ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Αξιοποίηση νευρωνικών δικτύων γράφων για την πρόβλεψη του αλγορίθμου συσταδοποίησης με την καλύτερη επίδοση»

Εμμανουήλ Διλμπέρης

Επιβλέπων Καθηγητής:

Δουλκερίδης Χρήστος

ΠΕΙΡΑΙΑΣ

ΦΕΒΡΟΥΑΡΙΟΣ 2025

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αξιοποίηση νευρωνικών δικτύων γράφων για την πρόβλεψη του αλγορίθμου συσταδοποίησης με την βέλτιστη επίδοση

Διλμπέρης Εμμανουήλ

Α.Μ.: ΜΕ2104

ΠΕΡΙΛΗΨΗ

Στον σημερινό ταχύτατα εξελισσόμενο κόσμο που ζούμε, η τεχνητή νοημοσύνη ενσωματώνεται καθημερινά όλο και περισσότερο στη ζωή μας, φέρνοντας επανάσταση στον τρόπο με τον οποίο αλληλοεπιδρούμε με την τεχνολογία. Ωστόσο η ανάπτυξη μοντέλων πρόβλεψης απαιτεί τεχνογνωσία και εξειδικευμένες γνώσεις, κάτι που την καθιστά απρόσιτη για την πλειονότητα των ανθρώπων. Αυτό το πρόβλημα προσπαθεί να επιλύσει η αυτοματοποιημένη μηχανική μάθηση, μειώνοντας σημαντικά την ανθρώπινη παρέμβαση, έχοντας ως αποτέλεσμα να γίνεται πιο προσιτή η χρήση μοντέλων πρόβλεψης από μεγαλύτερη μερίδα του κόσμου. Η παρούσα διπλωματική εργασία ασχολείται με τη δημιουργία ενός μοντέλου αυτοματοποιημένης μηχανικής μάθησης, το οποίο δέχεται σαν είσοδο ένα σύνολο δεδομένων και προσπαθεί να προβλέψει τον αλγόριθμο συσταδοποίησης με την καλύτερη απόδοση, χρησιμοποιώντας αρχιτεκτονικές νευρωνικών δικτύων για γράφους. Για τη υλοποίηση του παραπάνω μοντέλου, δημιουργήθηκε αρχικά η βιβλιοθήκη Dataset2Graph στη γλώσσα προγραμματισμού Python, η οποία παρέχει τη δυνατότητα μετατροπής ενός συνόλου δεδομένων σε γράφο, καθώς και τη χρήση αλγορίθμων για την διαχείριση και την απλοποίηση του παραγόμενου γράφου. Στη διάθεση μας είχαμε 50 σύνολα δεδομένων, τα οποία μετατράπηκαν σε γράφο. Χρησιμοποιώντας διαφορετικές τεχνικές απλοποίησης πάνω στους 50 γράφους, παράχθηκαν πάνω από 150 σύνολα δεδομένων γράφων. Στο τέλος, όλες αρχιτεκτονικές νευρωνικών δικτύων γράφων που αναπτύχθηκαν για το στάδιο της πρόβλεψης, αξιολογούνται πάνω στα διάφορα σύνολα γράφων και το βέλτιστο μοντέλο συγκρίνεται με τον κύριο ανταγωνιστή MARCO-GE.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Αυτοματοποιημένη Μηχανική Μάθηση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μηχανική Μάθηση, Βαθιά Μάθηση, Γράφοι, Νευρωνικά Δίκτυα Γράφων, Κατηγοριοποίηση, Συσταδοποίηση

ABSTRACT

In today's rapidly evolving world, artificial intelligence is becoming more and more integrated into our lives every day, revolutionizing the way we interact with technology. However, developing predictive models requires expertise and specialized knowledge, which makes it inaccessible to the majority of people. Automated machine learning (AutoML) attempts to solve this problem by reducing significantly human intervention, resulting in prediction models becoming more accessible to a larger portion of the world. This thesis deals with the implementation of an automated machine learning model, which takes as input a data set and tries to predict the best performing clustering algorithm utilizing graph neural network architectures. To implement the above model, the Dataset2Graph library was initially developed in the Python programming language, which provides the ability to convert a dataset into a graph and includes a variety of algorithms that process and simplify the generated graphs. At our disposal we had 50 datasets, which were converted into a graph. Different simplification techniques were used on the 50 graphs and over 150 graph datasets were generated. Finally, all graph neural network architectures developed for the prediction stage are evaluated on the different graph sets and the optimal model is compared with the main competitor MARCO-GE.

SUBJECT AREA: Automated Machine Learning

KEYWORDS: Machine Learning, Deep Learning, Graphs, Graph Neural Networks, Classification, Clustering

Στην οικογένεια και τους φίλους μου.

ΕΥΧΑΡΙΣΤΙΕΣ

Καταρχάς οφείλω να ευχαριστήσω από καρδιάς τον επιβλέποντα καθηγητή μου, κύριο Χρήστο Δουλκερίδη για τις πολύτιμες συμβουλές και την καθοδήγηση του κατά την εκπόνηση αυτής της εργασίας. Επιπλέον ένα θερμό ευχαριστώ στους υποψήφιους διδάκτορες και φίλους Ιωάννη Πουλάκη και Δημήτριο Πετράτο για την σημαντική συμβολή και βοήθεια τους. Τέλος είμαι βαθιά ευγνώμων στην οικογένεια μου και τους φίλους μου για την υπομονή, τη στήριξη και την κατανόηση που έδειξαν κατά τη διάρκεια των μεταπτυχιακών σπουδών μου. Ο κάθε ένας αποτέλεσε πηγή δύναμης ειδικά στις πιο δύσκολες στιγμές αυτού του ταξιδιού.

Πίνακας Περιεχομένων

Κεφάλαιο 1 - Εισαγωγή

1.1 Ανάλυση του προβλήματος

Στον σημερινό ταχύτατα εξελισσόμενο κόσμο που ζούμε, η μηχανική μάθηση (machine learning) ενσωματώνεται καθημερινά όλο και περισσότερο στη ζωή μας, φέρνοντας επανάσταση στον τρόπο με τον οποίο αλληλοεπιδρούμε με την τεχνολογία. Αυτή η υποκατηγορία της τεχνητής νοημοσύνης αποτελεί ένα πολύτιμο εργαλείο, το οποίο επικεντρώνεται στη δημιουργία αλγορίθμων, που επιτρέπουν στους υπολογιστές να εκπαιδεύονται από δεδομένα και στη συνέχεια να κάνουν προβλέψεις, να παίρνουν αποφάσεις, να εξάγουν πολύτιμη πληροφορία από αυτά ή ακόμα και να δημιουργούν νέο περιεχόμενο. Για όλους τους προαναφερθέντες λόγους η μηχανική μάθηση δεν περιορίζεται μόνο σε τεχνολογικούς κλάδους, αλλά εφαρμόζεται και σε άλλους τομείς, όπως για παράδειγμα στη φαρμακευτική, για την ανακάλυψη νέων φαρμάκων, αλλά ακόμα και στον τραπεζικό τομέα για την ανίχνευση οικονομικής απάτης ή ξεπλύματος μαύρου χρήματος.

Η συσταδοποίηση (clustering) είναι μια τεχνική μηχανικής μάθησης, η οποία στοχεύει στην εξόρυξη χρήσιμων πληροφοριών, διαχωρίζοντας σε ομάδες τις οντότητές ενός συνόλου δεδομένων, στο οποίο δεν υπάρχουν προκαθορισμένες ετικέτες κλάσεων. Μια πρόκληση που παρουσιάζεται σε αυτή την τεχνική είναι η επιλογή του κατάλληλου αλγορίθμου συσταδοποίησης, καθώς κάθε αλγόριθμος έχει τα δικά του πλεονεκτήματα και μειονεκτήματα και η απόδοσή τους εξαρτάται άμεσα από το μέγεθος, τις διαστάσεις, τις κατανομές και τον τύπο των χαρακτηριστικών του συνόλου δεδομένων. Για παράδειγμα ο αλγόριθμος K-Means θα αποδώσει καλύτερα σε δεδομένα που σχηματίζουν σφαιρικές συστάδες, ενώ αν τα δεδομένα σχημάτιζαν αφηρημένου σχήματος συστάδες, τότε μπορεί να μην ήταν τόσο αποδοτικός, όσο ένας αλγόριθμος που βασίζεται στην πυκνότητα όπως ο DBSCAN. Μια ακόμα πρόκληση αφορά την αξιολόγηση των αλγορίθμων, διότι η πληροφορία για τις κλάσεις των οντοτήτων δεν χρησιμοποιείται, επομένως δεν υπάρχει κάποιο προκαθορισμένο κριτήριο με το οποίο μπορεί να μετρηθεί η αποτελεσματικότητά ή το σφάλμα των αλγορίθμων. Επομένως για την αξιολόγηση μπορούν να χρησιμοποιηθούν μέθοδοι ενδογενής επικύρωσης (internal validation methods), που το κριτήριο τους αφορά τις ιδιότητες των ομάδων που σχηματίστηκαν, για παράδειγμα το πόσο συμπαγείς είναι ή το αν διαχωρίζονται καλά μεταξύ τους. Υπάρχουν σαφώς και μέθοδοι εξωγενής επικύρωσης (external validation methods), όπου τα αποτελέσματα της συσταδοποίησης μπορούν να συγκριθούν με προκαθορισμένη γνώση, όπως οι ετικέτες των κλάσεων των δεδομένων, εφόσον αυτές είναι διαθέσιμες. Οι δυο αυτές προκλήσεις καθιστούν απαραίτητη την εξειδίκευση στον τομέα, αλλά και τη βαθιά κατανόηση των αλγορίθμων συσταδοποίησης, ούτως ώστε να ερμηνευτούν και να αξιολογηθούν σωστά τα αποτελέσματα και τα συμπεράσματα που θα προκύψουν.

Η αυτοματοποιημένη μηχανική μάθηση (AutoML) έρχεται να αλλάξει τα δεδομένα, καθώς προσπαθεί να δημιουργήσει ένα σύστημα το οποίο θα είναι ικανό να αναπτύξει από μόνο του ένα μοντέλο μηχανικής μάθησης. Η πολύπλοκη αυτή διαδικασία, απλοποιείται σε τέτοιο βαθμό που επιτρέπει σε άτομα με ελάχιστη εμπειρία πάνω στον προγραμματισμό και τα μαθηματικά να δημιουργήσουν και να εκπαιδεύσουν μοντέλα τεχνητής νοημοσύνης σε σύντομο χρονικό διάστημα, τα οποία θα είναι βελτιστοποιημένα και αξιόπιστα. Τα στάδια που αυτοματοποιούνται σε ένα AutoML σύστημα είναι τα εξής:

1) Προεπεξεργασία Δεδομένων (Data Preprocessing): Διαχείριση ελλείπων τιμών και διπλοτύπων, κανονικοποίηση των τιμών, κ.α.
2) Σχεδιασμός χαρακτηριστικών (Feature Engineering): Επιλογή χαρακτηριστικών που θα χρησιμοποιηθούν στο επόμενο βήμα και εξαγωγή νέων χαρακτηριστικών που παράγονται από τα ήδη υπάρχον.
3) Επιλογή αλγορίθμου (Algorithm selection): Επιλογή του καταλληλότερου αλγορίθμου από μια σειρά αλγορίθμων, που θα χρησιμοποιηθεί για τη δημιουργία και την εκπαίδευση του τελικού μοντέλου.

4) Βελτιστοποίηση υπερπαραμέτρων (Hyperparameter tuning): Ρύθμιση των υπερπαραμέτρων του μοντέλου, με σκοπό την βελτιστοποίηση της απόδοσης του.

Η παρούσα διπλωματική εργασία ασχολείται με την προσέγγιση της δημιουργία ενός συστήματος AutoML, το οποίο περιορίζεται στα τρία πρώτα βήματα της αυτοματοποιημένης μηχανικής μάθησης και καλείται να λύσει το πρόβλημα της επιλογής του καταλληλότερου αλγορίθμου συσταδοποίησης για ένα νέο σύνολο δεδομένων. Η καινοτομία στο συγκεκριμένο σύστημα είναι ότι τα σύνολα δεδομένων μετατρέπονται από μορφή πίνακα (tabular) σε γράφο, μέσω της βιβλιοθήκης Dataset2Graph. Στη συνέχεια εκπαιδεύονται μοντέλα των οποίων η αρχιτεκτονική, βασίζεται σε state-of-the-art τεχνικές βαθιάς μάθησης, τα νευρωνικά δίκτυα για γράφους (Graph Neural Networks). Αυτά με τη σειρά τους προσπαθούν να προβλέψουν τον αλγόριθμο με τη υψηλότερη απόδοση για ένα νέο σύνολο δεδομένων, το οποίο δεν συμμετείχε στη διαδικασία της εκπαίδευσης.

Για την υλοποίηση αυτής της προσέγγισης χρησιμοποιήθηκαν 3 αλγόριθμοι συσταδοποίησης, οι οποίοι αξιολογήθηκαν σε 50 πραγματικά σύνολα δεδομένων με την external μετρική ARI (Adjusted Rand Index). Τα σύνολα δεδομένων μετατράπηκαν σε γράφους και στη συνέχεια εφαρμόστηκαν τεχνικές απλοποίησης πάνω στους αρχικούς γράφους, για να εξεταστεί το ενδεχόμενο της βελτίωσης της απόδοσης των μοντέλων. Τέλος αναπτύχθηκαν και εκπαιδεύτηκαν μοντέλα διαφορετικών αρχιτεκτονικών, κατάλληλα για την επεξεργασία δεδομένων σε μορφή γράφων και ικανά να προβλέπουν τον αλγόριθμο συσταδοποίησης με τις καλύτερες επιδόσεις για ένα σύνολο δεδομένων.

1.2 Δομή της διπλωματικής εργασίας

Η δομή της διπλωματικής εργασίας έχει ως εξής. Στο πρώτο κεφάλαιο γίνεται μια συνοπτική εισαγωγή στην αυτοματοποιημένη μηχανική μάθηση, τη συσταδοποίηση και παρουσιάζεται το αντικείμενο της παρούσας εργασίας. Το δεύτερο κεφάλαιο παρέχει το απαραίτητο θεωρητικό υπόβαθρο για τις κατηγορίες της μηχανικής μάθησης που αποτελούν τους πυλώνες του συστήματος που υλοποιήθηκε, τη συσταδοποίηση και τα μοντέλα βαθιάς μάθησης για γράφους καθώς και την τεχνική του Meta-Learning. Στο κεφάλαιο τρία αναφέρονται σχετικές εργασίες για συστήματα αυτοματοποιημένης μηχανικής μάθησης πάνω σε προβλήματα μη επιβλεπόμενης μάθησης. Στο κεφάλαιο τέσσερα αναλύεται η μεθοδολογία που ακολουθήθηκε για την ανάπτυξη του AutoML συστήματος. Στο πέμπτο κεφάλαιο παρουσιάζεται το σύνολο δεδομένων καθώς και οι διαφορετικές προσεγγίσεις στη πειραματική διαδικασία και τα αποτελέσματα τους. Στο ίδιο κεφάλαιο επιλέγεται το μοντέλο με την καλύτερη απόδοση και συγκρίνεται με τον κύριο ανταγωνιστή του το MARCO-GE. Στο έκτο και τελευταίο κεφάλαιο αποτυπώνονται τα συμπεράσματα στο πλαίσιο της ανάπτυξης ενός τέτοιου συστήματος και η μελλοντική εργασία για την περαιτέρω βελτίωση της παρούσας υλοποίησης.

Κεφάλαιο 2 - Θεωρητικό Υπόβαθρο

2.1 Μηχανική Μάθηση

Η μηχανική μάθησης, γνωστή και με τη συντομογραφία ML, είναι ένας από τους σημαντικότερους κλάδους της τεχνητής νοημοσύνης. Συνδυάζει την επιστήμη υπολογιστών, τα μαθηματικά και τη στατιστική, με σκοπό τη δημιουργία αλγορίθμων που επιτρέπουν σε έναν υπολογιστή να επεξεργαστεί τεράστιες ποσότητες ιστορικών δεδομένων, να μάθει από αυτά και έπειτα με τη γνώση που αποκτά να πάρει αποφάσεις ή να κάνει προβλέψεις. Οι αλγόριθμοι είναι σχεδιασμένοι να βελτιώνουν την απόδοσή τους με την πάροδο του χρόνου και να γίνονται πιο αποτελεσματικοί και ακριβείς όταν τροφοδοτούνται και επεξεργάζονται όλο και περισσότερα δεδομένα. Οι ικανότητες του ML το καθιστούν ένα ισχυρό και ευέλικτο εργαλείο και κρύβεται πίσω από πολλές τεχνολογικές καινοτομίες του σήμερα, όπως τα chatbots, τα αυτόνομα αυτοκίνητα, οι ψηφιακοί βοηθοί και τα συστήματα συστάσεων.

Η μηχανική μάθηση διαχωρίζεται σε τρία διαφορετικά είδη, την μάθηση με επίβλεψη, τη μάθηση χωρίς επίβλεψη και την ενισχυτική μάθηση. Κάθε ένα από τα παραπάνω είδη καλείται να επιλύσει προβλήματα διαφορετικής φύσεως. Παρακάτω αναλύονται τα δύο πρώτα είδη που αποτελούν και μέρος της προσέγγισης που έχει υλοποιηθεί.

2.1.1 Μάθηση με επίβλεψη (Supervised Learning)

Η μάθηση με επίβλεψη χρησιμοποιείται σε προβλήματα που απαιτούν να γίνει πρόβλεψη. Σε αυτή την προσέγγιση το μοντέλο εκπαιδεύεται πάνω σε δεδομένα, που φέρουν την πληροφορία, την οποία προσπαθεί να προβλέψει. Η πληροφορία αυτή είναι γνωστή και ως κλάση ή ετικέτα. Το μοντέλο κατά τη διάρκεια της εκπαίδευσης σχηματίζει κάποιες αντιστοιχίες μεταξύ των χαρακτηριστικών των δεδομένων και των κλάσεων. Αφού ολοκληρωθεί το στάδιο της εκπαίδευσης και οι εσωτερικοί παράμετροι του μοντέλου έχουν βελτιστοποιηθεί, τότε μπορεί να κάνει προβλέψεις σε δεδομένα που δεν έχει ξαναδεί.

Πριν ξεκινήσει η διαδικασία της εκπαίδευσης τα δεδομένα χωρίζονται σε δυο υποσύνολα, τα οποία δεν έχουνε καμία τομή μεταξύ τους. Το σύνολο εκπαίδευσης (train set) χρησιμοποιείται για την εκπαίδευση του μοντέλου ενώ το σύνολο επικύρωσης (test ή validation set) για την αξιολόγηση της απόδοσης του. Με αυτή την τεχνική μπορούν να εντοπιστούν ανεπιθύμητες συμπεριφορές, όπως το overfitting, και να αντιμετωπιστούν πριν το μοντέλο βγει σε παραγωγικό περιβάλλον. Στο φαινόμενο του overfitting το μοντέλο αποτυγχάνει να γενικεύσει, δηλαδή προσαρμόζεται σε μεγάλο βαθμό στα δεδομένα του συνόλου εκπαίδευσης και στη συνέχεια αδυνατεί να κάνει ακριβείς προβλέψεις στα δεδομένα του συνόλου επικύρωσης που δεν έχει επεξεργαστεί ξανά.

Η μάθηση με επίβλεψη περιλαμβάνει αλγορίθμους παλινδρόμησης (regression) και κατηγοριοποίησης (classification). Η κύρια διαφορά τους είναι ότι στην παλινδρόμηση προβλέπονται συνεχείς τιμές, όπως η ηλικία, η θερμοκρασία, το ύψος ενώ στην κατηγοριοποίηση οι ετικέτες έχουν διακριτές τιμές, για παράδειγμα σκύλος ή γάτα, υγιής ή άρρωστος.

2.1.2 Μάθηση χωρίς επίβλεψη (Unsupervised Learning)

Τα μοντέλα στη μάθηση χωρίς επίβλεψη, σε αντίθεση με την επιβλεπόμενη, επεξεργάζονται δεδομένα, τα οποία δεν έχουν την πρότερη γνώση της κλάσης και προσπαθούν από μόνα τους μέσω κάποιον κριτηρίων να ανακαλύψουν πρότυπα και δομές, άγνωστες μέχρι πρότινος. Ομαδοποιούν τα δεδομένα, στηριζόμενα στις ομοιότητες, τα μοτίβα και τις διαφορές των χαρακτηριστικών τους. Υπάρχουν τρεις τύποι μη επιβλεπόμενης μάθησης, η συσταδοποίηση (clustering), η εξόρυξη κανόνων συσχέτισης (association rules mining) και η μείωση των διαστάσεων (dimensionality reduction). Οι πιο συνηθισμένες περιπτώσεις που χρησιμοποιούνται τεχνικές μη επιβλεπόμενης μάθησης είναι η ανίχνευση ανωμαλιών και απάτης, η επεξεργασία φυσικής γλώσσας (NLP) για ομαδοποίηση άρθρων και σε έρευνες ιατρικής γενετικής για ανάλυση μοτίβων στο DNA.

2.2 Συσταδοποίηση (Clustering)

Η συσταδοποίηση είναι μια τεχνική μη επιβλεπόμενης μάθησης που προσπαθεί να εξορύξει πληροφορία από δεδομένα που δεν περιλαμβάνουν την πληροφορία της ετικέτας. Ουσιαστικά σε αυτή τη διαδικασία ένα αλγόριθμος διαχωρίζει τις οντότητες σε συστάδες, δηλαδή ομάδες, που παρουσιάζουν όμοια χαρακτηριστικά. Η συσταδοποίηση θεωρείται επιτυχημένη εάν η ομοιότητα των στοιχείων εντός της κάθε συστάδας (intra-cluster similarity) είναι υψηλή και παράλληλα μεταξύ των στοιχείων που ανήκουν σε διαφορετικές συστάδες η ομοιότητα (inter-cluster similarity) είναι χαμηλή. Ένα ακόμα προτέρημα της διαδικασίας της συσταδοποίησης είναι ότι καθιστά εφικτή την ερμηνεία συνόλων δεδομένων με μεγάλες διαστάσεις, τα οποία είναι αδύνατο να οπτικοποιηθούν.

2.2.1 Μέθοδοι και Αλγόριθμοι Συσταδοποίησης

Στη συσταδοποίηση δεν υπάρχει κάποιος καθολικά αποδεκτός αλγόριθμος που θα δουλέψει καλά σε οποιαδήποτε εργασία και αν χρησιμοποιηθεί. Κάθε αλγόριθμος σχηματίζει τις τελικές συστάδες με διαφορετική λογική. Επομένως το κριτήριο ομαδοποίησης του αλγορίθμου σε συνδυασμό με τις ιδιότητες του συνόλου δεδομένων, π.χ. πλήθος διαστάσεων, τύποι και κατανομές χαρακτηριστικών, είναι οι δύο κυριότεροι παράγοντες που θα καθορίσουν ικανοποιητική ή όχι την τελική απόδοση του. Οι αλγόριθμοι συσταδοποίησης διαχωρίζονται σε τρεις διαφορετικές κατηγορίες. Για τις ανάγκες της εργασίας επιλέχθηκε ένας αλγόριθμος από κάθε μια κατηγορία.

2.2.1.1 Centroid-based Clustering και K-Means

Τα στοιχεία του συνόλου δεδομένων ομαδοποιούνται λαμβάνοντας υπόψιν την εγγύτητά τους από το κέντρα των συστάδων, τα οποία διαμορφώνονται μέσω μιας επαναληπτικής διαδικασίας. Η ευκλείδεια απόσταση ή η απόσταση Manhattan είναι κάποιες από τις μετρικές ομοιότητας που χρησιμοποιούνται. Στους αλγορίθμους αυτού του τύπου είναι απαραίτητο να οριστεί εκ των προτέρων ο αριθμός των τελικών συστάδων.

Ο K-Means είναι ένας από τους πιο γνωστούς αλγορίθμους αυτής της κατηγορίας. Κάθε στοιχείο ανήκει μόνο σε μία συστάδα, δηλαδή δεν υπάρχουν επικαλύψεις. Αρχικά επιλέγονται τυχαία τα κέντρα των cluster, των οποίων ο αριθμός έχει προκαθοριστεί. Στη συνέχεια για κάθε οντότητα υπολογίζεται η απόσταση τους από όλα τα κέντρα και αναθέτονται στο πιο κοντινό. Έπειτα επανυπολογίζονται τα κέντρα κάθε συστάδας χρησιμοποιώντας τους μέσους όρους των χαρακτηριστικών των στοιχείων που ανήκουν σε αυτές. Η διαδικασία επαναλαμβάνεται μέχρι να μην υπάρξει κάποια αλλαγή στις συντεταγμένες των κέντρων.

2.2.1.2 Connectivity-based Clustering και Agglomerative

Οι αλγόριθμοι αυτού του τύπου, γνωστοί και ως ιεραρχικοί, βασίζονται στην ιδέα ότι το κάθε στοιχείο σχετίζεται περισσότερο με τα κοντινότερα του στοιχεία παρά με αυτά που βρίσκονται πιο μακριά. Ακολουθώντας αυτό το σκεπτικό, δημιουργείται μια ιεραρχική σειρά από εμφωλευμένες συστάδες, ο οποίες μπορούν να αναπαρασταθούν από ένα δενδροδιάγραμμα. Υπάρχουν δυο διαφορετικές προσεγγίσεις, η συσσωρευτική (agglomerative) και η διαιρετική (divisive). Η συσσωρευτική είναι μια bottom-up προσέγγιση, δηλαδή κάθε οντότητα θεωρείται αρχικά ένα ξεχωριστό μεμονωμένο cluster και μέσω μιας επαναληπτικής διαδικασίας συσσωρεύονται έως ότου όλα τα cluster να συγχωνευθούν σε μια ενιαία συστάδα. Αντίθετα η διαιρετική είναι μια top-down διαδικασία, όπου όλα τα στοιχεία στο σύνολο δεδομένων περιέχονται σε ένα cluster και στη συνέχεια αναδρομικά σπάνε σε μικρότερα clusters μέχρι το κάθε στοιχείο να αποτελεί από μόνο του μια συστάδα. Τέλος, στους ιεραρχικούς αλγορίθμους δεν χρειάζεται να δοθεί μια τιμή για το πλήθος των τελικών συστάδων. Η προσέγγιση του Agglomerative θα χρησιμοποιηθεί στην παρούσα διπλωματική εργασία.

2.2.1.3 Density-based Clustering και DBSCAN

Οι αλγόριθμοι που βασίζονται στην πυκνότητα προσδιορίζουν τις συστάδες ως συνεχείς περιοχές συγκεντρωμένης πυκνότητας από στοιχεία του συνόλου δεδομένων, που διαχωρίζονται μεταξύ τους από περιοχές χαμηλής πυκνότητας. Τα κυριότερα πλεονεκτήματά τους είναι ότι οι συστάδες που διαμορφώνονται μπορούν να έχουν αφηρημένο σχήμα και αποδίδουν καλά σε σύνολα δεδομένων

που περιέχουν θόρυβο και outliers, δηλαδή παρατηρήσεις που διαφέρουν σημαντικά από τις υπόλοιπες. Το τελικό πλήθος των ομάδων δεν απαιτείται να προκαθοριστεί.

Ο αλγόριθμος πυκνότητας DBSCAN ζητά από το χρήστη την τιμή σε δυο παραμέτρους. Την ακτίνα ενός κύκλου, που θα δημιουργηθεί γύρω από κάθε στοιχείο και τον αριθμό των ελάχιστων στοιχείων που πρέπει να περιέχονται μέσα στον κύκλο ώστε το στοιχείο να θεωρηθεί πυρήνας. Αν μέσα στον κύκλο περιέχονται λιγότερα στοιχεία από τον αριθμό που δόθηκε στη δεύτερη παράμετρο τότε θεωρείται συνοριακό στοιχείο, ενώ εάν δεν περιέχεται κανένα στοιχείο μέσα στον κύκλο τότε θεωρείται θόρυβος. Τα στοιχεία θορύβου δεν κατατάσσονται σε κάποια συστάδα.

2.3 Βαθιά Μάθηση (Deep Learning)

Η βαθιά μάθηση [1] είναι μια υποκατηγορία της μηχανικής μάθησης, που χρησιμοποιεί τεχνητά νευρωνικά δίκτυα (neural networks) για να μιμηθεί τις λειτουργίες μάθησης του ανθρωπίνου εγκεφάλου. Αυτά τα δίκτυα συγκροτούνται από στρώματα (layers) νευρώνων ή αλλιώς κόμβων, αλληλένδετα μεταξύ τους, που συνεργάζονται για την επεξεργασία της πληροφορίας και την λήψη αποφάσεων.

2.3.1 Νευρωνικά Δίκτυα (Neural Networks)

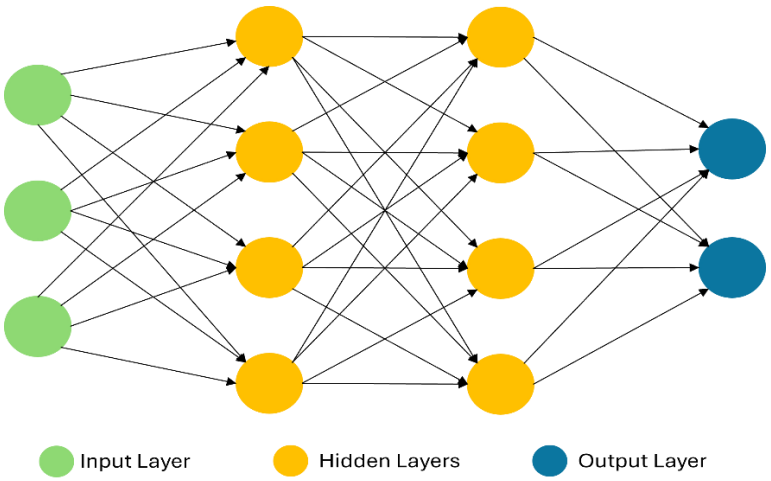
Μια βασική αρχιτεκτονική ενός νευρωνικού δικτύου είναι αυτή του Multilayer Perceptron (MLP) [2] που αποτελείται από τουλάχιστον τρία στρώματα. Αρχικά συναντάται το στρώμα εισόδου (input layer), όπου εισέρχεται η πληροφορία χωρίς καμία επεξεργασία. Σε αυτό το στάδιο οι νευρώνες του στρώματος επεξεργάζονται τα δεδομένα και τα περνούν στο επόμενο στρώμα, το κρυφό (hidden layer). Δεν υπάρχει κάποιος περιορισμός για το πλήθος των κρυφών στρωμάτων, ούτε των κόμβων μέσα σε αυτά. Ο αριθμός των κρυφών στρωμάτων, αντιπροσωπεύει το βάθος του δικτύου. Ομοίως και εδώ επεξεργάζονται και αναλύονται τα δεδομένα εξόδου του προηγούμενου στρώματος και μεταβιβάζονται στο επόμενο. Είναι σημαντικό να σημειωθεί πως όσο πιο βαθύ είναι ένα νευρωνικό δίκτυο, τόσο πιο ικανό είναι να μάθει πιο σύνθετα και πολύπλοκα χαρακτηριστικά και να επιλύσει πιο περίπλοκα προβλήματα. Το τελικό στρώμα, το στρώμα εξόδου (output layer) είναι υπεύθυνο για τις προβλέψεις. Το πλήθος των κόμβων εντός του διαφέρει ανάλογα με τη φύση του προβλήματος. Για παράδειγμα αν πρέπει να επιλυθεί ένα πρόβλημα δυαδικής κατηγοριοποίησης τότε το πλήθος των κόμβων μπορεί να είναι ίσο με το πλήθος των κλάσεων, δηλαδή δυο.

Σε κάθε κόμβο εφαρμόζεται μια συνάρτηση ενεργοποίησης (activation function), η οποία πραγματοποιεί μη-γραμμικούς μετασχηματισμούς στα δεδομένα εισόδου, μετατρέποντάς τα σε δεδομένα εξόδου. Οι μη-γραμμικοί μετασχηματισμοί είναι η αιτία που καθιστά τα νευρωνικά δίκτυα ικανά να επιλύουν σύνθετες και πολύπλοκες εργασίες. Χωρίς τις συναρτήσεις ενεργοποίησης θα είχαμε μοντέλα γραμμικής παλινδρόμησης. Η ReLU (rectified linear unit) είναι πια από τις πιο γνωστές μη-γραμμικές συναρτήσεις ενεργοποίησης και ορίζεται ως $f(x) = max(0, x)$.

Όπως προαναφέρθηκε, οι νευρώνες συνδέονται μεταξύ τους. Κάθε σύνδεση από έναν νευρώνα Χ σε έναν νευρώνα Υ έχει δυο παραμέτρους το βάρος και τη μεροληψία. Τα βάρη διαμορφώνονται κατά τη διαδικασία της εκπαίδευσης και προσδιορίζουν την επίδραση που έχει ο ένας νευρώνας προς τον άλλο. Αν αυτός ο αριθμός είναι θετικός τότε ο ένας κόμβος διεγείρει τον άλλο, ενώ αν είναι αρνητικός τον καταστέλλει. Η μεροληψία προστίθεται στο γινόμενο των χαρακτηριστικών εξόδου με τα βάρη για να τροποποιήσει το αποτέλεσμα της συνάρτησης ενεργοποίησης είτε θετικά είτε αρνητικά.

Η διαδικασία της εκπαίδευση του νευρωνικού δικτύου μπορεί να χωριστεί σε τρία στάδια. Το forward propagation, τον υπολογισμό της loss function και το backward propagation. Αρχικά αναθέτονται τυχαίες μη μηδενικές τιμές στα βάρη και τη μεροληψία. Έπειτα στο πρώτο στάδιο πραγματοποιούνται μια σειρά από πολλαπλασιασμούς πινάκων και ο υπολογισμός των συναρτήσεων ενεργοποίησης για κάθε νευρώνα σε κάθε στρώμα. Μόλις ολοκληρωθούν οι υπολογισμοί και στο στρώμα εξόδου τότε έχει ολοκληρωθεί η πρώτη επανάληψη του πρώτου σταδίου. Στο στάδιο του υπολογισμού της loss function, συγκρίνονται τα αποτελέσματα, δηλαδή οι προβλέψεις με τις πραγματικές τιμές (ground truth) και υπολογίζεται ένα loss σκορ. Το σκορ αυτό

αναπαριστά το σφάλμα του μοντέλου και αποτυπώνει την απόδοση του μοντέλου σε κάθε επανάληψη ή αλλιώς εποχή (epoch). Στο τελευταίο βήμα, με γνώμονα το σφάλμα οι τιμές των παραμέτρων του βάρους και της μεροληψίας ενημερώνονται με σκοπό τη μείωση του σφάλματος. Τα στάδια επαναλαμβάνονται από την αρχή στοχεύοντας στη βελτιστοποίηση των παραμέτρων που ως άμεσο αντίκτυπο έχει την ελαχιστοποίηση του σφάλματος.



Εικόνα 1: Παράδειγμα ενός Multilayer Perceptron με δυο κρυφά στρώματα.

Άλλες διαδεδομένες αρχιτεκτονικές νευρωνικών δικτύων είναι τα συνελικτικά νευρωνικά δίκτυα (CNN) και τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNN). Τα CNN [3] είναι ειδικά για προβλήματα εικόνας και βίντεο. Είναι το καταλληλότερο εργαλείο για ταξινόμηση εικόνας ή ανίχνευση αντικειμένων καθώς έχουν τη δυνατότητα να μαθαίνουν από μόνα τους χαρακτηριστικά πάνω σε εικόνες. Τα RNN [4] είναι η βέλτιστη λύση σε προβλήματα χρονοσειρών και επεξεργασίας φυσικής γλώσσας καθώς μπορούν να επεξεργαστούν διαδοχικά δεδομένα. Τέλος υπάρχει άλλος ένα τύπος νευρωνικών δικτύων, ο οποίος έχει εξελιχθεί ραγδαία τα τελευταία χρόνια. Τα νευρωνικά δίκτυα για γράφους (GNN) θεωρούνται δικαίως μια state-of-the-art τεχνική στην βαθιά μάθηση και αποτελεί κύριο αντικείμενο της παρούσας διπλωματικής εργασίας.

2.4 Γράφοι στην Βαθιά Μάθηση

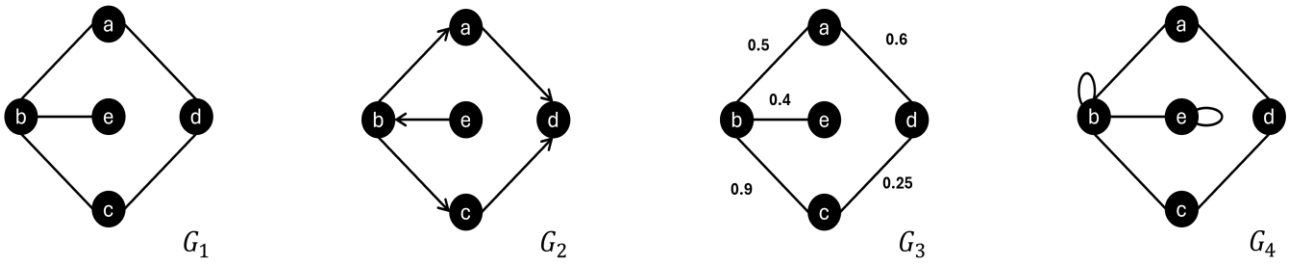
Η αδυναμία των παραδοσιακών τεχνικών μηχανικής μάθησης να διαχειριστούν δεδομένα που δεν βρίσκονται στον ευκλείδειο χώρο και η παρουσία δεδομένων σε μορφή γράφων στον πραγματικό κόσμο, είναι κάποιοι από τους λόγους που οδήγησαν στην ανάπτυξη νέων μεθόδων στον κλάδο της βαθιάς μάθησης. Τα Νευρωνικά Δίκτυα για Γράφους (Graph Neural Networks) [5] εμφανίζουν αξιοσημείωτες ικανότητες στην κατανόηση αυτών των πολύπλοκων δομών, κάτι που τα κατέστησε ιδανικά για την επίλυση προβλημάτων πάνω στους γράφους.

2.4.1 Εισαγωγή στη Θεωρία Γράφων

Οι γράφοι είναι μαθηματικές δομές που αναπαριστούν σχέσεις μεταξύ αντικειμένων. Καλούμε γράφο κάθε διατεταγμένο ζεύγος $G = (V, E)$, όπου V είναι ένα πεπερασμένο σύνολο κόμβων και E είναι ένα πεπερασμένο σύνολο ακμών, που περιέχει υποσύνολα δύο στοιχείων του V. Κάθε ακμή $e = \{u, v\} \in E$, ενώνει δύο κόμβους $u, v \in V$. Δυο κόμβοι που ενώνονται μέσω μιας ακμής θεωρούνται γειτονικοί (adjacent). Η γειτονιά ενός κόμβου $u \in V$ συμβολίζεται $N(u) = \{v \in V \mid \exists \{v, u\} \in E\}$. Εάν ισχύει ότι $N(u) = \emptyset$ τότε λέμε ότι ο u είναι απομονωμένος κόμβος (singleton). Επίσης το πλήθος το γειτόνων ισοδυναμεί με το βαθμό του κόμβου $deg(u) = |N(u)|$.

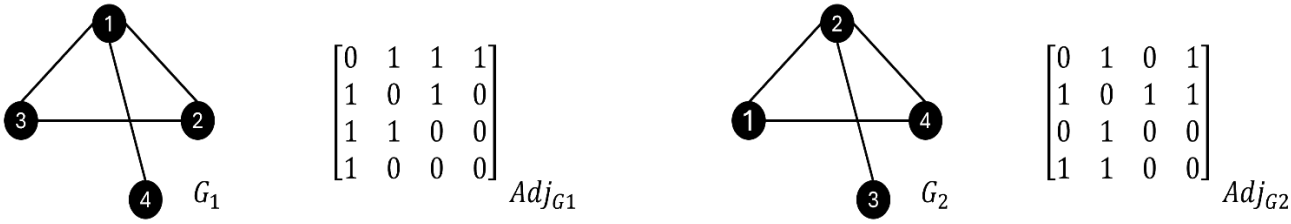
Ο ορισμός του γράφου δεν είναι μοναδικός, υπάρχουν πολλές παραλλαγές. Ένας γράφος μπορεί να είναι κατευθυνόμενος (directed), δηλαδή οι ακμές του είναι διατεταγμένα ζεύγη κορυφών. Επίσης αν ανατεθεί σε κάθε ακμή ενός γράφου ένα βάρος τότε καλείται βεβαρημένος (weighted). Μπορούν να επιτραπούν και ακμές που ξεκινούν και καταλήγουν στον ίδιο κόμβο, οι λεγόμενες θηλιές. Στην

παρούσα διπλωματική εργασία ο τύπος των γράφων που θα μας απασχολήσουν είναι μη κατευθυνόμενοι, βεβαρημένοι και χωρίς θηλιές.



Εικόνα 2: Ο μη κατευθυνόμενος γράφος G1, ο κατευθυνόμενος γράφος G2, ο βεβαρημένος γράφος G3 και ο γράφος με θηλιές G4.

Κάθε πεπερασμένος γράφος μπορεί να αναπαρασταθεί με τη μορφή ενός τετραγωνικού πίνακα, τον πίνακα γειτνίασης, $A \in \mathbb{Z}^{n\times n}$. Οι διαστάσεις του πίνακα είναι ίσες με το πλήθος των κόμβων του γράφου. Για παράδειγμα ο πίνακας γειτνίασης ενός γράφου με οκτώ κόμβους έχει διαστάσεις 8x8, δηλαδή οκτώ γραμμές και οκτώ στήλες. Τα στοιχεία του πίνακα παίρνουν τιμές 0 ή 1. Αν το στοιχείο στη θέση i_{uv} έχει τιμή ίση με ένα, τότε αυτό υποδηλώνει ότι οι κόμβοι $u, v \in V$ γειτονεύουν, επομένως υπάρχει κάποια ακμή στο σύνολο E που τις ενώνει. Εάν ο γράφος είναι βεβαρημένος τότε τα στοιχεία του πίνακα παίρνουν τις τιμές των βαρών των ακμών. Η διαγώνιος του πίνακα παίρνει τη μηδενική τιμή αν δεν υπάρχουν θηλιές στο γράφο. Ο πίνακας γειτνίασης ενός μη κατευθυνόμενου γράφου είναι συμμετρικός. Τέλος ένας γράφος μπορεί να αναπαρασταθεί με περισσότερους από έναν πίνακες γειτνίασης.



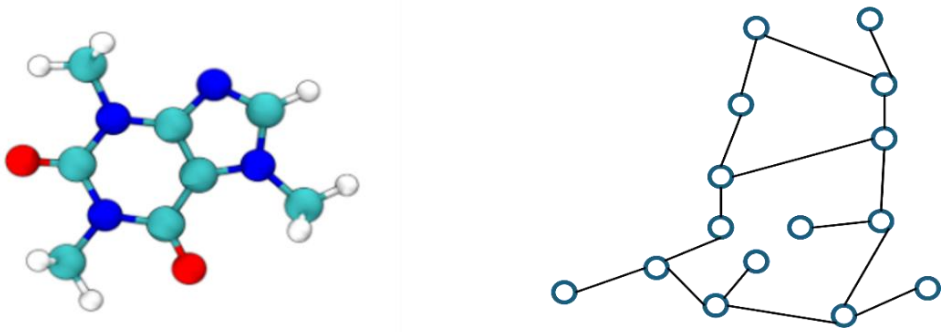
Εικόνα 3: Τα G1 και G2 αναπαριστούν ένα γράφο με την ίδια δομή, με μόνη διαφορά στην αρίθμηση των κόμβων. Επομένως ένας γράφος έχει εκφραστεί μέσω δυο διαφορετικών πινάκων γειτνίασης.

Δύο ακόμη πίνακες που χρησιμοποιούνται στους γράφους είναι ο πίνακας βαθμών (degree matrix) και ο πίνακας Laplace. Ο πρώτος είναι ένα διαγώνιος τετραγωνικός πίνακας $D \in \mathbb{Z}^{n\times n}$ που οι τιμές των στοιχείων είναι οι βαθμοί των κόμβων. Προηγουμένως είπαμε ότι ο βαθμός ενός κόμβου ισούται με το πλήθος των γειτόνων του. Αν όμως ο γράφος είναι βεβαρημένος τότε ο βαθμός εκφράζεται από το άθροισμα των βαρών των ακμών που προσπίπτουν στον κόμβο. Τέλος ο πίνακας Laplace είναι ομοίως ένας τετραγωνικός πίνακας $L \in \mathbb{Z}^{n\times n}$ και ορίζεται ως $L = D - A$.

2.4.2 Νευρωνικά Δίκτυα Γράφων (GNNs)

Τα νευρωνικά δίκτυα γράφων (GNNs) είναι μια κατηγορία μεθόδων βαθιάς μάθησης που έχουν σχεδιαστεί για να εξάγουν συμπεράσματα και να επιλύουν προβλήματα που αφορούν δεδομένα με τη δομή γράφου. Τα τελευταία χρόνια έχουν γίνει πολλές βελτιώσεις που έχουν ενισχύσει σε μεγάλο βαθμό τις δυνατότητές τους και την εκφραστική τους δύναμη. Μπορούν να εφαρμοστούν σε τομείς όπως η ανακάλυψη φαρμάκων [6], η ανίχνευση fake news [7], η πρόβλεψη κίνησης [8].

Οι γράφοι είναι ένα πανίσχυρο εργαλείο το οποίο μπορούμε να εκμεταλλευτούμε έτσι ώστε να μοντελοποιήσουμε δεδομένα, με πιο ετερογενή δομή, τα οποία δεν θα ήταν εύκολο να εκφραστούν στην κλασσική μορφή πίνακα. Ένα από τα προτερήματα τους είναι ότι πέρα από τα χαρακτηριστικά των στοιχείων που το αποτελούν, είναι διαθέσιμη και η πληροφορία για τις σχέσεις μεταξύ τους. Για παράδειγμα τα μόρια αποτελούνται από άτομα διαφορετικών στοιχείων, που ενώνονται μεταξύ τους. Η δομή ενός ατόμου είναι εύκολο να αναπαρασταθεί ως γράφος, όπου τα άτομα είναι οι κόμβοι και οι ενώσεις μεταξύ τους είναι οι ακμές. Κάθε κόμβος, αλλά και ακμή, μπορούν να περιγραφούν από ένα διάνυσμα χαρακτηριστικών. Στο συγκεκριμένο παράδειγμα τα χαρακτηριστικά των κόμβων μπορεί να είναι ο αριθμός των πρωτονίων, των νετρονίων και των ηλεκτρονίων του κάθε ατόμου, ενώ στις ακμές ο τύπος του δεσμού μεταξύ των ατόμων (π.χ. μονός δεσμός, διπλός δεσμός). Άλλοι τύποι δεδομένων που μπορούν να αναπαρασταθούν με τη μορφή γράφου είναι τα μέσα κοινωνικής δικτύωσης, το δίκτυο παραπομπών επιστημονικών δημοσιεύσεων, ακόμα και ένα κείμενο έχει τη δυνατότητα να εκφραστεί με αυτή τη δομή.



Εικόνα 4: Ένα άτομο καφεΐνης και η αναπαράστασή του σε γράφο.

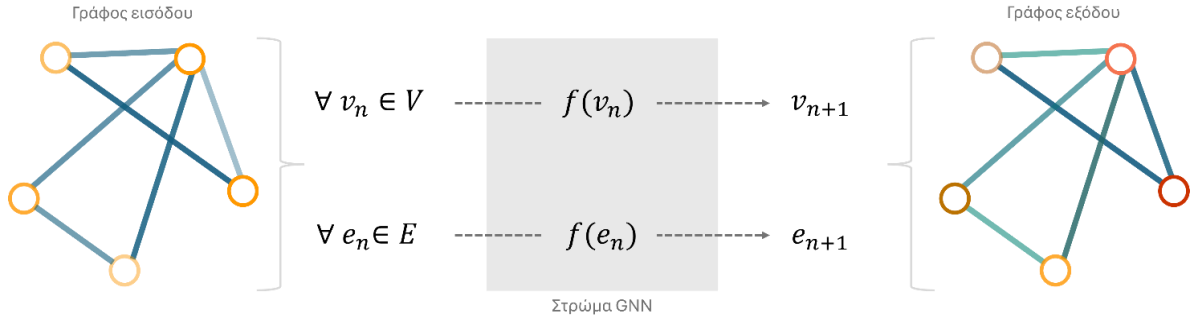
Μέσω των GNN μπορούν να επιλυθούν προβλήματα πρόβλεψης σε τρία διαφορετικά επίπεδα. Σε επίπεδο γράφου, το ζητούμενο είναι η πρόβλεψη μιας ιδιότητας ολόκληρου του γράφου. Για παράδειγμα έχουμε ένα σύνολο δεδομένων από γράφους μορίων και θέλουμε να προβλέψουμε ποιο από αυτά μπορεί να συνδεθεί με έναν υποδοχέα που εμπλέκεται σε μια ασθένεια. Σε επίπεδο κόμβου, προβλέπονται μια ή περισσότερες ιδιότητες για κάποιους από τους κόμβους εσωτερικά ενός γράφου. Ένα παράδειγμα τέτοιου προβλήματος είναι η πρόβλεψη του θέματος μιας δημοσίευσης μέσα σε ένα γράφο που περιέχει ένα δίκτυο με παραπομπές δημοσιεύσεων. Τέλος σε επίπεδο ακμής, πέρα από την πρόβλεψη ιδιοτήτων σε ακμές εντός του γράφου, ένα ακόμα ζητούμενο αφορά την πρόβλεψη της γενικότερης παρουσίας μιας ακμής μεταξύ δυο κόμβων. Ας πάρουμε ένα σύστημα συστάσεων με τη μορφή γράφου, όπου κάποιοι κόμβοι του αντιπροσωπεύουν χρήστες και κάποιοι άλλοι αντικείμενα. Η πρόβλεψης της σύστασης είναι ουσιαστικά η πρόβλεψη της ακμής μεταξύ ενός χρήστη και ενός αντικειμένου.

2.4.3 Τρόπος Λειτουργίας των GNN

Τα GNN υιοθετούν τη λογική “graph-in, graph-out” και αξιοποιούν όλη την πληροφορία που περιέχεται στους γράφους, δηλαδή τα χαρακτηριστικά των κόμβων, τη συνδεσιμότητα μεταξύ των κόμβων και τα χαρακτηριστικά των ακμών. Σταδιακά όλα αυτά τα χαρακτηριστικά μετασχηματίζονται, πέραν όμως της συνδεσιμότητας που παραμένει αναλλοίωτη. Μέρος της συνδεσιμότητας αποτελεί και το βάρος των ακμών, σε περιπτώσεις βεβαρημένων γράφων.

Μια από τις πιο απλές αρχιτεκτονικές των GNN είναι η χρήση ενός οποιουδήποτε μοντέλου MLP, το οποίο θα εφαρμοστεί πάνω σε όλα τα στοιχεία του γράφου. Εφαρμόζοντας το λοιπόν πάνω στα διανύσματα των κόμβων και των ακμών, θα μας επιστραφούν τα νέα χαρακτηριστικά (embeddings) που έχει μάθει το μοντέλο για το καθένα. Η συνδεσιμότητα μεταξύ των κόμβων στην παρούσα αρχιτεκτονική δεν έχει χρησιμοποιηθεί για την εξαγωγή των embeddings ούτε έχει μεταβληθεί στο

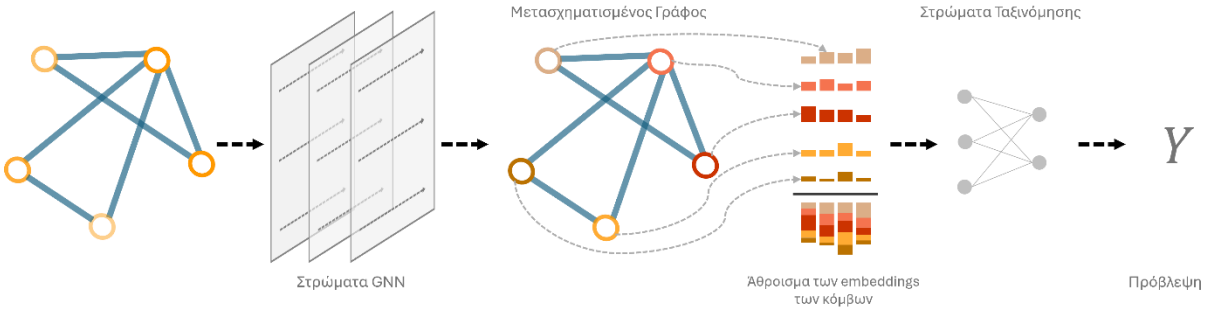
νέο γράφο που προέκυψε. Επομένως ο γράφος εισόδου με τον γράφο εξόδου εκφράζονται από τον ίδιο πίνακα γειτνίασης.



Εικόνα 5: Ένα στρώμα ενός απλού GNN. Το f αντιπροσωπεύει το μοντέλο νευρωνικού δικτύου που μετασχηματίζει τους κόμβους και τις ακμές.

2.4.4 Προβλέψεις σε Επίπεδο Γράφου

Σε ένα πρόβλημα ταξινόμησης σε επίπεδο κόμβων εάν ο γράφος περιέχει πληροφορία για τα χαρακτηριστικά των κόμβων είναι ξεκάθαρο πως η τελική ταξινόμηση των κόμβων θα παρθεί από τα embedding του τελευταίου στρώματος του GNN. Όμως η επίλυση ενός προβλήματος πρόβλεψης δεν είναι πάντα τόσο απλή. Υπάρχουν περιπτώσεις που ζητείται πρόβλεψη σε επίπεδο γράφου αλλά δεν υπάρχει κάποιο διάνυσμα που να περιγράφει γενικά το γράφο και η μόνη διαθέσιμη πληροφορία είναι αποθηκευμένη στους κόμβους του. Στα GNN είναι εφικτό να γίνουν προβλέψεις σε ένα επίπεδο αξιοποιώντας την πληροφορία που υπάρχει σε ένα άλλο, μέσω της διαδικασίας του pooling. Στο pooling συγκεντρώνεται η πληροφορία των απαραίτητων embeddings σε έναν πίνακα $m \times n$, όπου m το πλήθος των κόμβων του γράφου και n το πλήθος των εξαγόμενων χαρακτηριστικών τους. Στην συνέχεια εφαρμόζεται κάποιο aggregation, δηλαδή υπολογίζεται το άθροισμα ή ο μέσος όρος ανά χαρακτηριστικό, και εξάγεται ένα διάνυσμα $1 \times n$, το οποίο θα χρησιμοποιηθεί από τα ακόλουθα στρώματα που είναι υπεύθυνα για την τελική ταξινόμηση. Άρα στα προβλήματα επιπέδου γράφου, όπου η πληροφορία είναι συγκεντρωμένη στους κόμβους, θα συγκεντρωθούν όλα τα embeddings των κόμβων που έχουν προκύψει από το τελικό στρώμα του GNN σε ένα πίνακα και θα εφαρμοστεί μια συνάρτηση pooling για την παραγωγή του embedding που θα περιγράφει τον γράφο συνολικά.



Εικόνα 6: End-to-end διαδικασία πρόβλεψης ενός GNN μοντέλου σε επίπεδο γράφου, χρησιμοποιώντας την πληροφορία των κόμβων.

2.4.5 Διαβίβαση Μηνυμάτων μεταξύ Τμημάτων του Γράφου

Μια τεχνική που βελτίωσε σημαντικά την απόδοση των GNNs είναι η διαβίβαση μηνύματος που συστήθηκε από τον Gilmer [9], καθώς καταφέρνει να αποτυπώσει αποτελεσματικά τις αλληλεπιδράσεις και τις εξαρτήσεις μεταξύ των κόμβων. Επιτρέπει στους κόμβους να ανταλλάσσουν πληροφορίες με του γείτονές τους, επηρεάζοντας με αυτόν τον τρόπο ο ένας τον άλλο. Ουσιαστικά αυτή η τεχνική είναι η χρήση του pooling εντός του κάθε στρώματος στο GNN, όπου κάθε κόμβος συλλέγει τα χαρακτηριστικά της τοπικής του γειτονιάς και στη συνέχεια μέσω ενός aggregation διαμορφώνεται το ενημερωμένο embedding του.

Η διαβίβαση μηνυμάτων στα GNNs μπορεί να παρομοιαστεί με την τεχνική της συνέλιξης που χρησιμοποιούν τα CNNs σε προβλήματα με δεδομένα εικόνας. Και οι δυο αυτές τεχνικές συγκεντρώνουν την πληροφορία από τους γείτονες, τη συμπυκνώνουν και προσδίδουν νέα αξία στο στοιχείο. Είναι σημαντικό να σημειωθεί πως ο αριθμός των γειτόνων από κόμβο σε κόμβο σε ένα γράφο μπορεί να διαφέρει, σε αντίθεση με τις εικόνες όπου το κάθε pixel έχει ένα συγκεκριμένο αριθμό γειτόνων. Επίσης η διάταξη των κόμβων μπορεί να είναι αυθαίρετη και αυτό δεν θα έχει επίδραση στην απόδοση των GNNs, ενώ αν σε μια εικόνα μεταβληθεί η χωρική θέση των pixels τότε θα έχει αρνητικό αντίκτυπο στο μοντέλο.

Χρησιμοποιώντας διαδοχικά στρώματα GNN που κάνουν χρήση της τεχνικής της διαβίβασης μηνυμάτων ένα κόμβος μπορεί να ενσωματώσει στο τελικό του embedding πληροφορία που θα προέρχεται από κάθε στοιχείο του γράφου, χωρίς να περιορίζεται μόνο στους γείτονές του. Μετά από δύο διαδοχικά στρώματα ένας κόμβος θα έχει πάρει πληροφορία από το γείτονα του γείτονα του, ενώ μετά από τρία διαδοχικά στρώματα ο κόμβος θα έχει πληροφορία που απέχει τρία βήματα μακριά από αυτόν.

2.4.6 Αρχιτεκτονικές για τον Υπολογισμό Embeddings Κόμβων

Η εξαγωγή των embeddings εξαρτάται από τον τρόπο που επιλέγει η κάθε αρχιτεκτονική να εκμεταλλευτεί την πληροφορία της γειτονιάς του κάθε κόμβου. Οι προσεγγίσεις χωρίζονται σε δύο κατηγορίες, φασματικές και μη φασματικές. Η κύρια διαφορά τους έγκειται στον τρόπο με τον οποίο επεξεργάζονται τους γράφους. Τα φασματικά GNN λειτουργούν στο φασματικό πεδίο αξιοποιώντας τις ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα Laplace, ενώ τα μη φασματικά εστιάζουν στην τοπική δομή του γράφου. Στην παρούσα διπλωματική εργασία θα ασχοληθούμε μόνο με αρχιτεκτονικές που ανήκουν στη δεύτερη κατηγορία. Ο λόγος για τον οποίο δεν υλοποιήθηκαν μοντέλα που ανήκουν στην πρώτη κατηγορία είναι κάποια μειονεκτήματα που υπάρχουν σε αυτές τις αρχιτεκτονικές. Αρχικά ο υπολογισμός των ιδιοτιμών και των ιδιοδιανυσμάτων αλλά και οι επαναλαμβανόμενοι πολλαπλασιασμοί μεταξύ πινάκων μεγάλων διαστάσεων έχουν τεράστιο υπολογιστικό κόστος. Η πιο σημαντική όμως πρόκληση είναι η γενίκευση αυτών των μοντέλων. Τα βάρη τους διαμορφώνονται βάση των ιδιοτιμών και των ιδιοδιανυσμάτων των πινάκων Laplace των γράφων στο σύνολο εκπαίδευσης, τα οποία είναι μοναδικά για κάθε γράφο. Αυτό σημαίνει πως το μοντέλο δεν θα μπορέσει να μεταφέρει τη γνώση του σε νέα γραφήματα με διαφορετικό πίνακα Laplace. Αντίθετα τα μη φασματικά μοντέλα επικεντρώνονται στη συγκέντρωση πληροφοριών από τις τοπικές γειτονιές, καθιστώντας τα πιο ευέλικτα σε γράφους με διαφορετικές δομές.

2.4.6.1 Graph Convolution Network (GCN)

Μια από τις πιο διαδεδομένες αρχιτεκτονικές είναι αυτή των Graph Convolution Networks (GCN) [10], άμεσα βασισμένα στη λογική των CNN. Σε κάθε στρώμα k του GCN για $k = 1,2,..,n$, υπολογίζονται τα νέα χαρακτηριστικά για κάθε κόμβο $u \in V$ από τον τύπο:

$$h_u^{(k)} = \sigma^{(k)}\left(W^{(k)} * \sum_{v \in N(u)} \frac{e_{u,v} * h_v^{(k-1)}}{\sqrt{|N(u)| * |N(v)|}} + B^{(k)} * h_u^{(k-1)}\right)$$

όπου το σ συμβολίζει τη συνάρτηση ενεργοποίησης, τα W και B τις παραμέτρους εκμάθησης του βάρους και της μεροληψίας αντίστοιχα και το $e_{u,v}$ το βάρος της ακμής.

2.4.6.2 Graph Attention Network (GAT)

Τα Graph Attention Networks είναι μια αρχιτεκτονική η οποία προσπαθεί να αντικαταστήσει τον συντελεστή $e_{u,v}/\sqrt{|N(u)| * |N(v)|}$ με έναν συντελεστή εκμάθησης που βασίζεται σε ένα μηχανισμό προσοχής (attention mechanism). Ο μηχανισμός επιδεικνύει τη σημαντικότητα των γειτόνων ενός κόμβου χωρίς να εκμεταλλεύεται την πρότερη γνώση του πίνακα βαθμών (degree matrix) και τα βάρη των ακμών. Σύμφωνα με τη δημοσίευση [11] το πρώτο βήμα είναι ένας γραμμικός μετασχηματισμός των χαρακτηριστικών εισόδου σε κάθε στρώμα, με τη βοήθεια ενός πίνακα βαρών W. Τα στοιχεία αυτού του πίνακα ορίζονται τυχαία στην αρχή και ενημερώνονται σε κάθε επανάληψη της εκπαίδευσης. Στη συνέχεια ξανά εφαρμόζεται ένα γραμμικός σχηματισμός με ένα διάνυσμα $\overrightarrow{W_{att}}$ που τα στοιχεία του είναι επίσης βάρη εκμάθησης και έχει μέγεθος ίσο με το πλήθος τον χαρακτηριστικών των κόμβων. Το αποτέλεσμα των γραμμικών μετασχηματισμών εισάγεται σε μια συνάρτηση ενεργοποίησης. Η LeakyReLU είναι μια μη γραμμική συνάρτηση ενεργοποίησης, η οποία στις αρνητικές τιμές έχει μια μικρή αρνητική κλίση σε αντίθεση με τη ReLU που έχει επίπεδη κλίση:

$$e_{uv} = LeakyReLU\left(\overrightarrow{W_{att}}^{T}\,[Wh_u \parallel Wh_v]\right)$$

όπου ‖ ο τελεστής συνένωσης. Παρακάτω όλοι οι συντελεστές των γειτόνων ενός κόμβου κανονικοποιούνται μέσω της softmax, αθροίζοντας στο 1, ώστε να μπορούν να είναι συγκρίσιμοι.

$$a_{u,v} = softmax(e_{uv}) = \frac{\exp(e_{uv})}{\sum_{k\in N(u)}\exp(e_{uk})}$$

Στη συνέχεια τα νέα χαρακτηριστικά για κάθε κόμβο υπολογίζονται από την εξίσωση:

$$h_u^{(k)} = \sigma^{(k)}\left(W^{(k)} * \left[\sum_{v\in N(u)} {\alpha_{u,v}}^{(k-1)} * h_v^{(k-1)} + {\alpha_{u,u}}^{(k-1)} * h_u^{(k-1)}\right]\right)$$

Στα GATs υπάρχει η δυνατότητα να χρησιμοποιηθεί παραπάνω από ένας μηχανισμός προσοχής. Ορίζοντας τυχαία n διαφορετικά $\overrightarrow{W_{att}}$ και W μπορούμε να υπολογίσουμε για κάθε κόμβο την παραπάνω εξίσωση n φορές και είτε να πάρουμε τους μέσους όρους των νέων χαρακτηριστικών που θα προκύψουν, είτε να συνενώσουμε τα χαρακτηριστικά σε ένα νέο διάνυσμα. Αυτή η προσέγγιση μειώνει το ρίσκο της εμφάνισης του overfitting και σταθεροποιεί την διαδικασία εκμάθησης.

Μια βελτιστοποίηση των στρωμάτων GAT προτάθηκε το 2021 [12], όπου τροποποιείται η σειρά των πράξεων στον υπολογισμό του e_{uv}. Ο μετασχηματισμός με τον πίνακα W γίνεται μετά τη συνένωση των χαρακτηριστικών των γειτόνων και ο μετασχηματισμός με το διάνυσμα $\overrightarrow{W_{att}}$ μετά τη LeakyReLU.

$$e_{uv} = \overrightarrow{W_{att}}^{T} LeakyReLU(W[h_u \parallel h_v])$$

2.4.6.3 Graph Isomorphism Network

Δυο γράφοι G και H καλούνται ισόμορφοι αν υπάρχει μια 1-1 και επί συνάρτηση $\varphi: V(G) \to V(H)$, τέτοια ώστε για κάθε $u, v \in V(G)$ με $u \neq v$, ισχύει ότι $\{u, v\} \in E(G) \Leftrightarrow \{\varphi(u), \varphi(v)\} \in E(H)$. Με πιο απλά λόγια δυο γράφοι θεωρούνται ισόμορφοι αν έχουν την ίδια τοπολογική δομή.

Στη θεωρία των γράφων υπάρχει το τεστ ισομορφισμού των Weisfeiler-Lehman [13], το οποίο μπορεί να προσδιορίσει αν δυο γράφοι δεν είναι ισόμορφοι. Υπάρχουν πολλές παραλλαγές αυτού του αναδρομικού αλγορίθμου. Στην πιο απλή εκδοχή του, σε κάθε επανάληψη τα χαρακτηριστικά του κάθε κόμβου ενημερώνονται με τα aggregated χαρακτηριστικά των γειτόνων τους. Σε αυτά τα aggregations χρησιμοποιείται μια συνάρτηση 1-1, η οποία κάθε διαφορετικού τύπου γειτονιά την εκφράζει με ένα ξεχωριστό διάνυσμα. Αυτή η συνάρτηση είναι που καθιστά το τεστ αυτό τόσο ισχυρό. Ο αλγόριθμος ολοκληρώνεται μετά από προκαθορισμένο αριθμό πεπερασμένων επαναλήψεων ή

όταν σε δύο διαδοχικές επαναλήψεις δεν μεταβληθεί κανένα από τα χαρακτηριστικά των κόμβων. Αυτή η διαδικασία τρέχει παράλληλα και στους δύο γράφους και στο τέλος γίνεται ο έλεγχος αν είναι ισόμορφοι ή όχι.

Αυτή η διαδικασία θυμίζει πολύ τη διαβίβαση μηνυμάτων στα GNNs, όμως οι συναρτήσεις των aggregations που συναντούσαμε μέχρι στιγμής δεν ήταν 1-1. Στη δημοσίευση [14] παρουσιάζεται η αρχιτεκτονική των Graph Isomorphism Network (GIN) που καταφέρνουν να γενικεύσουν τον παραπάνω αλγόριθμο στα GNN. Τα χαρακτηριστικά του κάθε κόμβου στα GIN υπολογίζονται από τον τύπο:

$$h_u^{(k)} = MLP^{(k)}\left(\left(1 + e^{(k)}\right) * h_u^{(k-1)} + \sum_{v \in N(u)} h_v^{(k-1)}\right)$$

Η 1-1 συνάρτηση στα GIN αντικαταστάθηκε από ένα MLP και το e είναι μια παράμετρος εκμάθησης.

2.5 Meta-Learning

Το Meta-Learning [15] είναι μια υποκατηγορία της μηχανικής μάθησης. Στο παραδοσιακό και συμβατικό τρόπο λειτουργίας της μηχανικής μάθησης, τα μοντέλα εκπαιδεύονται πάνω σε ένα συγκεκριμένο σύνολο δεδομένων για την επίλυση ενός συγκεκριμένου προβλήματος. Αντίθετα τα μοντέλα του Meta-Learning εκπαιδεύονται πάνω σε μια πληθώρα προβλημάτων ή εργασιών, έτσι ώστε να αποκτήσουν γενικευμένη γνώση και να μην περιορίζονται μόνο σε ένα συγκεκριμένο.

Η διαδικασία του Meta-Learning είναι ιδανική για τη δημιουργία ενός συστήματος AutoML που θα επιλέγει τον βέλτιστο αλγόριθμο. Αρχικά παράγονται νέα δεδομένα, τα λεγόμενα meta-data ή μετα-χαρακτηριστικά (meta-features), που περιγράφουν κάποια από τα χαρακτηριστικά των συνόλων δεδομένων, καθώς και την απόδοση του κάθε αλγορίθμου που μας ενδιαφέρει πάνω σε αυτά. Στη συνέχεια εκπαιδεύεται ένα μοντέλο, meta-model, πάνω στο νέο σύνολο δεδομένων που δημιουργήθηκε από τα μετα-χαρακτηριστικά και προσπαθεί να αναγνωρίσει και να μάθει συσχετίσεις μεταξύ των χαρακτηριστικών και των αποδόσεων των αλγορίθμων. Αφού ολοκληρωθεί η διαδικασία της εκπαίδευσης, το meta-model θα μπορεί να δέχεται σαν είσοδο τα meta-features που θα εξάγονται από τα νέα σύνολα δεδομένων και θα επιστρέφει τον βέλτιστο αλγόριθμο.

Ένα από τα πιο απαιτητικά στάδια της διαδικασίας, είναι ο τρόπος εξαγωγής των μετα-χαρακτηριστικών. Η επιλογή τους είναι καίριας σημασίας διότι έχει άμεσο αντίκτυπο στην απόδοση του meta-model. Επιλέγοντας τα σωστά μετα-χαρακτηριστικά θα δημιουργηθεί ένα πιο γενικευμένο μοντέλο με αυξημένη απόδοση.

2.6 Συναφείς Ερευνητικές Εργασίες

2.6.1 AutoClust

Το AutoClust [16] είναι ένα AutoML σύστημα το οποίο επιλύει το πρόβλημα επιλογής αλγορίθμου και διαδοχικά στη βελτιστοποίηση των παραμέτρων του. Τα μετα-χαρακτηριστικά που περιγράφουν ένα σύνολο δεδομένων σε αυτή την υλοποίηση είναι οι τιμές διάφορων ενδογενών δεικτών συσταδοποίησης, που προκύπτουν από την εφαρμογή του αλγορίθμου Mean Shift πάνω σε αυτό. Ο συγκεκριμένος αλγόριθμος επιλέγεται λόγω της μη παραμετρικής του φύσης. Οι ενδογενείς δείκτες που επιλέχθηκαν είναι οι Silhouette Score, Dunn Index, C-Index, Calinski-Harabasz, Davies Bouldin, SDbw, CDBW, Tau Index, Ratkowsky Lance και McClain Rao. Η σύσταση του βέλτιστου αλγορίθμου συσταδοποίησης γίνεται με ένα εκπαιδευμένο KNN meta-model. Για το πρόβλημα της βελτιστοποίησης των υπερπαραμέτρων των αλγορίθμων δημιουργούνται νευρωνικά παλινδρόμησης, τα οποία δέχονται σαν είσοδο διανύσματα με τις τιμές των ενδογενών δεικτών και εκτιμούν τον εξωγενή δείκτη ARI. Με τη βελτιστοποίηση κατά Bayes και συγκεκριμένα με τη χρήση Tree-Parzen εκτιμητών, γίνεται η βελτιστοποίηση μεγιστοποιώντας το ARI που έχει προβλεφθεί.

2.6.2 MARCO-GE

Η δημοσίευση [17]από την Noy Cohen-Shapira παρουσιάζει ένα πρωτοποριακό AutoML σύστημα το MARCO-GE, σχεδιασμένο να αυτοματοποιεί την διαδικασία επιλογής ενός αλγορίθμου

συσταδοποίησης. Σε αυτή τη προσέγγιση τα σύνολα δεδομένων μετατρέπονται σε γράφους. Αρχικά σε όλα τα σύνολα δεδομένων εφαρμόζεται η τεχνική PCA μειώνοντας τις διαστάσεις των χαρακτηριστικών. Έπειτα οι παρατηρήσεις αντιμετωπίζονται ως κόμβοι και σχηματίζονται βεβαρημένες ακμές μεταξύ τους χρησιμοποιώντας τη μετρική cosine similarity πάνω στα χαρακτηριστικά που προέκυψαν από την PCA. Οι γράφοι παίρνουν την τελική τους μορφή αφαιρώντας τις ακμές που έχουν βάρος μικρότερο ενός κατωφλιού με την τιμή 0.9. Στη συνέχεια εξάγονται χαρακτηριστικά κόμβων μέσω του αλγορίθμου DeepWalk και αξιοποιούνται νευρωνικά δίκτυα για γράφους, συγκεκριμένα τα Graph Convolutional Neural Networks, μέσω των οποίων εξάγονται χαρακτηριστικά γράφων (graph embeddings). Τέλος τα embeddings θα χρησιμοποιηθούν ως σύνολο εκπαίδευσης για ένα μετα-μοντέλο XGBoost το οποίο είναι υπεύθυνο για τη πρόβλεψη της κατάταξης των αλγορίθμων συσταδοποίησης.

Χρησιμοποιήθηκαν 210 σύνολα δεδομένων, πάνω στα οποία αξιολογήθηκαν 17 αλγόριθμοι συσταδοποίησης σε 10 διαφορετικές ενδογενείς μετρικές. Διεξάχθηκαν πειράματα για κάθε κατάταξη αλγορίθμων που προέκυψε από τις διαφορετικές μετρικές αλλά και για μια μέση κατάταξη στην οποία λήφθηκαν υπόψιν όλες οι παραπάνω κατατάξεις.

2.6.3 Μετα-Χαρακτηριστικά Βασισμένα στην Απόσταση

Οι D.G. Ferrari και L. N. de Castro παρουσιάζουν μια καινοτόμα προσέγγιση στα πλαίσια του Meta-Learning. Σε αυτή την εργασία [18]οι συγγραφείς δημιουργούν ένα διάνυσμα d το οποίο περιέχει τις ευκλείδειες αποστάσεις μεταξύ όλων των παρατηρήσεων που ανήκουν σε ένα σύνολο δεδομένων D:

$$d = [d_{1,2}, \dots, d_{i,j}, \dots, d_{n-1,n}]$$

όπου $d_{i,j} = dist(X_i^{(D)}, X_j^{(D)})$ η ευκλείδεια απόσταση της παρατήρησης i με την j.

Στη συνέχεια το διάνυσμα κανονικοποιείται στο διάστημα [0,1] και υπολογίζονται κάποιες μετρικές περιγραφικής στατιστικής, οι οποίες χρησιμοποιούνται ως μετα-χαρακτηριστικά για την περιγραφή του συνόλου δεδομένων. Συγκεκριμένα υπολογίζονται οι παρακάτω 19 στατιστικές μετρικές:

Πίνακας 1: Μετα-Χαρακτηριστικά της προσέγγισης των Ferrari και de Castro.

Α/Α	Μετα-Χαρακτηριστικά
1	Μέσος όρος του d
2	Διακύμανση του d
3	Τυπική απόκλιση του d
4	Ασσυμετρία του d
5	Κύρτωση του d
6	% των τιμών στο διάστημα $[0,0.1]$
7	% των τιμών στο διάστημα $(0.1\ ,0.2]$
8	% των τιμών στο διάστημα $(0.2\ ,0.3]$
9	% των τιμών στο διάστημα $(0.3\ ,0.4]$
10	% των τιμών στο διάστημα $(0.4\ ,0.5]$
11	% των τιμών στο διάστημα $(0.5\ ,0.6]$
12	% των τιμών στο διάστημα $(0.6\ ,0.7]$
13	% των τιμών στο διάστημα $(0.7\ ,0.8]$
14	% των τιμών στο διάστημα $(0.8\ ,0.9]$
15	% των τιμών στο διάστημα $(0.9\ ,1]$
16	% των τιμών με απόλυτη τυπική τιμή στο διάστημα $[0,1)$
17	% των τιμών με απόλυτη τυπική τιμή στο διάστημα $[1,2)$
18	% των τιμών με απόλυτη τυπική τιμή στο διάστημα $[2,3)$
19	% των τιμών με απόλυτη τυπική τιμή στο διάστημα $[3,+\infty)$

Σε αυτή τη προσέγγιση, 84 σύνολα δεδομένων, τα οποία συλλέχθηκαν από το αποθετήριο UCI, αξιολογήθηκαν σε 7 αλγορίθμους συσταδοποίησης, των οποίων η απόδοση μετρήθηκε από 10 ενδογενείς μετρικές.

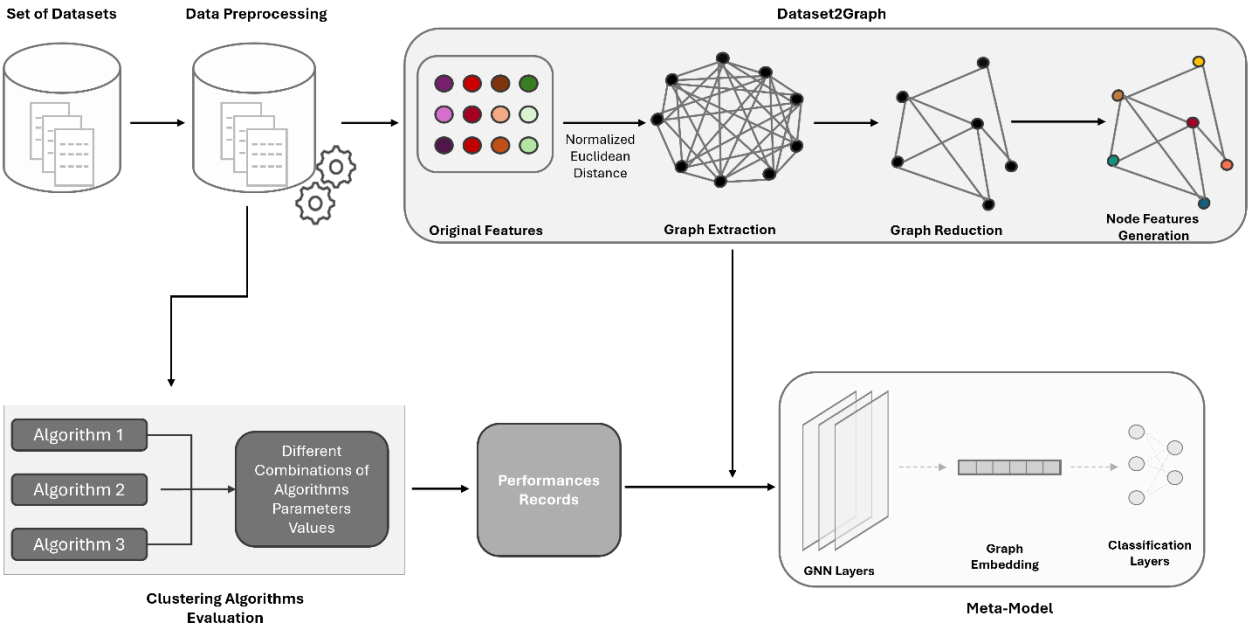
Κεφάλαιο 3 - Μεθοδολογία

Το σύστημα AutoML που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας στοχεύει στην επίλυση του προβλήματος της επιλογής του βέλτιστου αλγορίθμου συσταδοποίησης. Η προσέγγιση αυτή εμπνεύστηκε από την εργασία του MARGO-GE, όπου οι συγγραφείς εκμεταλλεύτηκαν τις πανίσχυρες δυνατότητες που παρέχουν τα νευρωνικά δίκτυα για γράφους για την επίλυση του παραπάνω προβλήματος. Αυτή η έρευνα επεκτείνεται στην παρούσα υλοποίηση, δημιουργώντας νέους τρόπους εξαγωγής και διαχείρισης γράφων αλλά και εξερευνώντας νέες αρχιτεκτονικές νευρωνικών δικτύων.

Το σύστημα που αναπτύχθηκε διαχωρίζεται σε δύο φάσεις. Τη φάσης της εκπαίδευσης, όπου γίνονται οι απαραίτητες προεργασίες με απώτερο σκοπό την δημιουργία του τελικού μοντέλου, το οποίο θα μάθει να επιλύει προβλήματα επιλογής αλγορίθμου. Στο δεύτερο στάδιο το στάδιο online, το μοντέλο θα κάνει προβλέψεις πάνω σε εργασίες, δηλαδή σε γράφους που δεν έχει ξαναδεί.

3.1 Στάδιο Εκπαίδευσης και Αρχιτεκτονική

Σε αυτή την υποενότητα περιγράφονται λεπτομερώς όλα τα βήματα του σταδίου εκπαίδευσης. Τα βήματα που συμμετέχουν σε αυτή τη διαδικασία είναι η επεξεργασία και ο καθαρισμός των αρχικών δεδομένων, η αξιολόγηση των αλγορίθμων συσταδοποίησης πάνω στα σύνολα δεδομένων, ο μετασχηματισμός των δεδομένων σε γράφο, η εξαγωγή meta-features και η εκπαίδευση ενός meta-model.



Εικόνα 7: Όλα τα βήματα της διαδικασίας της εκπαίδευσης από την αρχή μέχρι το τέλος.

3.1.1 Προεπεξεργασία Δεδομένων

Η προεπεξεργασία των δεδομένων αποτελεί ένα από τα πιο σημαντικά βήματα στη διαδικασία ανάπτυξης ενός οποιουδήποτε συστήματος μηχανικής μάθησης, καθώς έχει άμεσο αντίκτυπο σε μεταγενέστερα στάδια. Ένα σύνολο δεδομένων γεμάτο θόρυβο, ακραίες τιμές, διπλότυπα και τιμές που λείπουν υστερεί σε ποιότητα και αυτό θα υποβαθμίσει την τελική απόδοση των μοντέλων πρόβλεψης. Γι' αυτό το λόγο στο στάδιο της προεπεξεργασία καθαρίζοντας και προετοιμάζοντας τα σωστά διασφαλίζεται η συνοχή, η αξιοπιστία και η ποιότητα τους.

Σε κάθε σύνολο δεδομένων της συλλογής μας εφαρμόστηκαν με τη σειρά οι παρακάτω ενέργειες προεπεξεργασίας:

1. Τα χαρακτηριστικά που περιείχαν μια μοναδική τιμή σε όλες τις παρατηρήσεις του συνόλου δεδομένων, αφαιρέθηκαν καθώς δεν προσφέρουν καμία πληροφορία.
2. Χαρακτηριστικά με κατηγορικά δεδομένα κωδικοποιήθηκαν σε αριθμητική τιμή.
3. Εάν το ποσοστό των ελλειπουσών τιμών σε ένα χαρακτηριστικό ξεπερνούσε το 30% του συνολικού μεγέθους, τότε αυτό αφαιρείται. Σε τέτοιες περιπτώσεις υπάρχει ανεπαρκής πληροφορία και αν επιχειρήσει κάποιος να χρησιμοποιήσει κάποια μέθοδο αντικατάστασης τιμών, τότε υπάρχει κίνδυνος να εμφανιστεί μεροληψία στα δεδομένα.
4. Στα χαρακτηριστικά που περιλαμβάνουν ελλείπουσες τιμές αλλά δεν ανήκουν στην προηγούμενη κατηγορία εφαρμόστηκε μια τεχνική αντικατάστασης. Οι ελλείπουσες τιμές της κάθε παρατήρησης υπολογίζονται από τη μέση τιμή των k κοντινότερων γειτόνων της. Όπου το k στην παρούσα υλοποίηση είναι ίσο με 10.
5. Τα δεδομένα κανονικοποιήθηκαν στο διάστημα [0,1]. Η κανονικοποίηση ή αλλιώς η κλιμάκωση των δεδομένων παίζει σημαντικό ρόλο στη διαδικασία εκπαίδευσης των μοντέλων. Σε ένα μη-κανονικοποιημένο σύνολο δεδομένων, ένα χαρακτηριστικό με διακύμανση που είναι τάξεις μεγέθους μεγαλύτερη από τις άλλες μπορεί να καταστήσει το μοντέλο ανίκανο να μάθει και από τα υπόλοιπα χαρακτηριστικά, όπως θα αναμενόταν κανονικά.

3.1.2 Αξιολόγηση Απόδοσης Αλγορίθμων Συσταδοποίησης

Στο παρόν βήμα γίνεται η ποσοτικοποίηση της απόδοσης των αλγορίθμων συσταδοποίησης πάνω στα σύνολα δεδομένων και για διαφορετικούς συνδυασμούς τιμών παραμέτρων. Συμβολίζουμε με το γράμμα $D = \{D_1, D_2, \dots, D_n\}$ τη συλλογή των συνόλων δεδομένων, όπου D_i κάποιο σύνολο δεδομένων και $A = \{A_1, A_2, \dots, A_m\}$ το σύνολο με τους αλγορίθμους συσταδοποίησης, όπου A_j κάποιος αλγόριθμος συσταδοποίησης. Επίσης συμβολίζουμε με P_j ένα σύνολο από συνδυασμούς τιμών παραμέτρων που αφορούν των αλγόριθμο A_j, όπου $P_j = \{\left(p_{j1}{}^{(1)}, p_{j2}{}^{(1)}, \dots, p_{jk_j}{}^{(1)}\right), \left(p_{j1}{}^{(2)}, p_{j2}{}^{(2)}, \dots, p_{jk_j}{}^{(2)}\right), \dots, \left(p_{j1}{}^{(h)}, p_{j2}{}^{(h)}, \dots, p_{jk_j}{}^{(h)}\right)\}$ και k_j το πλήθος των παραμέτρων του αλγορίθμου A_j. Το σύνολο όλων των πιθανών συνδυασμών μπορεί να εκφραστεί ως το καρτεσιανό γινόμενο αυτών των συνόλων, δηλαδή:

$$C = D \times \bigcup_{j=1}^{m} (A_j \times P_j)$$

Κάθε στοιχείο του συνόλου C έχει τη μορφή $c = (D_i, A_j, \left(p_{j1}{}^{(h)}, p_{j2}{}^{(h)}, \dots, p_{jk_j}{}^{(h)}\right))$. Τέλος η απόδοση για κάθε συνδυασμό $c \in C$ μπορεί να ποσοτικοποιηθεί με τη χρήση μιας συνάρτησης $Perf((D_i, A_j, \left(p_{j1}{}^{(h)}, p_{j2}{}^{(h)}, \dots, p_{jk_j}{}^{(h)}\right)))$.

Σε δεύτερο χρόνο αφού έχει γίνει ο υπολογισμός της συνάρτησης για όλους τους συνδυασμούς, για κάθε D_i και A_j κρατάμε μόνο τους συνδυασμούς παραμέτρων που είχαν τη μέγιστη απόδοση, δηλαδή:

$$BestPerf\left(D_i, A_j\right) = \max{}_h(Perf((D_i, A_j, \left(p_{j1}{}^{(h)}, p_{j2}{}^{(h)}, \dots, p_{jk_j}{}^{(h)}\right))))$$

Έπειτα αφού έχουμε συγκεντρώσει ανά D_i τις καλύτερες επιδόσεις για κάθε αλγόριθμο A_j δημιουργείται μια κατάταξη από αυτές με βάση την παραπάνω συνάρτηση. Άρα την πρώτη θέση καταλαμβάνει ο αλγόριθμος με τη υψηλότερη απόδοση πάνω στο συγκεκριμένο σύνολο.

3.1.3 Μετατροπή σε Γράφο

Τα σύνολα δεδομένων πινακικής μορφής αποτελούνται από έναν αριθμό παρατηρήσεων και τα χαρακτηριστικά τους, όμως δεν υπάρχει πουθενά η πληροφορία για τις σχέσεις μεταξύ των παρατηρήσεων. Μετατρέποντας ένα σύνολο δεδομένων σε γράφο δίνεται η δυνατότητα καταγραφής αυτών των σχέσεων. Η κάθε μια παρατήρηση αντιπροσωπεύεται ως κόμβος στον παραγόμενο γράφο και οι σχέσεις μεταξύ τους ως ακμές. Η διαίσθηση πίσω από αυτή τη προσέγγιση είναι πως οι γράφοι διαθέτουν πλούσια πληροφορία δομικών ιδιοτήτων, όπως για παράδειγμα οι δομές των τοπικών γειτονιών, ο σχηματισμός κοινοτήτων, οι ομοιότητες μεταξύ κόμβων. Αυτή την πληροφορία είναι πιθανόν να μπορεί ένα πανίσχυρο εργαλείο όπως τα νευρωνικά δίκτυα για γράφους να την κωδικοποιήσει σε ένα embedding και στη συνέχεια να χρησιμοποιηθεί για γίνουν προβλέψεις. Τα μεταγενέστερα στρώματα που είναι υπεύθυνα για την πρόβλεψη θα ανακαλύψουν συσχετίσεις μεταξύ των embeddings και του κατάλληλου αλγορίθμου συσταδοποίησης για κάθε σύνολο δεδομένων.

Η διαδικασία της μετατροπής ενός συνόλου δεδομένων σε γράφο ξεκινά με τη δημιουργία ενός μη κατευθυνόμενου, χωρίς θηλιές, βεβαρημένου γράφου $G = (V, E)$, όπου κάθε κόμβος $v_i \in V$ αντιστοιχεί στην αρχική παρατήρηση x_i του συνόλου. Σε πρώτη φάση ο γράφος είναι πλήρως συνδεδεμένος, δηλαδή ό κάθε κόμβος του είναι συνδεδεμένος με όλους τους υπόλοιπους. Για την άντληση των βαρών των ακμών ακολουθούμε την παρακάτω διαδικασία. Αρχικά υπολογίζονται όλες οι αποστάσεις των παρατηρήσεων ανά δυο. Κάθε σύνολο δεδομένων διαθέτει διαφορετικό πλήθος χαρακτηριστικών, εξαιτίας αυτού για τον υπολογισμό των αποστάσεων των παρατηρήσεων χρησιμοποιήθηκε μια μετρική η οποία δεν επηρεάζεται από τη διαφορά των διαστάσεων. Η μετρική αυτή είναι μια παραλλαγή της ευκλείδειας απόστασης. Στο πρώτο βήμα υπολογίζεται η ευκλείδεια απόσταση μεταξύ δύο παρατηρήσεων x_i, x_j:

$$Euclidean\_Dist(x_i, x_j) = \sqrt[2]{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{in} - x_{jn})^2}$$

όπου n το πλήθος των διαστάσεων του συνόλου δεδομένων και x_{ik} η τιμή της παρατήρησης i στο χαρακτηριστικό k. Έπειτα αυτή η τιμή κανονικοποιείται στο διάστημα $[0,1]$ διαιρώντας τη με την τετραγωνική ρίζα του πλήθους των διαστάσεων:

$$Normalized\_Euclidean\_Dist(x_i, x_j) = \frac{Euclidean\_Dist(x_i, x_j)}{\sqrt[2]{n}}$$

Στους γράφους τις κοντινές παρατηρήσεις θέλουμε να τις αναπαραστήσουμε με έναν ισχυρό δεσμό, δηλαδή με ακμές υψηλού βάρους. Όμως όσο πιο κοντά είναι δύο σημεία στον ευκλείδειο χώρο τόσο πιο μικρή είναι η τιμή της απόστασης τους. Γι' αυτό το λόγο το τελικό βάρος τις κάθε ακμής διαμορφώθηκε από τον τύπο:

$$Edge\_Weight(v_i, v_j) = 1 - Normalized\_Euclidean\_Dist(x_i, x_j)$$

Άρα όσο μικρότερη η απόσταση μεταξύ δύο παρατηρήσεων τόσο ισχυρότερος ο δεσμός των αντίστοιχων κόμβων. Όσο πιο κοντά στο 1 είναι το βάρος της ακμής τόσο πιο ισχυρός ο δεσμός, ενώ κοντά στο 0 θεωρείται ασθενής.

3.1.4 Απλοποίηση των Γράφων

Όπως αναφέρθηκε στο προηγούμενο υποκεφάλαιο, ο γράφος που παράγεται είναι πλήρως συνδεδεμένος. Αν χρησιμοποιηθούν γράφοι με τέτοιες πυκνές δομές σαν είσοδο στα GNN, είναι μεγάλη η πιθανότητα να υπάρξουν επιπτώσεις στην ικανότητα των μοντέλων να επεξεργαστούν αποτελεσματικά την πληροφορία και να τη γενικεύσουν. Ένα κύριος λόγος είναι ότι χάνεται η έννοια της τοπικής δομής, δηλαδή της γειτονιάς. Αφού πια όλος ο γράφος αποτελεί μια μεμονωμένη γειτονία, ένα GNN θα αδυνατεί να αναγνωρίσει τοπικά μοτίβα και συσχετίσεις μεταξύ γειτονιών. Επίσης τέτοιες δομές μπορούν να εντείνουν φαινόμενα όπως αυτό του oversmoothing[19], όπου τα embeddings όλων των κόμβων συγκλίνουν σε παρόμοιες τιμές, καθιστώντας δύσκολο το

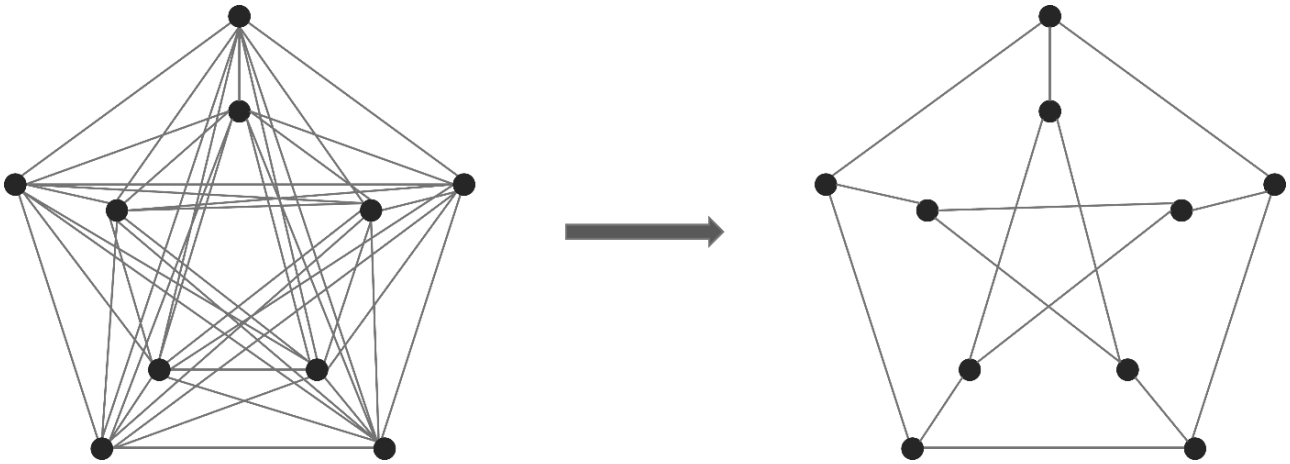
διαχωρισμό τους. Πέρα όμως από τα ποιοτικά προβλήματα που μπορεί να προκύψουν, δημιουργούνται και προβλήματα υπολογιστικής πολυπλοκότητας. Το πλήθος των ακμών ενός πλήρως συνδεδεμένου γράφου ισοδυναμεί με:

$$|S| = \frac{|V|(|V|-1)}{2}$$

Επομένως ένας γράφος με 100 κόμβους θα έχει 4,950 ακμές ενώ ένας με 200 κόμβους θα έχει 19,900. Αυτό σημαίνει πως ο αριθμός των ακμών αυξάνεται εκθετικά όσο αυξάνονται οι κόμβοι, άρα όσο μεγαλύτερη η κλίμακα ενός γράφου τόσο μεγαλύτερο το υπολογιστικό κόστος.

Μια αποτελεσματική λύση για την αντιμετώπιση των προαναφερθέντων προβλημάτων είναι η εφαρμογή τεχνικών απλοποίησης [20]. Η χρήση αυτών των μεθόδων βοηθάει στην απόκτηση μιας πιο απλουστευμένης μορφής της δομής του γράφου, διατηρώντας παράλληλα τις φασματικές και βασικές του τοπικές ιδιότητες. Δύο από τις πιο γνωστές κατηγορίες τεχνικών είναι η αραίωση του ακμών (sparsification) και η σύμπτυξη κόμβων (coarsening) [21][22].

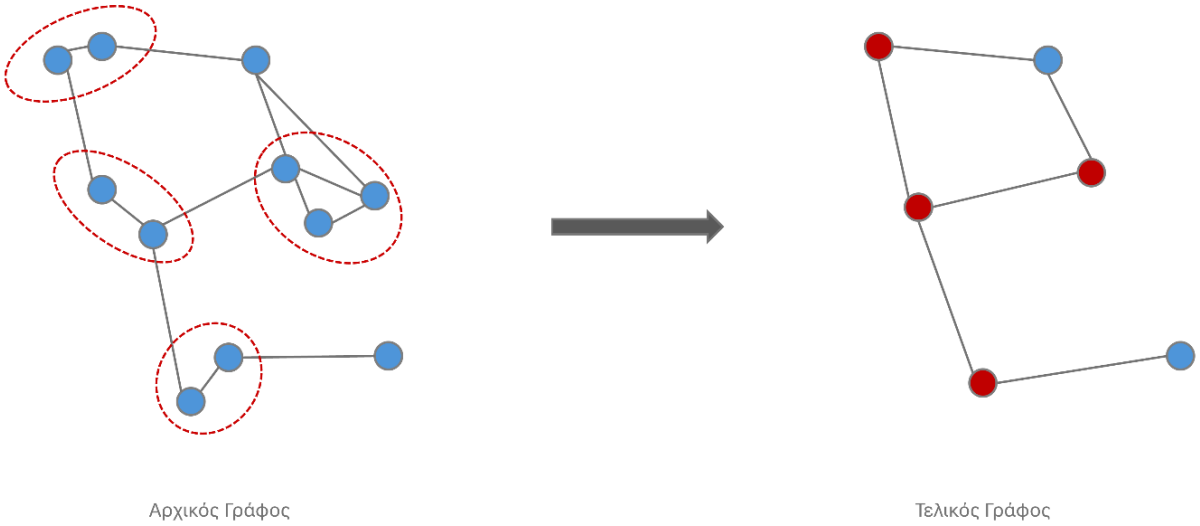
Οι μέθοδοι αραίωσης ακμών επικεντρώνονται στην αφαίρεση ακμών βασισμένες σε κάποια κριτήρια, με απώτερο σκοπό τη διατήρηση των πιο σημαντικών. Εφαρμόζοντας μια τέτοια μέθοδο σε έναν γράφο $G = (V, E)$ θα επιστραφεί ένας υπογράφος $G' = (V, E')$, όπου $E' \subseteq E$. Διατηρώντας τις πιο σημαντικές ακμές, καταφέρνουμε να εξαλείψουμε τον θόρυβο που υπάρχει στην αρχική πυκνή δομή, βοηθώντας ένα GNN μοντέλο να επικεντρωθεί στις ουσιαστικές συνδέσεις και να αποφύγει εμφάνισης του φαινομένου oversmoothing [23]. Επίσης παραμένουν σχεδόν ανεπηρέαστες οι φασματικές ιδιότητες του γράφου. Με τον όρο φασματικές ιδιότητες εννοούμε τις ιδιότητες που προκύπτουν από τον πίνακα Laplace. Ένα παράδειγμα είναι οι ιδιοτιμές του πίνακα που δείχνουν πόσο καλά είναι συνδεδεμένος ένας γράφος.



Εικόνα 8: Τεχνική αραίωση ακμών.

Η εφαρμογή μιας μεθόδου σύμπτυξης κόμβων σε ένα γράφο $G = (V, E)$ θα δημιουργήσει μια νέα απεικόνιση του γράφου $G_c = (V_c, E_c)$, όπου $|V_c| < |V|$ και $|E_c| < |E|$. Συγκεκριμένα σκοπός της τεχνικής είναι η δημιουργία μιας μικρότερης αναπαράστασης του αρχικού γράφου , η οποία είναι πιο διαχειρίσιμη και διατηρεί τις αρχικές του ιδιότητες. Σε αντίθεση με την προηγούμενη κατηγορία, το αποτέλεσμα δεν είναι ένας υπογράφος, καθώς δεν γίνεται επιλογή κόμβων αλλά ένωση. Μέσα από μια επαναληπτική διαδικασία επιλέγονται βάση κάποιων κριτηρίων ζευγάρια κόμβων που πρέπει να ενωθούν, έως ότου φτάσουμε στο επιθυμητό πλήθος κόμβων. Κάθε ζευγάρι θα σχηματίσει έναν υπερκόμβο. Οι υπερκόμβοι κληρονομούν τις ακμές των παιδιών τους με νέα βάρη, τα οποία είναι ο μέσος όρος των δυο προηγούμενων. Παρόμοια είναι τα οφέλη και σε αυτή την τεχνική καθώς μειώνεται ο θόρυβος και μπορούν να αποτυπωθούν πιο ξεκάθαρες δομές, που θα κλιμακώσουν την απόδοση των νευρωνικών δικτύων [24]. Τέλος πρέπει να είμαστε προσεκτικοί, διότι η υπερβολική

σύμπτυξη κόμβων μπορεί να οδηγήσει σε απώλεια πληροφορίας κάτι που θα έχει συνέπειες στην απόδοση των μοντέλων GNN μεταγενέστερα.



Εικόνα 9: Τεχνική σύμπτυξης κόμβων. Στον τελικό γράφο με κόκκινο χρώμα απεικονίζονται οι υπερκόμβοι.

Κάθε φορά που ένας γράφος απλοποιείται με κάποια από τις παραπάνω τεχνικές, δημιουργείται και ένας νέος πίνακας γειτνίασης που να εκφράζει τη νέα συνδεσιμότητα. Στην περίπτωση της αραίωσης μηδενίζονται τα στοιχεία των ακμών που απαλείφονται χωρίς να επηρεάζονται οι διαστάσεις του πίνακα. Αντίθετα ο πίνακας γειτνίασης που θα προκύψει μετά τη σύμπτυξη κόμβων θα είναι ένα τετραγωνικός πίνακας μικρότερων διαστάσεων από τον προηγούμενο και τα βάρη των ακμών προκύπτουν όπως αναφέραμε προηγουμένως από aggregation συναρτήσεις.

3.1.4.1 Ορισμός Κατωφλιού

Για την απλοποίηση των γράφων ακολουθήθηκε μια διαδικασία η οποία αποτελείται από τρία βήματα. Το πρώτο βήμα εφαρμόζεται αναγκαστικά σε κάθε προσέγγιση για υπολογιστικούς λόγους. Αφορά μια άπληστη τεχνική αραίωσης, όπου ορίζεται ένα κατώτατο όριο βάρους (κατώφλι) και στη συνέχεια απαλείφεται κάθε ακμή με βάρος μικρότερο του κατώτατου ορίου. Συγκεκριμένα το νέο σύνολο ακμών του αραιωμένου γράφου είναι:

$$E' = \{(u, v) \in E \mid Edge_{Weight(u,v)} \geq Weight\_Threshold\}$$

Όλοι αυτοί οι μετασχηματισμοί αποτυπώνονται στον πίνακα γειτνίασης του γράφου, μηδενίζοντας τα στοιχεία που δεν ικανοποιούν την παραπάνω συνθήκη. Ένα τέτοιο είδος απλοποίησης όμως μπορεί να οδηγήσει στη δημιουργία απομονωμένων κόμβων ή ακόμα χειρότερα σε ένα μη συνεκτικό γράφο. Οι μη συνεκτικοί γράφοι δυσχεραίνουν την αποτελεσματική διαβίβαση μηνυμάτων μεταξύ κόμβων, κάτι που μπορεί να έχει επιπτώσεις στο τελικό embedding.

Ένα προληπτικό μέτρο που λήφθηκε έτσι ώστε να μην εξάγονται μη συνεκτικοί γράφοι είναι ο υπολογισμός του Maximum Spanning Tree (MaxST). Μια παραλλαγή του αλγορίθμου του Kruskal για το Minimum Spanning Tree [26], ενός αλγορίθμου που επιστρέφει έναν πλήρως συνεκτικό υπογράφο του αρχικού, που δεν περιέχει κύκλους. Στόχος είναι να σχηματιστεί μια δομή δέντρου με το ελάχιστο δυνατό συνολικό βάρος ακμών. Η παραλλαγή του Maximum Spanning Tree επιστρέφει έναν υπογράφο με δομή δέντρου με το μέγιστο δυνατό συνολικό βάρος ακμών. Αυτό επιτυγχάνεται πολλαπλασιάζοντας όλα τα βάρη των ακμών με -1, εφαρμόζοντας των αλγόριθμο του Kruskal και στην συνέχεια ξανά πολλαπλασιάζοντας τα βάρη των ακμών με -1. Το MaxST αποτελεί την ραχοκοκαλιά του αρχικού γράφου, επομένως αν στο πρώτο βήμα αφαιρεθούν ακμές που ανήκουν σε αυτό, τότε θα ξανά προστεθούν διατηρώντας με αυτό τον τρόπο τη συνεκτικότητα.

Με την διατήρηση των ακμών του MaxST παράλληλα διατηρούνται και οι απομονωμένοι κόμβοι που προέκυψαν στην αρχική απλοποίηση, των οποίων η αφαίρεση δεν επηρεάζει τη συνεκτικότητα του γραφήματος. Οι κόμβοι αυτοί είναι όσοι έχουν μόνο ένα γείτονα και η ακμή που τους ενώνει έχει βάρος μικρότερο από το προκαθορισμένο κατώφλι. Επειδή αντιμετωπίζονται ως θόρυβος, θα πρέπει να εντοπιστούν και να εξαιρεθούν από τον υπογράφο.

3.1.4.2 Αραίωση Ακμών και Σύμπτυξη Κόμβων

Αφού αφαιρεθούν οι ακμές του γράφου που δεν ικανοποιούν το κατώτατο όριο και οι μεμονωμένοι κόμβοι, προχωράμε στο δεύτερο βήμα. Τα σενάρια σε αυτό το βήμα είναι τρία. Θα χρησιμοποιηθεί είτε ένας αλγόριθμος αραίωσης, είτε ένας αλγόριθμος σύμπτυξης κόμβων ή ένας συνδυασμός αλγορίθμων από κάθε κατηγορία με σκοπό την περαιτέρω απλοποίηση του γράφου. Σε αντίθεση με το προηγούμενο βήμα, οι ακμές ή οι κόμβοι που θα απλοποιηθούν επιλέγονται με κάποιο κριτήριο από τον εκάστοτε αλγόριθμο και όχι με τον ορισμό ενός κατωφλιού.

Κάποιοι αλγόριθμοι αραίωσης που δοκιμάστηκαν, ενέχουν κινδύνους απαλοιφής ακμών που θα διαταράξουν τη συνδεσιμότητα του γράφου. Για τον λόγο μετά την ολοκλήρωσή τους γινόταν ο έλεγχος με το MaxST. Παρακάτω δίνεται η περιγραφή τους:

- **k-Neighbor:** Για κάθε κόμβο επιλέγονται οι k ακμές με πιθανότητα ανάλογη των βαρών τους. Σε περίπτωσης που κάποιος κόμβος έχει πλήθος ακμών μικρότερο του k, τότε συμπεριλαμβάνονται όλες οι ακμές. Η παράμετρος k πρέπει να οριστεί από τον χρήστη.
- **Rank Degree:** Σε αυτή τη μέθοδο [27] επιλέγεται αρχικά ένα τυχαίο σύνολο κόμβων, οι οποίοι ονομάζονται ‘σπόροι’. Στη συνέχεια για κάθε σπόρο ξεχωριστά, ταξινομούνται οι γείτονές τους σε φθίνουσα σειρά με τον βαθμό τους. Ένα ποσοστό των ακμών που ενώνουν τους σπόρους με τους κορυφαίους γείτονες της κατάταξης ενσωματώνονται στον αραιωμένο γράφο. Οι κορυφαίοι γειτονικοί κόμβοι χρησιμεύουν ως οι νέοι ‘σπόροι’ και η διαδικασία επαναλαμβάνεται έως ότου ο νέος υπογράφος έχει την επιθυμητή πυκνότητα. Η πυκνότητα ορίζεται ως το κλάσμα του πλήθους των ακμών διά του πλήθους των κόμβων και εισέρχεται ως παράμετρος στον αλγόριθμο. Ο Ranked Degree μεροληπτεί υπέρ των κόμβων με υψηλό βαθμό καθώς θεωρούνται κομβικές, οπότε τείνει να διατηρεί ακμές που προσπίπτουν σε αυτούς.
- **Local Degree:** Παρόμοιος με τον προηγούμενο αλγόριθμο, ο Local Degree [28] διατηρεί τις ακμές που προσπίπτουν σε κόμβους με υψηλό βαθμό αλλά με ένα πιο ντετερμινιστικό τρόπο. Για κάθε κόμβο $u \in V$, ενσωματώνονται στον υπογράφο οι ακμές που τους ενώνουν με τους κορυφαίους $\deg(u)^{\alpha}$ γείτονες που κατατάσσονται με βάση το βαθμός τους σε φθίνουσα σειρά. Το $\alpha \in [0,1]$ ελέγχει το βαθμό στον οποίο αραιώνεται ο γράφος. Ο αλγόριθμος διασφαλίζει ότι κάθε κόμβος θα έχει τουλάχιστον μία ακμή, διατηρώντας έτσι τη συνεκτικότητα του γράφου.
- **t-Spanner:** Ο αλγόριθμος [29] δημιουργεί έναν αραιό υπογράφο απαλείφοντας ακμές που ενώνουν ένα ζευγάρι κόμβων $u, v \in V$, αν υπάρχει ένα μονοπάτι που τις ενώνει έμμεσα και ικανοποιεί τη συνθήκη $Shortest\_Path\_Weight(u,v) \leq t * Edge\_Weight(u,v)$. Το t δηλώνει έναν παράγοντα τεντώματος. Η διαδικασία ξεκινά από έναν τυχαίο κόμβο και ολοκληρώνεται έως ότου έχουν εξεταστεί όλες οι ακμές.
- **L-Spar:** Ο αλγόριθμος [30] είναι μια παραλλαγή του Local Degree. Η μόνη τους διαφορά είναι ότι η ταξινόμηση των γειτονικών κόμβων γίνεται με βάση της ομοιότητας Jaccard, η οποία ορίζεται ως:

$$Jaccard\_Similarity(v,u) = \frac{|N(v) \cap N(u)|}{|N(v) \cup N(u)|}$$

όπου N το σύνολο των γειτόνων ενός κόμβου. Εστιάζοντας στις τοπικές ομοιότητες μεταξύ των κορυφών, αυτός ο αλγόριθμος μπορεί να παρέχει μια πιο ακριβή αναπαράσταση του αρχικών τοπικών ιδιοτήτων.

Μετά τη διαδικασία της αραίωσης των γράφων ακολουθεί η διαδικασία της σύμπτυξης κόμβων. Οι αλγόριθμοι αυτής της κατηγορίας δέχονται σαν είσοδο το πλήθος κόμβων που θα έχει ο νέος γράφος και είναι επαναληπτικοί. Σε κάθε επανάληψη επιστρέφουν μια λίστα από ζευγάρια κόμβων που

υπολογίζονται βάση ενός κριτηρίου και στη συνέχεια συμπτύσσονται σε ένα νέο υπερ-κόμβο. Οι αλγόριθμοι ολοκληρώνονται μόλις ο νέος γράφος φτάσει το επιθυμητό πλήθος κόμβων. Παρακάτω δίνεται η περιγραφή τους:

- **Heavy Edge Matching:** Αποτελεί μια απλή άπληστη προσέγγιση [31], η οποία συγχωνεύει έναν κόμβο με τον γείτονά του, με τον οποίο διατηρεί την πιο βαριά ακμή. Ο αλγόριθμος ξεκινάει σαρώνοντας όλες τις ακμές του γράφου κατά φθίνουσα σειρά βάσει των βαρών τους. Οι κόμβοι που αποτελούν τα άκρα των ακμών, δημιουργούν ένα ζευγάρι κόμβων. Αν έστω ένας από τα δύο άκρα υπάρχει ήδη σε κάποιο ζευγάρι που δημιουργήθηκε προηγουμένως, τότε ο αλγόριθμος συνεχίζει στην επόμενη ακμή. Μόλις ελεγχθούν όλες, τότε όλα τα ζευγάρια συγχωνεύονται και δημιουργούν τους νέους υπερκόμβους. Μετά τη συγχώνευση τα βάρη των ακμών ενημερώνονται ώστε να αντικατοπτρίζουν τη νέα δομή.
- **Algebraic Distance:** Σύμφωνα με αυτή την ιδέα [32] υπολογίζονται οι αλγεβρικές αποστάσεις $s_{v,u}$ για κάθε ζευγάρι κόμβων $(v,u) \in E$. Οι αποστάσεις αυτές εξαρτώνται από δύο παραμέτρους, το ω μια παράμετρο χαλάρωσης και το k τον αριθμό των επαναλήψεων. Καθώς το $k \to \infty$, οι αλγεβρικές αποστάσεις συγκλίνουν στο μηδέν. Η ταχύτητα με την οποία μια αλγεβρική απόσταση $s_{v,u}$ συγκλίνει στο μηδέν είναι ένας δείκτης του πόσο ισχυρή είναι η σύνδεση μεταξύ του v και του u.
- **Local Variation (Edges):** Ο αλγόριθμος [21] βασίζεται στην συγχώνευση των κόμβων των οποίων τα βάρη των ακμών παρουσιάζουν ελάχιστη μεταβολή σε σύγκριση με την τοπική γειτονιά τους. Η μέθοδος αυτή αξιολογεί τη διαφορά στα βάρη ακμών μεταξύ ενός κόμβου και των γειτόνων του, συγχωνεύοντας εκείνους με σχετικά παρόμοια βάρη. Στόχος είναι η διατήρηση των τοπικών δομικών ιδιοτήτων μειώνοντας ταυτόχρονα το μέγεθος του γράφου.

3.1.5 Εξαγωγή Χαρακτηριστικών σε Επίπεδο Κόμβων

Μετά την απλοποίηση του γράφου σειρά έχει το βήμα της εξαγωγής χαρακτηριστικών των κόμβων. Τα αρχικά χαρακτηριστικά των παρατηρήσεων πριν μετατραπούν σε κόμβο, χρησιμοποιούνται μόνο για τον υπολογισμό των βαρών των ακμών, μετά από αυτό το βήμα δεν ξανά λαμβάνονται υπόψιν. Ένας από τους λόγους που τα αφήνουμε εκτός της υπόλοιπης διαδικασίας είναι ότι οι διαστάσεις των χαρακτηριστικών διαφέρουν από σύνολο σε σύνολο και ένα μοντέλο GNN απαιτεί όλοι οι κόμβοι εισόδου να διαθέτουν το ίδιο πλήθος χαρακτηριστικών. Επίσης οι διαφορές στα μεγέθη των τιμών τους, αλλά και ο τομέας στον οποίο αναφέρονται είναι κάποιοι επιπλέον λόγοι που δικαιολογούν την απόφασή μας.

Οι δυο βασικές πληροφορίες που χρησιμοποιούνται σαν είσοδο στα μοντέλα με αρχιτεκτονικές νευρωνικών δικτύων για γράφους είναι η συνδεσιμότητα και τα χαρακτηριστικά των κόμβων. Η πληροφορία της συνδεσιμότητας προέρχεται από τον ήδη διαθέσιμο πίνακα γειτνίασης, όμως τα χαρακτηριστικά σε επίπεδο κόμβων δεν υπάρχουν μέχρι στιγμής. Γι' αυτό το λόγο εξάγουμε χαρακτηριστικά προκαθορισμένης διάστασης, που πηγάζουν μέσα από το γράφο και εκφράζουν τις δομικές ιδιότητες των κόμβων. Ο βαθμός, η εκκεντρότητα, ο τοπικός συντελεστής συσταδοποίησης είναι κάποια από τα χαρακτηριστικά τα οποία εμπεριέχουν τοπική αλλά και γενική πληροφορία για τους κόμβους. Ενσωματώνοντας τέτοιου τύπου χαρακτηριστικά σε ένα GNN, μπορούμε να το βοηθήσουμε να κατανοήσει καλύτερα την τοπολογία των γράφων. Σύμφωνα με τη δημοσίευση [25] τα δομικά χαρακτηριστικά ενισχύουν σημαντικά την απόδοση των GNN.

3.1.6 Μέτα-Χαρακτηριστικά και Μέτα-Μοντέλο

Σε αυτό το βήμα αναπτύσσονται και εκπαιδεύονται μοντέλα βαθιάς μάθησης, τα οποία είναι υπεύθυνα για την εξαγωγή μετα-χαρακτηριστικών και την πρόβλεψη του καλύτερου αλγορίθμου συσταδοποίησης. Τα μοντέλα αυτά δέχονται σαν είσοδο μια συλλογή από σύνολα δεδομένων που έχουν μετασχηματιστεί σε γράφους, $Graph\_Collection = \{G_1, G_2, \dots, G_n\}$. Για όλα τα στοιχεία της συλλογής έχει ακολουθηθεί ακριβώς η ίδια διαδικασίας απλοποίησης και εξαγωγής χαρακτηριστικών σε επίπεδο κόμβων. Κάθε $G_i = (X_i, Edgelist_i, EdgeWeights_i, l_i)$ αποτελείται από:

1) Έναν πίνακα X_i που περιέχει τα χαρακτηριστικά των κόμβων, διαστάσεων $n \times d$, όπου $n = |V_i|$ και d το πλήθος των χαρακτηριστικών.
2) Μια $Edgelist_i$, είναι ένας διαφορετικός τρόπος καταγραφής των ακμών. Είναι μια λίστα από λίστες που περιέχουν ζεύγη κόμβων αναπαριστώντας με αυτό τον τρόπο την ακμή που υπάρχει μεταξύ τους.
3) Ένα διάνυσμα $EdgeWeights_i$ με τα βάρη των ακμών m διαστάσεων, όπου $m = |E|$.
4) Η τιμή l_i αναφέρεται στην ετικέτα του γράφου. Η ετικέτα είναι ο καλύτερος αλγόριθμος συσταδοποίησης που έχει προκύψει από το βήμα της αξιολόγησης, ο οποίος έχει κωδικοποιηθεί σε αριθμητική τιμή.

Όσο αφορά την αρχιτεκτονική των GNN μοντέλων στην αρχή βρίσκεται το στρώμα εισόδου (input layer), το οποίο τροφοδοτείται με τον πίνακα X_i, την $Edgelist_i$ και αν χρειάζεται η πληροφορία του βάρους των ακμών στους υπολογισμούς, τότε συμπεριλαμβάνεται και το διάνυσμα $EdgeWeights_i$. Στα επόμενα κρυφά στρώματα λαμβάνει χώρα η διαδικασία της διαβίβασης μηνυμάτων, όπου τα χαρακτηριστικά του κάθε κόμβου επαναυπολογίζονται, ενσωματώνοντας σε αυτά μέσω ενός aggregation και την πληροφορία των γειτόνων. Κάθε στρώμα ορίζεται ως εξής:

$$H_i^{(k+1)} = \sigma(AGGREGATE\left(H_i^{(k)}, Edgelist_i, W^{(k)}\right))$$

όπου το $H^{(k)}$ είναι ο πίνακας των χαρακτηριστικών των κόμβων που εξήχθη από το στρώμα k, $AGGREGATE$ είναι η συνάρτηση διαβίβασης του μηνύματος η οποία διαφέρει με βάση τον τύπο αρχιτεκτονικής του GNN, W^k ένας πίνακας με τα βάρη εκμάθησης και σ μια συνάρτηση ενεργοποίησης. Ισχύει πως $H^0 = X_i$. Πρέπει να τονιστεί πως η λίστα $Edgelist$ παραμένει αμετάβλητη σε όλα τα στρώματα του δικτύου.

Για να μπορέσει να γίνει η πρόβλεψη σε επίπεδο γράφου, εφαρμόζεται μια διαδικασία pooling στα χαρακτηριστικά των κόμβων, τα οποία συνοψίζονται σε ένα embedding που περιγράφει ολόκληρο το γράφο.

$$Graph\_Emdedding_i = Average\_Pooling(H_i^{(K)})$$

όπου $H^{(K)}$ τα χαρακτηριστικά των κόμβων που προκύπτουν από το τελευταίο στρώμα και $Average\_Pooling$ μια aggregation συνάρτηση που επιστρέφει το μέσο όρο κάθε χαρακτηριστικού. Το embedding του γράφου είναι ουσιαστικά τα μετα-χαρακτηριστικά της υλοποίησης, που εκφράζουν το αρχικό σύνολο δεδομένων.

Στη συνέχεια υπάρχουν κάποια στρώματα τα οποία εφαρμόζουν κάποιους γραμμικούς μετασχηματισμούς στο embedding του γράφου και τέλος το στρώμα εξόδου στο οποίο γίνεται η πρόβλεψη με τον παρακάτω τρόπο:

$$\widehat{l_i} = max\ (softmax(f(Graph\_Emdedding_i)))$$

όπου f η συνάρτηση των γραμμικών μετασχηματισμών και $softmax$ μία μαθηματική συνάρτηση, η οποία μετατρέπει το διάνυσμα σε κατανομή πιθανοτήτων. Το στοιχείο με την μεγαλύτερη πιθανότητα ταυτίζεται με την πρόβλεψη του μοντέλου.

Είναι σημαντικό να σημειωθεί πως όλα τα GNN είναι αμετάβλητα σε μεταθέσεις, αυτό σημαίνει πως η έξοδος τους δεν επηρεάζεται από τη σειρά με την οποία θα εισαχθούν οι κόμβοι ενός γράφου. Αυτή η ιδιότητα εξασφαλίζει πως τα συγκεκριμένα μοντέλα αντιμετωπίζουν τους γράφους βάση της δομής τους και όχι της αυθαίρετης ευρετηρίασης των κόμβων.

3.2 Online Στάδιο

Στο online στάδιο γίνονται προβλέψεις πάνω σε σύνολα δεδομένων τα οποία δεν έχει ξαναδεί προηγουμένως το μοντέλο. Τα μοντέλο που χρησιμοποιήθηκε είναι αυτό που εκπαιδεύτηκε από τη συλλογή γράφων στο προηγούμενο στάδιο και παρουσίασε την καλύτερη απόδοση. Αρχικά τα νέα σύνολα δεδομένων μετατρέπονται σε γράφο και στη συνέχεια επιδέχονται ακριβώς την ίδια διαδικασία απλοποίησης και εξαγωγής χαρακτηριστικών με αυτή της συλλογής εκπαίδευσης. Κάθε

νέος γράφος εισάγεται στο εκπαιδευμένο μοντέλο, όπου γίνεται η εξαγωγή των embedding τους και στη συνέχεια η πρόβλεψη του κατάλληλου αλγορίθμου συσταδοποίησης.

3.3 Προδιαγραφές Σχεδιασμού Συστήματος AutoML

Το παρόν σύστημα αυτοματοποιημένης μηχανικής μάθησης αναπτύχθηκε με τη χρήση της γλώσσας προγραμματισμού Python. Η επιλογή αυτής της γλώσσα οφείλεται στη μεγάλη ποικιλία εργαλείων και βιβλιοθηκών που διαθέτει, κατάλληλες για τη διαχείριση δεδομένων με μορφή γράφου και την ανάπτυξη μοντέλων πρόβλεψης.

Για τη διαδικασία της μετατροπής των συνόλων δεδομένων σε γράφο και την μετέπειτα απλοποίηση τους αναπτύχθηκε μια βιβλιοθήκη με το όνομα Dataset2Graph. Οι γράφοι στο Dataset2Vec εκφράζονται μέσω ενός πίνακα γειτνίασης. Η βιβλιοθήκη περιέχει βελτιστοποιημένες συναρτήσεις που υλοποιήθηκαν από το μηδέν με τη βοήθεια της NumPy, ένα πακέτο κατάλληλο για τη διαχείριση πινάκων και πράξεων πάνω σε αυτούς. Επίσης αξιοποιούνται και βιβλιοθήκες όπως το NetworKit για την αραίωση των γράφων και το NetworkX για την εξαγωγή χαρακτηριστικών σε επίπεδο κόμβων.

Η ανάπτυξη των νευρωνικών δικτύων για γράφους έγινε με τη βοήθεια της βιβλιοθήκης PyTorch Geometric. Διαθέτει προκατασκευασμένα στρώματα GNN όπως τα GCN, GAT και GIN, οι οποίες αποτελούν state of the art αρχιτεκτονικές στον τομέα της βαθιάς μάθησης.

Κεφάλαιο 4 - Πειραματισμός και αποτελέσματα

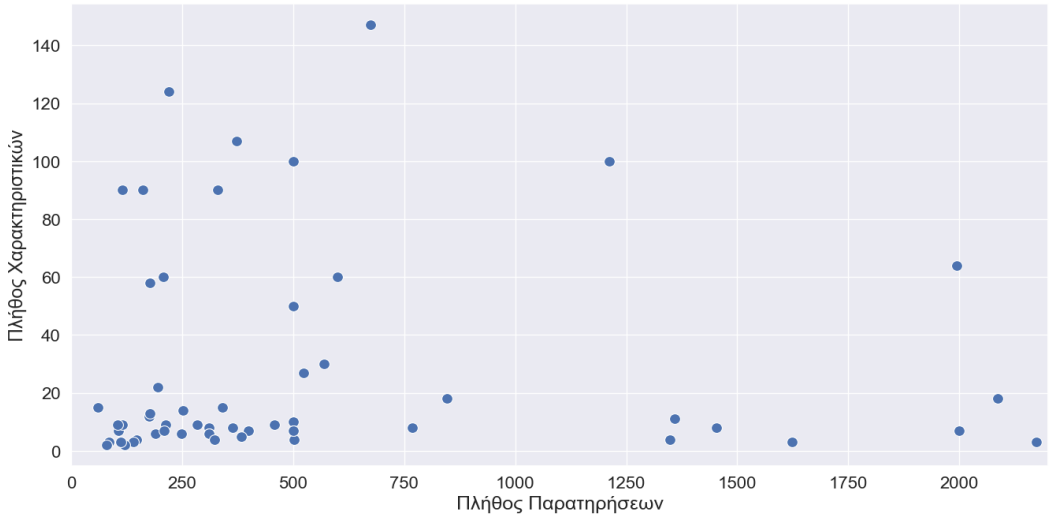
Σε αυτό το κεφάλαιο παρουσιάζεται η πειραματική μελέτη που διενεργήθηκε για την αξιολόγηση του συστήματος αυτοματοποιημένης μηχανικής μάθησης με τη χρήση πραγματικών δεδομένων. Καλύπτονται όλες οι τεχνικές λεπτομέριες, όπως τα σύνολα δεδομένων που συλλέχθηκαν, οι αλγόριθμοι συσταδοποίησης, οι διάφοροι συνδυασμοί τεχνικών απλοποίησης των γράφων που χρησιμοποιήθηκαν για την παραγωγή συλλογών εκπαίδευσης και οι διαφορετικές αρχιτεκτονικές νευρωνικών δικτύων για γράφους.

Αρχικά γίνεται μια αξιολόγηση των μοντέλων πάνω σε συλλογές γράφων, που απλοποιήθηκαν με διαφορετικό τρόπο η κάθε μια. Στη συνέχεια αυτές οι συλλογές εμπλουτίζονται με χαρακτηριστικά σε επίπεδο κόμβων και επαναλαμβάνετε η διαδικασία της αξιολόγησης των μοντέλων. Επίσης παρουσιάζονται και τα αποτελέσματα των μοντέλων πάνω στην αρχική συλλογή γράφων, η οποία δεν έχει υποστεί καμία επεξεργασία. Τέλος η προσέγγισή μας συγκρίνεται με τον κύριο ανταγωνιστή της, το MARCO-GE.

4.1 Σύνολα Δεδομένων

Η διεξαγωγή των πειραμάτων έγινε σε μια συλλογή με 50 σύνολα πραγματικών δεδομένων, το οποία ήταν διαθέσιμα στα ηλεκτρονικά αποθετήρια UCI Machine Learning Repository, Kaggle και OpenML. Τα σύνολα δεδομένων αναφέρονται σε ένα ευρύ φάσμα τομέων, όπως η ιατρική, τα οικονομικά, το περιβάλλον και εμφανίζουν διαφορές ως προς το πλήθος των παρατηρήσεων και των χαρακτηριστικών. Επιπρόσθετα είναι διαθέσιμη και η πληροφορία της ετικέτας για τα δεδομένα.

Τα σύνολα δεδομένων που χρησιμοποιήθηκαν στην πειραματική διαδικασία είναι τα appendicitis, blood_transfusion, breast_cancer_coimbra, breast_cancer_wisconsin, breast_cancer_wisconsin_diagnostic, breast_tissue, chscase_census2, chscase_census5, Engine1, forest_type_mapping, fri_c2_500_50, fri_c3_500_10, fri_c3_500_50, fri_c4_500_100, iris, jEdit_4, leaf, libras, lupus, mu284, robot_failures_lp4, robot_failures_lp5, seeds, sonar, synthetic_control, triazines_cl, urban_land_cover, banknote_authentication, hill_valley, image_segmentation, mfeat_karhunen, newton_hema, parkinsons, planning_relax, prnn_fglass, quake, rabe_266, vehicle, vertebral_column_2classes, vertebral_column_3classes, vinnie, visualizing_environmental, visualizing_galaxy, volcanoes_a2, wine, winequality_red, winequality_white, wireless_indoor_localization, yeast.



Εικόνα 10: Διαστάσεις των συνόλων δεδομένων. Ο οριζόντιος άξονας αναπαριστά το πλήθος των παρατηρήσεων ενώ ο κάθετος το πλήθος των χαρακτηριστικών.

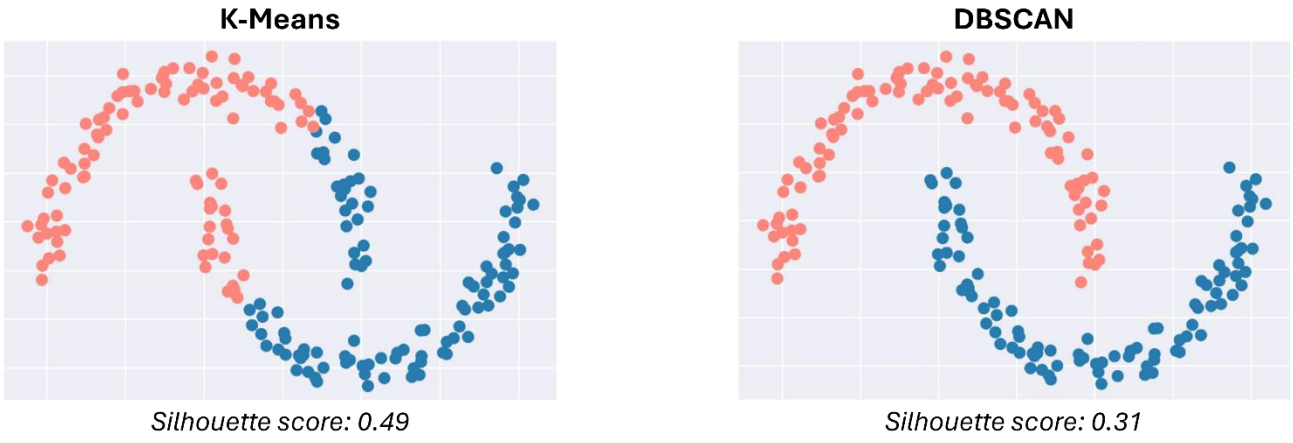
4.2 Αλγόριθμοι Συσταδοποίησης και Μετρική Αξιολόγησης Απόδοσης

Πάνω στα 50 σύνολα δεδομένων αξιολογήθηκε η απόδοση τριών αλγορίθμων συσταδοποίησης. Επιλέχθηκε ένας από κάθε οικογένεια αλγορίθμων, δηλαδή ο K-Means από την οικογένεια των centroid-based, ο DBSCAN από την οικογένεια των density-based και ο Agglomerative από την οικογένεια των connectivity-based. Μια εξαντλητική αναζήτηση των προαναφερθέντων αλγορίθμων εφαρμόστηκε πάνω στα σύνολα δεδομένων, για διαφορετικούς συνδυασμούς τιμών στις παραμέτρους τους. Σε 25 σύνολα δεδομένων απέδωσε καλύτερα ο K-Means, σε 15 ο DBSCAN και σε 10 ο Agglomerative.

Πίνακας 2: Παράμετροι αλγορίθμων συσταδοποίησης

Αλγόριθμος	Παράμετροι Αλγορίθμου
K-Means	n_clusters: Αριθμός των τελικών συστάδων
DBSCAN	eps: Η μέγιστη απόσταση μεταξύ δύο παρατηρήσεων για να θεωρηθεί ότι ανήκουν στην ίδια γειτονιά min_samples: Ο ελάχιστος αριθμός παρατηρήσεων σε μια γειτονιά για να θεωρηθεί το σημείο πυρήνας
Agglomerative	n_clusters: Αριθμός των τελικών συστάδων linkage: Κριτήριο σύνδεσης (π.χ. το κριτήριο ward ελαχιστοποιεί τη διακύμανση των συστάδων που συγχωνεύονται)

Η αξιολόγηση των αποδόσεων των αλγορίθμων συσταδοποίησης μπορείς να γίνει με δύο τρόπους. Με τη χρήση ενδογενών μετρικών και με τη χρήση εξωγενών. Όμως οι ενδογενείς μετρικές τείνουν να ευνοούν τους αλγορίθμους που χρησιμοποιούν παρόμοια συνάρτηση βελτιστοποίησης με αυτές. Ένα χαρακτηριστικό παράδειγμα είναι η μετρική silhouette score που μετράει τη συνοχή μιας συστάδας και τον διαχωρισμό της από τις υπόλοιπες. Η συγκεκριμένη ενδογενής μετρική έχει συνήθως υψηλότερη βαθμολογία για σφαιρικές συστάδες.



Εικόνα 11: Αξιολόγηση των αλγορίθμων K-means και DBSCAN με τη ενδογενή μετρική silhouette score.

Όπως φαίνεται και στην *Εικόνα 11*, παρόλο που ο τρόπος με τον οποίο ομαδοποίησε ο K-means τα δεδομένα είναι φαινομενικά χειρότερος από του DBSCAN, η επίδοση του με βάση τη συγκεκριμένη μετρική είναι υψηλότερη.

Για τον παραπάνω λόγο χρησιμοποιήθηκε μια εξωγενής μετρική, το Adjusted Rand Index (ARI). Το ARI προϋποθέτει να υπάρχουν διαθέσιμες στο σύνολο δεδομένων οι ετικέτες των παρατηρήσεων, καθώς μετράει την ομοιότητας τους με τις συστάδες που παράχθηκαν. Ο μαθηματικός τύπος είναι ο εξής:

$$ARI(X,Y) = \frac{\sum_{ij}\binom{n_{x_i y_i}}{2} - \sum_i \binom{n_{x_i}}{2}\sum_j \binom{n_{y_j}}{2}/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_{x_i}}{2} + \sum_j \binom{n_{y_j}}{2}\right] - \left[\sum_i \binom{n_{x_i}}{2}\sum_j \binom{n_{y_j}}{2}\right]/\binom{n}{2}}$$

όπου $X = \{x_1, x_2, \dots, x_i\}$ αντιπροσωπεύει ένα σύνολο από συστάδες που εντόπισε ένας αλγόριθμος συσταδοποίησης, $Y = \{y_1, y_2, \dots, y_j\}$ το σύνολο των ground-truth συστάδων, n ο συνολικός αριθμός παρατηρήσεων, $n_{x_i} = |x_i|$ και $n_{x_i y_i} = |x_i \cap y_j|$. Οι τιμές του ARI βρίσκονται στο διάστημα $[-1, 1]$, όπου το 1 σημαίνει πλήρης ταύτιση, ενώ -1 καμία ταύτιση.

4.3 Δημιουργία Συλλογών με Γράφους

Από τη μετατροπή ενός συνόλου δεδομένων προκύπτει ένας γράφος πλήρως συνδεδεμένος. Μια συλλογή που περιέχει μόνο πλήρως συνδεδεμένους γράφους και θα χρησιμοποιηθεί σαν είσοδος σε μοντέλα GNN, ενέχει κινδύνους όπως το oversmoothing και επιπρόσθετα τεράστιο υπολογιστικό κόστος. Γι' αυτό το λόγο στην αρχική συλλογή γράφων εφαρμόζονται τεχνικές απλοποίησης με σκοπό την βελτίωση της απόδοσης των GNN, αλλά και την επιτάχυνση των υπολογισμών. Η απλοποίηση διαχωρίζεται σε δύο κατηγορίες την αραίωση και την σύμπτυξη κόμβων, όπως αναφέρθηκε και στο υποκεφάλαιο 3.1.4. και κάθε μια από τις κατηγορίες περιλαμβάνει επιμέρους αλγορίθμους. Οι αλγόριθμοι κάθε κατηγορίας είναι εφικτό να χρησιμοποιηθούν είτε μεμονωμένα είτε σε συνδυασμό με τους αλγορίθμους της άλλης κατηγορίας. Κάθε τέτοια προσέγγιση θα παραγάγει μια διαφορετική συλλογή γράφων. Ο απώτερος σκοπός είναι να βρεθεί ο βέλτιστος συνδυασμός συλλογής γράφων εισόδου και GNN μοντέλου, δηλαδή ένα σύνολο δεδομένων που θα περιέχει όλη την ουσιαστική πληροφορία των γράφων έτσι ώστε να βοηθήσει το μοντέλο να ανακαλύψει συσχετίσεις μεταξύ των δομικών ιδιοτήτων τους και τον καλύτερο αλγόριθμο συσταδοποίησης.

Πίνακας 3: Πολυπλοκότητα Αλγορίθμων Αραίωσης και Δοκιμαστικές Τιμές Παραμέτρων

Αλγόριθμοι Αραίωσης	Παράμετροι και Τιμές	Πολυπλοκότητα*
k-Neighbor	$\#\ \gamma\varepsilon\iota\tau\acute{o}\nu\omega\nu \in [5,10,15]$	$O(\|E\|)$
Rank Degree	$\pi\upsilon\kappa\nu\acute{o}\tau\eta\tau\alpha \in [5,10,15]$	$O(\|E\|) - O(p\|E\|\log(p\|E\|))$
Local Degree	$\pi\upsilon\kappa\nu\acute{o}\tau\eta\tau\alpha \in [5,10,15]$	$O(\|E\|) - O(\|E\|\log(\|E\|))$
t-Spanner	$t \in [5,7,9]$	$O(\|V\|^2\log(\|V\|))$
L-Spar	$\pi\upsilon\kappa\nu\acute{o}\tau\eta\tau\alpha \in [5,10,15]$	$O(k\|E\|)$

*$|V|$ = πλήθος κόμβων, $|E|$ = πλήθος ακμών, p = ποσοστό κλαδέματος, k= πλήθους ελάχιστους κατακερματισμού

Πίνακας 4: Πολυπλοκότητα Αλγορίθμων Σύμπτυξης Κόμβων

Αλγόριθμοι Σύμπτυξης Κόμβων	Παράμετροι και Τιμές	Πολυπλοκότητα
Heavy Edge Matching	$\#\kappa\acute{o}\mu\beta\omega\nu \in [50,100,200,300]$	$O(\|V\| + \|E\|)$
Algebraic Distance	$\#\kappa\acute{o}\mu\beta\omega\nu \in [50,100,200,300]$	$O(k(\|V\| + \|E\|))$
Local Variation (Edges)	$\#\kappa\acute{o}\mu\beta\omega\nu \in [50,100,200,300]$	$O(\|V\| + \|E\|)$

Το σύνολο των συλλογών γράφων που δημιουργήθηκαν από όλους τους πιθανούς συνδυασμούς των αλγορίθμων απλοποίησης ανέρχεται στις 159. Κάποιες από αυτές έχουν μέγεθος μικρότερο των 50 γράφων, διότι κάποιοι συνδυασμοί απλοποιήσεων εκφυλίζουν σε μεγάλο βαθμό κάποιους

γράφους, οι οποίοι αφαιρούνται. Κάθε μια από αυτές χρησιμοποιήθηκε σαν σύνολο εκπαίδευσης στα μοντέλα που αναπτύχθηκαν στο επόμενο στάδιο.

4.4 Δομή Μοντέλων

Κατά τη διαδικασία της εκπαίδευσης σχεδιάστηκαν και υλοποιήθηκαν πολλαπλά μοντέλα πρόβλεψης. Η γενική δομή τους αποτελείται από ένα στρώμα εισόδου, από πολλαπλά στρώματα νευρωνικών δικτύων για γράφους (GNN), ένα στρώμα που εξάγει το graph embedding (readout layer) και έναν στρώμα ταξινόμησης, υπεύθυνο για την τελική πρόβλεψη. Σαν συνάρτηση ενεργοποίησης ενδιάμεσα των στρωμάτων χρησιμοποιείται η ReLU, σαν loss function η Cross Entropy και σαν αλγόριθμος βελτιστοποίησης των βαρών στο στάδιο του back propagation ο Adam.

Η κύρια διαφορά μεταξύ των μοντέλων εντοπίζεται στον τύπο των GNN στρωμάτων. Οι GNN αρχιτεκτονικές που εξετάστηκαν είναι το Graph Convolution Network (GCN), το Graph Attention Network (GAT) και το Graph Isomorphism Network (GIN). Πέρα από τον τύπο των στρωμάτων , έγιναν και πειραματισμοί ως προς το πλήθος των GNN στρωμάτων και των διαστάσεων των graph embeddings. Οι τιμές που εξετάστηκαν για το πλήθος των στρωμάτων είναι 2,3,4 και για τις διαστάσεις των embeddings 32,64,128, 256. Τέλος το πλήθος των εποχών εκπαίδευσης σε όλα τα πειράματα ήταν ίσο με 40.

4.5 Μετρικές Απόδοσης

Για την αξιολόγηση του προβλήματος επιλογής αλγορίθμου χρησιμοποιήθηκαν δύο μετρικές που πηγάζουν από τη μετρική F_1-Score, κατάλληλη για την αξιολόγηση μοντέλων που εκπαιδεύονται πάνω σε σύνολα δεδομένων όπου οι παρατηρήσεις τους δεν έχουν ισορροπημένα πλήθη κλάσεων. Το F_1-Score είναι ο αρμονικός μέσος των precision και recall:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

όπου το $Precision$ μετράει το ποσοστό των αληθώς θετικών προβλέψεων μεταξύ όλων των θετικών προβλέψεων και το $Recall$ μετράει το ποσοστό των θετικών προβλέψεων μεταξύ όλων των πραγματικά θετικών περιπτώσεων. Παρακάτω δίνονται οι τύποι τους:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}, \qquad Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Συγκεκριμένα χρησιμοποιήθηκαν το macro F_1-Score και το weighted F_1-Score. Το macro F_1-Score είναι ο αριθμητικός μέσος των F_1-Score της κάθε κλάσης:

$$F_{1_{macro}} = \frac{F_1^1 + F_1^2 + \cdots + F_1^n}{n}$$

ενώ το weighted F_1-Score είναι το άθροισμα των F_1-Scores για κάθε κλάση, λαμβάνοντας υπόψιν και το βάρος της κάθε κλάσης:

$$F_{1_{weighted}} = weight_1 * F_1^1 + weight_2 * F_1^2 + \cdots + weight_n * F_1^n$$

Η απόδοση των μοντέλων στα πλαίσια των F_1 μετρικών συγκρίθηκε με την απόδοση που θα είχε ένας τυχαίος ταξινομητής. Η απόδοση ενός τυχαίου ταξινομητή με βάση την κατανομή των κλάσεων είναι $F_{1_{macro}}$= 0.333 και $F_{1_{weighted}}$= 0.38.

4.6 Αξιολόγηση GNN Μοντέλων

Σε αυτή την υποενότητα παρουσιάζονται τα αποτελέσματα της αξιολόγησης των GNN μετα-μοντέλων για το πρόβλημα της επιλογής αλγορίθμου. Τα GNN, με τις αρχιτεκτονικές GCN, GAT και GIN, εκπαιδεύονται πάνω στις διάφορες συλλογές δεδομένων που παράχθηκαν και η απόδοσή τους αξιολογήθηκε χρησιμοποιώντας τη μέθοδο leave-one-out, λόγω του μικρού πλήθους των διαθέσιμων γράφων. Στο leave-one-out κάθε μεμονωμένος γράφος της συλλογής χρησιμοποιείται μια φορά ως σύνολο επικύρωσης, ενώ οι υπόλοιπου χρησιμεύουν ως σύνολο εκπαίδευσης. Η

μέθοδος ολοκληρώνεται όταν όλα οι γράφοι της συλλογής έχουν χρησιμοποιηθεί ως σύνολο επικύρωσης.

Μέχρι στιγμής η πληροφορία που φέρει κάθε γράφος είναι η συνδεσιμότητά των κόμβων του και τα βάρη των ακμών. Για τη σωστή λειτουργία των μοντέλων είναι υποχρεωτικό κάθε κόμβος να έχει τουλάχιστον ένα χαρακτηριστικό. Γι' αυτό το λόγο ο βαθμός του κάθε κόμβου, δηλαδή το πλήθος των γειτόνων, χρησιμοποιήθηκε σαν χαρακτηριστικό.

Αρχικά γίνεται η αξιολόγηση των προβλεπτικών GNN μοντέλων που η αρχιτεκτονική τους περιέχει στρώματα GCN. Οι διαφορές μεταξύ των μοντέλων αυτής της κατηγορίας, εντοπίζονται στο πλήθος των κρυφών GCN στρωμάτων αλλά και στις διαστάσεις των graph embeddings που παράγονται από τα κρυφά επίπεδα. Τα μοντέλα αξιολογούνται πάνω στις διάφορες συλλογές απλοποιημένων γράφων, προσπαθώντας να εντοπιστεί ο βέλτιστος συνδυασμός συλλογής και μοντέλου. Οι ονομασίες των συλλογών προέρχονται από τη διαδικασία προεπεξεργασίας που υπέστη το αρχικό σύνολο δεδομένων με γράφους. Για παράδειγμα η συλλογή με όνομα LS10_AG100 σημαίνει ότι οι γράφοι απλοποιήθηκαν με τη μέθοδο αραίωσης L-Spar, με την παράμετρο πυκνότητας ίση με 10 και στη συνέχεια εφαρμόστηκε ο Algebraic Distance με στόχο οι κόμβοι να μειωθούν στους 100.

Πίνακας 5: Οι 5 συλλογές με τις υψηλότερες αποδόσεις ανά μοντέλο με διαφορετικό πλήθος GCN στρωμάτων.

Dataset	#Embeddings	Train F1-Macro	Test F1-Macro	Train F1-Weighted	Test F1-Weighted
1 GCN layer					
LS15_AG300	128	0.372	0.389	0.466	0.481
LS5_VA300	32	0.372	0.391	0.460	0.480
LS15_AG200	128	0.367	0.366	0.458	0.456
LS10_VA300	128	0.362	0.367	0.443	0.447
KN10_AG200	128	0.330	0.342	0.427	0.438
2 GCN layers					
TS7_AG200	32	0.404	0.420	0.483	0.498
TS7_HE200	32	0.402	0.420	0.480	0.498
LS5_VA300	128	0.383	0.389	0.471	0.480
LS15_AG200	256	0.413	0.376	0.485	0.459
LS10_VA300	128	0.364	0.367	0.444	0.447
3 GCN layers					
LS15_AG200	64	0.396	**0.467**	0.475	**0.513**
LS5_VA300	64	0.379	0.389	0.467	0.480
LS10_VA300	128	0.359	0.367	0.446	0.447
LS15_AG300	32	0.361	0.360	0.448	0.446
TS7_HE200	32	0.355	0.363	0.433	0.441

Από τον παραπάνω πίνακα είναι φανερό πως την καλύτερη απόδοση την είχε το GNN με 3 στρώματα GCN το οποίο εκπαιδεύτηκε με τη συλλογή η οποία απλοποιήθηκε με τη μέθοδο αραίωσης L-Spar με την παράμετρο πυκνότητας ίση με 15 και ύστερα οι κόμβοι απλοποιήθηκαν με τη μέθοδο Algebraic Distance μειώνοντας τους σε 200. Αυτός ο συνδυασμός είχε το υψηλότερο σκορ

και στην F_1-Macro και στην F_1-Weighted μετρική στο σύνολο επικύρωσης. Επίσης σύμφωνα με τα αποτελέσματα φαίνεται πως τα μοντέλα αποδίδουν καλύτερα ως επί το πλείστων στις συλλογές όπου οι ακμές απλοποιήθηκαν με την τεχνική L-Spar και οι κόμβοι τους δεν ελαττώθηκαν κάτω από τους 200. Οι συλλογές των οποίων οι κόμβοι είχαν πλήθος 50 ή 100 είχαν αποδόσεις κάτω από το 0.333 στις F_1 μετρικές, το οποίο σημαίνει ότι δεν ξεπερνούν την τυχαιότητα. Αυτό ίσως οφείλεται στο γεγονός ότι αφαιρέθηκε χρήσιμη πληροφορία από τους γράφους, ελαττώνοντας σε τέτοιο βαθμό το πλήθος των κόμβων. Τέλος το πλήθος των graph embeddings δεν δείχνει να παίζει κάποιο σημαντικό ρόλο, καθώς υπάρχουν περιπτώσεις όπου τα μοντέλα που εξάγουν graph embeddings 32 διαστάσεων αποδίδουν το ίδιο καλά με αντίστοιχα μοντέλα των 128 διαστάσεων.

Στη συνέχεια αξιολογήθηκαν τα μοντέλα που στην αρχιτεκτονική τους περιείχαν GAT στρώματα. Λόγω του ότι σε αυτά τα στρώματα το υπολογιστικό κόστος είναι κατά πολύ μεγαλύτερο από αυτό των GCN έγιναν δοκιμές μόνο για πλήθος στρωμάτων 1 και 2.

Πίνακας 6: Οι 5 συλλογές με τις υψηλότερες αποδόσεις ανά μοντέλο με διαφορετικό πλήθος GAT στρωμάτων.

Dataset	#Embeddings	Train F1-Macro	Test F1-Macro	Train F1-Weighted	Test F1-Weighted
1 GAT layer					
LS5_VA200	256	0.243	0.345	0.335	0.411
KN15_AG300	128	0.253	0.348	0.353	0.407
TS7_AG200	32	0.396	0.342	0.470	0.403
LOC15_HE200	256	0.242	0.360	0.338	0.402
KN10_AG300	256	0.221	0.374	0.323	0.397
2 GAT layers					
TS7_HE200	32	0.391	0.381	0.466	**0.455**
LS10_HE200	32	0.383	**0.426**	0.439	0.436
TS7_AG200	128	0.376	0.371	0.449	0.435
LS10_VA300	64	0.444	0.421	0.496	0.429
KN5_VA200	32	0.328	0.364	0.425	0.427

Σε αυτή την κατηγορία των μοντέλων, απέδωσαν καλύτερα αυτά που στην αρχιτεκτονική τους είχαν δύο GAT στρώματα. Δεν παρατηρείται να ξεχωρίζει κάποιος αλγόριθμος απλοποίησης που να ενισχύει τις αποδόσεις του μοντέλου, όπως παρατηρήθηκε στα GCN μοντέλα. Ομοίως όμως και σε αυτή την περίπτωση η υπερβολική μείωση των κόμβων οδηγούσε σε φτωχές αποδόσεις που δεν ξεπερνούν την τυχαιότητα.

Τέλος αξιολογήθηκαν τα μοντέλα με GIN στρώματα. Παρόμοια με τα GAT περιοριστήκαμε στο ίδιο πλήθος στρωμάτων καθώς και σε αυτή την κατηγορία το υπολογιστικό κόστος είναι μεγάλο.

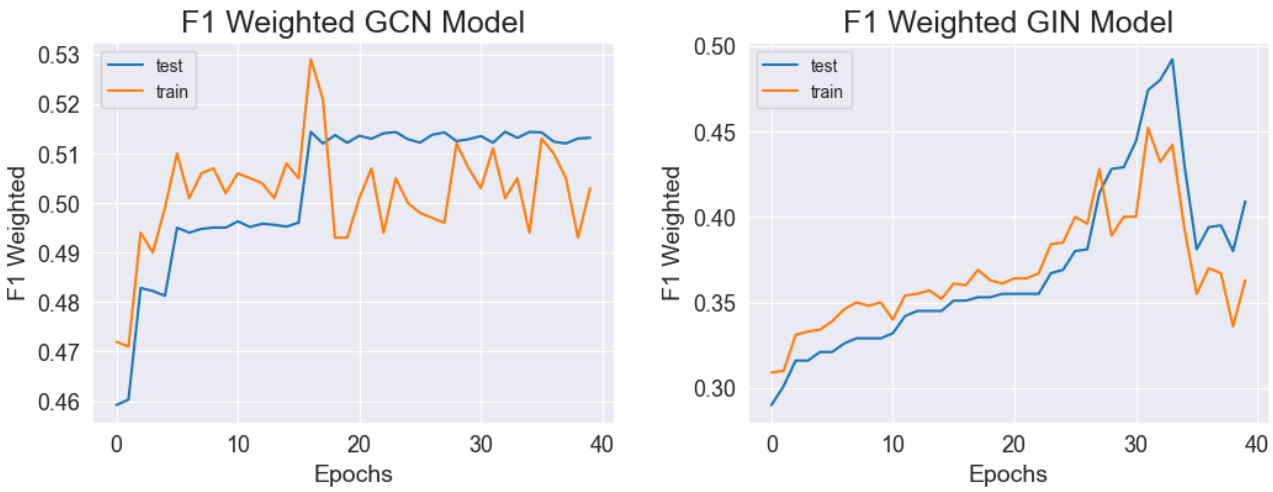
Πίνακας 7: Οι 5 συλλογές με τις υψηλότερες αποδόσεις ανά μοντέλο με διαφορετικό πλήθος GAT στρωμάτων.

Dataset	#Embeddings	Train F1-Macro	Test F1-Macro	Train F1-Weighted	Test F1-Weighted
1 GIN layer					

LOC10_AG300	32	0.213	0.213	0.299	0.298
LS10_HE200	256	0.213	0.213	0.299	0.298
LS10_VA100	128	0.217	0.213	0.302	0.298
LS10_VA300	128	0.213	0.213	0.299	0.298
TS7_AG200	128	0.208	0.208	0.284	0.284
2 GIN layers					
LS15_AG300	128	0.349	0.396	0.441	**0.492**
TS10_HE200	64	0.329	**0.401**	0.400	0.476
TS9_VA300	32	0.293	0.285	0.390	0.381
LS15_AG200	64	0.297	0.257	0.392	0.355
LS10_HE200	128	0.258	0.249	0.339	0.333

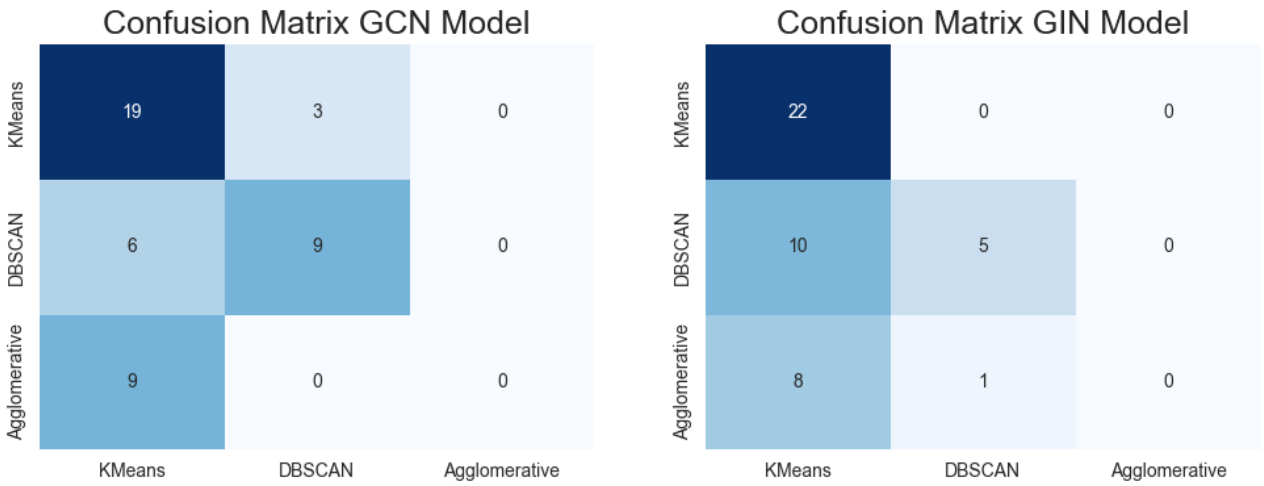
Από τον παραπάνω πίνακα συμπεραίνουμε πως τα μοντέλα με ένα στρώμα GIN δεν είναι ικανά να ξεπεράσουν την τυχαιότητα. Αντίθετα τα μοντέλα με δύο στρώματα GIN έδειξαν πολύ πιο θετικά αποτελέσματα.

Συγκρίνοντας και τις τρεις κατηγορίες μεταξύ τους είναι ξεκάθαρο πως ο νικητής είναι ο συνδυασμός του μοντέλου με 3 στρώματα GCN πάνω που εκπαιδεύτηκε με το σύνολο δεδομένων το οποίο απλοποιήθηκε με της μεθόδους LSpar και Algebraic Distance. Στη δεύτερη θέση έρχεται το μοντέλο με 2 GIN στρώματα το οποίο εκπαιδεύτηκε πάνω στο σύνολο το οποίο απλοποιήθηκε με τις ίδιες ακριβώς μεθόδους και η μόνη αλλαγή είναι ότι σε αυτό το σύνολο οι κόμβοι μειώθηκαν στους 300 και όχι στους 200 όπως στο GCN μοντέλο. Τα μοντέλα GAT δεν καταλαμβάνουν καν την 3η θέση καθώς υπάρχουν και άλλα μοντέλα από τις δύο προαναφερθέντες κατηγορίες που τα ξεπερνούν.



Εικόνα 12: Εξέλιξη αποδόσεων των δυο καλύτερων μοντέλων ανά εποχή.

Κατά τη διάρκεια της εκπαίδευσης το GCN μοντέλο μόλις από την 18η εποχή φαίνεται πως η απόδοσή τους ως προς τη μετρική F_1 Weighted έχει σταθεροποιηθεί σε αντίθεση με το GIN το οποίο έπιασε τη μέγιστη απόδοση του στην 32η εποχή. Αυτό ίσως οφείλεται στο γεγονός πως το GCN ξεκίνησε από την πρώτη εποχή με πολύ υψηλότερη απόδοση σε σχέση με το GIN. Πραγματοποιήθηκαν και πειραματισμοί όπου η εκπαίδευση των δυο παραπάνω μοντέλων ορίστηκε να γίνει σε 100 εποχές, αλλά σύμφωνα με τα αποτελέσματα, οι παραπάνω εποχές δεν τα βοηθούν να βελτιώσουν την απόδοσή τους.



Εικόνα 13: Πίνακες σύγχυσης.

Σύμφωνα με τον πίνακα σύγχυσης και τα δύο μοντέλα είναι ικανά να προβλέπουν σωστά σε υψηλό ποσοστό την κλάση όπου ο K-Means είναι ο βέλτιστος αλγόριθμος. Συγκεκριμένα το GCN μοντέλο κατάφερε να κάνει σωστές προβλέψει στους 19 από τα 22 από τους 22 γράφους, ενώ το GIN μοντέλο κατάφερε να προβλέψει σωστά και τις 22 περιπτώσεις. Όσο αναφορά τις περιπτώσεις όπου οι γράφοι ανήκουν στην κλάση του DBSCAN το GCN μοντέλο κατάφερε να παρουσιάσει καλύτερα αποτελέσματα από το GIN μοντέλο. Από τους 15 συνολικά γράφους που έχουν ως αλγόριθμο με την καλύτερη απόδοση τον DBSCAN το GCN μοντέλο είχε 9 σωστές προβλέψεις ενώ το GIN 6. Και τα δύο μοντέλα τις λάθος τους προβλέψεις τις κατέταξαν στην κλάση του K-Means. Τέλος καμία από τις δυο αρχιτεκτονικές δεν κατάφερε να κάνει έστω μια σωστή πρόβλεψη για τους γράφους που σαν ετικέτα είχανε τον Agglomerative αλγόριθμο. Τα μοντέλα κατέταξαν όλες αυτές τις περιπτώσεις στον αλγόριθμο του K-Means, πέρα από μία.

4.7 Ενίσχυση Μοντέλων με τη Δημιουργία Επιπλέον Χαρακτηριστικών σε Επίπεδο Κόμβων

Στην προηγούμενη υποενότητα τα μοντέλα εκπαιδεύτηκαν πάνω σε κόμβους οι οποίοι διέθεταν μόνο ένα χαρακτηριστικό, το βαθμό τους. Σκοπός μας σε αυτή την υποενότητα είναι η ενίσχυση της απόδοσης των μοντέλων, εξάγοντας περισσότερα χαρακτηριστικά σε επίπεδο κόμβων. Τα νέα αυτά δομικά χαρακτηριστικά χωρίζονται σε δύο κατηγορίες. Αυτά που μέσω ενός κριτηρίου υπολογίζεται πόσο σημαντικός είναι ένας κόμβος μέσα στο γράφημα (node centrality features) και αυτά που περιγράφουν την τοπική δομή του κόμβου. Παρακάτω δίνεται μια περιγραφή για το καθένα:

- Eigenvector Centrality: Είναι μια μετρική που δείχνει πόση επιρροή έχει ένας κόμβος μέσα στο γράφο, λαμβάνοντας υπόψιν τόσο το πλήθος των γειτόνων αλλά και την ποιότητα τους. Ένας κόμβος θεωρείται σημαντικός αν συνδέεται με άλλους σημαντικούς κόμβους. Η κεντρικότητα x_i ενός κόμβου υπολογίζεται από την εξίσωση:

$$x_u = \frac{1}{\lambda} \sum_{v \in N(u)} e_{u,v} x_v$$

  όπου λ μια θετική σταθερά, συνήθως χρησιμοποιείται η μέγιστη ιδιοτιμή του πίνακα γειτνίασης, $e_{u,v}$ το βάρος της ακμής και x το ιδιοδιάνυσμα που αντιστοιχεί στην μέγιστη ιδιοτιμή.

- Betweenness Centrality: Το συγκεκριμένο κριτήριο ποσοτικοποιεί τη σημασία ενός κόμβου με βάση τη συμμετοχή του στα ελάχιστα μονοπάτια ενός γράφου. Κόμβοι με υψηλό betweenness centrality θεωρούνται κρίσιμοι μεσάζοντες καθώς επηρεάζουν τη ροή της πληροφορίας. Υπολογίζεται από τον τύπο:

$$x_u = \sum_{v \neq u \neq t} \frac{\#\ \text{ελάχιστων μονοπατιών μεταξύ των } v \text{ και } t \text{ που περιέχουν το } u}{\#\text{ελάχιστων μονοπατιών μεταξύ των } v \text{ και } t}$$

- Closeness Centrality: Μετρά το πόσο αποτελεσματικά ένας κόμβος μπορεί να έχει πρόσβαση σε όλους τους άλλους κόμβους σε ένα γράφημα. Υπολογίζεται ως το αντίστροφο του μήκους των ελάχιστων μονοπατιών μεταξύ του συγκεκριμένου κόμβου και όλων των υπόλοιπων. Δίνεται από την εξίσωση:

$$x_u = \frac{1}{\sum_{u \neq v} \text{μήκος ελάχιστους μονοπατιού μεταξύ } u \text{ και } v}$$

- Clustering Coefficient: Ποσοτικοποιεί την τάση των κόμβων να σχηματίζουν συστάδες, εξετάζοντας τη σύνδεση μεταξύ των γειτόνων ενός κόμβου.

$$x_u = \frac{\#\ \text{ακμών μεταξύ των γειτόνων του κόμβου } u}{\binom{\#\ \text{γειτόνων του κόμβου } u}{2}}$$

Αυτά τα τέσσερα χαρακτηριστικά καθώς και ο βαθμός κάθε κόμβου χρησιμοποιήθηκαν στην εκπαίδευση των GNN μοντέλων. Αρχικά γίνεται μια σύγκριση των τριών καλύτερων επιδόσεων όπως αυτές προέκυψαν από τον συνδυασμό συνόλου δεδομένων και μοντέλων, για κάθε αρχιτεκτονική. Οι συγκρίσεις έγιναν με βάση τις μετρικές F_1-Macro και F_1-Weighted στο σύνολο επικύρωσης.

Πίνακας 8: Σύγκριση απόδοσης μοντέλων με και χωρίς τη χρήση χαρακτηριστικών σε επίπεδο κόμβου.

Dataset	F1-Macro	F1-Macro Node Features	F1-Weighted	F1-Weighted Node Features
GCN Models				
LS15_AG200	0.467	**0.527**	0.512	**0.547**
TS7_AG200	0.420	**0.457**	0.498	**0.519**
TS7_HE200	0.420	**0.492**	0.498	**0.545**
GAT Models				
TS7_HE200	**0.381**	0.302	**0.455**	0.282
LS10_HE200	**0.426**	0.212	**0.436**	0.298
TS7_AG200	**0.371**	0.366	**0.435**	0.352
GIN Models				
LS15_AG300	**0.396**	0.218	**0.492**	0.321
TS7_HE200	**0.401**	0.256	**0.476**	0.331
TS9_VA300	0.285	**0.403**	0.381	**0.499**

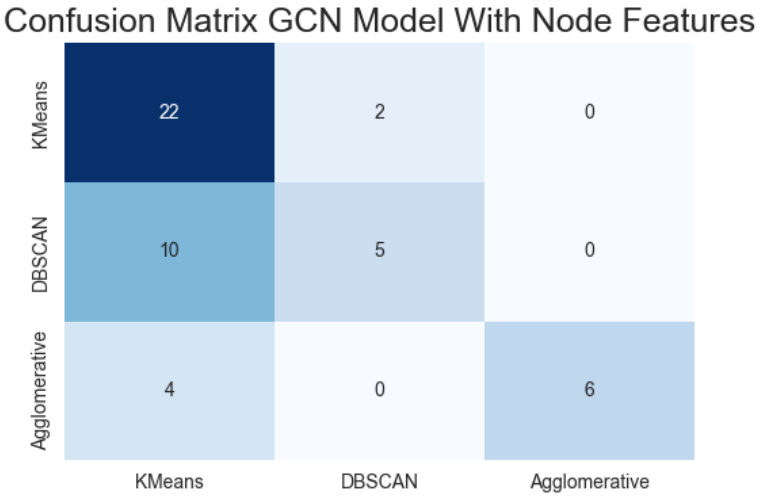
Η προσθήκη των δομικών χαρακτηριστικών των κόμβων στα μοντέλα με αρχιτεκτονική GCN είχε θετική επίδραση αυξάνοντας την απόδοσή τους έως και 0.05 παραπάνω. Αντίθετα στα GAT μοντέλα αυτή η ενέργεια φαίνεται πως είχε αρνητικό αντίκτυπο. Τέλος στα GIN μοντέλα συναντάμε και τις δυο περιπτώσεις, δηλαδή σε κάποιες συλλογές παρατηρείται μείωση της απόδοσης όμως στη συλλογή TS9_VA300 η απόδοση του μοντέλου αυξήθηκε κατά 0.118 και στις δύο F_1 μετρικές.

Στη συνέχεια καθώς παρατηρήθηκε ότι για κάποιες συλλογές αυξάνεται η απόδοση των μοντέλων, έτρεξαν τα ίδια πειράματα με την προηγούμενη υποενότητα με όλες τις συλλογές εμπλουτισμένες με τα νέα χαρακτηριστικά κόμβων.

Πίνακας 9: Αποτελέσματα μοντέλων που εκπαιδεύτηκαν πάνω σε συλλογές εμπλουτισμένες με χαρακτηριστικά κόμβων.

Dataset	#Embeddings	Train F1-Macro	Test F1-Macro	Train F1-Weighted	Test F1-Weighted
GCN Models					
KN10_VA100	128	0.676	**0.630**	0.696	**0.647**
TS7_HE200	32	0.482	0.549	0.537	0.596
TS9_VA300	128	0.389	0.493	0.421	0.552
GAT Models					
TS9_VA300	32	0.399	0.403	0.471	**0.486**
LOC15_VA300	64	0.405	**0.419**	0.446	**0.486**
LS10_VA100	32	0.359	0.494	0.402	**0.486**
GIN Models					
TS9_VA300	32	0.334	**0.404**	0.424	**0.499**
LSp5_VA200	64	0.319	0.359	0.414	0.454
TS7_AG200	32	0.455	0.378	0.497	0.453

Γενικότερα σε όλες τις κατηγορίες μοντέλων, οι αποδόσεις αυξήθηκαν με τη βοήθεια των νέων χαρακτηριστικών και επιτεύχθηκαν καλύτερα αποτελέσματα. Αξιοσημείωτο είναι το γεγονός πως αυτή η κίνηση κατέστησε ικανές και τις συλλογές στις οποίες οι κόμβοι μειώθηκαν στους 100 να βοηθήσουν τα μοντέλα να αυξήσουν σε μεγάλο βαθμό την απόδοσή τους. Είχε αναφερθεί στην προηγούμενη υποενότητα πως οι συλλογές με αντίστοιχη μείωση κόμβων πως δεν επέστρεφαν καλά αποτελέσματα. Την καλύτερη απόδοση μέχρι στιγμής έχει σημειώσει η συλλογή KN10_VA100 πάνω στην οποία εκπαιδεύτηκε ένα GNN μοντέλο με 2 στρώματα GCN τα οποία εξήγαγαν 128 graph embeddings. Ο ίδιος συνδυασμός χωρίς τα δομικά χαρακτηριστικά είχε F_1-Macro ίση με 0.219 και F_1-Weighted ίση με 0.322.



Εικόνα 14: Ο πίνακας σύγχυσης του GCN μοντέλου που εκπαιδεύτηκε με χαρακτηριστικά κόμβων

Ένα ακόμα θετικό αντίκτυπο που είχε η εξαγωγή δομικών χαρακτηριστικών είναι ότι κατάφερε να βοηθήσει τα μοντέλα να μπορέσουν να κάνουν και σωστές προβλέψεις για τους γράφους που ανήκουν στην κλάση του Agglomerative, αλλά μειώθηκε το ποσοστό σωστών προβλέψεων για την κλάση του DBSCAN. Στην προηγούμενη ενότητα κανένα μοντέλο δεν μπόρεσε να κάνει ούτε μια σωστή πρόβλεψη για αυτή την κλάση.

4.8 Επαύξηση Συλλογής Εκπαίδευσης

Ένα ακόμα στάδιο της πειραματικής διαδικασίας είναι να εξεταστεί αν η επαύξηση της συλλογής εκπαίδευσης επηρεάσει θετικά την απόδοση των μοντέλων. Αυτό επιτυγχάνεται εκπαιδεύοντας τα μοντέλα με ένα σύνολο από συλλογές γράφων. Επιλέγοντας για παράδειγμα τρεις διαφορετικές συλλογές, οι οποίες δημιουργήθηκαν με διαφορετικό τρόπο απλοποίησης, όπως η LS10_HE200, η TS9_VA300 και η LS15_AG200 δημιουργούμε μια νέα συλλογή εκπαίδευσης. Επομένως ένα αρχικό σύνολο δεδομένων αναπαρίσταται από τρία διαφορετικά γραφήματα μέσα σε αυτή τη συλλογή. Πίσω από αυτή την προσέγγιση υπάρχει η διαίσθηση πως τα μοντέλα θα μπορέσουν να τροφοδοτηθούν με περισσότερες παραστάσεις απλοποίησης και αυτό θα τα βοηθήσει να δημιουργήσουν καλύτερους συσχετισμούς μεταξύ των απλοποιημένων δομών των γράφων και του καλύτερου αλγορίθμου συσταδοποίησης.

Για την εκπαίδευση των μοντέλων χρησιμοποιήθηκε μια παραλλαγή της μεθόδου Leave One Out. Αντί κάθε φορά το σύνολο επικύρωσης να περιέχει μόνο ένα γράφο, θα περιέχει όλους τους γράφους που αφορούν το αρχικό σύνολο δεδομένων. Στη συνέχεια το μοντέλο θα εκπαιδεύεται με όλους τους υπόλοιπους και θα κάνει προβλέψεις πάνω σε κάθε γράφο του συνόλου επικύρωσης. Η κλάση που θα αποτελέσει την πλειοψηφία των προβλέψεων θα είναι και αυτή που θα ανατεθεί ως τελική κλάση για το σύνολο επικύρωσης.

Στο παρόν πείραμα εκπαιδεύσαμε την κάθε κατηγορία μοντέλων με σύνολα εκπαίδευσης που δημιουργήθηκαν από τις 5 συλλογές με την καλύτερη απόδοση στην κάθε κατηγορία αρχιτεκτονικής. Επίσης έγιναν δοκιμές με και χωρίς τη χρήση των χαρακτηριστικών σε επίπεδο κόμβων.

Πίνακας 10: Αποδόσεις των εκπαιδευμένων μοντέλων στην επαυξημένη συλλογή.

GNN Model	# Layers	Train F1-Macro	Test F1-Macro	Train F1-Weighted	Test F1-Weighted
Without Node Features					
GCN	3	0.387	0.371	0.467	**0.465**
GAT	2	0.354	**0.377**	0.379	0.417
GIN	2	0.357	0.356	0.425	0.453
With Node Features					
GCN	3	0.385	0.435	0.401	0.451
GAT	2	0.343	0.357	0.395	0.418
GIN	2	0.544	**0.584**	0.590	**0.611**

Αρχικά τα μοντέλα που εκπαιδεύτηκαν πάνω στην επαυξημένη συλλογή, χωρίς τη χρήση των δομικών χαρακτηριστικών των κόμβων, δεν μπόρεσαν να ξεπεράσουν σε αποδόσεις όλα τα μοντέλα που εκπαιδεύτηκαν πάνω σε μη-επαυξημένες συλλογές. Αυτό ισχύει και για τις τρεις κατηγορίες μοντέλων. Ύστερα αξιοποιώντας και τα δομικά χαρακτηριστικά των κόμβων, δεν παρατηρούνται σημαντικές βελτιώσεις για τις αρχιτεκτονικές GCN και GAT, όμως για το GIN μοντέλο παρατηρείται αύξηση κατά 0.22 και 0.15 για τις μετρικές F_1-Macro και F_1-Weighted αντίστοιχα. Τα αποτελέσματα του συγκεκριμένου μοντέλου σημειώνουν τη δεύτερη καλύτερη απόδοση μέχρι στιγμής στην πειραματική διαδικασία.

4.9 Αξιολόγηση GNN Μοντέλων σε μη Απλοποιημένους Γράφους.

Στις προηγούμενες υποενότητες όλες οι συλλογές περιείχαν γράφους οι οποίοι είχαν υποστεί απλοποιήσεις σε ακμές και κόμβους μέσω διάφορων μεθόδων. Αυτές οι απλοποιήσεις έγιναν με σκοπό τη διατήρηση της ουσιαστικής δομής του γράφου, απαλείφοντας τον θόρυβο. Όμως η απαλοιφή αυτών των στοιχείων σημαίνει πως αφαιρείται ένα μέρος της πληροφορίας του γράφου χωρίς να γνωρίζουμε αν αυτό είναι κομβικό ή όχι. Για τον παραπάνω λόγο δημιουργήθηκε μια συλλογή η οποία περιέχει γράφους από τους οποίους έχουν αφαιρεθεί μόνο ακμές με τη χρήση ενός κατωφλιού για υπολογιστικούς λόγους, χωρίς κάποιον περεταίρω εκφυλισμό. Το κατώφλι ορίστηκε στην τιμή 0.6 για το βάρος των ακμών.

Πίνακας 11: Αποδόσεις των εκπαιδευμένων μοντέλων στην μη απλοποιημένη συλλογή

GNN Model	# Layers	Train F1-Macro	Test F1-Macro	Train F1-Weighted	Test F1-Weighted
Without Node Features					
GCN	2	0.330	0.362	0.428	**0.461**
GAT	2	0.348	**0.390**	0.446	0.427
GIN	2	0.290	0.267	0.387	0.369
With Node Features					
GCN	2	0.332	**0.343**	0.431	**0.445**
GAT	1	0.341	**0.343**	0.439	0.388
GIN	2	0.255	0.266	0.355	0.367

Βάση των αποτελεσμάτων είναι εμφανές πως η απλοποίηση των γράφων ήταν ένα ουσιαστικό βήμα στην παρούσα προσέγγιση, καθώς λειτούργησε θετικά μη επηρεάζοντας τη θεμελιώδη πληροφορία. Ειδικότερα στα GIN μοντέλα η συλλογή με την πλεονάζουσα δομική πληροφορία οδήγησε στο φαινόμενο του oversmoothing, δηλαδή τα embeddings των γράφων ήταν πολύ όμοια μεταξύ τους, το οποίο είχε ως αποτέλεσμα την χαμηλή απόδοση του μοντέλου.

4.10 Σιαμαία Νευρωνικά Δίκτυα

Τα σιαμαία νευρωνικά δίκτυα (SNN) είναι μια κατηγορία νευρωνικών δικτύων τα οποία σχεδιάστηκαν για τη σύγκριση και την αξιολόγηση της ομοιότητας δυο εισόδων. Η αρχιτεκτονική τους περιέχει δυο πανομοιότυπα υπό-δίκτυα με τις ίδιες παραμέτρους και τα ίδια βάρη. Το μοντέλο δέχεται δύο εισόδους, όπου η κάθε μια εισέρχεται σε ένα υπό-δίκτυο, το οποίο την επεξεργάζεται ανεξάρτητα και εξάγει embeddings. Σε αυτό το σημείο έρχεται η Contrastive loss, μια συνάρτηση που υπολογίζει το σφάλμα βάση της απόστασης. Χρησιμοποιείται για την εκμάθηση embeddings τα οποία αν ανήκουν στην ίδια κλάση, θα έχουν μικρή ευκλείδεια απόσταση μεταξύ τους ενώ αν ανήκουν σε διαφορετική κλάση η ευκλείδεια απόστασή τους θα είναι μεγάλη. Η συνάρτηση δίνεται από τον τύπο:

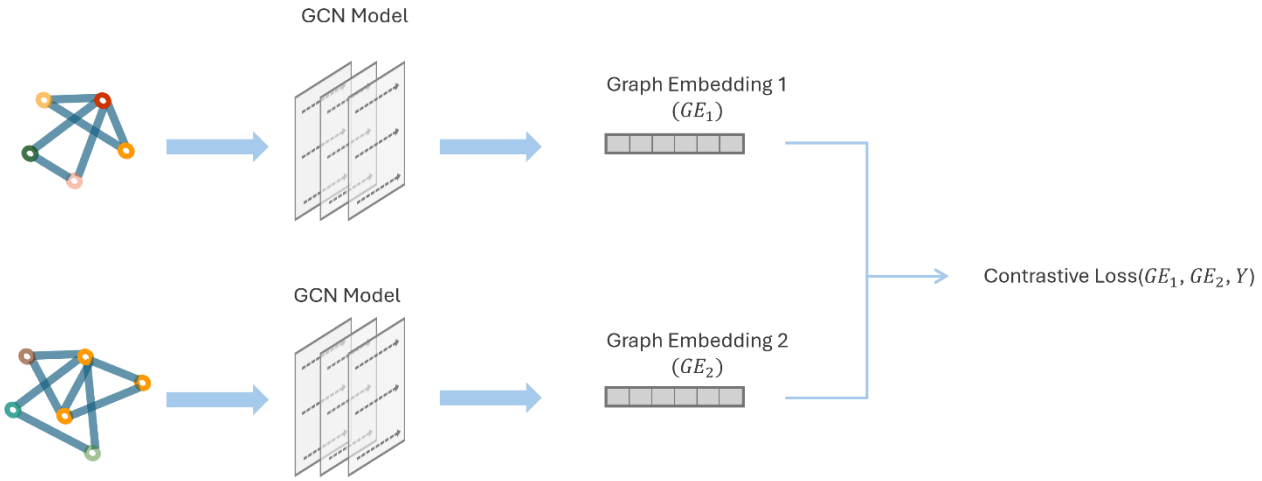
$$\mathcal{L} = \sum_{i \neq j} L(D_i, D_j)$$

όπου

$$L(D_i, D_j) = (1 - Y) * \frac{1}{2} * d(D_i, D_j) + Y * \frac{1}{2} * (\max(0, m - d(D_i, D_j)))^2$$

Το Y αναφέρεται στην κλάση ομοιότητας των συνόλων δεδομένων D, παίρνει την τιμή 0 αν D_i και D_j ανήκουν στην ίδια κλάση, διαφορετικά την τιμή 1. Το $d(D_i, D_j)$ υποδηλώνει την απόσταση μεταξύ των embeddings που προέκυψαν από τα νευρωνικά δίκτυα.

Σκοπός αυτού του πειράματος δεν είναι να υλοποιηθεί άλλο ένα μοντέλο που θα προβλέπει τον βέλτιστο αλγόριθμο συσταδοποίησης, αλλά να εξεταστούν οι ικανότητες αυτής της αρχιτεκτονικής στο πόσο καλά μπορεί να εντοπίσει ομοιότητες ή διαφορές μεταξύ γράφων που ανήκουν στην ίδια ή σε διαφορετική κλάση αντίστοιχα.



Εικόνα 15: Αρχιτεκτονική σιαμαίου νευρωνικού δικτύου με δυο υπό-δίκτυα GCN.

Στην παρούσα υλοποίηση θα αξιοποιήσουμε τις δυνατότητες που μας προσφέρουν τα SNN σε συνδυασμό με τα GNN μοντέλα. Δηλαδή θα αναπτυχθεί ένα σιαμαίο νευρωνικό δίκτυο το οποίο στην αρχιτεκτονική του θα περιέχει δύο GCN μοντέλα ίδιων παραμέτρων και βαρών. Συγκεκριμένα το GCN μοντέλα θα είναι πανομοιότυπα με αυτό από την ενότητα 5.7 το οποίο μέχρι στιγμής είχε τις βέλτιστες επιδόσεις. Αποτελείται από τρία GCN στρώματα που εξάγουν ένα graph embedding 128 διαστάσεων. Επιπρόσθετα η συλλογή εκπαίδευσης θα είναι αυτή που απλοποιήθηκε με τις μεθόδους K-Neighbors με πλήθος γειτόνων ίσο με 10 και Variation Edges μειώνοντας το πλήθος των κόμβων στους 100. Το μοντέλο εκπαιδεύτηκε για 20 εποχές και εμφάνισε το μικρότερο σφάλμα στη 19η εποχή, ίσο με 0.013.

Πίνακας 12: Απόσταση μεταξύ των graph embeddings για διάφορα σύνολα δεδομένων

Σύνολο Δεδομένων 1	Κλάση 1	Σύνολο Δεδομένων 2	Κλάση 2	Ευκλείδεια Απόσταση
seeds	Kmeans	vinnie	Kmeans	0.0654
banknote_authentication	DBSCAN	vehicle	DBSCAN	0.07293
mfeat_karhunen	Agglomerative	libras	Agglomerative	0.097793
appendicitis	Kmeans	blood_transfusion	DBSCAN	0.377209
leaf	Kmeans	lupus	Agglomerative	0.314100
vehicle	DBSCAN	winequality_red	Agglomerative	0.069277

Σύμφωνα με τον παραπάνω πίνακα φαίνεται πως το μοντέλο είναι ικανό να δημιουργήσει graph embeddings τα οποία έχουν μικρή απόσταση, για γράφους ίδιας κλάσης. Επίσης τα graph

embeddings διαφορετικών κλάσεων παρουσιάζουν μεγαλύτερες αποστάσεις μεταξύ τους. Δεν λείπουν όμως και περιπτώσεις, όπως στην τελευταία γραμμή, όπου δύο γράφοι ανήκουν σε διαφορετικές κλάσεις αλλά βρίσκονται πολύ κοντά μεταξύ τους.

4.11 Σύγκριση με το MARCO-GE

Η προσέγγιση της παρούσας διπλωματικής εργασίας είναι πανομοιότυπη με το MARCO-GE καθώς και οι δυο αφορούν την δημιουργία ενός AutoML συστήματος για την επιλογή του βέλτιστου αλγορίθμου συσταδοποίησης, αξιοποιώντας τα GNN. Οι κύριες διαφορές των δύο υλοποιήσεων εμφανίζονται στο στάδιο της μετατροπής ενός συνόλου δεδομένων σε γράφο.

Η διαδικασία που ακολουθεί το MARCO-GE για τη μετατροπή ενός συνόλου δεδομένων σε γράφο είναι δημιουργία ακμών με τη μετρική cosine similarity, αραίωση ακμών με τη χρήση ενός κατωφλιού του οποίου η τιμή είναι ίση με 0.9 και τέλος εξαγωγή χαρακτηριστικών σε επίπεδο κόμβων με τη μέθοδο του DeepWalk. Κάθε ένα από τα παραπάνω βήματα είναι δυνατόν να ενέχει κάποιους κινδύνους που ίσως να έχουν αρνητικό αντίκτυπο στην απόδοση του AutoML συστήματος.

Ξεκινώντας από το τρόπο με τον οποίο εξάγονται οι ακμές, να τονίσουμε πως οι αλγόριθμοι συσταδοποίησης είναι αλγόριθμοι που βασίζονται κυρίως στην απόσταση και όχι στην γωνία που σχηματίζεται μεταξύ δυο σημείων. Επομένως σχηματίζοντας ακμές με το cosine similarity θα δημιουργηθεί ένας γράφος με κόμβους οι οποίοι θα ενώνονται με ακμές μεγάλου βάρους, ενώ οι αρχικές παρατηρήσεις παρουσίαζαν μεγάλη απόσταση μεταξύ τους αλλά σχημάτιζαν μικρή γωνία.

Στο βήμα της αραιοποίησης, ένα κατώφλι στο βάρος των ακμών μπορεί να οδηγήσει σε δύο ανεπιθύμητα σενάρια. Το πρώτο είναι η δημιουργία ενός μη συνεκτικού γράφου. Εάν ο γράφος είναι μη συνεκτικός μπορεί να μην αποτυπωθούν ουσιαστικές σχέσεις μεταξύ των κόμβων στο τελικό embedding, γεγονός που θα παρεμποδίσει την απόδοση του GNN μοντέλου. Το δεύτερο σενάριο αφορά την αφαίρεση πληροφορίας που ίσως είναι σημαντική. Οι πιο ελαφριές ακμές δεν σημαίνει ότι δεν κατέχουν σημαντική δομική πληροφορία, καθώς κάποιες από αυτές θα μπορούσαν να αποτελούν γέφυρες ανάμεσα σε δύο κοινότητες ενός γράφου.

Τέλος η εξαγωγή χαρακτηριστικών σε επίπεδο κόμβων με το DeepWalk βασίζεται σε τυχαίους περιπάτους πάνω στο γράφο. Αυτό σημαίνει πως η μέθοδος αυτή μπορεί ακούσια να αποτύχει να αποτυπώσει σημαντικές δομικές ιδιότητες του γράφου πάνω στα χαρακτηριστικά. Οι τυχαίοι περίπατοι τείνουν να επισκέπτονται συχνότερα κόμβους υψηλού βαθμού, γεγονός που μπορεί να οδηγήσει σε χαρακτηριστικά που υπερτονίζουν αυτούς τους κόμβους. Αυτή η προκατάληψη μπορεί να επισκιάσει τη σημασία των κόμβων χαμηλού βαθμού, οι οποίοι μπορεί να εξακολουθούν να παίζουν ουσιαστικό ρόλο στο γράφημα.

Πίνακας 13: Διαφορές στην διαδικασία εξαγωγής γράφων

Στάδια Εξαγωγής Γράφου	Dataset2Graph	MARCO-GE
Δημιουργία Ακμών	Κανονικοποιημένη Ευκλείδεια Απόσταση	Cosine Similarity
Απλοποίηση του Γράφου	Τεχνικές Αραίωσης και Σύμπτυξης Κόμβων	Κατώφλι Βάρους Ακμών
Εξαγωγή Χαρακτηριστικών Κόμβων	Εξαγωγή Δομικών Χαρακτηριστικών	Εξαγωγή Χαρακτηριστικών με Τυχαίους Περιπάτους

Εντοπίζεται ακόμη μία διαφορά με την προσέγγιση της Cohen Shapira στο βήμα των μοντέλων πρόβλεψης. Το GNN μοντέλο χρησιμοποιείται μόνο για την εξαγωγή των graph embeddings και όχι για την πρόβλεψη του βέλτιστου αλγορίθμου. Στη συνέχεια ένα μετα-μοντέλο XGBoost εκπαιδεύεται με τα graph embeddings και είναι υπεύθυνο για τις τελικές προβλέψεις.

Υλοποιήθηκε το MARCO-GE στη γλώσσα προγραμματισμού Python, σύμφωνα με τη δημοσίευση [17], με απώτερο σκοπό τη σύγκριση του με την προσέγγιση της παρούσας διπλωματικής. Με αυτό τον τρόπο θα εξακριβωθεί εάν ο διαφορετικός τρόπος που επιλέξαμε για την εξαγωγή των γράφων επέφερε καλύτερα αποτελέσματα. Από τη μεριά μας επιλέχθηκε το GCN μοντέλο από το Κεφάλαιο 4.7, το οποίο παρουσίασε τα καλύτερα αποτελέσματα στην πειραματική διαδικασία. Στον παρακάτω πίνακα αναφέρεται με την ονομασία Dataset2Graph.

Πίνακας 14: Σύγκριση αποδόσεων μεταξύ του Dataset2Graph και του MARCO-GE

Μετρικές	**Dataset2Graph**	**MARCO-GE**
F1-Macro	**0.630**	0.373
F1-Weighted	**0.647**	0.438

Είναι εμφανές πως η παρούσα υλοποίηση ξεπερνά και στις δύο μετρικές κατά πολύ το MARCO-GE, εξακριβώνοντας πως οι αλλαγές στον τρόπο εξαγωγής των γράφων είχε θετικό αντίκτυπο. Επίσης κατά τη διάρκεια της εκπαίδευσης του μοντέλου του MARCO-GE το μοντέλο εμφάνισε το φαινόμενο του overfitting.

Κεφάλαιο 5 - Συμπεράσματα και Μελλοντική Έρευνα

Στην παρούσα διπλωματική εργασία διερευνήθηκε το πρόβλημα της αυτοματοποιημένης μηχανικής μάθησης για την πρόβλεψη του αλγορίθμου συσταδοποίησης με την καλύτερη επίδοση για ένα σύνολο δεδομένων. Υλοποιήθηκε ένα σύστημα το οποίο μετατρέπει το σύνολα δεδομένων σε γράφους, με την κατάλληλη επεξεργασία και αξιοποιεί τις δυνατότητες διάφορων αρχιτεκτονικών νευρωνικών δικτύων για γράφους, τα οποία είναι υπεύθυνα για την τελική πρόβλεψη. Δοκιμάστηκαν διάφορες τεχνικές παραγωγής γράφων, αλλά και μοντέλα με διαφορετικές υπερπαραμέτρους και αρχιτεκτονικές, με σκοπό την αξιολόγηση τους και την εύρεση της βέλτιστης προσέγγισης. Ως μετρικές αξιολόγησης χρησιμοποιήθηκαν οι $F_{1_{macro}}$ και $F_{1_{weighted}}$ και αρχικά ο κύριος ανταγωνιστής ήταν ο τυχαίος ταξινομητής.

Η έρευνα μας έδειξε πως η διαδικασία της απλοποίησης των πλήρως συνδεδεμένων γράφων είχε θετικό αντίκτυπο στην απόδοση όλων των μοντέλων. Είναι σημαντικό να υπογραμμιστεί πως γενικότερα τα μοντέλα με αρχιτεκτονική GCN ξεπέρασαν τις αποδόσεις όλων των GAT μοντέλων με τους μηχανισμούς προσοχής καθώς και τα GIN μοντέλα. Επιπλέον η απόδοση των GCN μοντέλων αυξήθηκε ακόμα περισσότερο όταν οι γράφοι εμπλουτίστηκαν με δομικά χαρακτηριστικά κόμβων. Επιπρόσθετα τα αποτελέσματα έδειξαν πως τα μοντέλα που είχαν μεγαλύτερο βάθος, δηλαδή περισσότερα στρώματα, παρουσίασαν καλύτερες αποδόσεις.

Ύστερα από την εύρεση της βέλτιστης προσέγγισης, έγινε και η σύγκριση του συστήματός μας με τον κύριο ανταγωνιστής του, το MARCO-GE. Ένα επίσης σύστημα αυτοματοποιημένης μηχανικής μάθησης, το οποίο προσπαθεί να επιλύσει το ίδιο ακριβώς πρόβλημα με τη χρήση GNN. Η κύρια διαφορά μας βρίσκεται στην επεξεργασία των γράφων. Το μοντέλο μας σημείωσε υψηλότερες αποδόσεις, κάτι που αποδεικνύει πως η προσέγγιση μας αποτυπώνει πιο αποτελεσματικά τις δομικές ιδιότητες των γράφων, βοηθώντας με αυτό τον τρόπο και το GNN να κάνει πιο σωστές προβλέψεις.

Στο μέλλον η έρευνα μας θα επεκταθεί περιλαμβάνοντας πρόσθετες αρχιτεκτονικές GNN για τη δημιουργία μοντέλων, φασματικές και μη. Επίσης θα εξεταστεί αν μια προσέγγιση με τη χρήση Autoencoders, όπως τα VGAEs, καταφέρνουν να κωδικοποιήσουν καλύτερα τους γράφους σε σχέση με τα παραδοσιακά GNN. Επιπλέον, θα γίνει διερεύνηση νέων τεχνικών επεξεργασίας και απλοποίησης γράφων, που θα μπορούν να αποτυπώνουν πιο ξεκάθαρα τις κρίσιμες δομικές πληροφορίες. Τέλος, θα ενσωματωθεί στο σύστημα μας και το βήμα της βελτιστοποίησης των υπερπαραμέτρων του προβλεπόμενου αλγορίθμου συσταδοποίησης.

Κεφάλαιο 6 - Βιβλιογραφικές Αναφορές

[1] X. Du, Y. Cai, S. Wang, and L. Zhang, “Overview of deep learning,” in *Proceedings - 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, YAC 2016*, 2017. doi: 10.1109/YAC.2016.7804882.

[2] K. L. Du, C. S. Leung, W. H. Mow, and M. N. S. Swamy, “Perceptron: Learning, Generalization, Model Selection, Fault Tolerance, and Role in the Deep Learning Era,” 2022. doi: 10.3390/math10244730.

[3] M. Bai and M. Li, “A Presentation of Structures and Applications of Convolutional Neural Networks,” *Highlights in Science, Engineering and Technology*, vol. 61, 2023, doi: 10.54097/hset.v61i.10291.

[4] A. Chinea, “Understanding the principles of recursive neural networks: A generative approach to tackle model complexity,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009. doi: 10.1007/978-3-642-04274-4_98.

[5] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Trans Neural Netw*, vol. 20, no. 1, 2009, doi: 10.1109/TNN.2008.2005605.

[6] P. Bongini, M. Bianchini, and F. Scarselli, “Molecular generative Graph Neural Networks for Drug Discovery,” *Neurocomputing*, vol. 450, 2021, doi: 10.1016/j.neucom.2021.04.039.

[7] H. T. Phan, N. T. Nguyen, and D. Hwang, “Fake news detection: A survey of graph neural network methods,” 2023. doi: 10.1016/j.asoc.2023.110235.

[8] W. Jiang and J. Luo, “Graph neural network for traffic forecasting: A survey,” 2022. doi: 10.1016/j.eswa.2022.117921.

[9] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *34th International Conference on Machine Learning, ICML 2017*, 2017.

[10] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.

[11] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, and Y. Bengio, “Graph attention networks,” in *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, 2018. doi: 10.1007/978-3-031-01587-8_7.

[12] S. Brody, U. Alon, and E. Yahav, “HOW ATTENTIVE ARE GRAPH ATTENTION NETWORKS?,” in *ICLR 2022 - 10th International Conference on Learning Representations*, 2022.

[13] B. Y. Weisfeiler and A. A. Leman, “A reduction of a graph to a canonical form and an algebra arising from this reduction,” *Nauchno-Technicheskaya Informatsia*, vol. 2, no. 9, 1968.

[14] K. Xu, S. Jegelka, W. Hu, and J. Leskovec, “How powerful are graph neural networks?,” in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[15] J. Vanschoren, “Meta-Learning,” 2019. doi: 10.1007/978-3-030-05318-5_2.

[16] Y. Poulakis, C. Doulkeridis, and D. Kyriazis, “AutoClust: A framework for automated clustering based on cluster validity indices,” in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2020. doi: 10.1109/ICDM50108.2020.00153.

[17] N. Cohen Shapira and L. Rokach, “Automatic selection of clustering algorithms using supervised graph embedding,” *Inf Sci (N Y)*, vol. 577, 2021, doi: 10.1016/j.ins.2021.08.028.

[18] D. G. Ferrari and L. N. De Castro, “Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods,” *Inf Sci (N Y)*, vol. 301, 2015, doi: 10.1016/j.ins.2014.12.044.

[19] X. Zhang, Y. Xu, W. He, W. Guo, and L. Cui, “A Comprehensive Review of the Oversmoothing in Graph Neural Networks,” in *Communications in Computer and Information Science*, 2024. doi: 10.1007/978-981-99-9637-7_33.

[20] M. Hashemi, S. Gong, J. Ni, W. Fan, B. A. Prakash, and W. Jin, “A Comprehensive Survey on Graph Reduction: Sparsification, Coarsening, and Condensation,” 2024.

[21] A. Loukas, “Graph reduction with spectral and cut guarantees,” *Journal of Machine Learning Research*, vol. 20, 2019.

[22] J. Chen, Y. Saad, and Z. Zhang, “Graph coarsening: from scientific computing to machine learning,” 2022. doi: 10.1007/s40324-021-00282-x.

[23] T. Hossain, K. M. Saifuddin, M. I. K. Islam, F. Tanvir, and E. Akbas, “Tackling Oversmoothing in GNN via Graph Sparsification: A Truss-based Approach,” Jul. 2024, [Online]. Available: http://arxiv.org/abs/2407.11928

[24] Z. Huang, S. Zhang, C. Xi, T. Liu, and M. Zhou, “Scaling Up Graph Neural Networks Via Graph Coarsening; Scaling Up Graph Neural Networks Via Graph Coarsening,” vol. 10, doi: 10.1145/3447548.

[25] H. Cui, Z. Lu, P. Li, and C. Yang, “On Positional and Structural Node Features for Graph Neural Networks on Non-attributed Graphs,” in *International Conference on Information and Knowledge Management, Proceedings*, 2022. doi: 10.1145/3511808.3557661.

[26] Paryati and K. Salahddine, “The Implementation of Kruskal’s Algorithm for Minimum Spanning Tree in a Graph,” *MATEC Web of Conferences*, vol. 348, 2021, doi: 10.1051/matecconf/202134801001.

[27] E. Voudigari, N. Salamanos, T. Papageorgiou, and E. J. Yannakoudakis, “Rank degree: An efficient algorithm for graph sampling,” in *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016*, 2016. doi: 10.1109/ASONAM.2016.7752223.

[28] M. Hamann, G. Lindner, H. Meyerhenke, C. L. Staudt, and D. Wagner, “Structure-preserving sparsification methods for social networks,” *Soc Netw Anal Min*, vol. 6, no. 1, 2016, doi: 10.1007/s13278-016-0332-2.

[29] S. Baswana and S. Sen, “A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs,” *Random Struct Algorithms*, vol. 30, no. 4, 2007, doi: 10.1002/rsa.20130.

[30] V. Satuluri, S. Parthasarathy, and Y. Ruan, “Local graph sparsification for scalable clustering,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2011. doi: 10.1145/1989323.1989399.

[31] G. Karypis and V. Kumar, “Multilevel Graph Partitioning Schemes,” in *[ICPP’95] International Conference on Parallel Processing*, 1995.

[32] J. Chen and I. Safro, “Algebraic distance on graphs,” *SIAM Journal on Scientific Computing*, vol. 33, no. 6, 2011, doi: 10.1137/090775087.