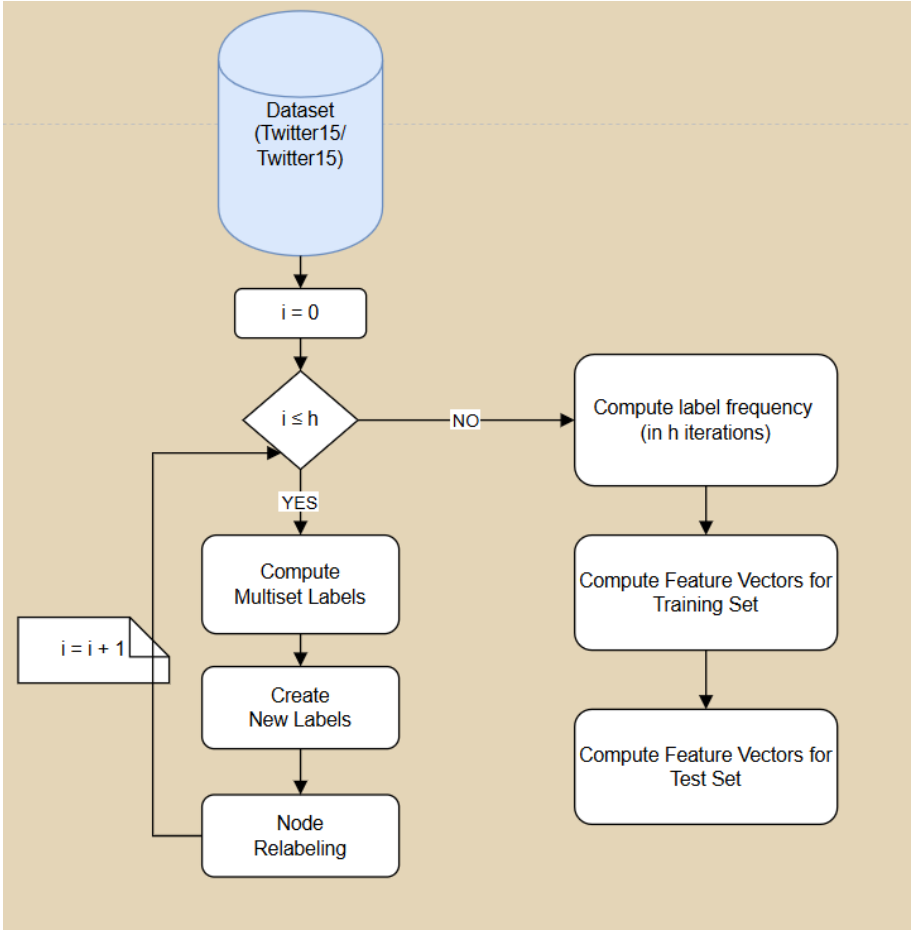
Αυτή η αναπαράσταση μπορεί να ερμηνευτεί ως ένα μοντέλο ¨σάκου μοτίβων¨ (bag-of-words) [43], όπου:

- Κάθε μοτίβο αντιστοιχεί σε ένα υποδέντρο ανάπτυξης βάθους $i \leq h$.
- Ο πίνακας χαρακτηριστικών έχει σχεδιαστεί έτσι ώστε, αν δύο γράφοι είναι ισομορφικοί, διαφέρουν μόνο στο πώς έχουν ονομαστεί ή ταξινομηθεί οι κόμβοι τους και αντιστοιχίζονται ακριβώς στο ίδιο διάνυσμα χαρακτηριστικών. Με άλλα λόγια, η αναπαράσταση είναι «τυφλή» στις μετονομασίες των επιγραφών των κόμβων και εστιάζει μόνο στη δομή του γραφήματος[2].

Η προσέγγιση αυτή παρουσιάζει αναλογίες με την κλασική αναπαράσταση κειμένων μέσω bag-of-words [43].

Επιπλέον, κατά τη σύνθεση των διανυσμάτων χαρακτηριστικών για το σύνολο ελέγχου, εφαρμόζεται η διορθωτική μετρική απόστασης δέντρων (*tree edit distance*). Συγκεκριμένα, υπολογίζονται οι αποστάσεις μεταξύ των δέντρων ανάπτυξης του συνόλου ελέγχου και εκείνων του συνόλου εκπαίδευσης. Η διαδικασία αυτή εξασφαλίζει:

- **Συνεκτικότητα αναπαράστασης**: Διατήρηση της ίδιας μετρικής βάσης και για τα δύο σύνολα
- **Γενίκευση της μεθόδου**: Δυνατότητα επέκτασης σε νέα, μη παρατηρηθέντα αντικείμενα ανίχνευσης



Σχήμα 4.7: *Διαδιακασία υπολογισμού διανυσμάτων χαρακτηριστικών με χρήση πυρήνα Weisfeiler-Lehman.*

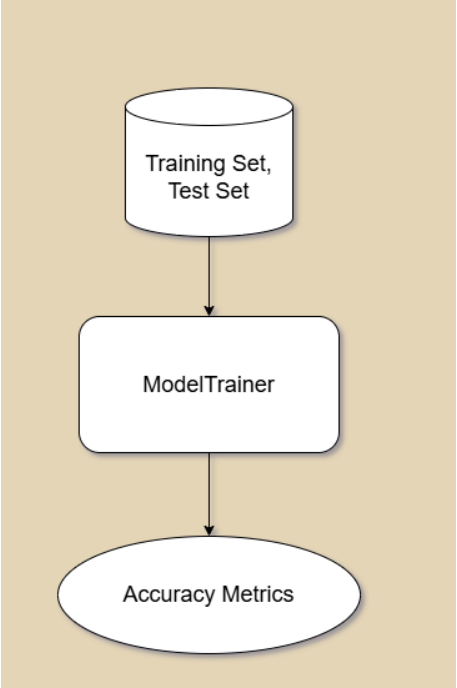
Υποσύστημα *ModelTrainer*

Το υποσύστημα **ModelTrainer** λαμβάνει ως είσοδο τα ζεύγη $(\mathbf{x}, y)$ του συνόλου εκπαίδευσης και ελέγχου (*training set*, *test set*), όπου $\mathbf{x} \in \mathbb{R}^d$ είναι το διάνυσμα χαρακτηριστικών (ήδη παραχθέν, π. χ. από τον αλγόριθμο Weisfeiler–Lehman[1]) και $y \in \{0, 1\}$ η ετικέτα γνησιότητας. Η ροή επεξεργασίας αποτυπώνεται στο Σχήμα 4.8.

- **Κανονικοποίηση:** Τα χαρακτηριστικά κλιμακώνονται με `StandardScaler` ώστε κάθε διάσταση να έχει μηδενικό μέσο και μοναδιαία διακύμανση (`prepare_data`).
- **Εκπαίδευση μοντέλων:** Ως μοντέλα εκπαίδευσης χρησιμοποιούνται:
  1. *Logistic Regression* (γραμμικός ταξινομητής) και
  2. *Gradient Boosting Classifier* (μη γραμμικός ταξινομητής) με πλήθος εκτιμητών `n_estimators` = 100, ρυθμός μάθησης (`learning_rate`=0.1, μέγιστο βάθος `max_depth` = 3 (`train_models`).
- **Αξιολόγηση:** Για κάθε μοντέλο υπολογίζονται *ακρίβεια* (*accuracy*) και αναλυτική αναφορά ταξινόμησης (*precision*, *recall*, F_1) πάνω στο σύνολο ελέγχου (`evaluate_models`).

[1] Η τιμή του βάθους h επιλέγεται στο στάδιο εξαγωγής χαρακτηριστικών και *δεν* αλλάζει μέσα στον `ModelTrainer`.

Οι υπερπαράμετροι των ταξινομητών ορίζονται εντός της κλάσης, επιτρέποντας τη βελτιστοποίησή τους μέσω τεχνικών όπως η `grid search` ή η `random search`.. Η κλάση επανεγγράφει στην έξοδο της κονσόλας τα αποτελέσματα, όμως μπορεί εύκολα να επεκταθεί ώστε να επιστρέφει τις μετρικές σε μορφή πινάκων για αυτοματοποιημένη σύγκριση πειραμάτων.



Σχήμα 4.8: *Υποσύστημα Υπολογισμού Ακριβείας Μοντέλου.*

4.5 Σύνοψη

Στο παρόν κεφάλαιο παρουσιάστηκε η λεπτομερής υλοποίηση του συστήματος ανίχνευσης ψευδών ειδήσεων βασισμένης στη γραφοθεωρητική ανάλυση των δέντρων διάδοσης πληροφορίας. Αρχικά αναπτύχθηκε το θεωρητικό πλαίσιο μοντελοποίησης του προβλήματος, το οποίο στηρίζεται στην αντιστοίχιση κάθε δέντρου διάδοσης σε ένα διανυσματικό χώρο χαρακτηριστικών, και στη συνέχεια αξιοποιήθηκε για την εκπαίδευση δυαδικών ταξινομητών.

Παρουσιάστηκαν τρεις εναλλακτικές προσεγγίσεις για την εξαγωγή των διανυσματικών αναπαραστάσεων των δέντρων διάδοσης:

1. **Βασική προσέγγιση**, που στηρίζεται σε δομικές μετρικές και βαθμούς κόμβων.
2. **Προσέγγιση με Απλό Πυρήνα Weisfeiler-Lehman**, η οποία αξιοποιεί την επαναληπτική αναρίθμηση των επιγραφών των κόμβων και την εξαγωγή ιστογραμμάτων επιγραφών.
3. **Προσέγγιση με Γενικευμένο Πυρήνα Weisfeiler-Lehman**, η οποία συνδυάζει τη διαδικασία του απλού πυρήνα με ιεραρχική συσταδοποίηση υποδέντρων βάσει της διορθωτικής απόστασης δέντρων (*Tree Edit Distance*) και της μεθόδου συσταδοποίησης *Wasserstein k-means*.

Συστημική Αρχιτεκτονική: Το σύστημα σχεδιάστηκε με βάση την αρχή του διαχωρισμού των ευαισθησιών, αποτελούμενο από τα ακόλουθα διακριτά υποσυστήματα:

- Εισαγωγής και προεπεξεργασίας δεδομένων
- Ετικετοποίησης και αναγνώρισης κόμβων
- Μετασχηματισμού δομών δέντρων σε διανυσματικές αναπαραστάσεις
- Εκπαίδευσης και αξιολόγησης ταξινομητών

Κεφάλαιο 5

Λεπτομέρειες Υλοποίησης

5.1 Εισαγωγή

Στην ενότητα αυτή αναλύονται όλες οι τεχνικές λεπτομέρειες της υλοποίησης της γραφοθεωρητικής μεθόδου πρόγνωσης. Παρουσιάζονται οι δομές δεδομένων, οι βιβλιοθήκες που χρησιμοποιήθηκαν, οι προκλήσεις που αναδείχθηκαν κατά τη διάρκεια της ανάπτυξης και η πλήρης περιγραφή των κλάσεων που υλοποιήθηκαν.

5.2 Δομές Δεδομένων

Στην παρούσα ενότητα περιγράφονται οι βασικές δομές δεδομένων που υλοποιούνται όπως αυτές προκύπτουν από τα αρχεία πηγαίου κώδικα.

5.2.1 Κατευθυνόμενα δέντρα αναπαράστασης γραφημάτων διάδοσης

- **Δομή:** Προσανατολισμένος γράφος $T = (V, E)$ με ρίζα, που υλοποιείται ως `networkx.DiGraph`. Κάθε κόμβος $v \in V$ φέρει ετικέτα `label` (`type int`).

- **Βασικές λειτουργίες:**

  1. *Κωδικοποίηση δέντρων διάδοσης* τα δέντρα διάδοσης που είναι αποθηκευμένα σε πίνακες μετά την ανάγνωση του συνόλου δεδομένων Τα αρχεία των Twitter-15/Twitter-16 είναι σε αρχεία .txt. Στη συνέχεια η κλάση readDataSetFromFullPath αποθηκεύει το σύνολο δεδομένων σε πίνακες Pandas. Μέσω των πινάκων αυτών δημιουργούμε τα δέντρα διάδοσης. Τα δέντρα διάδοσης είναι τύπου nx.Graph.
  2. *Υπολογισμός Δέντρων Ανάπτυξης* (*unfolding trees*)- επαναφορά των κωδικοποιημένων υποδέντρων στο πλήρες τους σχήμα για επόμενης φάσης επεξεργασία.

5.2.2 Διανυσματική κωδικοποίηση και ομαδοποίηση δέντρων

- **Διανυσματατική μορφή Δέντρων Ξεδίπλωσης**

$$\mathbf{x}_T = \underbrace{\mathbf{e}_{\text{root}}}_{\text{one-hot}} \,\|\, \underbrace{\mathbf{c}_{\text{subtrees}}}_{\text{κατανομή υποδέντρων}} \,\|\, \underbrace{(2d - \deg(\text{root})),}_{\text{συνάρτηση βαθμού}}$$

όπου $\|$ δηλώνει σύμπτυξη, d το μέγιστο επιτρεπτό πλήθος παιδιών και $\mathbf{e}_{\text{root}} \in \mathbb{R}^L$ διάνυσμα μονού σημαντικού (one-hot vector) που κωδικοποιεί την επιγραφή της ρίζας. Είναι τύπου list ώστε να υπολογίζονται τα επιμέρους στοιχεία και να συνενώνονται όλα μαζί στο τελικό διάνυσμα γρήγορα και αποδοτικά.

- **Λεξικογραφική κωδικοποίηση** *(string-encoding)* κάθε δέντρου κωδικοποιημένο σε συμβολοσειρά, που αναπαρίσταται ως κλειδί στην διανυσματική αναπαράσταση του `dict⟨str, np.ndarray⟩` για εύρεση της συστάδας στη οποία ανήκει το κάθε δέντρο ανάπτυξης μετά την εφαρμογή του αλγορίθμου Wasserstein k-Means.

- **Πίνακες κόστους**

  1. $M_r \in \mathbb{R}^{(L+1)\times(L+1)}$ για πίνακα κόστους μεταξύ των επιγραφών τύπου numpy.ndarray
  2. $M_c \in \mathbb{R}^{(m+1)\times(m+1)}$ για κόστη επεξεργασίας υποδέντρων, όπου m ο αριθμός μη-ισομορφικών υποδέντρων στα δέντρα ξεδίπλωσης ύψους 2 (συμμετρικός, μη αρνητικός) τύπου numpy.ndarray

- **Ομαδοποίηση Wasserstein k-Means** Εφαρμόζεται σε λίστα N διανυσμάτων $\{\mathbf{x}_{T_i}\}_{i=1}^{N}$ με χρήση των παραπάνω πινάκων κόστους (M_r, M_c), παράγοντας:

  – `dict⟨int, list[np.ndarray]⟩` με τα κεντροειδή σύνολα.

  – `dict⟨str, int⟩` που αντιστοιχεί κάθε κωδικοποιημένο σε συμβολοσειρά δέντρο στην ανάλογη σύστάδα (`treeEncodedToClusterMap`).

5.2.3 Σύνολα Εκπαίδευσης – Ελέγχου

Οι δομές `trainingSetDataSet` και `testSetDataSet` είναι λίστες πλειάδων

$$(\mathbf{x}_T,\ y)$$

όπου $\mathbf{x}_T$ το διάνυσμα χαρακτηριστικών τύπου np.ndarray, $y \in \{0, 1\}$ η ετικέτα εγκυρότητας. Η κλάση `ModelTrainer` αναλαμβάνει την κανονικοποίηση μέσω `StandardScaler` και εκπαίδευση δύο ταξινομητών (`LogisticRegression`, `GradientBoostingClassifier`), με απευθείας επεξεργασία των δεδομένων χωρίς δημιουργία αντιγράφων.

5.3 Βιβλιοθήκες & Εργαλεία Λογισμικού

5.3.1 Βιβλιοθήκες

Για την υλοποίηση της προτεινόμενης μεθόδου χρησιμοποιήθηκε η γλώσσα **Python 3.10** . Τα βασικά πακέτα που αξιοποιήθηκαν είναι τα ακόλουθα:

- **Διαχείριση & ανάλυση γραφημάτων**: `NetworkX`.
- **Αριθμητικές/γραμμικές πράξεις**: `NumPy`, `SciPy`.
- **Μηχανική μάθηση**: `Scikit-learn`.

- **Γραφικά & απεικόνιση**: `Matplotlib, Pylab`.
- **Βέλτιστη μεταφορά μάζας (Wasserstein/EMD)**: `POT - Python Optimal Transport`.

5.3.2 Εκδόσεις & Επιμέρους Ρυθμίσεις

Στον Πίνακα 5.1 συνοψίζονται οι ακριβείς εκδόσεις καθώς και ειδικές ρυθμίσεις που ενεργοποιήθηκαν για αποδοτικότερη εκτέλεση.

Πίνακας 5.1: *Κύριες βιβλιοθήκες και ρυθμίσεις λογισμικού.*

Βιβλιοθήκη / Εργαλείο	**Έκδοση & Σχόλια**
`networkx`	3.3 — χρήση *graph_views* για ελαχιστοποίηση μνήμης
`numpy`	1.26
`scipy`	1.13 — υπολογισμοί Wasserstein
`scikit-learn`	1.4 — *LogisticRegression*, *GradientBoostingClassifier*
`POT`	0.9.3 — ταχύς υπολογισμός Earth Mover's Distance (EMD)
`joblib`	1.4 — παράλληλη εκτέλεση (`n_jobs = 8`)

5.4 Προκλήσεις Ανάπτυξης και Τεχνικά Ζητήματα

Κατά την υλοποίηση του συστήματος αναδύθηκαν τρία κεντρικά ζητήματα πρακτικής φύσεως. Στη συνέχεια περιγράφουμε *πώς* και *γιατί* προέκυψε κάθε πρόκληση και συνοπτικά τη στρατηγική που επιλέχθηκε για την αντιμετώπισή της.

5.4.1 Διαχείριση Μεγάλου Όγκου Δεδομένων

Τα σύνολα δεδομένων *Twitter15/16* περιλαμβάνουν περίπου χίλια δέντρα διάδοσης για κάθε ένα από τα δύο σύνολα. Ο συνολικός αριθμός κόμβων σε κάθε σύνολο υπερβαίνει σημαντικά τις διαθέσιμες μνημονικές δυνατότητες ενός τυπικού φορητού υπολογιστή, καθώς η απαίτηση σε μνήμη υπερβαίνει τα 16 GB.

Μέθοδος Διαχείρισης. Για την αντιμετώπιση αυτής της πρόκλησης, υιοθετήθηκε μια *στρατηγική επεξεργασίας σε ροή παρτίδων* (*mini-batch streaming*):

- Τα δέντρα φορτώνονται και επεξεργάζονται τμηματικά
- Μετατρέπονται σε ενδιάμεση αναπαράσταση χρησιμοποιώντας τη βιβλιοθήκη `NetworkX`
- Αποθηκεύονται προσωρινά σε μεταβλητές του κύριου προγράμματος (main.py)

Πλεονεκτήματα της Προσέγγισης:

1. Σημαντική μείωση της απαίτησης σε μνήμη (memory footprint)
2. Δυνατότητα τμηματικής κανονικοποίησης και κλιμάκωσης των δεδομένων
3. Αποτελεσματική διαχείριση πόρων - οι παρτίδες διαγράφονται μετά την εξαγωγή των απαραίτητων χαρακτηριστικών

Μέσω αυτής της μεθοδολογίας, η μέγιστη ταυτόχρονη απαίτηση σε μνήμη RAM περιορίστηκε σε λιγότερο από 3 GB, επιτρέποντας την εκτέλεση του πειράματος σε τυπικό υλικό.

5.4.2 Αντιμετώπιση Ισομορφισμών Γραφημάτων

Η σύγκριση υποδέντρων που υπάρχουν ως μοτίβα στα δέντρα διάδοσης βασίζεται στην ταξινόμησή τους σε κλάσεις ισομορφισμού. Δύο δέντρα που διαφέρουν μόνο ως προς την αρίθμηση των κόμβων τους οφείλουν να παράγουν *ταυτόσημα* διανύσματα χαρακτηριστικών. Η απουσία ειδικής επεξεργασίας οδηγεί σε εκρηκτική αύξηση του πλήθους των ετικετών και στη δημιουργία αραιών διανυσμάτων χαρακτηριστικών.

Μεθοδολογία. Για την αντιμετώπιση αυτού του προβλήματος, υλοποιήθηκε η κλάση `TreeIsomorphismTransformer`, η οποία εκτελεί τους ακόλουθους μετασχηματισμούς:

- Εφαρμογή *κανονικής επισήμανσης με BFS* (canonical BFS labelling)
- Ταξινόμηση κόμβων με βάση:
  1. Τον εσωτερικό βαθμό εισόδου (ρίζα: in-degree = 0)
  2. Λεξικογραφική διάταξη των μονοπατιών προς τη ρίζα

Πλεονεκτήματα:

1. **Αποδοτικός έλεγχος ισομορφισμού**: Ο αλγόριθμος επιτυγχάνει βέλτιστη πολυπλοκότητα $O(|V|)$ για τον έλεγχο ισομορφισμού, όπου $|V|$ ο αριθμός κορυφών.
2. **Βελτιστοποίηση χώρου**: Οι επόμενες διαδικασίες (όπως η ετικετοποίηση Weisfeiler-Lehman) επωφελούνται από τη συμπιεσμένη αναπαράσταση, με αποτέλεσμα σημαντική βελτίωση της υπολογιστικής απόδοσης.

5.4.3 Υπολογιστική Πολυπλοκότητα Πυρήνων Γραφημάτων

Ο απλός πυρήνας γραφημάτων Weisfeiler–Lehman παρουσιάζει υπολογιστική πολυπλοκότητα $O(h|E|)$ ανά ζεύγος γραφημάτων, όπου h είναι ο αριθμός επαναλήψεων του αλγορίθμου και $|E|$ το πλήθος των ακμών. Ωστόσο, η εφαρμογή του σε μεγάλα σύνολα δεδομένων, τα οποία περιέχουν χιλιάδες δέντρα, καθίσταται υπολογιστικά απαγορευτική, καθώς απαιτείται η εκτέλεση εκατομμυρίων υπολογισμών ομοιότητας μεταξύ ζευγών γραφημάτων.

Αποδοτική Υπολογιστική Προσέγγιση. Για την αντιμετώπιση του παραπάνω προβλήματος, αντί της πλήρους κατασκευής του πίνακα πυρήνα, υιοθετείται μια αποδοτικότερη στρατηγική: τα διανύσματα χαρακτηριστικών (feature vectors) κάθε δέντρου εξάγονται απευθείας, μέσω μίας μόνο διέλευσης του δέντρου, και αποθηκεύονται σε συμπιεσμένη μορφή (*hash map*, για αποδοτική αντιστοίχιση ετικέτας → πλήθος εμφανίσεων).

Η ομοιότητα μεταξύ δύο δέντρων, με διανύσματα χαρακτηριστικών $\mathbf{x}$ και $\mathbf{z}$, υπολογίζεται ως το εσωτερικό γινόμενο $\langle \mathbf{x}, \mathbf{z} \rangle$, με πολυπλοκότητα $O(\min\{\|\mathbf{x}\|_0, \|\mathbf{z}\|_0\})$, όπου $\|\cdot\|_0$ δηλώνει το πλήθος των μη μηδενικών συνιστωσών.

Πρακτικά Αποτελέσματα: Η προτεινόμενη μεθοδολογία καθιστά εφικτή την εκπαίδευση αλγορίθμων όπως το Gradient Boosting σε σύνολα δεδομένων μεγέθους έως 5.000 δέντρων, με χρόνο εκτέλεσης της τάξης των λίγων λεπτών σε τυπικό υπολογιστικό σύστημα (CPU), χωρίς να απαιτείται επιτάχυνση μέσω GPU.

Για να επιταχυνθεί ο υπολογισμός των διανυσμάτων χαρακτηριστικών για το σύνολο ελέγχου, εφαρμόζεται μια μέθοδος προσέγγισης που εντοπίζει τα πλησιέστερα δέντρα του συνόλου εκπαίδευσης. Η διαδικασία έχει ως εξής:

1. **Ομαδοποίηση Δέντρων Εκπαίδευσης:** Αρχικά, όλα τα δέντρα ανάπτυξης που ανήκουν στο **σύνολο εκπαίδευσης** ομαδοποιούνται με βάση τον αριθμό των κόμβων τους. Δημιουργούνται δηλαδή ομάδες δέντρων με ίδιο πλήθος κόμβων.

2. **Επεξεργασία Νέου Δέντρου Ελέγχου:** Για κάθε δέντρο ανάπτυξης T από το **σύνολο ελέγχου** που έχει n κόμβους και δεν έχει εμφανιστεί ξανά, ακολουθούνται τα παρακάτω υπο-βήματα:

   (α′) **Πρώτος Υπολογισμός Απόστασης:** Υπολογίζεται η «διορθωτική απόστασης δέντρου» του T και όλων των δέντρων της ομάδας εκπαίδευσης που έχουν **ακριβώς** n **κόμβους**(tree edit distance), την οποία ονομάζουμε δ.

   (β′) **Διευρυμένη Αναζήτηση:** Στη συνέχεια, η αναζήτηση επεκτείνεται. Εξετάζονται όλα τα δέντρα του συνόλου εκπαίδευσης των οποίων ο αριθμός των κόμβων κυμαίνεται στο εύρος $[n - \delta, n + \delta]$. Από αυτή τη μεγαλύτερη ομάδα, εντοπίζονται τα δέντρα που έχουν τη μικρότερη απόσταση από το T.

3. **Δημιουργία Διανύσματος Χαρακτηριστικών:** Αν από την τελική αναζήτηση προκύψουν k δέντρα που ισαπέχουν από το T με την ελάχιστη απόσταση, τότε η μονάδα **κατανέμεται ισοβαρώς** σε αυτά. Δηλαδή, οι k αντίστοιχες θέσεις στο διάνυσμα χαρακτηριστικών του T λαμβάνουν η καθεμία την τιμή $\frac{1}{k}$.

5.5 Περιγραφή Υλοποιημένων Κλάσεων

Παρουσιάζονται οι κύριες κλάσεις του συστήματος, οι λειτουργίες που επιτελούν και η μεταξύ τους αλληλεπίδραση.

5.5.1 Κλάση `WLGraph`

Η `WLGraph` υλοποιεί την *πρωτογενή* οντότητα-γράφο που τροφοδοτείται στους αλγορίθμους **Weisfeiler–Lehman (WL)**. Κάθε στιγμιότυπο περιέχει ένα `networkx.Graph` $G = (V, E)$, στον οποίο κάθε κόμβος $v \in V$ φέρει μία ακέραιη ετικέτα $\ell(v) \in \mathbb{N}$.

Κύριες αρμοδιότητες.

- *Αναπαράσταση γραφήματος*: αποθήκευση κόμβων, ακμών και επιγραφών.
- *Παραγωγή πολυσύνολων επιγραφών* — `computeMultiLabelString()` — υπολογίζει το πολυσύνολο $\{\ell(v), \text{multiset } \ell(N(v))\}_{v \in V}$, ταξινομημένο λεξικογραφικά.
- *Επαναληπτική αναρίθμηση WL* — `relabelingNodes()` — αντικαθιστά κάθε κόμβο με νέα επιγραφή βάσει λεξικού `old_string` → `new_label`.

Σημαντικές μέθοδοι.

`computeStrOfNode(node)` Κατασκευάζει τη συμβολοσειρά $\ell(v).\ell(u_1).\ldots.\ell(u_{\deg(v)})$ σε αύξουσα σειρά των γειτονικών ετικετών του κόμβου v. Η πολυπλοκότητα της μεθόδου αυτής είναι $O(\deg(N(v)) \log \deg(N(v)))$.

`computeMultiLabelString()` Εφαρμόζει την συνάρτηση computeStrOfNode(node) σε όλους τους κόμβους. Χρησιμοποιείται για τον υπολογισμό όλων των επιγραφών πολυσυνόλου σε όλα τα γραφήματα του συνόλου εκπαίδευσης.

`relabelingNodes(newLabelsMap)` Αντιστοιχίζει σε κάθε κόμβο κάθε γραφήματος την επιγραφή $\ell'(v) =$ `newLabelsMap[string(`v`)]`, ολοκληρώνοντας ένα βήμα της διαδικιασίας WL.

`getGraphLabels()` Επιστρέφει το διάνυσμα $[\ell(v_1), \ldots, \ell(v_{|V|})]$ που χρησιμοποιείται για ιστογράμματα.

Τυπικός κύκλος μιας WL επανάληψης

$$\text{labels}^{(t+1)} = \texttt{WLGraph.relabelingNodes}\Big(\textsc{Hash}\big(\texttt{WLGraph.computeMultiLabelString()}\big)\Big).$$

Οφέλη σχεδιασμού.

1. *Καθαρή διεπαφή·* όλες οι λειτουργίες που σχετίζονται με ένα μεμονωμένο γράφημα ενσωματώνονται σε μία ενιαία κλάση.

2. *Ευκολία οπτικοποίησης·* `plotWithLabels()` χρησιμοποιεί `graphviz` για άμεση αποσφαλμάτωση (debugging).

5.5.2 Κλάση `WLGraphListKernel`

Η κλάση `WLGraphListKernel` υλοποιεί τον **απλό πυρήνα Weisfeiler–Lehman (WL)** για μια *λίστα γραφημάτων*, παράγοντας τελικά τα διανύσματα χαρακτηριστικών των στοιχείων του συνόλου δεδομένων που απαιτείται από μοντέλα μηχανικής μάθησης (π.χ. SVM, Kernel Ridge). Επιπλέον, αποθηκεύει πληροφορίες που είναι χρήσιμες για τον υπολογισμό των διανυσμάτων χαρακτηριστικών του συνόλου ελέγχου.

Είσοδοι.

- `graphList`: συλλογή m γραφημάτων G_i.
- h: αριθμός επαναλήψεων WL.

1. Μετατροπή σε `WLGraph`. Η μέθοδος `toWLGraph()` μετατρέπει κάθε γράφημα σε αντικείμενο της κλάσης `WLGraph` (§5.5.1), παρέχοντας απαραίτητες λειτουργίες για τον αλγόριθμο WL.

2. Επανάληψη WL (h+1 γύροι). Η `mainFun()` εκτελεί:

1. **Συλλογή επιγραφών** — `addLabelsToMatrix()` — προσθέτει όλες τις τρέχουσες επιγραφές κάθε γράφου στο `allLabels[i]`.

2. **Συμπίεση ετικετών** — `labelCompression()` — αντιστοιχεί το πολυσύνολο συμβολοσειρών $\ell(v).\text{multiset}(\ell(N(v)))$ σε *νέους* ακεραίους, ξεκινώντας μετά τη μέγιστη αρχική επιγραφή· εγγυάται λεξικογραφική σειρά (μέσω της συνάρτησης `custom_sort` που ταξινομεί λεξικογραφικά τις επιγραφές πολυσυνόλων).

3. **Επανατοποθέτηση Επιγραφών στους Κόμβους** — `graphsRelabeling()` — Επαναπροσδιορίζει τις επιγραφές όλων των κόμβων του γραφήματος χρησιμοποιώντας ένα λεξικό αντιστοίχισης που υπολογίζεται στην τρέχουσα κλάση. Η υλοποίηση γίνεται μέσω της μεθόδου relabelNodes() της κλάσης WLGraph, η οποία δέχεται ως είσοδο το παραγόμενο λεξικό και ενημερώνει τις επιγραφές όλων των κόμβων όλων των γραφημάτων του συνόλου graphList.

Ο βρόχος εκτελείται για $h + 1$ επαναλήψεις, ώστε να συμπεριληφθεί και η αρχική ετικέτα ($t = 0$).

3. Μετατροπή σε διανύσματα συχνοτήτων. Η μετατροπή των γραφημάτων σε διανύσματα χαρακτηριστικών μέσω αυτής της μεθόδου γίνεται βάσει της ακόλουθης σχέσης:

$$\mathbf{x}_i[j] = \#\{\ell \in \texttt{allLabels}[i] \mid \ell = j\}, \quad 1 \le j \le L_{\max},$$

όπου $L_{\max} = \max(\texttt{allLabels})$. Η συνάρτηση `computeFrequencies()` παράγει το τελικό $\mathbf{x}_i \in \mathbb{R}^{L_{\max}}$, το οποίο αποθηκεύεται στη λίστα `graphsToVec`.

Πολυπλοκότητα. Η πολυπλοκότητας της συγκεκριμένης κλάσης για m γραφήματα και πλήθος επαναλήψεων h είναι:

$$O((h + 1) \sum_{i=1}^{m} |E_i| + m\, L_{\max}).$$

Το πρώτο μέρος είναι η επανάληψη WL, το δεύτερο η παραγωγή διανυσμάτων.

Η `WLGraphListKernel` λειτουργεί ως συνδετική κλάση μεταξύ της γραφοθεωρητικής αναπαράστασης και των κλασικών αλγορίθμων μηχανικής μάθησης, παρέχοντας πυκνά, υψηλής εκφραστικότητας διανύσματα με ελάχιστο υπολογιστικό κόστος.

5.5.3 Κλάση `readDataSetFromFullPath`

Η κλάση `readDataSetFromFullPath` αποτελεί το σημείο εισόδου για τη *φόρτωση*, τον *καθαρισμό* και τη *μετατροπή* των αρχικών αρχείων του φακέλου `tree/` σε δομές γραφημάτων κατάλληλες για περαιτέρω επεξεργασία. Ακολουθεί αναλυτική περιγραφή της λειτουργικότητάς της.

Αρμοδιότητες.

1. **Χαρτογράφηση ετικετών** Διαβάζει το `label.txt`, αντιστοιχίζοντας κάθε `tweetId` σε μία από τις τιμές `true` ή `false`. Εγγραφές που φέρουν `non-rumor` ή `unverified` αγνοούνται, ώστε να διατηρηθεί η δυαδική μορφή του προβλήματος.

2. **Φιλτράρισμα σφαλμάτων** Υπάρχει χειροκίνητη λίστα `notCorrectTweets` με *tweetIds* που παρουσιάζουν ελλιπή ή λανθασμένα αρχεία δέντρων· οι εγγραφές αυτές παραλείπονται για να αποφευχθούν εξαιρέσεις κατά την κατασκευή των γράφων.

3. **Κατασκευή γράφημάτων με χρονικό χαρακτηριστικό** Ο μετασχηματισμός `fullGraphWithTimeFeatures()` δημιουργεί κατευθυνόμενο γράφημα `DiGraph` όπου:

   - κάθε κόμβος φέρει ως επιγραφή τον αναγνωριστικό κωδικό κάθε χρήση(label = userId) και επιπλέον επιγραφή το χρονικό σήμα `time` (συσσώρευση χρονικής καθυστέρησης),
   - οι ακμές κωδικοποιούν τη σχέση διάδοσης «*χρήστης A* → *χρήστης B*».

   Έτσι, το γράφημα διατηρεί τόσο τη δομή όσο και τη σχετική χρονική σειρά των αναδημοσιεύσεων.

Σημαντικές Μέθοδοι.

`returnVeracityPerTweetId()` Επιστρέφει λεξικό ―`tweetId: veracity`″. Η πολυπλοκότητα είναι $O(N)$, όπου N το πλήθος γραμμών του `label.txt`.

`createDataSet()` Διατρέχει τον φάκελο `tree/`, κατασκευάζοντας τον δέντρο διάδοσης κάθε δημοσίευσης και επιστρέφοντας τη λίστα από πλειάδες της μορφής `(graph, label)`. Παραλείπει τα *tweetIds* που δεν ανήκουν σε μία από την κατηγορία true/false (δηλαδή είναι non-rumors/unverified-rumors) ή ανήκουν στη λίστα σφαλμάτων.

`fromGraphFileToMatrix()` Μετατρέπει το μη επεξεργασμένο αρχείο της μορφής `[userId, tweetId, ...]` -᾽ [...] σε πίνακα λιστών. Στη συνέχεια, εφαρμόζεται ανάλυση σε μία διέλευση (one-pass parsing) ώστε να ελαχιστοποιηθεί η κατανάλωση της μνήμης.

`fullGraphWithTimeFeatures()` Κατασκευάζει το τελικό `DiGraph` για κάθε δέντρο διάδοσης· ο πρώτος κόμβος (ρίζα) αρχικοποιείται με `time = 0`, ενώ κάθε επόμενος κόμβος προσθέτει τη δική του χρονική καθυστέρηση στη συνολική διάρκεια του γονέα του.

Πλεονεκτήματα Σχεδιασμού.

- *Απλότητα και σαφήνεια*: Η διαδικασία φόρτωσης ολοκληρώνται μέσω ενός ενιαίου API, καθώς τα δεδομένα είναι αποθηκευμένα στο cloud. Αυτό απλοποιεί σημαντικά την ενσωμάτωση του συστήματος σε υπάρχουσες αλυσίδες επεξεργασίας (pipelines).
- *Ενσωματωμένο φιλτράρισμα*: η απομάκρυνση προβληματικών εγγραφών νωρίς στον κύκλο μειώνει τις εξαιρέσεις κατά την εκπαίδευση.
- *Χρονικό χαρακτηριστικό*: επιτρέπει την εισαγωγή δυναμικών πληροφοριών (ρυθμός εξάπλωσης) στα επόμενα βήματα εξαγωγής χαρακτηριστικών.

Πολυπλοκότητα. Η συνολική χρονική πολυπλοκότητα είναι $O(|F|+|E|)$, όπου $|F|$ το πλήθος αρχείων και $|E|$ το πλήθος αναδημοσιεύσεων σε όλα τα δέντρα, καθώς κάθε ακμή αναλύεται ακριβώς μία φορά. Η χρήση μνήμης είναι $O(|V| + |E|)$ εντός των αντικειμένων `NetworkX`.

5.5.4 Κλάση `canonicalTreeTransformer`

Η κλάση `canonicalTreeTransformer` υλοποιεί έναν αλγόριθμο *κανονικοποίησης δέντρων* με επιγραφές στους κόμβους τους. Στόχος της είναι η μετατροπή *οποιουδήποτε* δέντρου με επιγραφές T σε μία μοναδική, **ισομορφικά αναλλοίωτη** αναπαράσταση T^*· κάθε ζεύγος ισομορφικών δέντρων μετασχηματίζεται στην ίδια τελική μορφή T^*.

Η ιδιότητα αυτή επιτρέπει αποδοτική αποθήκευση, σύγκριση και ομαδοποίηση υποδέντρων σε μεταγενέστερα βήματα, καθώς –όπως αναφέραμε και στην Ενότητα 3.2.3– δίνεται η δυνατότητα να ελέγξουμε σε χρόνο $O(|V|)$ αν δύο δέντρα είναι ισομορφικά ως προς τις επιγραφές τους. Επιπλέον, ο διαμερισμός σε κλάσεις ισοδυναμίας βάσει του ισομορφισμού είναι κρίσιμος, διότι:

- ελαχιστοποιεί τον αριθμό των διακριτών προτύπων που χρειάζεται να αποθηκεύσουμε·
- επιτρέπει τη δημιουργία συνεκτικών 'ομάδων' δέντρων, μέσα στις οποίες κάθε στοιχείο είναι δομικά ταυτόσημο
- διευκολύνει τη στατιστική επεξεργασία, αφού κάθε ομάδα μπορεί να αντιμετωπιστεί ως ένα ενιαίο, αντιπροσωπευτικό σύνολο χαρακτηριστικών.

Με άλλα λόγια, χάρη στον έλεγχο ισομορφισμού μπορούμε να οικοδομήσουμε σαφώς ορισμένες **ομάδες μοτίβων** · κάθε νέο δέντρο τοποθετείται αυτόματα στη κατάλληλη κατηγορία, βελτιώνοντας τόσο την ταχύτητα των υπολογισμών όσο και την ποιότητα των παραγόμενων χαρακτηριστικών.

Υψηλού επιπέδου ροή.

1. **Εξίσωση βάθους (*padding*).** Τα φύλλα μικρότερου ύψους επεκτείνονται με κόμβους ετικέτας 0 ώστε κάθε μονοπάτι ρίζας–φύλλου να αποκτήσει κοινό μήκος $d_{\max}$. Ωστόσο, το βήμα αυτό είναι περιττό στα δέντρα ξεδίπλωσης ύψους i καθώς όλα τα φύλλα βρίσκονται στο ίδιο ύψος.
2. **Λεξικογραφική ταξινόμηση ετικετών.** Για κάθε εσωτερικό κόμβο αναδιατάσσονται οι ετικέτες των παιδιών σε αύξουσα σειρά.
3. **Σύμπτυξη υποδέντρων.**
   (α΄) Δημιουργούνται συμβολοσειρές τύπου `root.child_1.child_2...` για όλους τους κόμβους γονείς φύλλων.
   (β΄) Οι συμβολοσειρές ταξινομούνται και αντιστοιχίζονται σε νέες ετικέτες.
   (γ΄) Ο κόμβος γονέας λαμβάνει τη νέα ετικέτα και οι κατιόντες φύλλα αφαιρούνται.

   Η διαδικασία επαναλαμβάνεται μέχρι το δέντρο να περιοριστεί σε ύψος 1.

4. **Ξεδίπλωση δέντρου (*unfolding*).** Το συμπτυγμένο δέντρο «ξεδιπλώνεται», εισάγοντας ξανά τα απαλειφθέντα φύλλα, αυτή τη φορά με τις νέες ετικέτες, ώστε να ανακτηθεί πλήρης δομή χωρίς τοπικές ασυμφωνίες.

5. **Επαναρίθμηση κόμβων.** Οι κόμβοι επαναριθμούνται από 1 έως $|V|$ και αφαιρούνται τυχόν κόμβοι ετικέτας 0, παράγοντας το τελικό κανονικοποιημένο γράφημα T^*.

Βασικές μέθοδοι.

`padWithZeroNodes()` Ισοσταθμίζει το βάθος όλων των φύλλων.

`sortNodeLabels()` Λεξικογραφική αναδιάταξη επιγραφών παιδιών των κόμβων που βρίσκονται σε βάθος $d_{\max} - 1$.

`createSubtreeStrEnum()` Δημιουργία λεξικού `signature` → νέα ετικέτα.

`nodeReplacement()` Αντικαθιστά τους γονείς φύλλων με νέες ετικέτες από το λεξικό `subtreeMap`, συμπτυκτώνοντας τα υποδέντρα τους σε μοναδικές ετικέτες, και διαγράφει τους απογόνους τους για να διατηρηθεί η συμπυκνωμένη δομή.

`unfoldTree()` Ανακατασκευάζει το δέντρο επεκτείνοντάς το, εισάγοντας νέους κόμβους βάσει των συμβολοσειρών που αποθηκεύονται στον χάρτη `labelSubtreeString`, αποκαθιστώντας την αρχική ιεραρχία και το βάθος.

Πολυπλοκότητα.

- Χρονική: $O(h\,(|V| + |E|) \;+\; |V| \log |V|)$, όπου h ο αριθμός επαναλήψεων σύμπτυξης. Ο όρος $|V| \log |V|$ προέρχεται από τις ταξινομήσεις ετικετών.
- Χωρική: $O(|V| + |E|)$, καθώς ο αλγόριθμος αποθηκεύει μόνον τα προσωρινά υποδέντρα και ένα λεξικό αντιστοίχισης.

5.5.5 Κλάση `nodeLabelingOfDataSet`

Σκοπός. Πριν εφαρμοστούν οι Weisfeiler–Lehman πυρήνες ή οποιοσδήποτε ταξινομητής που απαιτεί *διακριτές* επιγραφές κόμβων, απαιτείται η αντιστοίχιση κάθε κορυφής των γραφημάτων διάδοσης σε μία ακέραια ετικέτα. Η κλάση `nodeLabelingOfDataSet` υλοποιεί το σύστημα τοποθέτησης επιγραφών (labeling) που επιτρέπει τόσο απλές στρατηγικές τοποθέτησης επιγραφών που βασίζεται στο ίδιο το γράφημα (cascade-based), όσο και αξιοποίηση εξωγενούς πληροφορίας (graph-based). Χρησιμοποιείται αμέσως μετά τη φόρτωση των δεδομένων και *πριν* από την εξαγωγή χαρακτηριστικών ή τον υπολογισμό των διανυσματικών αναπαραστάσεων.

Είσοδος. Σύνολο Δεδομένων:

$$[(G_i,\ y_i)]_{i=1}^{N}, \quad G_i = (V_i, E_i),\ y_i \in \{true, false\},$$

όπου κάθε G_i είναι δέντρο ή γράφος διάδοσης (NetworkX DiGraph) και y_i η ετικέτα αληθούς/ψευδούς είδησης.

Βασικές μέθοδοι.

- ¨`init`¨(`dataSet, method="cascade-based-log-bin"`, k=4) αποθηκεύει το σύνολο, επιλέγει στρατηγική και καλεί `run()`.
- `run()` μέθοδος που τρέχει πάνω σε όλα τα γραφήματα· υπολογίζει τοπικές μετρικές (out-degree, βάθος, ¨ απογόνων κ.λπ.) και εκχωρεί την ετικέτα $\ell(v)$ σε κάθε κόμβο.
- `logBinBucket(x)` λογαριθμική ομαδοποίηση $x \mapsto \lfloor \log_{10} x \rfloor + 1$.
- `kmeans1d(arr, k)` k-Means σε 1-διάστατο πίνακα, επιστρέφει $\{1, \ldots, k\}$ ετικέτες.

Υποστηριζόμενες στρατηγικές.

1. **Cascade-based**
   - `cascade-based-log-bin:` λογαριθμική ομαδοποίηση εξωτερικού βαθμού κάθε κόμβου εντός του *ίδιου* δέντρου· στα φύλλα τοποθετούμε την επιγραφή 1.
   - `cascade-based-kmeans:` k-μέσων (προεπιλογή $k = 4$) στους στους εξωτερικούς βαθμούς του δέντρου. Ως αλγόρριθμος συσταδοποίησης χρησιμοποιείται ο k-Means σε 1 διάσταση.
   - `cascade-based-max-depth:` βάθος κόμβου από τη ρίζα, κανονικοποιώντας το.
   - `cascade-based-kmeans-tot-descendants:` Η τιμή της συστάδας στο πλήθος απογόνων (descendants) μετά από εφαρμογή του αλγορίθμου k-Means σε 1 διάσταση.
   - `cascade-based-tot-leaves-kmeans` Η τιμή της συστάδας στο πλήθος φύλλων που ανήκουν στον υποδένδρο μετά από εφαρμογή του αλγορίθμου k-Means σε 1 διάσταση.
   - `cascade-based-max-leaf-root-depth:` απόσταση κόμβου από το βαθύτερο φύλλο.
2. **Graph-based**
   - `graph-based-out-deg-sum:` , `-mean`, `-median` λογαριθμική δειγματοληψία συνολικού, μέσου ή διάμεσου εξωτερικού βαθμού ανά χρήστη σε *όλο* το σύνολο δεδομένων.
   - `graph-based-k-means:` συστάδα στην οποία ανήκει η συχνότητα συμμετοχής κάθε χρήστη σε όλα τα δέντρα διάδοσης μετά από εφαρμογή αλγορίθμου Κ-Μέσων σε 1-Διάσταση.
   - `out-deg-statistics:` τριπλή κατηγορία κόμβων βάσει σύγκρισης του εξωτερικού βαθμού τους με τη μέση τιμή αληθών ή των ψευδών δέντρων διάδοσης.

Πολυπλοκότητα. Για κάθε γράφο $G_i = (V_i, E_i)$:

$$O(|V_i| + |E_i|) \; + \; \begin{cases} O(k|V_i|) & \text{σε περίπτωση k-Means} \\ O(|V_i|) & \text{σε περίπτωση λογ. ομαδοποίησης} \end{cases}$$

Επομένως, για ολόκληρο το σύνολο η πολυπλοκότητα είναι $O(\sum_i |V_i|+|E_i|)$, δηλαδή γραμμική στο συνολικό πλήθος κόμβων/ακμών.

Εφαρμογή. Η `nodeLabelingOfDataSet` καλείται:

1. Μετά το διάβασμα των αρχείων `tree.txt/label.txt`.
2. Πριν από τη μετατροπή των $\tilde{G}_i$ σε διανύσματα χαρακτηριστικών με τον απλό ή τον γενικευμένο Πυρήνα Γραφημάτων Weisfeiler–Lehman.

Με τον τρόπο αυτό εξασφαλίζεται ότι *όλοι* οι κόμβοι φέρουν συνεπείς, διακριτές και αριθμητικές επιγραφές, καθιστώντας εφικτή την αποδοτική συγκριτική ανάλυση των δέντρων διάδοσης.

5.5.6 Κλάση `SDTedClass`

Σκοπός. Η `SDTedClass` παρέχει έναν αποδοτικό υπολογισμό της *Structure-and-Depth Preserving Tree Edit Distance* (SD-TED): ενός μέτρου ομοιότητας που λαμβάνει υπόψη τόσο τις επιγραφές κόμβων όσο και τη σχετική *θέση* τους στο δέντρο διάδοσης. Η απόσταση αυτή αποτελεί τη βασική μετρική στους αλγορίθμους ομαδοποίησης υποδέντρων και στην αναζήτηση εγγύτερων δέντρων κατά τη φάση της δημιουργίας των διανυσμάτων χαρακτηριστικών των γραφημάτων του συνόλου ελέγχου.

Είσοδος.

- Δύο κατευθυνόμενα δέντρα $T_1 = (V_1, E_1)$, $T_2 = (V_2, E_2)$ σε μορφή `networkx.DiGraph`.
- Πίνακας κόστους $\Sigma \in \mathbb{R}^{(\ell+1)\times(\ell+1)}$ (οι τελευταίες γραμμή/στήλη αντιστοιχούν σε πράξεις *εισαγωγής* και *διαγραφής* insert/delete).

Κύρια ροή λειτουργίας.

1. **Εξαγωγή υποδέντρων**: εντοπίζονται οι άμεσοι απόγονοι της ρίζας και δημιουργούνται τα αντίστοιχα υποδέντρα. Για ισοστάθμιση προστίθενται κενά δέντρα.
2. **Κόστος υποδέντρων**: κατασκευάζεται ο πίνακας M διαστάσεων $d_1 \times d_2$ (ή τετραγωνικός μετά τη συμπλήρωση), όπου

$$M_{ij} = \begin{cases} \Sigma(\ell(T_1[i]), \ell(T_2[j])) & \text{για αντικατάση επιγραφών ρίζας,} \\ \texttt{SDTED}(T_1[i], T_2[j]) & \text{διαφορετικά,} \\ \Sigma(\ell, \langle del \rangle) & \text{για διαγραφή,} \\ \Sigma(\langle ins \rangle, \ell) & \text{για εισαγωγή.} \end{cases}$$

Οι κλήσεις SDTED() αποθηκεύονται σε λεξικό tedSubtreeCost (memoization).

3. **Ελάχιστο ταίριασμα**: εφαρμόζεται ο Hungarian αλγόριθμος (linear_sum_assignment) για τον ελάχιστο συνολικό κόστος $\sum_{(i,j)} M_{ij}$.

4. **Συνδυασμός**: Το τελικό αποτέλεσμα είναι το:

$$\mathrm{SDTED}(T_1, T_2) = \Sigma(\ell(\mathrm{root}_{T_1}), \ell(\mathrm{root}_{T_2})) + \min_{\pi} \sum_{(i,j)\in\pi} M_{ij}.$$

Βασικές μέθοδοι υλοποίησης.

`__init__`() Φόρτωση δέντρων, πίνακα Σ, αρχικοποίηση λεξικού tedSubtreeCost.

`computeDistance()` Αναδρομική υλοποίηση των βημάτων (1)–(4) με caching.

`bfs_tree()` Δημιουργία υποδέντρων μέσω Breadth-First Search.

Τεχνικές βελτιστοποίησης.

- **Memoization**: το λεξικό tedSubtreeCost μειώνει την πολυπλοκότητα από εκθετική σε ψευδοπολυωνυμική στο πλήθος μοναδικών υποδέντρων.
- **Κενά δέντρα**: ενσωματώνουν πράξεις εισαγωγής/διαγραφής (insert/delete) απευθείας στον πίνακα M.

Πολυπλοκότητα. Έστω d_1, d_2 τα πλήθη άμεσων απογόνων των ριζών.

$$T_{\mathrm{SDTED}}(T_1, T_2) = O(d_1\, d_2 \;+\; \max\{d_1, d_2\}^3),$$

όπου ο πρώτος όρος προέρχεται από τις αναδρομικές κλήσεις και ο δεύτερος από τον αλγόριθμο Hungarian που χρησιμοποιείται στο κομμάτι του ελαχίστου κόστους διμερούς ταιριάσματος. Στα δέντρα Twitter15/16 με $d_i \leq 10$ το κόστος παραμένει πρακτικά γραμμικό ως προς το μέγεθος του δέντρου.

Χρήση στην εργασία. Η SDTedClass:

1. χρησιμοποιείται στον ModifiedWassersteinKMeans για τον ορισμό της απόστασης μεταξύ κεντροειδών και δειγμάτων,
2. καλείται από την FindNearestTrees για την εύρεση των πλησιέστερων υποδέντρων κατά τη διαδιακασία της δημιουργίας των διανυσμάτων χαρακτηριστικών του συνόλου ελέγχου.

5.5.7 Κλάση ModifiedWassersteinKMeans

Η κλάση ModifiedWassersteinKMeans υλοποιεί έναν αλγόριθμο συσταδοποίησης στον χώρο των **μεταφορών κατά Wasserstein** (*Wasserstein k-means*), όπως προτάθηκε από τους Schultz *et al.* [3]. Η βασική ιδέα συνίσταται στην αντικατάσταση της Ευκλείδειας απόστασης

με την *απόσταση μεταφοράς μάζας* (*Earth Mover's Distance, EMD*), προκειμένου να λαμβάνονται υπόψη οι δομικές ομοιότητες μεταξύ των διανυσματικών αναπαραστάσεων των δέντρων ανάπτυξης. Με τον τρόπο αυτό επιτυγχάνεται η ομαδοποίηση των δέντρων σε προκαθορισμένο πλήθος συστάδων, με βάση τις δομικές τους ιδιότητες.

Είσοδοι.

- `vecs`: συλλογή διανυσμάτων $\mathbf{x}_i \in \mathbb{R}^{n_r+n_c}$. Σε κάθε διάνυσμα είναι κωδικοποιημένη η πληροφορία που αφορά τα προς συσταδοποίηση δέντρα ξεδίπλωσης.
- $\mathbf{M}_r, \mathbf{M}_c$: πίνακες κόστους Wasserstein για τις επιγραφές των κόμβων και των υποδέντρων που έχουν ως ρίζα τα παιδιά της ρίζας κάθε δέντρου ξεδίπλωσης.
- k: αριθμός συστάδων, n: πλήθος επαναλήψεων, $\langle n_r, n_c \rangle$: διαστάσεις επιμέρους ιστογραμμάτων.

Λειτουργία αλγορίθμου.

1. *Κεντρική αρχικοποίηση* Επιλέγονται τυχαία k δείγματα και κανονικοποιούνται σε ιστόγραμμα πιθανότητας.
2. *Ανάθεση σημείων σε συστάδες* Για κάθε διάνυσμα υπολογίζεται το άθροισμα δύο αποστάσεων EMD:

$$d_W(\mathbf{x}, C_j) = \mathrm{EMD}(\mathbf{x}^{(r)}, C_j^{(r)}, \mathbf{M}_r) + \mathrm{EMD}(\mathbf{x}^{(c)}, C_j^{(c)}, \mathbf{M}_c),$$

   όπου C_j το j-οστό κέντρο.
3. *Επανυπολογισμός των κέντρων* Κάθε συστάδα αντικαθίσταται από το *Wasserstein βαρυκεντρικό μέσο* των στοιχείων της, επιλυόμενο με αλγόριθμο Sinkhorn (τακτική παράμετρος $\lambda = 10^{-2}$) και μέγιστο $N_{\mathrm{it}} = 20$ επαναλήψεις.
4. *Τερματισμός* Επανάληψη των βημάτων 2–3 για n επανάληψεις ή μέχρι την σύγκλιση των αναθέσεων.

Τεχνικές υλοποίησης.

- **Κανονικοποίηση Ιστογραμμάτων** Η `normalize_histogram()` εξασφαλίζει ότι κάθε διάνυσμα είναι μη αρνητικό και έχει άθροισμα 1.
- **Σταθεροποίηση αριθμητικού σφάλματος** Τα διανύσματα περικόπτονται από κάτω στο $\varepsilon = 10^{-15}$ πριν περάσουν στην `POT/emd2()`.
- **Παράλληλη ανάθεση** Οι αποστάσεις EMD κάθε δείγματος από τα κέντρα υπολογίζονται με `joblib.Parallel` για ταχύτερη εκτέλεση του αλγορίθμου.
- **Αποθήκευση βαρυκέντρων** Τα κέντρα των συστάδων αποθηκεύονται σε ξεχωριστούς πίνακες (`centers_r`, `centers_c`), επιτρέποντας τον ανεξάρτητο υπολογισμό των δύο βαρυκέντρων. Ωστόσο αυτό το βήμα μειώνει τον χρόνο σύγκλισης.

Πολυπλοκότητα. Ένας κύκλος ανάθεσης–βαρυκέντρου κοστίζει $\Theta(Nk\,\mathrm{EMD}_{n_r+n_c})$, όπου EMD_d η πολυπλοκότητα EMD σε διάσταση d. Η παράλληλη υλοποίηση κλιμακώνεται σχεδόν γραμμικά ως προς τον αριθμό πυρήνων CPU.

5.5.8 Κλάση `compute1UnfTreeToClusterMap`

Η κλάση `compute1UnfTreeToClusterMap` υλοποιεί την αντιστοίχιση του *σύνολο όλων* των **υποδέντρων βάθους 1** (*one-unfolding trees*) σε *κλάσεις ομοιοτήτας*, χρησιμοποιώντας το τροποποιημένο Wasserstein k-means (βλ. §5.5.7). Η διαδικασία παράγει ένα λεξικό:

$$\mathcal{M}\colon \mathrm{treeLabelEncoding}(T) \longrightarrow \{0,\dots,k-1\},$$

το οποίο χρησιμοποιείται στις φάσεις της ομαδοποίησης χαρακτηριστικών και του υπολογισμού πυρήνων.

Είσοδοι.

- `unfoldingTrees`: λίστα N δέντρων ανάπτυξης T_i βάθους 1.
- `maxLabel`: μέγιστη επιγραφή όλων των κόμβων ($\ell_{\max}$).
- `d`: μέγιστος βαθμός σε όλα τα δέντρα διάδοσης.
- `k`: πλήθος διαφορετικών συστάδων.

Βήματα αλγορίθμου.

1. **Λεξικογραφική ταξινόμηση** — `sortUnfTreesListLexicographically()` — εγγυάται ντετερμινιστική επεξεργασία των υποδέντρων ανεξαρτήτως της αρίθμησης των κόμβων.
2. **Διανυσματική αναπαράσταση υποδέντρου T.** Για να κωδικοποιήσουμε, ταυτόχρονα, (i) την ετικέτα της ρίζας και (ii) την κατανομή των παιδιών της, αντιστοιχίζουμε σε κάθε δέντρο ανάπτυξης το διάνυσμα [3]

$$\mathbf{v}(T) = \underbrace{\mathbf{e}_{\ell(r)}}_{\text{one-hot vec (root label)}} \;\|\; \underbrace{(c_1, c_2, \dots, c_{\ell_{\max}},\; 2d - \deg^+(r))}_{\text{κατανομή ετικετών παιδιών + όρος κανονικοποίησης}} \in \mathbb{R}^{2\ell_{\max}+1},$$

όπου

- $\mathbf{e}_{\ell(r)} \in \{0,1\}^{\ell_{\max}}$ είναι το διάνυσμα μοναδιαίου σημαντικού (one-hot vector) της ετικέτας της ρίζας r,
- $c_\ell = \#\{\text{παιδιά της ρίζας με ετικέτα } \ell\}$ μετρά πόσα παιδιά της ρίζας φέρουν κάθε δυνατή ετικέτα,
- $\deg^+(r)$ είναι ο εξωτερικός βαθμός της ρίζας, και
- ο τελευταίος όρος $2d - \deg^+(r)$ συμπληρώνει τα υποδέντρα που λείπουν μέχρι τον μέγιστο επιτρεπτό αριθμό $2d$, ώστε όλα τα διανύσματα να έχουν κοινό άθροισμα συνιστωσών (ℓ_1-κανονιποιημένα).

Αναπαράσταση Δομής Δέντρου Η μέθοδος αυτή καταγράφει πλήρως την τοπολογία του δέντρου μέσω:

- Της **ετικέτας της ρίζας**, που κωδικοποιείται απευθείας
- Ενός **κανονικοποιημένου ιστογράμματος** που αναπαριστά:
  - Τις συχνότητες εμφάνισης κάθε δυνατής ετικέτας παιδιών
  - Την αναλογική κατανομή τους (εξομοιώνοντας δέντρα διαφορετικών μεγεθών)

Αυτή η προσέγγιση διασφαλίζει τη *συγκρισιμότητα* μεταξύ όλων των δέντρων $T_1, T_2 \in \mathcal{T}$, ανεξαρτήτως του πλήθους παιδιών της ρίζας, δηλαδή:

$$\text{Συγκρισιμότητα} \quad \forall\, T_1, T_2 \in \mathcal{T}, \quad \text{ανεξάρτητα του } |\text{Children}(T_i)|$$

όπου $\mathcal{T}$ είναι το σύνολο των υποδέντρων και $|\text{Children}(T_i)|$ ο αριθμός παιδιών της ρίζας του T_i.

3. **Συσταδοποίηση με Wasserstein k-means** Τα ιστογράμματα χωρίζονται σε V_r και V_c ($n_r = n_c = \ell_{\max} + 1$) και ο αλγόριθμος `ModifiedWassersteinKMeans` επιστρέφει λεξικό `res[cluster_id]` $\rightarrow$ λίστα διανυσμάτων.

4. **Αντιστοίχιση υποδέντρων $\rightarrow$ συστάδες** Για κάθε υποδέντρο χρησιμοποιείται η μέθοδος `treeLabelEncoding()` ως κλειδί· η συνάρτηση `createMapFromUnfoldingTreeToCluster()` παράγει το τελικό λεξικό $\mathcal{M}$.

5. **Έλεγχος κενών συστάδων.** Αν μετά το πρώτο πέρασμα κάποια συστάδα είναι κενή, ο αλγόριθμος ξανατρέχει με νέα τυχαία κέντρα, ώστε να εξασφαλίσει πλήρη κάλυψη του χώρου.

Πολυπλοκότητα. Η χρονική πολυπλοκότητα της δημιουργίας των διανυσματικών αναπαραστάσεων των δέντρων ξεδίπλωσης είναι γραμμική ως προς το πλήθος των υποδέντρων, δηλαδή $\mathcal{O}(N(\ell_{\max} + d))$. Ο υπολογιστικά βαρύς αλγόριθμος είναι ο *Wasserstein k-means*· σε κάθε επανάληψη εκτελεί Nk υπολογισμούς EMD σε χώρο $2(\ell_{\max} + 1)$ διαστάσεων. Αν συμβολίσουμε με EMD_m το κόστος υπολογισμού της Απόστασης Μεταφοράς Μάζας (*Earth Mover's Distance*) σε m διαστάσεις, η πολυπλοκότητα ανά επανάληψη είναι ίση με:

$$\mathcal{O}(Nk \cdot \text{EMD}_{2(\ell_{\max}+1)}).$$

5.5.9 Κλάση `compute2UnfTreeToClusterMap`

Η κλάση `compute2UnfTreeToClusterMap` επεκτείνει τη λογική του προηγούμενου βήματος `compute1UnfTreeToClusterMap`, δημιουργώντας ένα λεξικό

$$\mathcal{M}_2 : \{\text{treeLabelEncoding}(T^{(2)})\} \longrightarrow \{0, \ldots, k-1\},$$

όπου $T^{(2)}$ δηλώνει **Δέντρα-Ανάπτυξης βάθους 2** (*two-unfolding trees*). Η δημιουργία της αντιστοίχισης δημιουργείται συνδυάζοντας: α) γραφικά χαρακτηριστικά ρίζας/υποδέντρων (με ρίζα τα παιδιά της ρίζας), β) πίνακα κόστους υποδέντρων βάθους 1 και γ)Αλγορίθμου Wasserstein k-means.

Είσοδοι.

- `unfTreesList`: πλήρης λίστα N υποδέντρων βάθους 2.
- $\ell_{\max}$ (= `maxLabel`): μέγιστη επιγραφή σε όλους τους κόμβους.
- d: μέγιστος βαθμός ρίζας σε όλα τα δέντρα ξεδίπλωσης.
- k: πλήθος των συστάδων.

1. Κατάλογος μοναδικών υποδέντρων βάθους 1. Αρχικά παράγεται το σύνολο $\mathcal{S}_1$ *μη ισομορφικών* υποδέντρων βάθους 1 που έχουν ως ρίζα τα παιδιά της ρίζας όλων των δέτρων ανάπτυξης ύψους 2 (μέθοδος `computeSubTreeAndSubtreesToIndexMapping()`). Κάθε τέτοιο υποδέντρο λαμβάνει μοναδικό δείκτη $\mathrm{i} \in \{0, \ldots, |\mathcal{S}_1| - 1\}$.

2. Διάνυσμα χαρακτηριστικών $\mathbf{x}(T^{(2)})$. Για κάθε δέντρο ξεδίπλωσης βάθους 2:

$$\mathbf{x} = \underbrace{\mathbf{e}_{\ell(r)}}_{\text{ρίζα}} \;\|\; \underbrace{(\#\mathcal{S}_1^{(0)}, \ldots, \#\mathcal{S}_1^{(|\mathcal{S}_1|-1)},\ 2d - \deg^+(r))}_{\text{ιστόγραμμα υποδέντρων βάθους 1 + όρος κανονικοποίησης}}$$

όπου $\#\mathcal{S}_1^{(i)}$ είναι το πλήθος παιδιών της ρίζας που είναι ισομορφικά με το i-οστό μέλος της $\mathcal{S}_1$.

3. Πίνακας κόστους $\mathbf{M}_c$ για τα ιστογράμματα βάθους 1. Κατασκευάζεται ο πίνακας $\mathbf{M}_c \in \mathbb{R}^{(|\mathcal{S}_1|+1)\times(|\mathcal{S}_1|+1)}$ ο οποίος υπολογίζεται ως εξής:

$$\mathbf{M}_c[i,j] = \begin{cases} \mathrm{TED}(S_i, S_j), & 0 \le i,j < |\mathcal{S}_1| \\ |V(S_i)|, & j = |\mathcal{S}_1| \\ |V(S_j)|, & i = |\mathcal{S}_1| \end{cases}$$

όπου `TED` καλεί τον `SDTedClass` για απόσταση υποδέντρων του συνόλου Σ_1.

4. Συσταδοποίηση με Wasserstein k-means. Χρησιμοποιείται η `ModifiedWassersteinKMeans` με:

$$\mathbf{M}_r = \mathbf{1} - \mathbf{I}_{\ell_{\max}+1}, \quad \mathbf{M}_c \text{ όπως παραπάνω}, \quad \textit{numOfIters} = 5 \text{ επαναλήψεις}.$$

Εάν μία συστάδα βρεθεί κενή, ο αλγόριθμος επανεκκινεί με νέα αρχικοποίηση κέντρων (μέθοδος `emptyCluster()`).

5. Παραγωγή λεξικού. Το λεξικό $\mathcal{M}_2$ που προκύπτει από τη μέθοδο `findClusterOfEachUnfoldingTree()` αντιστοιχεί κάθε υποδέντρο στη συστάδα όπου ανήκει η διανυσματική του αναπαράσταση.

Πολυπλοκότητα. Η συνολική πολυπλοκότητα του προτεινόμενου αλγορίθμου ισούται με:

$$O(N(\ell_{\max} + |\mathcal{S}_1|) \; + \; |\mathcal{S}_1|^2 \, \mathrm{TED} \; + \; Nk \, \mathrm{EMD}_{\ell_{\max}+|\mathcal{S}_1|+2}),$$

όπου:

- N: το πλήθος των υποδέντρων (unfolding–trees) που παράγονται από το σύνολο των δέντρων διάδοσης.
- $\ell_{\max}$: ο μέγιστος αριθμός διαφορετικών ετικετών ρίζας· ισοδυναμεί με το μήκος του διανύσματος μονού σημαντικού (one-hot vector) $\mathbf{e}_{\ell(r)}$.
- $\mathcal{S}_1$: το σύνολο των *μοναδικών* υποδέντρων βάθους 1 Το μέγεθός του, $|\mathcal{S}_1|$, χρησιμοποιείται τόσο στη διάσταση των διανυσμάτων χαρακτηριστικών όσο και στους συγκριτικούς υπολογισμούς ομοιότητας.
- TED: το κόστος υπολογισμού της *Tree-Edit Distance* ανά ζεύγος δέντρων ανάπτυξης· στην παράσταση εμφανίζεται ως $|\mathcal{S}_1|^2$TED, διότι απαιτείται ο πλήρης πίνακας αποστάσεων μεταξύ όλων των υποδέντρων βάθους 1.
- k: ο αριθμός συστάδων (k στους οποίους χωρίζονται τα υποδέντρα.
- EMD_m: η πολυπλοκότητα υπολογισμού της *Earth Mover's Distance* σε m-διάστατο χώρο. Στην περίπτωσή μας ο αριθμός διαστάσεων είναι $\ell_{\max} + |\mathcal{S}_1| + 2$

Ο πρώτος όρος $N(\ell_{\max} + |\mathcal{S}_1|)$ αντιστοιχεί στην κατασκευή των διανυσμάτων χαρακτηριστικών για όλα τα υποδέντρα· ο δεύτερος αφορά τη δημιουργία του πίνακα αποστάσεων με βάση την TED (Διορθωτική Απόσταση Δέντρων)· ο τρίτος, $Nk \, \mathrm{EMD}_{\ell_{\max}+|\mathcal{S}_1|+2}$, καθορίζει τον χρόνο εκτέλεσης του αλγορίθμου *Wasserstein k-means*, καθώς σε κάθε επανάληψη απαιτούνται Nk υπολογισμοί της Earth Mover's Distance (Μετρικής Βέλτιστης Μεταφοράς) στον αντίστοιχο διανυσματικό χώρο.

5.5.10 Κλάση `trainingSetTestSetSplitter`

Η κλάση `trainingSetTestSetSplitter` αναλαμβάνει τον διαχωρισμό σε ισο πλήθος στοιχείων ανά κατηγορία (true/false) του συνόλου δεδομένων σε *σύνολο εκπαίδευσης* και *σύνολο ελέγχου*. Στο σύστημα ορίζεται ως παράμετρος το μέγιστο επιτρεπτό μέγεθος του γραφήματος, λειτουργώντας ως επιπλέον φίλτρο προκειμένου να αποκλειστούν ακραία μεγάλα δέντρα που θα επιβάρυναν τον χρόνο εκπαίδευσης και που θεωρούνται ακραίες περιπτώσεις (outliers).

Είσοδοι.

- `dataset`: λίστα ζευγών (G_i, y_i), με $y_i \in \{true, false\}$ και G_i είναι τύπου nx.Graph()
- `perc`$\in [0, 1]$: ποσοστό κάθε κλάσης που θα κατανεμηθεί στο *σύνολο εκπαίδευσης*.
- `maxSize`: μέγιστο επιτρεπτό πλήθος κόμβων δέντρων διάδοσης · γραφήματα μεγαλύτερα δεν συμπεριλαμβάνονται στο τελικό σύνολο δεδομένων που θα χρησιμοποιηθεί για την διαδικασία των πειραμάτων.

Βήματα αλγορίθμου.

1. **Φιλτράρισμα μεγέθους** — `filtering()` — διατηρεί μόνο στοιχεία του συνόλου με $|V(G_i)| \leq$ `maxSize`.

2. **Υπολογισμός στόχου ανά κλάση** $n = \lfloor \frac{|\mathcal{D}|}{2} \rfloor$ (υποθέτοντας περίπου ισόρροπο σύνολο) $\rightarrow$ `totTrue` = `totFalse` = `perc` $\cdot\, n$.

3. **Τυχαία δειγματοληψία** `getTrueElems()`, `getFalseElems()` ανακατεύουν τυχαία (`random.shuffle`) και επιστρέφουν *train*, *test* υπολίστες μεγέθους `perc` $\cdot len(D), (1 - perc) \cdot len(D)$.

4. **Σύνθεση τελικών συνόλων** $\mathcal{T}_{\text{train}} = trueTrain \cup falseTrain$, $\mathcal{T}_{\text{test}} = trueTest \cup falseTest$. Σε αυτό το τελευταίο βήμα, γίνεται ανακάτεμμα (`shuffle`) για την τυχαία ανάμειξη συνόλων εκπαίδευσης και ελέγχου.

Ιδιότητες.

- *Ισοζυγισμένος διαχωρισμός*: εξασφαλίζει ίσο πλήθος δειγμάτων ανά κλάση και στα δύο σύνολα.

- *Έλεγχος μεγέθους γραφήματος*: αποτρέπει ακραίες τιμές που θα προκαλούσαν σφάλματα μνήμης ή χρονική ανωμαλία.

- *Τυχαιότητα*: η χρήση του ανακατέμματος (`shuffling`) μειώνει το ρίσκο συστηματικής μεροληψίας.

Πολυπλοκότητα. Όλες οι λειτουργίες (`filtering`, `shuffling`, splitting) είναι γραμμικές: $O(|\mathcal{D}|)$ σε χρόνο και $O(|\mathcal{D}|)$ σε μνήμη, καθώς αντιγράφουν δείκτες χωρίς επιπλέον αντιγραφή γραφημάτων.

5.5.11 Κλάση `computeGraphFeatureVecs`

Η κλάση `computeGraphFeatureVecs` συνθέτει το **τελικό σύνολο δεδομένων** που χρησιμοποιείται για την εκπαίδευση και δοκιμή των ταξινομητών. Συνθέτει τα διανύσματα χαρακτηριστικών από τα ακόλουθα τρία είδη πληροφορίας των γραφημάτων:

1. *ιστόγραμμα ετικετών κόμβων*,

2. *κατανομή υποδέντρων βάθους 1*

3. *κατανομή υποδέντρων βάθους 2*

1. Διαχωρισμός εκπαίδευσης–δοκιμής. Με χρήση της `trainingSetTestSetSplitter` θέτοντας παραμετρικά το ποσοστό διαχωρισμού σε σύνολο εκπαίδευσης και ελέγχου (`perc`) και την παράμετρο `maxSize`, παράγονται τα σύνολα $\mathcal{T}_{\text{train}}, \mathcal{T}_{\text{test}}$.

2. Παράγωγα καθολικά μεγέθη.

- **Μέγιστη ετικέτα** $\ell_{\max} = \max_{G \in \mathcal{T}_{\text{train}}} \left(\max_{v \in V(G)} \ell(v) \right)$
- **Μέγιστος βαθμός** $d = \max_{G \in \mathcal{T}_{\text{train}}} \left(\max_{v \in V(G)} \deg(v) \right)$.
- **Πίνακας κόστους TED** $\Sigma = \mathbf{1} - \mathbf{I}_{\ell_{\max} x \ell_{\max}}$ (χρησιμοποιείται στο SD-TED ως πίνακας κόστους μετατροπής μιας επιγραφής $l_i \longrightarrow l_j$).

3. Δημιουργία λεξικών υποδέντρων Στον *Γενικευμένο Πυρήνα Γραφημάτων* (*Generalised WL-Kernel*) τα δέντρα ξεδίπλωσης (*unfolding trees*) ομαδοποιούνται σε συστάδες (*clusters*) βάσει της δομικής τους ομοιότητας. Για να είναι εφικτή η γρήγορη αναζήτηση και ανάθεση ενός νέου υποδέντρου στη σωστή συστάδα, ορίζουμε δύο λεξικά (δομές αντιστοίχισης) τα οποία αντιστοιχίζουν το κωδικοποιημένο σε συμβολοσειρά δέντρο (*tree string encoding*) στον αντίστοιχο αναγνωριστικό κωδικό κάθε συστάδας.

1. **Βάθος 1.** Καλούμε τη ρουτίνα `compute1UnfTreeToClusterMap`, η οποία παράγει k_1 συστάδες. Ορίζεται έτσι η συνάρτηση

$$\mathcal{M}_1 : \text{oneUnfTreeToStr} \longrightarrow \{0, 1, \ldots, k_1 - 1\},$$

   που αντιστοιχεί κάθε κωδικοποίημένο δέντρο σε συμβολοσειρά βάθους 1 στον δείκτη της συστάδας του.

2. **Βάθος 2.** Με παρόμοιο τρόπο, η ρουτίνα `compute2UnfTreeToClusterMap` δημιουργεί k_2 συστάδες για τα δέντρα βάθους 2· επομένως προκύπτει η συνάρτηση

$$\mathcal{M}_2 : \text{twoUnfTreeToStr} \longrightarrow \{0, 1, \ldots, k_2 - 1\}.$$

Τα λεξικά $\mathcal{M}_1$ και $\mathcal{M}_2$ χρησιμοποιούνται σε όλα τα επόμενα στάδια του αλγορίθμου για: (i) τη γρήγορη ταυτοποίηση της συστάδας στην οποία ανήκει κάθε νέο υπό εξέταση υποδέντρο και (ii) την κατασκευή των τελικών διανυσμάτων χαρακτηριστικών, όπου η συμβολή κάθε υποδέντρου υπολογίζεται σύμφωνα με τη συστάδα του.

4. Υπολογισμός διανυσμάτων εκπαίδευσης. Για κάθε γράφημα $G \in \mathcal{T}_{\text{train}}$ εξάγεται το διάνυσμα

$$\mathbf{x}(G) = \underbrace{[\#\ell = 1, \ldots, \#\ell = \ell_{\max}]}_{\text{Label Histogram}} \parallel \underbrace{[\#\mathcal{M}_1^{(0)}, \ldots, \#\mathcal{M}_1^{(k_1-1)}]}_{\text{Depth-1}} \parallel \underbrace{[\#\mathcal{M}_2^{(0)}, \ldots, \#\mathcal{M}_2^{(k_2-1)}]}_{\text{Depth-2}}.$$

Οι μετρήσεις $\#\mathcal{M}_d^{(j)}$ προκύπτουν από την `oneUnfTreesVec()` και `computeTwoUnfTreesForATreeVec()`.

5. Υπολογισμός διανυσμάτων δοκιμής. Για το $\mathcal{T}_{\text{test}}$ ακολουθείται η ίδια διαδικασία Για ένα υποδέντρο T που δεν εμφανίζονται στα λεξικά υπολογίζεται ο **εγγύτερος γείτονας** του

δέντρου ανάπτυξης σύμφωνα με την σχέση:

$$\text{TED–NN}(T) = \arg \min_{S \in \mathcal{S}_1 \text{ ή } \mathcal{S}_2} \text{SDTED}(T, S).$$

Αν υπάρχουν πολλαπλά ελάχιστα, η συχνότητα κατανέμεται ομοιόμορφα.

6. Επιστροφή διανυσμάτων χαρακτηριστικών. Παράγονται οι λίστες `trainingSetDataSet` και `testSetDataSet`, όπου κάθε στοιχείο είναι ζεύγος $(\mathbf{x}(G), y)$ με $y \in 0, 1$(`true`$\rightarrow 1$, `false`$\rightarrow 0$).

Πολυπλοκότητα. Η πολυπλότητα αυτής της κλάσης είναι:

$$O(|\mathcal{T}_{\text{train}}|(\,|V| + \text{Unf}_1 + \text{Unf}_2) \;+\; |\mathcal{T}_{\text{test}}|\,\text{TED-NN}),$$

όπου Unf_d η πολυπλοκότητα παραγωγής και κατηγοριοποίησης υποδέντρων βάθους d.

5.5.12 Κλάση `FindNearestTrees`

Σκοπός. Δεδομένου ενός Δέντρου Ανάπτυξης T (unfolding tree) και ενός συνόλου δέντρων ανάπτυξης $\mathcal{U} = \{U_1, \ldots, U_N\}$, η κλάση `FindNearestTrees` επιστρέφει *όλα* δέντρα από το σύνολο $\mathcal{U}$ που έχουν την ελάχιστη Διορθωτική Απόσταση Δέντρων (Tree-Edit Distance (TED)) από το T, αξιοποιώντας μία διαδικασία *ταχύτερης αναζήτησης* με **ομαδοποίηση** (bucketing) βασισμένο στο πλήθος κόμβων.

Είσοδοι.

- unfTrees ($\mathcal{U}$) — λίστα δέντρων ανάπτυξης που υπολογίστηκαν στο σύνολο εκπαίδευσης.
- T — δέντρο ανάπτυξης που προκύπτει από το σύνολο ελέγχου και δεν ανήκει στο $\mathcal{U}$.
- CostMatrix — πίνακας κόστους για την μετατροπή των επιγραφών των κόμβων για τον υπολογισμό της Διορθωτικής Απόστασης Δέντρων (TED).

Κεντρική ιδέα.

1. **Ομαδοποίηση κατά μέγεθος.** Τα υποδέντρα τοποθετούνται σε *ομάδες* (buckets) με κλειδί το πλήθος κόμβων $|V(U_i)|$.
2. **Τοπική αναζήτηση.**

   (α′) Υπολογίζουμε πρώτα τη μικρότερη απόσταση δ_* μεταξύ T και των δέντρων της *ίδιας* ομάδας ($|V(T)|$).

   (β′) Αν η ομάδα είναι άδεια, αναζητούμε την *πλησιέστερη γειτονική* ομάδα (ως προς το μέγεθος του δέντρου).

   (γ′) Εάν βρεθεί δ_*, εξετάζουμε *μόνο* ομάδες των οποίων το μέγεθος ανήκει στο διάστημα $\left[\,|V(T)| - \delta_*,\, |V(T)| + \delta_*\right]$.

3. **Ελαχιστοποίηση.** Για κάθε υποψήφιο δέντρο U, υπολογίζουμε την απόσταση $\mathrm{TED}(T, U)$.
   - Εάν $\exists U$ τέτοιο ώστε $\mathrm{TED}(T, U) = \delta_*$, όπου $\delta_* = \min_U \mathrm{TED}(T, U)$, τότε κρατάμε όλα τα δέντρα U με $\mathrm{TED}(T, U) = \delta_*$.
   - Διαφορετικά, κρατάμε τα δέντρα U με τη μικρότερη διαθέσιμη απόσταση $\mathrm{TED}(T, U)$.

Σημαντικές μέθοδοι.

`createBucketing()` Ομαδοποιεί τα δέντρα σε λεξικό $\{|V| \mapsto \text{λίστα δέντρων}\}$.

`computeSmallestDistInTheSameBuck()` Υπολογίζει την ελάχιστη απόσταση δέντρων (TED) μέσα στην ομάδα $|V(T)|$.

`returnPossibleTrees()` Επιστρέφει *μόνο* τα δέντρα που πρέπει να εξεταστούν, με βάση τον κανόνα $|V(U)| \in [\,|V(T)| - \delta_*,\ |V(T)| + \delta_*\,]$.

`main()` Αποτελεί την κεντρική μέθοδο στην οποία εκτελούνται όλες οι απαιτούμενες διεργασίες παράγοντας τη λίστα των πλησιέστερων υποδέντρων.

Πολυπλοκότητα.

- *Δημιουργία ομάδων*: $O(N)$.
- *Αρχικός υπολογισμός* δ_* — το πολύ $B_{|V|}$ υπολογισμοί TED, όπου $B_{|V|}$ το πλήθος δέντρων στην ομάδα $|V(T)|$.
- *Τελική αναζήτηση:* το πολύ k_{cand} υποδέντρα από τις επιλεγμένες ομάδες, δηλαδή $O(k_{\text{cand}} \cdot \mathrm{TED})$ αντί της $O(N \cdot \mathrm{TED})$ σε περίπτωση ενός αλγορίθμου ωμής βίας (brute-force).

Με αυτό τον τρόπο βελτιστοποιούμε την αναζήτηση των πλησιέστερου ¨γείτονα¨ ενός δέντρου ανάπτυξης που προέκυψε στο σύνολο ελέγχου με όλα τα δέντρα που προέκυψαν στο σύνολο εκπαίδευσης.

5.5.13 Κλάση `ModelTrainer`

Η κλάση `ModelTrainer` υλοποιεί την τελική φάση της πειραματικής διαδικασίας, αποτελούμενη από τρία κύρια στάδια:

1. **Προεπεξεργασία δεδομένων**: Κανονικοποίηση των διανυσμάτων χαρακτηριστικών που παράγονται για το σύνολο εκπαίδευσης και το σύνολο δοκιμής.
2. **Εκπαίδευση μοντέλων**: Χρήση δύο διαφορετικών ταξινομητών:
   - Γραμμικός ταξινομητής (*Logistic Regression*)
   - Μη γραμμικός ταξινομητής (*Gradient Boosting Trees*)
3. **Αξιολόγηση απόδοσης**: Συστηματική μέτρηση των μετρικών απόδοσης και σύγκριση των δύο προσεγγίσεων

Σύνοψη ροής.

1. *Προετοιμασία δεδομένων* — `prepare_data()` — εξάγει πίνακες χαρακτηριστικών $\mathbf{X}_{\text{train}}, \mathbf{X}_{\text{test}}$ και ετικέτες $\mathbf{y}_{\text{train}}, \mathbf{y}_{\text{test}}$. Εφαρμόζει **τυπική κανονικοποίηση** (`StandardScaler`) ώστε κάθε συνιστώσα να έχει μηδενικό μέσο και μοναδιαία διασπορά.

2. *Εκπαίδευση μοντέλων* — `train_models()` —

   - **Logistic Regression**: υλοποίηση `scikit-learn`, *solver* LBFGS, προεπιλεγμένη ℓ_2 κανονικοποίηση.
   - **Gradient Boosting Trees**: 100 δέντρα, ρυθμός μάθησης 0.1, μέγιστο βάθος 3, κατά `Friedman MSE`.

3. *Αξιολόγηση* — `evaluate_models()` — παράγει **accuracy** και **classification report** (precision/recall/F_1) στο σύνολο δοκιμής, επιτρέποντας συγκριτική αξιολόγηση των δύο ταξινομητών.

Ανάλυση Πολυπλοκότητας Η συνολική υπολογιστική πολυπλοκότητα της μεθόδου δίνεται από:

$$O(n_{\text{train}} \cdot d + n_{\text{tree}} \cdot \text{depth} \cdot \log n_{\text{train}})$$

όπου:

- d: Διάσταση διανύσματος χαρακτηριστικών
- n_{train}: Πλήθος δειγμάτων εκπαίδευσης
- $n_{\text{tree}} = 100$: Πλήθος δέντρων στο μοντέλο GBT
- depth: Μέγιστο βάθος δέντρων (3)

Η φάση προεπεξεργασίας (κανονικοποίηση) εμφανίζει γραμμική πολυπλοκότητα $O(n_{\text{train}} \cdot d)$, με n_{train} να αντιπροσωπεύει τον αριθμό των δειγμάτων εκπαίδευσης και d τη διάσταση των χαρακτηριστικών. Ο αλγόριθμος λογιστικής παλινδρόμησης διατηρεί γραμμική πολυπλοκότητα ως προς το μέγεθος του συνόλου δεδομένων, ενώ ο αλγόριθμος GBT εκτιμάται με λογαριθμική πολυπλοκότητα $O(\log n_{\text{train}})$ λόγω της ιδιότητας διάσχισης δυαδικών δέντρων.

5.6 Σύνοψη

Στο παρόν κεφαλαιό αναλύθηκαν διεξοδικά όλες οι τεχνικές λεπτομέρειες που σχετίζονται με την υλοποίηση της προτεινόμενης γραφοθεωρητικής μεθόδου πρόγνωσης. Συγκεκριμένα, παρουσιάστηκαν οι βασικές δομές δεδομένων που χρησιμοποιήθηκαν, όπως τα κατευθυνόμενα δέντρα αναπαράστασης γραφημάτων διάδοσης και οι μέθοδοι διανυσματικής κωδικοποίησης και ομαδοποίησης των δέντρων ανάπτυξης. Επιπλέον, περιγράφηκαν αναλυτικά οι βιβλιοθήκες και τα εργαλεία λογισμικού που αξιοποιήθηκαν κατά την ανάπτυξη του συστήματος.

Ιδιαίτερη έμφαση δόθηκε στις σημαντικότερες προκλήσεις που προέκυψαν, όπως η διαχείριση μεγάλου όγκου δεδομένων, η αντιμετώπιση προβλημάτων ισομορφισμού γραφημάτων και η βελτιστοποίηση της υπολογιστικής πολυπλοκότητας των πυρήνων γραφημάτων. Επισημάνθηκαν οι μέθοδοι που εφαρμόστηκαν για την αποτελεσματική αντιμετώπισή τους, με έμφαση σε πρακτικές όπως η επεξεργασία δεδομένων σε ροή παρτίδων και η κανονικοποίηση των δομών.

Παρουσιάστηκαν, επίσης, οι κυριότερες κλάσεις που υλοποιήθηκαν στο πλαίσιο της διπλωματικής εργασίας, αναδεικνύοντας τις βασικές λειτουργίες και τον τρόπο που αλληλεπιδρούν μεταξύ τους. Οι κλάσεις αυτές καλύπτουν όλο το εύρος της διαδικασίας, από τη φόρτωση και την επεξεργασία των αρχικών δεδομένων μέχρι τη δημιουργία διανυσματικών αναπαραστάσεων, τη συσταδοποίηση και την τελική ταξινόμηση με σκοπό τον υπολογισμό των διανυσμάτων χαρακτηριστικών του συνόλου δεδομένων.

Κεφάλαιο 6

Πειραματική Αξιολόγηση

6.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφονται λεπτομερώς η διαδικασία και οι παράμετροι διεξαγωγής των πειραμάτων. Αρχικά, γίνεται ανάλυση των συνόλων δεδομένων Twitter15/Twitter16, με έμφαση στα χαρακτηριστικά και τη δομή τους. Ακολουθεί η παρουσίαση των πειραματικών ρυθμίσεων, συμπεριλαμβανομένων των τεχνικών προδιαγραφών του υλικού και του λειτουργικού συστήματος που αξιοποιήθηκαν.Στο τελικό μέρος, για κάθε προτεινόμενη μέθοδο, παρουσιάζονται τα αποτελέσματα μέσω πολλαπλών μετρικών αξιολόγησης, ενώ πραγματοποιείται συγκριτική ανάλυση των μεθόδων.

6.2 Σχεδίαση Πειραμάτων

6.2.1 Πειραματικά Δεδομένα

Στην ενότητα αυτή παρουσιάζεται αναλυτικά το υλικό που χρησιμοποιήθηκε στα πειράματα, τα δημόσια διαθέσιμα σύνολα δεδομένων **Twitter15** και **Twitter16** που εισήχθησαν από τους Ma *et al.* [11]. Τα δύο σύνολα θεωρούνται σήμερα *σημεία αναφοράς* στον τομέα της ανίχνευσης φημών και έχουν αξιοποιηθεί σε πληθώρα εργασιών–ενδεικτικά [44]

Πηγές

Τα πρωτογενή δεδομένα προέρχονται από τον δικτυακό ιστότοπο *Twitter*. Στην πράξη αξιοποιούμε το έτοιμο Σύνολο Δεδομένων [1], το οποίο περιέχει ήδη:

- τις αρχικές δημοσιεύσεις στο Twitter (**source tweets**)·
- όλες οι *αναδημοσιεύσεις, απαντήσεις και παραθέσεις* που συνθέτουν το *δέντρο διάδοσης* κάθε αρχικής δημοσίευσης·
- τα αρχεία μεταδεδομένων και τις ετικέτες εγκυρότητας.

[1] Διαθέσιμο στη διεύθυνση https://www.dropbox.com/s/7ewzdrbelpmrnxu/rumdetect2017.zip?dl=0

Στατιστικά Συνόλου Δεδομένων

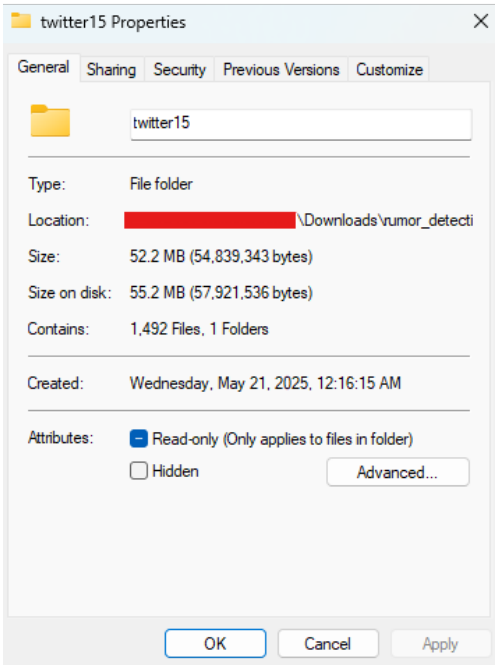
Τα κύρια χαρακτηριστικά των συνόλων δεδομένων *Twitter15/Twitter16* συνοψίζονται στον Πίνακα 6.1.

Πίνακας 6.1: *Βασικά στατιστικά των συνόλων Twitter15/Twitter16 [5].*

	Twitter15	**Twitter16**
Πλήθος δέντρων διάδοσης (cascades)	1490	818
Σύνολο δημοσιεύσεων (tweets)	331.612	204.820
Πλήθος Κανονικών Δημοσιεύσεων (non-rumours)	374	205
Πλήθος Ψευδών Δημοσιεύσεων (false-rumours)	370	205
Πλήθος Αληθών Δημοσιεύσεων (true-rumours)	372	207
Πλήθος μη-επιβεβαιωμένων Δημοσιεύσεων (unverified-rumours)	374	201

6.2.2 Απαιτήσεις Μνήμης

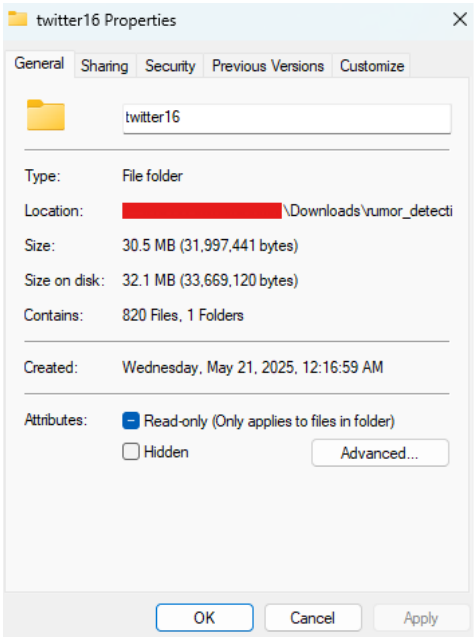
Σύνολο *Twitter15*. Το συνολικό μέγεθος του συνόλου δεδομένων *Twitter15*, όπως καταγράφεται από το λειτουργικό σύστημα, ανέρχεται σε περίπου 52,2 MB (*logical size*). Ο πραγματικός χώρος που καταλαμβάνεται στο δίσκο (*size on disk*) είναι μεγαλύτερος — περίπου 55,2 MB — λόγω των ιδιαιτεροτήτων διαχείρισης μπλοκ από το αρχείο συστήματος, καθώς και της στοίχισης (*block alignment*) σε αποθηκευτικά μέσα.



Σχήμα 6.1: *Απαιτήσεις Μνήμης Συνόλου Δεδομένων Twitter15.*

Σύνολο *Twitter16*. Αντίστοιχα, το σύνολο δεδομένων *Twitter16* έχει λογικό μέγεθος περίπου 30,5 MB και συνολικό αποτύπωμα στο δίσκο περίπου 32,1 MB. Η διαφορά αυτή,

όπως και στην περίπτωση του *Twitter15*, οφείλεται σε παραμέτρους που σχετίζονται με την εσωτερική οργάνωση του λειτουργικού συστήματος και δεν αντικατοπτρίζει απαραίτητα την πληροφοριακή πυκνότητα των αρχείων.



Σχήμα 6.2: *Απαιτήσεις Μνήμης Συνόλου Δεδομένων Twitter16.*

6.2.3 Περιγραφή Δομής Συνόλου Δεδομένων

Τα σύνολα δεδομένων *Twitter15* και *Twitter16* περιλαμβάνουν συλλογές από γεγονότα διάδοσης πληροφορίας (π.χ. φήμες, ειδήσεις), τα οποία δομούνται ως ιεραρχικές δομές διάδοσης. Κάθε γεγονός αντιστοιχεί σε ένα *δέντρο διάδοσης* (cascade tree) που προκύπτει από μία αρχική δημοσίευση και τις αντίστοιχες αλληλεπιδράσεις (αναδημοσιεύσεις, απαντήσεις) χρηστών στο δίκτυο *Twitter*.

Η γενική δομή των αρχείων των συνόλων δεδομένων απεικονίζεται στο Σχήμα 6.3.

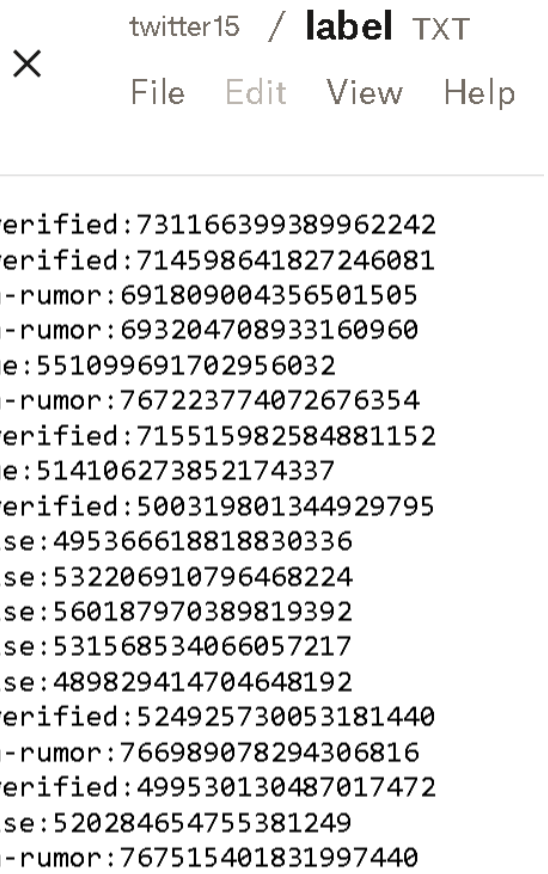
Σχήμα 6.3: *Δομή φακέλων του Συνόλου Δεδομένων Twitter15.*

Αρχείο `label.txt`. Το αρχείο αυτό περιέχει αντιστοιχίσεις μεταξύ μοναδικών αναγνωριστικών δημοσιεύσεων (`tweetIds`) και της κατάστασης εγκυρότητας του εκάστοτε γεγονότος, με

βάση τις επισημάνσεις ειδικών ή διασταυρωμένων πηγών (Σχήμα 6.4). Οι κατηγορίες που εμφανίζονται είναι οι εξής:

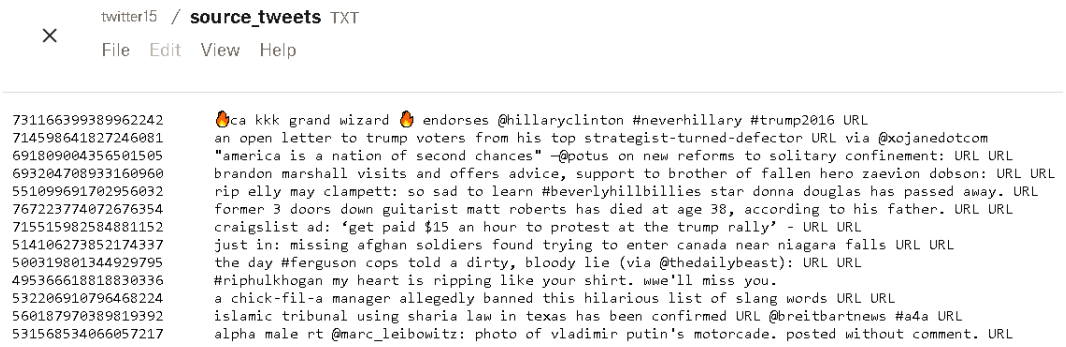
1. **true**: επαληθευμένη είδηση
2. **false**: αποδεδειγμένα ψευδής πληροφορία
3. **non-rumor**: γενική πληροφορία χωρίς χαρακτηριστικά φήμης
4. **unverified**: πληροφορία μη επιβεβαιωμένης εγκυρότητας

Οι ετικέτες αυτές αξιοποιούνται για την επίβλεψη της ταξινόμησης κατά την εκπαίδευση μοντέλων.



Σχήμα 6.4: *Ενδεικτική εγγραφή του αρχείου* `label.txt`.

Αρχείο `source_tweets.txt`. Περιέχει το περιεχόμενο (κείμενο) κάθε δημοσίευσης-πηγής, δηλαδή των αρχικών tweets. Κάθε εγγραφή συνοδεύεται από το μοναδικό `tweetId` και το πλήρες κείμενο της δημοσίευσης. Άλλα μεταδεδομένα, όπως το `userId` του χρήστη που το δημοσίευσε, βρίσκονται σε άλλα σχετικά αρχεία του συνόλου δεδομένων. Ωστόσο, στο πλαίσιο της παρούσας εργασίας, το εν λόγω αρχείο δεν χρησιμοποιείται, καθώς η προσέγγισή μας βασίζεται αποκλειστικά σε δομικά χαρακτηριστικά και όχι σε λεκτικό περιεχόμενο [43].

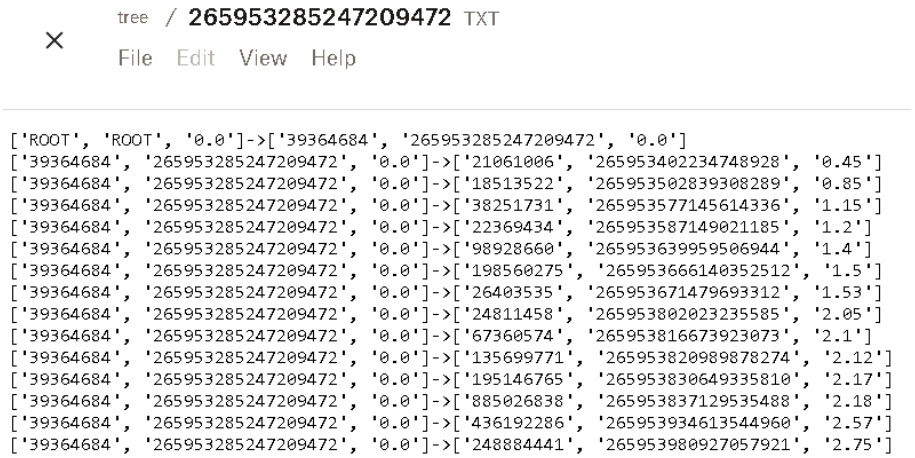


Σχήμα 6.5: *Παράδειγμα εγγραφής στο αρχείο* `source_tweets.txt`.

Φάκελος `tree/`. Ο φάκελος `tree/` περιλαμβάνει για κάθε `tweetId` ένα ξεχωριστό αρχείο κειμένου, στο οποίο περιγράφονται οι επιμέρους πληροφορίες που σχετίζονται με τη διάδοση της δημοσίευσης. Κάθε γραμμή του αρχείου αντιπροσωπεύει μια ζεύξη μετάδοσης πληροφορίας μεταξύ δύο χρηστών, σύμφωνα με τη μορφή:

1. `userId` του χρήστη-πηγή,
2. `tweetId` της ανάρτησης,
3. χρονική διαφορά από τη δημοσίευση της προηγούμενης ανάρτησης (σε λεπτά).

Η πρώτη γραμμή υποδηλώνει την αρχική δημοσίευση, με χρονική σήμανση ίση με μηδέν. Στις επόμενες γραμμές, κάθε αναδημοσίευση (ή απάντηση) αναπαρίσταται ως διάδοχος της αρχικής πηγής ή άλλου ενδιάμεσου χρήστη. Η δομή αυτή καθιστά εφικτή την κατασκευή κατευθυνόμενων δέντρων διάδοσης (*propagation graphs*), τα οποία χρησιμοποιούνται ως είσοδος στη γραφοθεωρητική ανάλυση.



Σχήμα 6.6: *Δομή ενός αρχείου από τον φάκελο* `tree/`. *Κάθε γραμμή απεικονίζει μια σχέση διάδοσης.*

Σχήμα 6.7: *Δομή αρχείου του φακέλου* `tree/`. *Κάθε γραμμή απεικονίζει μια σχέση διάδοσης.*

Προεπεξεργασία και φιλτράρισμα. Για τις ανάγκες της παρούσας εργασίας, λαμβάνονται υπόψιν μόνο τα γεγονότα που φέρουν ετικέτα `true` ή `false`, προκειμένου να διατηρηθεί το πρόβλημα σε μορφή δυαδικής ταξινόμησης. Τα γεγονότα με ετικέτα `non-rumor` ή `unverified` παραλείπονται.

Συνολικά, η δομή των δεδομένων παρέχει μια πλήρη απεικόνιση της χρονικής και δομικής εξέλιξης κάθε γεγονότος στο κοινωνικό δίκτυο, παρέχοντας το απαραίτητο υπόβαθρο για την εφαρμογή τεχνικών ανάλυσης βάσει γραφημάτων. Τα χρονικά σήματα ωστόσο παραλείπονται ως πληροφορία στη μελέτης μας.

6.2.4 Πειραματικές Ρυθμίσεις

Διαμέριση Εκπαίδευσης–Δοκιμής

Για την ποσοτική αξιολόγηση των μοντέλων πραγματοποιήθηκε **διαδοχική διαμέριση** του αρχικού συνόλου δεδομένων σε δύο μη επικαλυπτόμενα τμήματα: *σύνολο εκπαίδευσης* και *σύνολο δοκιμής*. Το ποσοστό των παραδειγμάτων που αποδίδεται στο σύνολο εκπαίδευσης μεταβάλλεται από 50 % έως 90 %, με βήμα 10 %.

Ο κύριος σκοπός αυτής της παραμετροποίησης είναι η διερεύνηση της *ευαισθησίας της ακρίβειας* ως προς τη διαθέσιμη ποσότητα εκπαιδευτικών δειγμάτων. Με άλλα λόγια, μετράται η σταθερότητα του μοντέλου όταν του παρέχονται προοδευτικά περισσότερα σύνολα δεδομένων για την εκπαίδευση του μοντέλου. Για κάθε ποσοστό p εφαρμόζεται η εξής διαδικασία:

1. **Τυχαίος δειγματοληπτικός διαχωρισμός** ($p\%$): επιλέγονται τυχαία $p\%$ των γεγονότων για εκπαίδευση, ενώ τα υπόλοιπα $(100 - p)\%$ χρησιμοποιούνται αποκλειστικά για τον έλεγχο.

2. **Εκπαίδευση μοντέλου**: το μοντέλο προσαρμόζεται στα δεδομένα του συνόλου εκπαίδευσης.

3. **Αξιολόγηση**: υπολογίζονται μετρικές απόδοσης (*accuracy*, *precision*, *recall*, $F_1-score$) στο σύνολο δοκιμής.

Χρησιμοποιηθέντες Ταξινομητές

1. **Λογιστική Παλινδρόμηση (Logistic Regression)** Ένα γραμμικό μοντέλο μέγιστης πιθανοφάνειας, το οποίο υπολογίζει την πιθανότητα $P(y = 1 \mid \mathbf{x})$ μέσω της *σιγμοειδούς συνάρτησης* (sigmoid function):

$$P(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b), \qquad \sigma(z) = \tfrac{1}{1+e^{-z}}.$$

   Χρησιμοποιήθηκε κανονικοποίηση ℓ_2 με αντίστροφη παράμετρο $C \in \{0.1, 1, 10\}$, επιλεγμένη με την *Μέθοδος Αναζήτησης Πλέγματος* (grid search) σε δεκαπλάσια διασταυρούμενη επικύρωση. Η βελτιστοποίηση πραγματοποιήθηκε με τον αλγόριθμο *LBFGS* (*Limited-memory Broyden-Fletcher-Goldfarb-Shanno*), όπως υλοποιείται στη βιβλιοθήκη `scikit-learn` (έκδοση 1.4).

2. **Δέντρα Ενίσχυσης Βαθμίδας (Gradient Boosting Trees)** Μη γραμμικός ταξινομητής που κατασκευάζει ένα σύνολο ασθενών δέντρων απόφασης

$$f_M(\mathbf{x}) = \textstyle\sum_{m=1}^{M} \gamma_m h_m(\mathbf{x})$$

   με διαδοχική ελαχιστοποίηση της λογιστικής συνάρτησης κόστους. Οι θεμελιώδεις υπερπαράμετροι αναζητήθηκαν στο πεδίο $M \in \{100, 200, 400\}$ (αριθμός δέντρων), *ρυθμός μάθησης* (learning rate) $\eta \in \{0.05, 0.1\}$, και *μέγιστο βάθος* (max depth) $\in \{3, 4\}$. Ενεργοποιήθηκε *πρόωρος τερματισμός* (early stopping) με παράθυρο 20 εποχών για αποφυγή υπερπροσαρμογής (overfitting).

Τα παραπάνω μοντέλα επιλέχθηκαν για τους εξής λόγους:

- *Συγκρισιμότητα γραμμικής vs μη γραμμικής απόφασης*: η λογιστική παλινδρόμηση λειτουργεί ως βασικό γραμμικό μοντέλο, ενώ το Gradient Boosting ανιχνεύει μη γραμμικές συσχετίσεις μεταξύ των γραφοθεωρητικών χαρακτηριστικών.
- *Στατιστική ερμηνευσιμότητα*: ο συντελεστής $\mathbf{w}$ της λογιστικής παλινδρόμησης επιτρέπει ανάλυση σημαντικότητας χαρακτηριστικών, χρήσιμη για την κατανόηση της ευαισθησίας του συστήματος.
- *Αποδεδειγμένη αποτελεσματικότητα σε παρόμοια έργα*, όπου τα δύο μοντέλα έχουν χρησιμοποιηθεί ευρέως για την ανίχνευση ψευδών ειδήσεων [43].

Μέθοδοι Μέτρησης Απόδοσης

Για την ποσοτική αξιολόγηση τόσο των ταξινομητικών μοντέλων όσο και των αλγοριθμικών μεθόδων μετασχηματισμού δέντρων διάδοσης σε διανυσματικές αναπαραστάσεις, εφαρμόστηκε ένα πλήρες σύστημα μετρικών. Αυτό το σύστημα επιτρέπει:

- Την εκτίμηση της συνολικής προγνωστικής ικανότητας (overall accuracy)
- Την ανάλυση της απόδοσης ανά κλάση.

1. **Ακρίβεια (Accuracy)** Εκφράζει το ποσοστό των σωστών προβλέψεων επί του συνόλου των δειγμάτων, παρέχοντας μια συνολική εικόνα της επίδοσης.

2. F_1**-μέτρο (F1-score)** Αποτελεί τον αρμονικό μέσο μεταξύ ακρίβειας (*precision*) και ευαισθησίας (*recall*), εξισορροπώντας τις δύο ποσότητες σε μία ενιαία τιμή. Είναι ιδιαίτερα χρήσιμη μετρική σε περιπτώσεις όπου το κόστος των ψευδώς θετικών και των ψευδώς αρνητικών δεν είναι συμμετρικό.

3. **Πίνακας Σύγχυσης (Confusion Matrix)** Παρουσιάζει αναλυτικά τα αληθώς θετικά, ψευδώς θετικά, αληθώς αρνητικά και ψευδώς αρνητικά, διευκολύνοντας τον εντοπισμό συστηματικής μεροληψίας του μοντέλου υπέρ ή κατά κάποιας κλάσης.

4. **Ευαισθησία, Εξειδίκευση και Ακρίβεια** *Ευαισθησία (recall)*: ικανότητα ανίχνευσης των πραγματικά θετικών περιπτώσεων (π.. ψευδών ειδήσεων). *Εξειδίκευση*: ικανότητα σωστής αναγνώρισης των πραγματικά αρνητικών (π.. αληθών ειδήσεων). *Ακρίβεια (precision)*: ποσοστό των προβλέψεων που χαρακτηρίστηκαν ως θετικές και ήταν πράγματι θετικές.

Ο συνδυασμός των παραπάνω μετρικών επιτρέπει μια συνολική αποτίμηση, διασφαλίζοντας ότι η τελική επιλογή μοντέλου βασίζεται όχι μόνο σε μία συνολική τιμή, αλλά και στην ικανότητα του συστήματος να χειρίζεται σωστά καθεμία από τις διαφορετικές κατηγορίες.

Υλικό & Λειτουργικό Σύστημα.

- **CPU**: AMD EPYC 7552 48-Core Processor
  (48 πυρήνες)

- **RAM**: 251GB DDR4
  (Χρησιμοποιείται 17GB, διαθέσιμα 231GB)

- **Αποθήκευση**:
  - Κύριος δίσκος: ST4000NM017A 3.6TB (HDD)

- **Λειτουργικό**: Ubuntu 22.04.3 LTS (Jammy Jellyfish)
  Πυρήνας: 5.15.x (default for Ubuntu 22.04)

6.3 Αξιολόγηση και Πειραματικά Αποτελέσματα

6.3.1 Αποτελέσματα Βασικής Προσέγγισης

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα που προκύπτουν από τη χρήση απλών δομικών χαρακτηριστικών των δέντρων διάδοσης (baseline features) για την ανίχνευση ψευδών ειδήσεων, μέσω των κλασικών ταξινομητών Logistic Regression και Gradient

Boosting Trees. Τα χαρακτηριστικά που εξετάζονται περιλαμβάνουν την ακολουθία βαθμών (degree sequence), κανονικοποιημένες εκδοχές αυτής, καθώς και συνδυασμούς βασικών δομικών μετρικών (μέγιστο βάθος, μέγιστο πλάτος, δομική ιογένεια).

Διανύσματα Ακολουθίας Βαθμών. Η κατανομή των βαθμών των κόμβων κάθε δέντρου αναπαρίσταται ως ένα διανυσματικό χαρακτηριστικό 12 διαστάσεων. Η κατασκευή αυτού του διανύσματος πραγματοποιείται ως εξής: κάθε κόμβος ταξινομείται σε μία από 12 προκαθορισμένες κατηγορίες, ανάλογα με τον βαθμό του. Συγκεκριμένα, οι κατηγορίες είναι οι εξής:

- Βαθμοί 1–9: κάθε βαθμός αντιστοιχεί σε μία διάσταση (θέσεις 1 έως 9).
- Βαθμοί 10–99: τοποθετούνται στη 10η διάσταση.
- Βαθμοί 100–999: τοποθετούνται στην 11η διάσταση.
- Βαθμοί μεγαλύτεροι ή ίσοι του 1000: συγκεντρώνονται στη 12η διάσταση.

Η παραπάνω διαδικασία έχει ως αποτέλεσμα την παραγωγή ενός διανύσματος συχνοτήτων, που εκφράζει την κατανομή των βαθμών του κάθε δέντρου. Το τελικό σύνολο δεδομένων που χρησιμοποιείται για την εκπαίδευση των ταξινομητών δημιουργείται προσθέτοντας σε κάθε διάνυσμα μία τελική συνιστώσα που αντιστοιχεί στην ετικέτα της κλάσης (1 για `true`, 0 για `false`). Οι επιδόσεις των ταξινομητών συνοψίζονται στον Πίνακα 6.2.

Πίνακας 6.2: *Επιδόσεις βασισμένες στην ακολουθία βαθμών.*

Μοντέλο	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.53	0.55	0.63	0.59
Gradient Boosting Trees	0.55	0.59	0.51	0.55

Βελτιστοποίηση ακολουθίας βαθμών. Στην παρούσα πειραματική διαδικασία διερευνάται η επίδραση του μήκους της διανυσματικής αναπαράστασης της ακολουθίας βαθμών των κόμβων κάθε γραφήματος στην ακρίβεια των μοντέλων μηχανικής μάθησης. Πιο συγκεκριμένα, για κάθε γράφημα, υπολογίζεται η ταξινομημένη σε φθίνουσα σειρά ακολουθία των βαθμών των κόμβων, επιλέγοντας τα πρώτα n στοιχεία. Δημιουργείται έτσι ένα διάνυσμα χαρακτηριστικών σταθερού μήκους n, που χρησιμοποιείται για την εκπαίδευση και αξιολόγηση των μοντέλων ταξινόμησης.

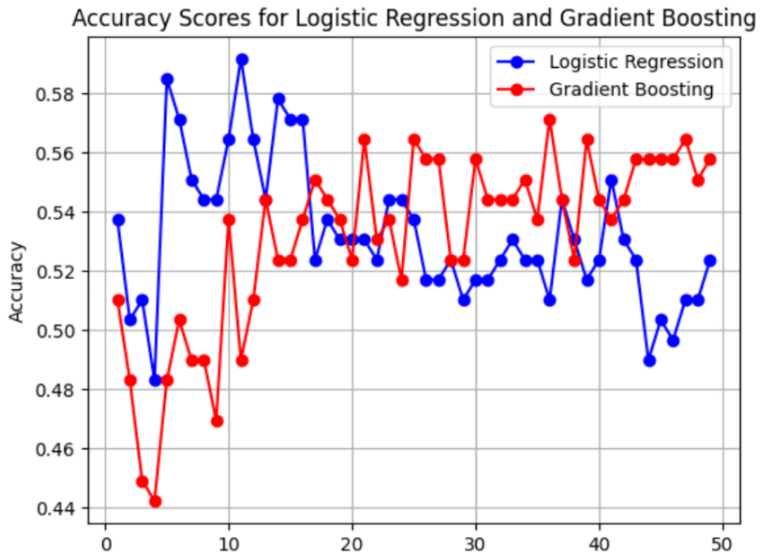
Η παραπάνω διαδικασία επαναλήφθηκε για εύρος τιμών $n \in \{1, \ldots, 50\}$, και για κάθε περίπτωση καταγράφηκε η ακρίβεια των ταξινομητών (Logistic Regression και Gradient Boosting Trees). Στον Πίνακα 6.3 παρουσιάζονται οι βέλτιστες ακρίβειες που επιτεύχθηκαν, καθώς και οι αντίστοιχες τιμές του μήκους n.

Παρατηρούμε ότι η μεγαλύτερη ακρίβεια για το μοντέλο Logistic Regression επιτυγχάνεται για $n = 11$, ενώ για το μοντέλο Gradient Boosting Trees για $n = 17$. Σημειώνεται επίσης ότι, πέρα από τα σημεία αυτά, η περαιτέρω αύξηση του μήκους του διανύσματος δεν επιφέρει σημαντική βελτίωση της ακρίβειας. Επιπλέον, στο Σχήμα 6.8 παρουσιάζεται το διάγραμμα

Πίνακας 6.3: *Καλύτερες ακρίβειες ως προς το μήκος ακολουθίας βαθμών.*

Μοντέλο	**Βέλτιστη ακρίβεια**	**Μήκος διανύσματος (n)**
Logistic Regression	0.592	11
Gradient Boosting Trees	0.571	17

της ακρίβειας των δύο μοντέλων συναρτήσει του μήκους διανύσματος της ακολουθίας βαθμών.

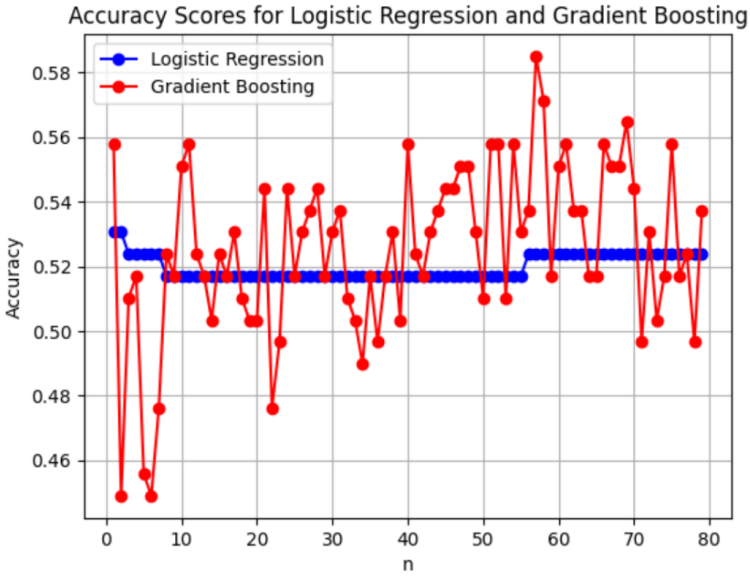


Σχήμα 6.8: *Ακρίβεια των μοντέλων συναρτήσει του μήκους διανύσματος της ακολουθίας βαθμών.*

Κανονικοποιημένη ακολουθία βαθμών. Στην περίπτωση των κανονικοποιημένων τιμών βαθμών, κάθε βαθμός κόμβου κανονικοποιείται διαιρώντας με τον μέγιστο δυνατό συνολικό βαθμό του δέντρου, που ισούται με 2.V.-2, όπου .V. είναι το πλήθος των κόμβων του γραφήματος. Έτσι, το χαρακτηριστικό διάνυσμα αποτελείται από τις πρώτες μεγαλύτερες κανονικοποιημένες τιμές βαθμών, ταξινομημένες σε φθίνουσα σειρά. Οι μέγιστες τιμές ακρίβειας για τα δύο μοντέλα παρουσιάζονται στον Πίνακα 6.4. Παρατηρείται ότι η βέλτιστη ακρίβεια του Gradient Boosting Trees (0.585) επιτυγχάνεται για μεγάλο μήκος διανύσματος (n=57), ενώ για το Logistic Regression (0.531) η βέλτιστη ακρίβεια εμφανίζεται με μόλις n=1.

Πίνακας 6.4: *Καλύτερες ακρίβειες για κανονικοποιημένη ακολουθία βαθμών.*

Μοντέλο	**Βέλτιστη ακρίβεια**	**Μήκος διανύσματος (n)**
Logistic Regression	0.531	1
Gradient Boosting Trees	0.585	57



Σχήμα 6.9: *Ακρίβεια των μοντέλων συναρτήσει του μήκους διανύσματος της κανονικοποιημένης ακολουθίας βαθμών.*

Το Σχήμα 6.9 παρουσιάζει αναλυτικά την εξέλιξη της ακρίβειας των δύο μοντέλων σε σχέση με το μήκος του διανύσματος της κανονικοποιημένης ακολουθίας βαθμών.

Βασικές δομικές μετρικές. Η χρήση τριών κλασικών μετρικών—κανονικοποιημένο μέγιστο βάθος, κανονικοποιημένο μέγιστο πλάτος και δομική ικότητα (structural virality)—δίνει παρόμοιες επιδόσεις στους δύο ταξινομητές:

Πίνακας 6.5: *Επιδόσεις με συνδυασμό βασικών δομικών χαρακτηριστικών.*

Μοντέλο	**Accuracy**
Logistic Regression	0.551
Gradient Boosting Trees	0.551

Μεμονωμένες εξειδικευμένες μετρικές. Για παράδειγμα, το χαρακτηριστικό «ελάχιστος αριθμός κόμβων που καλύπτουν άθροισμα βαθμών τουλάχιστον $n - 1$» (όπου n το πλήθος κόμβων) όταν χρησιμοποιείται για την κωδικοποίηση δέντρων ανάπτυξης σε ένα διάνυσμα μιας διάστασης παρουσιάζει τις εξής επιδόσεις:

Πίνακας 6.6: *Επιδόσεις για μεμονωμένη εξειδικευμένη μετρική.*

Μοντέλο	**Accuracy**
Logistic Regression	0.565
Gradient Boosting Trees	0.449

Σχολιασμός αποτελεσμάτων. Τα παραπάνω αποτελέσματα δείχνουν ότι ακόμη και απλά δομικά χαρακτηριστικά επιτρέπουν διαχωρισμό της τάξης του 55–59% (*accuracy*), με ελαφρώς καλύτερες επιδόσεις να επιτυγχάνονται με κατάλληλα διανύσματα ακολουθίας βαθμών

ή συνδυασμούς μετρικών. Το μοντέλο Gradient Boosting Trees υπερτερεί οριακά του Logistic Regression στη βέλτιστη τιμή κανονικοποιημένης ακολουθίας βαθμών. Έτσι, η βασική προσέγγιση (baseline approach) παρέχει μία ισχυρή γραμμή αναφοράς.

6.3.2 Αποτελέσματα Απλού Πυρήνα Weisfeiler-Lehman

Παρουσιάζονται τα αποτελέσματα του απλού πυρήνα γραφημάτων Weisfeiler-Lehman για ποσοστό συνόλου δεδομένων εκπαίδευσης 0.7% και χρήση ολόκληρου του συνόλου δεδομένων και πλήθος επαναλήψεων του αλγορίθμου h = 2.

Πίνακας 6.7: *Αποτελέσματα απλού πυρήνα Weisfeiler-Lehman (WL Kernel), $h = 2$.*

Μοντέλο	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.556	0.56	0.56	0.56
Gradient Boosting Trees	0.498	0.50	0.50	0.50

Όπως παρατηρείται, η χρήση του απλού πυρήνα Weisfeiler-Lehman οδηγεί σε παρόμοια αποτελέσματα με τις βασικές δομικές μετρικές που παρουσιάστηκαν στην προηγούμενη ενότητα. Η μέγιστη ακρίβεια που επιτυγχάνεται είναι περίπου 56% για Logistic Regression, ενώ οι υπόλοιπες μετρικές (precision, recall, f1-score) κυμαίνονται γύρω από 0.5. Συνεπώς, ο βασικός πυρήνας WL δεν προσφέρει ουσιαστική βελτίωση ως προς τη βασική προσέγγιση (baseline) για το συγκεκριμένο σύνολο δεδομένων και τη συγκεκριμένη πειραματική διαδικασία.

6.3.3 Αποτελέσματα Γενικευμένου Πυρήνα Weisfeiler-Lehman

Στον Πίνακα 6.8 παρουσιάζονται οι βέλτιστες επιδόσεις ταξινόμησης που επιτεύχθηκαν με τον γενικευμένο πυρήνα Weisfeiler-Lehman, για διάφορους συνδυασμούς των παραμέτρων:

- k_1: Πλήθος συστάδων 1- Δέντρων Ανάπτυξης.
- k_2: Πλήθος συστάδων 2- Δέντρων Ανάπτυξης.
- Ποσοστό στοιχείων εκπαίδευσης.
- Μέγιστο πλήθος κόμβων στο σύνολο δεδομένων.

Πίνακας 6.8: *Αποτελέσματα ταξινόμησης με τον γενικευμένο πυρήνα Weisfeiler-Lehman*

Παράμετροι	**Μοντέλο**	**Accuracy**	**F1 (0)**	**F1 (1)**	**Macro F1**	**N**
2_2_0.5_250	LR	0.56	0.55	0.57	0.56	227
2_2_0.5_250	GBT	0.54	0.47	0.59	0.53	227
3_3_0.8_559	LR	0.57	0.58	0.56	0.57	126
5_10_0.5_559	GBT	0.53	0.53	0.52	0.53	316
10_10_0.7_559	GBT	0.56	0.58	0.53	0.56	190

LR: Logistic Regression, GBT : Gradient Boosting Tree

N: Πλήθος δειγμάτων

Βασικά συμπεράσματα από την ανάλυση:

- Η Λογιστική Παλινδρόμηση παρουσιάζει ελαφρώς καλύτερες επιδόσεις (μέσο F1 0.56-0.57) σε μικρότερες παραμετρικές ρυθμίσεις (2_2_0.5_250, 3_3_0.8_559)
- Τα Gradient Boosting Trees εμφανίζουν παρόμοια αποτελέσματα (F1 0.52-0.56) σε μεγαλύτερα σύνολα δεδομένων
- Η αύξηση της πολυπλοκότητας (10_10_*) δεν φαίνεται να προσφέρει σημαντική βελτίωση της προβλεπτικής ικανότητας του μοντέλου.

6.4 Συγκριτική Ανάλυση

6.4.1 Σύγκριση Μεταξύ Προσεγγίσεων

Σε αυτό το τμήμα πραγματοποιείται συγκριτική ανάλυση των τριών μεθόδων που εφαρμόστηκαν: (α) των βασικών δομικών χαρακτηριστικών , (β) του απλού πυρήνα Weisfeiler-Lehman, και (γ) του γενικευμένου πυρήνα πυρήνα γραφημάτων Weisfeiler-Lehman. Η σύγκριση εστιάζει στην ακρίβεια ταξινόμησης, στην ευστάθεια των αποτελεσμάτων και στην υπολογιστική αποδοτικότητα κάθε προσέγγισης.

Πίνακας 6.9: *Συγκριτικός πίνακας των τριών μεθόδων ως προς την ακρίβεια ταξινόμησης (accuracy) και το κόστος υπολογισμού.*

Μέθοδος	**Logistic Regression**	**Gradient Boosting**	**Υπολογιστικό Κόστος**
Βασικά χαρακτηριστικά	0.55	0.55	Χαμηλό
Ακολουθία βαθμών	0.59	0.57	Πολύ Χαμηλό
Weisfeiler-Lehman	0.55	0.50	Μέτριο
Γενικευμένος Weisfeiler-Lehman	0.57	0.56	Υψηλό

Αποτελεσματικότητα. Η μέθοδος της **ακολουθίας βαθμών** πέτυχε την υψηλότερη ακρίβεια τόσο με το μοντέλο της Λογιστικής Παλινδρόμησης όσο και με το μοντέλο Gradient Boosting Trees, υποδηλώνοντας ότι τα χαρακτηριστικά που βασίζονται στην κατανομή των βαθμών των κόμβων είναι ιδιαίτερα διακριτικά για το συγκεκριμένο πρόβλημα.

Ευστάθεια. Παρατηρήθηκε ότι η απόδοση της μεθόδου ακολουθίας βαθμών παραμένει σχετικά σταθερή σε διαφορετικές διαμερίσεις συνόλων εκπαίδευσης και ελέγχου, ενώ οι μέθοδοι των βασικών χαρακτηριστικών και του Weisfeiler-Lehman παρουσιάζουν μεγαλύτερες διακυμάνσεις.

Υπολογιστική Αποδοτικότητα. Η βασική προσέγγιση είναι η πλέον αποδοτική υπολογιστικά, καθώς τα χαρακτηριστικά που υπολογίζονται (μέγιστο βάθος, πλήθος φύλλων κ.ά.) εξάγονται άμεσα με μικρό υπολογιστικό κόστος. Αντίθετα, η υλοποίηση του πυρήνα γραφημάτων Weisfeiler-Lehman απαιτεί περισσότερους υπολογιστικούς πόρους, κυρίως λόγω της αναδρομικής ενημέρωσης των ετικετών στους κόμβους και του υπολογισμού των διανυσμάτων χαρακτηριστικών του συνόλου ελέγχου με χρήση της διορθωτικής απόστασης

δέντρων. Ακόμα υπολογιστικά ακριβότερος είναι ο γενικευμένος πυρήνας γραφημάτων Weisfeiler-Lehman που χρησιμοποιεί την υπολογιστικά ακριβή απόσταση Wasserstein στη διαδικασία που πραγματοποιεί συσταδοποίηση των δέντρων ανάπτυξης καθώς και ενδιάμεσους αλγορίθμους γραφημάτων για την παραγωγή των διανυσμάτων χαρακτηριστικών που χρησιμοποιούνται ως είσοδοι για το μοντέλο δυαδική ταξινόμησης.

Συμπέρασμα. Συνολικά, η μέθοδος που βασίζεται στην ακολουθία βαθμών συνδυάζει ικανοποιητική διακριτική ικανότητα με υπολογιστική αποδοτικότητα και ευστάθεια αποτελεσμάτων, καθιστώντας την προτιμητέα για το πρόβλημα της ανίχνευσης παραπληροφόρησης βάσει δέντρων διάδοσης.

6.5 Σύνοψη

Στο κεφάλαιο αυτό πραγματοποιήθηκε μια εκτενής πειραματική αξιολόγηση μεθόδων που βασίζονται σε γραφήματα για την ταξινόμηση της διάδοσης πληροφορίας στα σύνολα δεδομένων Twitter15 και Twitter16. Στόχος ήταν η σύγκριση της αποτελεσματικότητας απλών δομικών χαρακτηριστικών έναντι πιο σύνθετων τεχνικών, όπως οι πυρήνες γραφημάτων Weisfeiler-Lehman.

Τα κύρια ευρήματα της αξιολόγησης είναι τα εξής:

1. **Υπεροχή των Βασικών Χαρακτηριστικών**: Η προσέγγιση που βασίζεται στην **ακολουθία βαθμών** των κόμβων ενός δέντρου διάδοσης αποδείχθηκε η πιο αποτελεσματική. Συγκεκριμένα, πέτυχε την υψηλότερη ακρίβεια ταξινόμησης, φτάνοντας το **59,2 %** με τον ταξινομητή Logistic Regression, ξεπερνώντας τόσο τις υπόλοιπες βασικές μετρικές όσο και τις πιο πολύπλοκες μεθόδους πυρήνων γραφημάτων.

2. **Περιορισμένη Απόδοση των Πυρήνων Γραφημάτων**: Ο απλός πυρήνας Ωεισφειλερ-Λεημαν παρουσίασε χαμηλή απόδοση, γεγονός που υποδηλώνει ότι η βασική του εκδοχή δεν είναι επαρκής για να συλλάβει τις διακριτικές δομικές διαφορές μεταξύ αληθών και ψευδών ειδήσεων. Ο γενικευμένος πυρήνας, αν και ελαφρώς βελτιωμένος, δεν κατάφερε να ξεπεράσει την προσέγγιση της ακολουθίας βαθμών, επιτυγχάνοντας μέγιστη ακρίβεια περίπου 57 %.

3. **Υπολογιστικό Κόστος και Αποδοτικότητα**: Παρατηρήθηκε μια σαφής αντιστάθμιση μεταξύ πολυπλοκότητας και απόδοσης. Οι μέθοδοι βασικών χαρακτηριστικών, και ιδίως η ακολουθία βαθμών, είναι υπολογιστικά πολύ αποδοτικές. Αντίθετα, οι πυρήνες γραφημάτων, ειδικά ο γενικευμένος, εισάγουν σημαντικό υπολογιστικό κόστος χωρίς να προσφέρουν ανάλογη βελτίωση στην προβλεπτική ικανότητα.

Συμπερασματικά, για το συγκεκριμένο πρόβλημα ταξινόμησης, η μελέτη καταδεικνύει ότι μια απλή, αλλά καλά σχεδιασμένη, γραφοθεωρητική αναπαράσταση, όπως η κατανομή των βαθμών των κόμβων, περιέχει ισχυρότερη διακριτική πληροφορία από τις πιο σύνθετες και υπολογιστικά απαιτητικές τεχνικές πυρήνων γραφημάτων.

Κεφάλαιο 7

Συμπεράσματα και Προοπτικές

7.1 Συμπεράσματα

"What we observe is not nature itself,
but nature exposed to our method of questioning."
Werner Heisenberg

Η παρούσα διπλωματική εργασία ξεκίνησε με την υπόθεση ότι η δομή της διάδοσης μιας είδησης στα κοινωνικά δίκτυα φέρει κρυμμένη πληροφορία για την εγκυρότητά της. Για να διερευνηθεί αυτή η υπόθεση, αναλύθηκαν τα δέντρα διάδοσης από τα σύνολα δεδομένων Twitter15 και Twitter16, μετατρέποντάς τα σε διανυσματικές αναπαραστάσεις κατάλληλες για μοντέλα δυαδικής ταξινόμησης.

Η κεντρική ερευνητική στρατηγική περιλάμβανε τη σύγκριση δύο διαφορετικών μεθόδων αντιστοίχισης γραφημάτων σε διανυσματικές αναπαραστάσεις:

1. **Απλές δομικές μετρικές**, όπως η ακολουθία βαθμών των κόμβων, η ο υπολογισμός διαφόρων μετρικών και η δημιουργία ενός διανύσματος χαρακτηριστικών για κάθε γράφημα διάδοσης.

2. **Πυρήνες γραφημάτων**, όπως ο απλός ή ο γενικευμένος πυρήνας γραφημάτων Weisfeiler-Lehman, που σχεδιάστηκαν για να ανιχνεύουν σύνθετα υπομοτίβα δέντρων μέσα στα γραφήματα διάδοσης.

Το κύριο και ίσως αντισυμβατικό συμπέρασμα που προέκυψε από την πειραματική αξιολόγηση είναι ότι η απλότητα υπερτερεί της πολυπλοκότητας για το συγκεκριμένο πρόβλημα. Η μέθοδος που βασίστηκε στην **ακολουθία βαθμών** πέτυχε την υψηλότερη ακρίβεια ταξινόμησης (έως και **59,2%**), ξεπερνώντας με τόσο τον απλό όσο και τον γενικευμένο πυρήνα Weisfeiler-Lehman. Οι τελευταίοι, παρά το υψηλό υπολογιστικό τους κόστος, δεν κατάφεραν να αξιοποιήσουν αποτελεσματικά τη δομική πληροφορία, δίνοντας χαμηλότερες ή, στην καλύτερη περίπτωση, οριακά συγκρίσιμες επιδόσεις.

Αυτό το εύρημα υπογραμμίζει ότι δεν είναι πάντα οι πιο εξεζητημένες τεχνικές αυτές που δίνουν τη λύση. Η άμεση μέτρηση της «ιογένειας» ενός δέντρου, όπως αυτή αποτυπώνεται στην κατανομή των αναδημοσιεύσεων (βαθμών), αποδείχθηκε ο ισχυρότερος και πιο αποδοτικός δείκτης για τον διαχωρισμό αληθών και ψευδών ειδήσεων.

7.2 Μελλοντικές Επεκτάσεις

Το σύστημα που αναπτύχθηκε στα πλαίσια αυτής της εργασίας αποτελεί μια ισχυρή γραμμή αναφοράς. Οι παρακάτω ιδέες θα μπορούσαν να το επεκτείνουν και να βελτιώσουν την απόδοσή του:

- **Εμπλουτισμός Χαρακτηριστικών:** Να συνδυαστεί η επιτυχημένη δομική αναπαράσταση με χαρακτηριστικά από το **περιεχόμενο** των δημοσιεύσεων (ανάλυση κειμένου, NLP), τα **μεταδεδομένα των χρηστών** (ηλικία λογαριασμού, αριθμός ακολούθων) και τα **χρονικά χαρακτηριστικά** της διάδοσης, δημιουργώντας πιο εύρωστα και πολυδιάστατα μοντέλα.

- **Δυναμική Ανάλυση σε Πραγματικό Χρόνο:** Ανάπτυξη μιας μεθόδου που να μπορεί να αξιολογεί μια φήμη καθώς αυτή εξελίσσεται. Αυτό απαιτεί τη δημιουργία μοντέλων που δεν περιμένουν το δέντρο διάδοσης να ολοκληρωθεί, αλλά ενημερώνουν δυναμικά την εκτίμησή τους με κάθε νέα αναδημοσίευση.

- **Προηγμένη Μείωση Διαστατικότητας:** Δεδομένου ότι ο γενικευμένος πυρήνας WL ήταν υπολογιστικά ακριβός λόγω του αλγορίθμου Wasserstein K-Means, θα μπορούσαν να διερευνηθούν εναλλακτικές, ταχύτερες τεχνικές μείωσης διαστατικότητας για την παραγωγή των διανυσμάτων χαρακτηριστικών.

- **Στατιστική Ανάλυση και Επιλογή Μοτίβων:** Να πραγματοποιηθεί στατιστική ανάλυση των δομικών μοτίβων που παράγονται από τους αλγορίθμους. Στόχος είναι να επιλεγούν μόνο εκείνα που παρουσιάζουν στατιστικά σημαντική διακριτική ικανότητα μεταξύ των κλάσεων (true/false), οδηγώντας σε πιο ερμηνεύσιμα και αποδοτικά μοντέλα.

- **Πειραματική Αξιολόγηση Στρατηγικών Τοποθέτησης Επιγραφών:** Να διεξαχθεί συγκριτική μελέτη των μεθοδολογιών τοποθέτησης επιγραφών στους κόμβους των γραφημάτων, με σκοπό τον ποσοτικό προσδιορισμό της επίδρασής τους στην προβλεπτική ικανότητα του τελικού μοντέλου.

Βιβλιογραφία

[1] Reza Zafarani και Xinyi Zhou. *A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities*. 2020.

[2] Nino Shervashidze, Pascal Schweitzer, Erik Janvan Leeuwen, Kurt Mehlhorn και Karsten M. Borgwardt. *Weisfeiler-Lehman Graph Kernels*. 2011.

[3] Till Hendrik Schulz, Tamás Horváth, Pascal Welke και Stefan Wrobel. *A generalized Weisfeiler-Lehman graph kernel*. 2022.

[4] Grigori Sidorov, Helena Gomez-Adorno, Ilia Markov, David Pinto και Nahun Loya. *Computing Text Similarity using Tree Edit Distance*. 2015.

[5] Jia Wu Luchen Liu Wang Xian Bin Qi Huang, Chuan Zhou. *Deep spatial-temporal structure learning for rumor detection on Twitter*. 2020.

[6] Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M. Rocha, Johan Bollen, Filippo Menczer και Alessandro Flammini. *Computational Fact Checking from Knowledge Networks*. 2015.

[7] Chao Shi και Tim Weninger. *Discriminative Predicate Path Mining for Fact Checking in Knowledge Graphs*. 2016.

[8] Veronica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre και Rada Mihalcea. *Automatic Detection of Fake News*. 2017.

[9] Song Feng, Ritwik Banerjee και Yejin Choi. *Syntactic Stylometry for Deception Detection*. 2012.

[10] Kai Shu, Deepak Mahudeswaran, Suhang Wang και Huan Liu. *Hierarchical Propagation Networks for Fake News Detection: Investigating News Spreading Paths*.

[11] Jing Ma, Wei Gao και Kam Fai Wong. *Detect rumors using time series of social context information on microblogging websites*. 2015.

[12] Carlos Castillo, Marcelo Mendoza και Barbara Poblete. *Information Credibility on Twitter*. 2011.

[13] Benjamin D. Horne, Jeppe Norregaard και Sibel Adali. *Different Spirals of Sameness: A Study of Content Sharing in Mainstream and Alternative Media*. 2019.

[14] William Yang Wang. *"Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection*. 2017.

[15] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee και Huan Liu. *FakeNewsNet: A Data Repository with News Content, Social Context and Spatiotemporal Information*. 2018.

[16] Tanushree Mitra και Eric Gilbert. *CREDBANK: A Large-scale Social Media Corpus with Associated Credibility Annotations*. 2015.

[17] Maximilian Nickel, Kevin Murphy, Volker Tresp και Evgeniy Gabrilovich. *A review of relational machine learning for knowledge graphs*. 2015.

[18] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thibault Strohmann, Shaohua Sun και Wei Zhang. *Knowledge vault: A web-scale approach to probabilistic knowledge fusion*. 2014.

[19] Nadia Conroy, Victoria Rubin και Yimin Chen. *Automatic deception detection: Methods for finding fake news*. 2015.

[20] Zhixuan Zhou, Huankang Guan, Meghana Moorthy Bhat και Justin Hsu. *Fake News Detection via NLP is Vulnerable to Adversarial Attacks*. 2019.

[21] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su και Jing Gao. *EANN: Event-Adversarial Neural Networks for Multi-Modal Fake News Detection*. 2018.

[22] Xinyi Zhou, Jindi Wu και Reza Zafarani. *SAFE: Similarity-Aware Multimodal Fake News Detection*. 2020.

[23] Zhiwei Jin, Juan Cao, Yongdong Zhang, Jianshe Zhou και Qi Tian. *Novel visual and statistical image features for microblogs news verification*. 2017.

[24] Jasper Nørregaard, Benjamin Horne και S. Adalı. *NELA-GT-2018: A Large Multi-Labelled News Dataset for The Study of Misinformation in News Articles*. 2019.

[25] Sha Yang Ye και Steven Skiena. *MediaRank: Computational Ranking of Online News Sources*. 2019.

[26] Emilio Ferrara. *Disinformation and Social Bot Operations in the Run Up to the 2017 French Presidential Election*. 2017.

[27] Onur Varol, Emilio Ferrara, Clayton Davis, Filippo Menczer και Alessandro Flammini. *Online Human-Bot Interactions: Detection, Estimation, and Characterization*. 2017.

[28] Onur Varol Kaicheng Yang Alessandro Flammini Chengcheng Shao, Giovanni Luca Ciampaglia και Filippo Menczer. *The spread of low-credibility content by social bots*. 2018.

[29] Sharad Goel, Ashton Anderson, Jake Hofman και Duncan Watts. *The Structural Virality of Online Diffusion*. 2016.

[30] Soroush Vosoughi, Deb Roy και Sinan Aral. *The Spread of True and False News Online*. 2018.

[31] Wei Gao Jing Ma και Kam Fai Wong. *Rumor Detection on Twitter with Tree-structured Recursive Neural Networks*. 2018.

[32] Zhiwei Jin, Juan Cao, Yongdong Zhang και Jiebo Luo. *News Verification by Exploiting Conflicting Social Viewpoints in Microblogs*. 2016.

[33] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang και Huan Liu. *Fake News Detection on Social Media: A Data Mining Perspective*. 2017.

[34] Xinyi Zhou, Reza Zafarani, Kai Shu και Huan Liu. *Fake News: Fundamental Theories, Detection Strategies and Challenges*. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, σελίδες 836–837. ACM, 2019.

[35] Anas Elghafari Aécio Santos Kien Pham Eduardo Nakamura Juliana Freire Sonia Castelo, Thais Almeida. *A Topic-Agnostic Approach for Detecting Misinformation*. *Proc. ACM WebSci*, 2019.

[36] Yang Liu και Hongning Wu. *Early Detection of Fake News on Social Media through Propagation Path Classification*. *IEEE Access*, 2018.

[37] C. Boehm. *The Validity Effect: A Search for Mediating Variables*. 1994.

[38] Naeemul Hassan, Fatma Arslan, Chengkai Li και Mark Tremayne. *Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBusters*. 2017.

[39] Mengnan Du, Ninghao Liu και Xia Hu. *Techniques for Interpretable Machine Learning*. 2019.

[40] Gabriele Vilone και Luca Longo. *Explainable Artificial Intelligence: A Systematic Review*. 2020.

[41] David Lazer, Matthew Baum, Yochai Benkler, Adam Berinsky και Kelly Greenhill. *The Science of Fake News*. 2018.

[42] Maria Lucia Sampoli Franco Scarselli1 Giuseppe Alessio D'Inverno1, Monica Bianchini. *A unifying point of view on expressive power of Graph Neural Networks*. 2021.

[43] Nir Rosenfeld, Aron Szanto και David C. Parkesl. *A Kernel of Truth: Determining Rumor Veracity on Twitter by Diffusion Pattern Alone*. 2020.

[44] Tian Bian, Xiang Wu, Yizhou Sun, Xiaojie Hu, Qi Liu και Yizhou Fu. *Rumor detection on social media with bi-directional graph convolutional networks*. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Απόδοση ξενόγλωσσων όρων

Απόδοση	Ξενόγλωσσος όρος
Ακολουθία	Sequence
Ακμή	Edge
Ακρίβεια	Accuracy
Αλγόριθμος Βαθμωτής Καθόδου	Gradient Descent Algorithm
Αλγόριθμος Κ-Μέσων	K-Means Algorithm
Αμφιμονοσήμαντη Αντιστοιχία	Bijection
Αναδρομή	Recursion
Αναδρομικά Νευρωνικά Δίκτυα	Recursive Neural Networks
Αναδρομική Σχέση	Recurrence Relation
Ανάλυση Δεδομένων	Data Analysis
Αναπαράσταση Γνώσης	Knowledge Representation
Αναπαράσταση Κόμβων	Node Representation
Αναπαράσταση Ύφους	Style Representation
Ανίχνευση Ψευδών Ειδήσεων	Fake News Detection
Αντιπρόσωπος Κλάσης	Class Representative
Αξιοπιστία	Reliability
Απόλυτη Συχνότητα	Absolute Frequency
Αυτόματος Έλεγχος Γεγονότων	Automatic Fact-checking
Βάθος	Depth
Βάση Γνώσης	Knowledge-base
Βαρύκεντρο	Barycenter
Βαθιά Νευρωνικά Δίκτυα	Deep Neural Networks
Βιοπληροφορική	Bioinformatics
Γονέας	Parent
Γράφημα	Graph
Γράφος Γνώσης	Knowledge-graph
Γραμμικός Συνδυασμός	Linear Combination
Δέντρα Ανάπτυξης	Unfolding Trees
Δέντρο	Tree
Δέντρα Διάδοσης	Propagation Trees
Δημοσίευση	Post
Διάσχιση κατά Βάθος	Depth First Search
Διάσχιση κατά Πλάτος	Breadth First Search
Διαγραφή	Deletion

Διανυσματική Αναπαράσταση	Vector Representation
Διανυσματικός Χώρος	Vector Space
Διασταυρωμένη Εντροπία	Cross Entropy
Διατεταγμένο Ζεύγος	Ordered pair
Διατομεακή Ανάλυση	Cross-domain Analysis
Διάνυσμα Χαρακτηριστικών	Feature Vector
Διγλωσσική Ανάλυση	Cross-lingual Analysis
Διμερές Ταίριασμα	Bipartite Matching
Διορθωτική Απόσταση Δέντρων	Tree Edit Distance
Δομική Ιογένεια	Structural Virality
Δυναμική ενημέρωση γράφων γνώσης	Dynamic Knowledge-graphs
Εγκυρότητα	Validity
Εισαγωγή	Insertion
Ελαχίστου Κόστους Διμερές Ταίριασμα	Minimum Cost Bipartite Matching
Ελέγξιμος Ισχυρισμός	Check-worthy claim
Ενισχυμένα Δέντρα Βαθμίδας	Gradient Boosted Trees
Ενσωματώσεις	Embeddings
Εξήγηση Γεγονότων	Fact-checking
Εξηγήσιμη Ανίχνευση	Explainable Detection
Επεξεργασία Φυσικής Γλώσσας	Natural Language Processing
Εσωτερικό Γινόμενο	Inner Product
Ετερογενείς Γράφοι	Heterogeneous Graphs
Ετικέτα	Label
Ετικετών Πολυσυνόλων	Multiset Label
Έλεγχος Σχέσης	Relation Verification
Ζεύγος	Pair
Θέση	Position
Ιδιοκατασκευασμένοι Γράφοι	Self-defined Graphs
Ιεραρχικά Δίκτυα	Hierarchical Networks
Ισομορφισμός Δέντρων	Tree Isomorphism
Ισομορφισμός	Isomorphism
Κακόβουλος Χρήστης	Malicious User
Κανονικός Χρήστης	Normal User
Κανονικοποιημένη Συχνότητα	Normalized Frequency
Κέντρο	Center
Κλάση	Class
Κοινωνικά Βοτς	Social Bots
Κοινωνική Συνοχή	Social Coherence
Κορυφή	Vertex
Κωδικοποίηση Δέντρου Ανάπτυξης	Unfolding Tree Encoding
Κωδικοποίηση Δέντρων	Tree Encoding
Λανθάνοντα Γνωρίσματα	Latent Features
Λεξικά Χαρακτηριστικά	Lexical Features

Λεξικογραφική Ταξινόμηση	Lexicographic Sorting
Λογισμικό	Software
Λογιστική Παλινδρόμηση	Logistic Regression
Μέσα Κοινωνικής Δικτύωσης	Social Media
Μετονομασία Ετικέτας	Relabeling
Μετρικός Χώρος	Metric Space
Μηχανική Μάθηση	Machine Learning
Μηχανική Μάθηση με Επίβλεψη	Supervised Machine Learning
Πίνακας Κόστους	Cost Matrix
Πίνακας Μεταφοράς	Transport Matrix
Πίνακας Σύγχυσης	Confusion Matrix
Παράμετρος Μεροληψίας	Bias
Προδιατεταγμένη διάσχιση	Pre-order Traversal
Πυρήνες Γράφημάτων	Graph Kernels
Πυρήνες συντομότερων μονοπατιών	Shortest Paths Graph Kernels
Πυρήνες τυχαίων περιπάτων	Random Walk Graph Kernels
Πυρήνες υποδέντρων	Subtree Graph Kernels
Ρίζα	Root
Ρυθμός Μάθησης	Learning Rate
Σιγμοειδής Συνάρτηση	Sigmoid Function
Συμβολοσειρά	String
Συνάρτηση Κόστους	Cost Function
Συνεκτική Συνιστώσα	Connected Component
Συστάδα	Cluster
Σύνολο Δεδομένων	Dataset
Σύνολο Εκπαίδευσης	Traing Set
Σύνολο Ελέγχου	Test Set
Ταξινομητής	Classifier
Ταξινόμηση	Sorting
Υποδέντρο	Subtree
Φυσικά Κοινωνικά Δίκτυα	Physical Social Networks
Φυσική Απαρίθμηση	Physical Enumaration
Φύλλο	Leaf
Χρήστες	Users
Ψευδείς Ειδήσεις	Fake News