ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μηχανική Μάθηση για την Ανάλυση Κοινωνικών Δικτύων - Μοντέλα για την Ανίχνευση Ψευδών Ειδήσεων με Συνελικτικά Νευρωνικά Δίκτυα Γράφων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χριστίνα Τσακανίκα

Επιβλέπων : Συμεών Παπαβασιλείου και Ειρήνη Κοιλανιώτη
Καθηγητής ΕΜΠ Ε.ΔΙ.Π ΕΜΠ

Αθήνα, Σεπτέμβριος 2022

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μηχανική Μάθηση για την Ανάλυση Κοινωνικών Δικτύων – Μοντέλα για την Ανίχνευση Ψευδών Ειδήσεων με Συνελικτικά Νευρωνικά Δίκτυα Γράφων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Χριστίνα Τσακανίκα

Επιβλέπων : Συμεών Παπαβασιλείου και Ειρήνη Κοιλανιώτη
Καθηγητής ΕΜΠ Ε.ΔΙ.Π ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30η Σεπτεμβρίου 2022

.....	.....	.....
Συμεών Παπαβασιλείου	Ιωάννα Ρουσσάκη	Γιώργος Ματσόπουλος
Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2022

.....

Χριστίνα Τσακανίκα

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

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

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

.....

Χριστίνα Τσακανίκα

Περίληψη

Στην εν λόγω διπλωματική εργασία αναλύεται η ανίχνευση ψευδών ειδήσεων με τη χρήση συνελικτικων νευρωνικων δικτύων σε μορφή γραφών, στο κοινωνικό δίκτυο του Twitter. Η συγκεκριμένη διαδικτυακή πλατφόρμα, επιλεγεται από πληθωρα χρηστών για την ενημέρωση και την έκφραση των ιδεών τους. Αρχικά, αποδίδεται το θεωρητικό υπόβαθρο και αναλύονται οι έννοιες "Τεχνητή Νοημοσύνη" και "Μηχανική Μάθηση", που καθιστούν την εργασία περισσότερο εύληπτη, και ύστερα παρουσιάζεται η δομή και το περιεχόμενο του επιλεγμένου συνόλου δεδομένων *FakeNewsNet*. Εν συνεχεία, αναλύονται τοσο η αρχιτεκτονική όσο και το μαθηματικό μέρος των νευρωνικών δικτυων καθώς και τα συνελικτικα νευρωνικά δίκτυα σε γράφους (GCNs). Η ανίχνευση ψευδών ειδήσεων βασίζεται σε τρεις μεθόδους. Επιλεγοντας ως μοντέλο το GCN πρώτα χρησιμοποιούνται ως είσοδος, για την εκπαίδευση και μετά για την αξιολόγηση, user related χαρακτηριστικά, έπειτα topic related και τέλος ένας συνδυασμός ενδογενους πληροφορίας, που αφορά την εξαγωγή διανυσμάτων χαρακτηριστικών από παλιές δημοσιεύσεις του χρήστη στο κοινωνικό δίκτυο του Twitter, και εξωγενούς πληροφορίας που προκύπτει από την εξαγωγή διανυσμάτων χαρακτηριστικών από την προς ανάλυση είδηση. Τέλος, προστίθενται στο αρχικό Νευρωνικό Δίκτυο μηχανισμός attention (Graph Attention Network, GAT) και δοκιμάζεται επίσης το συνελικτικό GraphSAGE μοντέλο, προς ενίσχυση των προηγούμενων επιδόσεων. Υστερα, παρατίθενται οι νέες επιδόσεις του μοντέλου, σε σύγκριση με τις προηγούμενης καθώς και με τις baseline μεθόδους των απλών νευρωνικων δικτύων. Το μοντέλο που υπερτερεί σε ακρίβεια και προτείνεται εν τέλει για την ανίχνευση ψευδών ειδήσεων είναι το Graph Attention Network με ακρίβεια (accuracy) που αγγίζει το 93%.

Λέξεις Κλειδιά:

Συνελικτικα Νευρωνικά Δίκτυα, Νευρωνικά Δίκτυα, Μηχανική Μάθηση, Τεχνητή Νοημοσύνη, Ανίχνευση Ψευδών Ειδήσεων, Εκπαιδευση Μοντέλου, Αξιολόγηση Μοντέλου, Ακρίβεια.

Abstract

This thesis analyzes the detection of fake news using Convolutional Neural Networks in the form of graphs, on the Twitter social media network. This specific online platform is chosen by a large number of users for information and the expression of a plethora of ideas. Firstly, the theoretical background and analysis of the concepts of artificial intelligence and machine learning are given, which make the work more susceptible, and then the structure and content of the selected FakeNewsNet database are presented. Subsequently, the architecture and the mathematical side of neural networks, as well as Graph Convolutional Neural Networks (GCNs), are analyzed. Fake news detection is based on three methods. Choosing GCN as a model, they are initially used as input, for training and then for evaluation, user-related features, then topic related and lastly a combination of endogenous and exogenous information of the user. Endogenous user information results from extracting feature vector from user's latest posts. Exogenous information results from extracting feature vectors from the news article. Finally, an attention mechanism (Graph Attention Network, GAT) is attached to the original Neural Network and afterward the convolutional GraphSAGE model is examined, to strengthen the previous performances, while the new performances of the model are subjoined and compared to the previous as well as to the baseline methods of simple neural networks. The model which excels in precision and is ultimately recommended for the detection of fake news is the Graph Attention Network (GAT) with an accuracy of 93%.

Keywords:

Convolutional Neural Networks, Neural Networks, Machine Learning, Artificial Intelligence, Fake News Detection, Model Training, Model Evaluation, Accuracy.

Ευχαριστίες

Ευχαριστώ θερμά τον επιβλέποντα κύριο καθηγητή Συμεών Παπαβασιλείου καθώς και την κυρία Ειρήνη Κοιλανιώτη, Ε.ΔΙ.Π. Ε.Μ.Π., για την άρτια κατατόπισή τους με την παροχή σχετικής βιβλιογραφίας, την συνέπεια και το ενδιαφέρον που επέδειξαν παρέχοντάς μου πολύτιμη βοήθεια κατά τη συγγραφή της διπλωματικής, καθώς και για τις εύστοχες παρατηρήσεις με σκοπό την βελτίωση του περιεχομένου και της μορφής της παρούσας εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου, που υπήρξαν οι θερμότεροι υποστηρικτές στα φοιτητικά μου χρόνια, αποτέλεσαν πηγή έμπνευσης και ενίσχυσαν περαιτέρω το κίνητρό μου προς εξέλιξη.

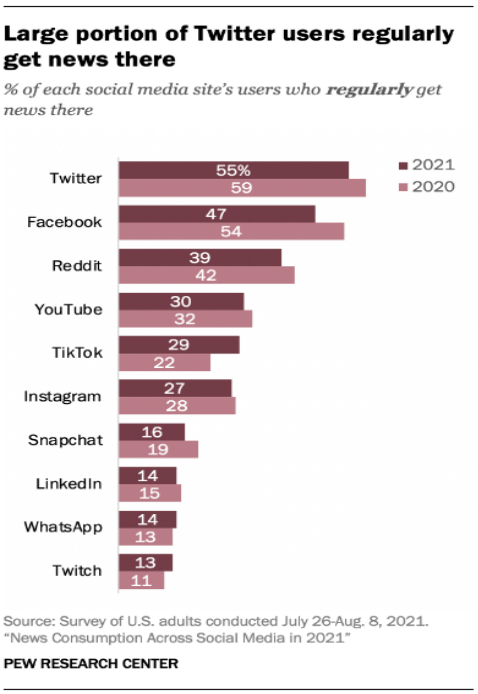
Αθήνα, Σεπτέμβριος 2022

Χριστίνα Τσακανίκα

1. Εισαγωγή

Τα τελευταία χρόνια, με την ραγδαία επέκταση του διαδικτύου στις ανθρώπινες ζωές, καθώς και των υπηρεσιών που το ίδιο προσφέρει, ολοένα και περισσότερα άτομα εξοικειώνονται με τη χρήση του προτιμώντας το για την ψυχαγωγία, την επικοινωνία και την ενημέρωσή τους. Με την ανάπτυξη του διαδικτύου, επήλθε και η διάδοση των κοινωνικών δικτύων (social media), τα οποία αποτελούν αναπόσπαστο κομμάτι του σύγχρονου τρόπου ζωής. Πρόκειται για μέσα αλληλεπίδρασης και επικοινωνίας ομάδων ανθρώπων μέσω διαδικτυακών κοινοτήτων. Αποτελούν κοινωνική διάδραση μεταξύ ανθρώπων οι οποίοι δημιουργούν, μοιράζονται ή ανταλλάσσουν πληροφορίες και ιδέες μέσα σε εικονικές κοινότητες και δίκτυα. Ως βασισμένες στο διαδίκτυο υπηρεσίες, τα διαδικτυακά κοινωνικά δίκτυα οι χρήστες δημιουργούν ένα δημόσιο ή ημι-δημόσιο (δημόσιο σε επιλεγμένους μόνο χρήστες) προφίλ μέσα σε ένα οριοθετημένο σύστημα και αλληλεπιδρούν με μια λίστα από άλλους χρήστες με τους οποίους μοιράζονται μια μορφή σύνδεσης (επαγγελματική, φιλική). Τα μέσα κοινωνικής δικτύωσης εκφράζουν τους τρόπους, τα εργαλεία με τα οποία διαμοιράζεται η πληροφορία και επικοινωνείται στο κοινό.

Κατά συνέπεια η διαδραστικότητα, η απλότητα στην χρήση και η ευρεία αποδοχή τους από το κοινό, συνάμα με την επιλογή τους ως μέσο διάδοσης ειδήσεων από κυβερνήσεις, οργανώσεις και δημόσια πρόσωπα συντελούν στο να επιλέγονται ως το κύριο μέσο για την ενημέρωση των ατόμων. Χάρη στα κοινωνικά δίκτυα, οι πληροφορίες διαδίδονται τάχιστα, σε χαμηλό κόστος και με άμεση πρόσβαση από κάθε χρήστη. Ωστόσο, τα θετικά αυτά χαρακτηριστικά για την διάδοση της πληροφορίας συνοδεύονται με το μειονέκτημα ότι χρόνο με το χρόνο παρατηρείται αύξηση στη διασπορά των ψευδών ειδήσεων (fake news) στα social media. Το φαινόμενο αυτό εξελίσσεται σε μάστιγα της εποχής, ενώ πυροδοτείται από την αύξηση της χρήσης των μέσων κοινωνικών δικτύωσης στον τομέα της ενημέρωσης. Τα στατιστικά δεδομένα που δημοσιεύτηκαν για το έτος 2022 πληροφορούν, πως το 58.4% του παγκόσμιου πληθυσμού, δηλαδή πάνω από το μέσο, χρησιμοποιεί τα social media, αφιερώνοντας σε αυτά 2 ώρες και 27 λεπτά κατά μέσο όρο. Το ποσοστό αυτό παρουσιάζει αύξηση κατά 10.1% από το έτος 2021 [1]. Στην Ελλάδα, για το έτος 2021, περισσότερο από το 69% του πληθυσμού επιλέγει την άντληση ενημέρωσης μέσω των κοινωνικών δικτύων [2], ποσοστό σημαντικά υψηλό για το μέγεθος και τον πληθυσμό της χώρας.
Στην Αμερική εν έτει 2021, το 48% του πληθυσμού δηλώνει πως χρησιμοποιεί τα social media για την ενημέρωσή του συχνά ή σπάνια, ποσοστό μειωμένο κατά 5% από το έτος 2020. Εξετάζοντας τα ποσοστά των χρηστών των social media που συχνά αντλούν ειδήσεις από αυτά, προκύπτει πώς ορισμένα μέσα χρησιμοποιούνται στοχευμένα για την ενημέρωση, παρόλο που εμφανίζουν λιγότερους συνδρομητές από τα άλλα μέσα. Επεξηγηματικά, μπορεί το 66% Αμερικανών να χρησιμοποιεί το Facebook, ενώ μόλις το 23% αυτών να χρησιμοποιεί το Twitter, ωστόσο αποδεικνύεται ότι μεγαλύτερο ποσοστό χρηστών χρησιμοποιούν το Twitter από ότι το Facebook για την ενημέρωση τους, προσδίδοντας έτσι στην πλατφόρμα του Twitter έναν ενημερωτικό χαρακτήρα [3].



Εικόνα 1 Το ποσοστό των χρηστών των κοινωνικών δικτύων Twitter, Facebook Reddit, YouTube, TikTok, Instagram, Snapchat, LinkedIn, WhatsApp, Twitch το οποίο αντλεί την ενημέρωσή από το εκάστοτε δίκτυο για τα έτη 2020 και 2021 [3].

Μια έρευνα που διεξήχθη στο Institute of Technology's Media του MIT, κατά την οποία εξετάστηκαν 126.000 ειδήσεις από 3 εκατομμύρια ανθρώπους από το 2006 έως το 2017 εξήγαγε το συμπέρασμα πως οι ψευδείς ειδήσεις είναι κατά 70% πιο πιθανό να διαδοθούν από ό,τι οι αληθείς [4]. Παράλληλα, μια έρευνα στην Αμερική το έτος 2020 κατέληξε στο συμπέρασμα πως 38.2% των ερωτηθέντων χρηστών του Twitter διέδωσε, εν αγνοία του, ψευδείς ειδήσεις [5].
Ως τελευταίο στατιστικό προστίθεται πως ο όρος "fake news", που χρησιμοποιήθηκε κατεξοχήν στις εκλογές της Αμερικής το 2016, επικράτησε ως ο πιο χρησιμοποιημένος όρος για το έτος 2017, σύμφωνα με το Collins Dictionary [6].

1.1 Συνεισφορά Διπλωματικής Εργασίας

Αναλογιζόμενοι τα παραπάνω, η ανάγκη ανάπτυξης μηχανισμών ελέγχου της εγκυρότητας των ειδήσεων προβάλλει πιο επιτακτική από ποτέ, προκειμένου να αποφευχθούν οι οδυνηρές συνέπειες του φαινομένου αυτού σε κοινωνικό και ατομικό επίπεδο.
Στην παρούσα διπλωματική εργασία, αναλύεται η ανίχνευση ψευδών ειδήσεων στα κοινωνικά δίκτυα με τη χρήση της Μηχανικής Μάθησης και δη των Graph Convolutional Networks (GCNs). Το μέσο κοινωνικής δικτύωσης από όπου αντλούνται τα δεδομένα της εκπαίδευσης, εκπαιδεύονται τα νευρωνικά δίκτυα στη μορφή γράφων και, τελικά, επιτελείται η αξιολόγησή τους είναι το Twitter. Συγκεκριμένα, το σύνολο δεδομένων (dataset) από το οποία εξετάζονται οι ειδήσεις κατά την ορθότητά τους ονομάζεται FakeNewsNet. Το εν λόγω dataset, αποτελείται από δύο επιμέρους datasets. Το GossipCop και το Politifact.
Στην εργασία αυτή, αναλύονται και εξετάζονται τόσο σύνθετα διανύσματα χαρακτηριστικών που χρησιμοποιούνται ως είσοδος στο GCN όσο και πιο σύνθετα νευρωνικά δίκτυα, με μηχανισμους *attention* (GAT model) και τεχνικές επαγωγικής εύρεσης διανυσμάτων χαρακτηριστικών στους κόμβους του γράφου (GraphSAGE model). Μάλιστα ενισχύονται περαιτέρω οι πολύ υψηλές επιδόσεις της μεθόδου User-Preference Fake News Detection [52] που αποτελεί καινοτόμα μέθοδο

στον τομέα της ανίχνευσης ψευδών ειδήσεων (δημοσιεύθηκε το Φεβρουάριο του 2021). Στην ανάλυσή μας, πέρα από την ενδογενή και εξωγενή πληροφορία του UPFD framework , λάβαμε υπόψιν να εισάγουμε στο GCN επιπλέον χαρακτηριστικά που αφορούν τον χρήστη, το θέμα της είδησης και εάν οι χρήστες του δικτύου αποτελούν bot ή όχι. Επιπλέον, ενισχύσαμε το μοντέλο GCN με 3 συνελικτικά επίπεδα, έναντι των δύο που παρουσιάζεται στο [52] και τέλος, δοκιμάσαμε τα σύνθετα χαρακτηριστικά εισόδου στα ακόμη πιο σύνθετα νευρωνικά δίκτυα GAT και GraphSAGE. Η ερευνητική αυτή μέθοδος παρουσιάζει έντονο ενδιαφέρον, διότι βασιζόμενη σε μία από τις πιο αποδοτικές μεθόδους ανίχνευσης ψευδών ειδήσεων, χρησιμοποιώντας GCNs επιτυγχάνει ακόμη καλύτερες επιδόσεις για το σύνολο δεδομένων του *FakeNewsNet.*
Πέρα από το ερευνητικό ενδιαφέρον, με τις διάφορες επεκτάσεις που μπορεί να λάβει η εν λόγω εργασία (παράγραφος 11), παρουσιάζει σημαντικό ανθρωπιστικό ενδιαφέρον, καθώς η ανάγκη προστασίας των ατόμων από τις ψευδείς ειδήσεις είναι ανεκτίμητης αξίας. Τα μοντέλα μηχανικής μάθησης έχουν το προνόμιο να αναγνωρίζουν μοτίβα και να αποφασίζουν εάν η είδηση πρόκειται για πραγματική ή ψευδή. Δεν ξεγελιούνται, ούτε απατούνται όπως ο άνθρωπος. Συνεπώς, η δημιουργία μοντέλων με ενισχυμένη ακόμα περισσότερο τη μετρική της ακρίβειας θα αποτελέσει σημαντικό επίτευγμα της επιστήμης σε ατομικό και συλλογικό επίπεδο.

1.2 Σύντομη Παρουσίαση Διπλωματικής

Το σύνολο δεδομένων (dataset) από το οποία εξετάζονται οι ειδήσεις κατά την ορθότητά τους ονομάζεται FakeNewsNet. Το εν λόγω dataset, αποτελείται από δύο επιμέρους datasets. Το GossipCop και το Politifact.
Αρχικά, κάθε ένα dataset εξετάζεται ξεχωριστά. Γίνεται επεξεργασία των χαρακτηριστικών που εκείνο προσφέρει και έπειτα ο διαχωρισμός αυτών σε δύο κατηγορίες. Τα user-related σχετικά με τον χρήστη, χαρακτηριστικά όπως αριθμός ακολούθων, αριθμός φίλων, αριθμός like, αριθμός retweet, αριθμός αναρτήσεων του χρήστη, καθώς και αν ο ίδιος είναι επικυρωμένος (verified). Τα topic related χαρακτηριστικά αφορούν το θέμα της είδησης όπως κατηγορία είδησης (πολιτική, πολεμική, ιατρική), η διαδικτυακή πηγή από την οποία αντλήθηκε η είδηση καθώς και η ημερομηνία δημοσίευσή της. Συγκεκριμένα, για το τελευταίο topic-related χαρακτηριστικό, εξετάζεται η χρονική απόσταση της είδησης από τις πιο πρόσφατες αμερικανικές εκλογές. Η ανάγκη εξαγωγής των user related χαρακτηριστικών βασίζεται στο ότι αποκλειστικά τα topic related χαρακτηριστικά, δεν μπορούν να μας δώσουν ικανοποιητικά αποτελέσματα στην ανίχνευση ψευδών ειδήσεων. Πλέον, κακόβουλοι χρήστες ή ακόμα ψεύτικοι λογαριασμοί αναρτούν περιεχόμενο άκρως ρεαλιστικό, μιμούμενο μια αληθινή είδηση, συνεπώς το περιεχόμενο μιας είδηση δεν αρκεί, για να αποφανθούμε εάν αυτή είναι έγκυρη.
Εν συνεχεία, για την επίτευξη υψηλότερων αποδόσεων ελέγχουμε ποια από αυτά τα χαρακτηριστικά παίζουν σπουδαιότερο ρόλο στην κατηγοριοποίηση μιας είδησης. Ο έλεγχος αυτός επιτυγχάνεται με τους Random Forest Regressor και Linear Regressor ταξινομητές (classifiers). Οι ταξινομητές αποτελούν αλγορίθμους μηχανικής μάθησης, που σκοπό έχουν, κατά κύριο λόγο, να κατηγοριοποιήσουν, να αποδώσουν μια ετικέτα στα δεδομένα εισόδου. Ωστόσο εμείς, στην εν λόγω εργασία, θα τους χρησιμοποιήσουμε για να συλλέξουμε τα σημαντικότερα user και topic related χαρακτηριστικά. Και οι δύο *classifiers* αναλύονται εκτενώς στις παραγράφους 8.3, 8.4. Μάλιστα, η ανάγκη μείωσης των διαθέσιμων από το dataset χαρακτηριστικών προβάλλει επιτακτική καθώς τα δεδομένα διαχείρισης είναι υπέρογκα και περιοριορίζοντας τα, πέρα από καλύτερη απόδοση επιτυγχάνουμε και σημαντική εξοικονόμηση χρόνου. Με περιορισμένα χαρακτηριστικά επεξεργασίας το εκάστοτε μοντέλο αποκτά υψηλότερη ικανότητα γενίκευσης, αποφεύγοντας το φαινόμενο υπερπροσμαρμογής (overfitting). Κατά το overfitting το μοντέλο υπερπροσαρμόζεται στα δεδομένα της εκπαίδευσης και αποδίδει βέλτιστα στο σύνολο αυτών, ωστόσο αδυνατεί να γενικεύσει και σημειώνει πτωχές επιδόσεις σε στα άγνωστα δεδομένα της εκπαίδευσης.

Έπειτα, εφαρμόζεται έλεγχος για το εάν οι λογαριασμοί που αναρτούν tweets είναι πραγματικοί χρήστες ή κακόβουλα λογισμικά, bots.
Με τον τρόπο αυτό, διαμορφώνουμε το Graph Convolutional Network, το εκπαιδεύουμε στη βάση δεδομένων και μας και ανιχνεύουμε αν η κάθε είδηση είναι πραγματική ή ψεύτικη. Το τελικό μοντέλο που προτείνεται εκφράζει μια καινοτόμα προσέγγιση στο ζήτημα της ανίχνευσης ψευδών ειδήσεων στα κοινωνικά δίκτυα. Καθοριστικό ρόλο στο να αποφανθούμε εάν ο χρήστης διαδίδει ψευδείς ειδήσεις διαδραματίζει η ενδογενής πληροφορία που υπάρχει διαθέσιμη στον ίδιο του το λογαριασμό. Η πληροφορία αυτή αξιοποιείται λαμβάνοντας υπόψιν τις πρόσφατες αναρτήσεις του εκάστοτε χρήστη. Οι δημοσιεύσεις αυτές, οι οποίες εκφράζουν σε μεγάλο βαθμό τις προτιμήσεις του χρήστη (*user preferences*) συνδυαστικά με τα user-related χαρακτηριστικά και το BotOrNot, κατόπιν επεξεργασίας προστίθενται στα λοιπά χαρακτηριστικά της είδησης μαζί με τα Topic Related χαρακτηριστικά και βάσει αυτών θα γίνει η εκπαίδευση του Graph Convolutional Network. Το νευρωνικό αυτό δίκτυο, διαφοροποιείται και επεκτείνεται του [52] λόγω των σύνθετων διανυσμάτων χαρακτηριστικών που καλείται να επεξεργαστεί. Επιπλέον, ενισχύεται περαιτέρω με ένα επιπλέον συνελικτικό επίπεδο, από ότι το [52]. Το framework που εστιάζει στην ενδογενή πληροφορία του χρήστη καλείται User Preference Fake news Detection (UPFD) και ξεπερνά σε επιδόσεις τόσο τα μέχρι τώρα μέσα ανίχνευσης ψευδών ειδήσεων (GCNs, RNNs) καθώς και τα πιο σύνθετα μοντέλα user και topic related που εξετάζονται και αναλύονται στο εν λόγω έργο. Λαμβάνοντας υπόψιν και τις προσθήκες τόσο στα διανύσματα χαρακτηριστικών όσο και στη δομή του νευρωνικού δικτύου ενισχύεται ακόμα περισσότερο η επίδοσή του.
Τέλος, για την περαιτέρω βελτίωση και εξέλιξη του GCN μοντέλου εισαγάγονται δύο νέοι μηχανισμοί που το ενισχύουν ακόμα περισσότερο. Ο SAmple and aggreGatE μηχανισμός, που αποδίδει το μοντέλο SAGE και ο μηχανισμός attention που αποδίδει το μοντέλο Graph Attention Network (GAT). Χάρη στα μοντέλα αυτά, ενισχύεται ακόμα περισσότερο η μετρική της ακρίβειας επιτυγχάνοντας μια πιο αποτελεσματική ανίχνευση ψευδών ειδήσεων. Όπως προαναφέρθηκε παραπάνω, η αναπαραγωγή ψευδών ειδήσεων επιτελείται με τρόπο ιδιαίτερα πετυχημένο, αφού οι ψευδείς ειδήσεις μιμούνται με πειστικότατο τρόπο τα πραγματικά γεγονότα, σε βαθμό τέτοιο ώστε τα ανθρώπινα όντα να αδυνατούν να διακρίνουν το ψευδές από το αληθές. Για το λόγο αυτό, αδήριτη κρίνεται η ανάγκη μοντέλων της μηχανικής μάθησης, για την ανίχνευση των παραπλανητικών ειδήσεων.

Εκκινούμε την ανάλυση παρουσιάζοντας τον μαθηματικό ορισμό του προβλήματος καθώς και το διάγραμμα ροής της διπλωματικής εργασίας.

Το προς εξέταση κοινωνικό δίκτυο αποτελείται από πολλούς αυτόνομους γράφους. Ο κάθε γράφος, έχει ως κόμβο κεφαλή το άρθρο της είδησης και οι υπόλοιποι κόμβοι είναι χρήστες που αναδημοσίευσαν το άρθρο αυτό. Οι ακμές του γράφου που ενώνουν την αρχική είδηση με τους χρήστες καθώς και τους χρήστες μεταξύ τους, εκφράζουν τη σχέση της αναδημοσίευσης. Συνεπώς, ο γράφος αυτός είναι κατευθυνόμενος φανερώνοντας ποια είδηση αναδημοσίευσε ο κάθε χρήστης καθώς και από ποιον χρήστη. Τέλος, ο γράφος αυτός είναι ομογενής, διότι όλοι οι κόμβοι του έχουν κοινό είδος (χρήστες του δικτύου) και επιπλέον οι ακμές του εκφράζουν την ενέργεια της αναδημοσίευσης. Οι κανόνες κατασκευής του γράφου διάδοσης καθώς και παραδείγματα αυτών παρουσιάζονται εκτενέστερα στις παραγράφους 8.5, 8.6, 8.7.

1.3 Ορισμός Προβλήματος Ψευδών Ειδήσεων

Δεδομένων των συσχετίσεων, E, N χρηστών σε ένα κοινωνικό δίκτυο που διαμορφώνονται βάσει του άρθρου ειδήσεων, A, η ανίχνευση ψευδών ειδήσεων αφορά την πρόβλεψη εάν η είδηση A είναι ψευδής ή πραγματική.

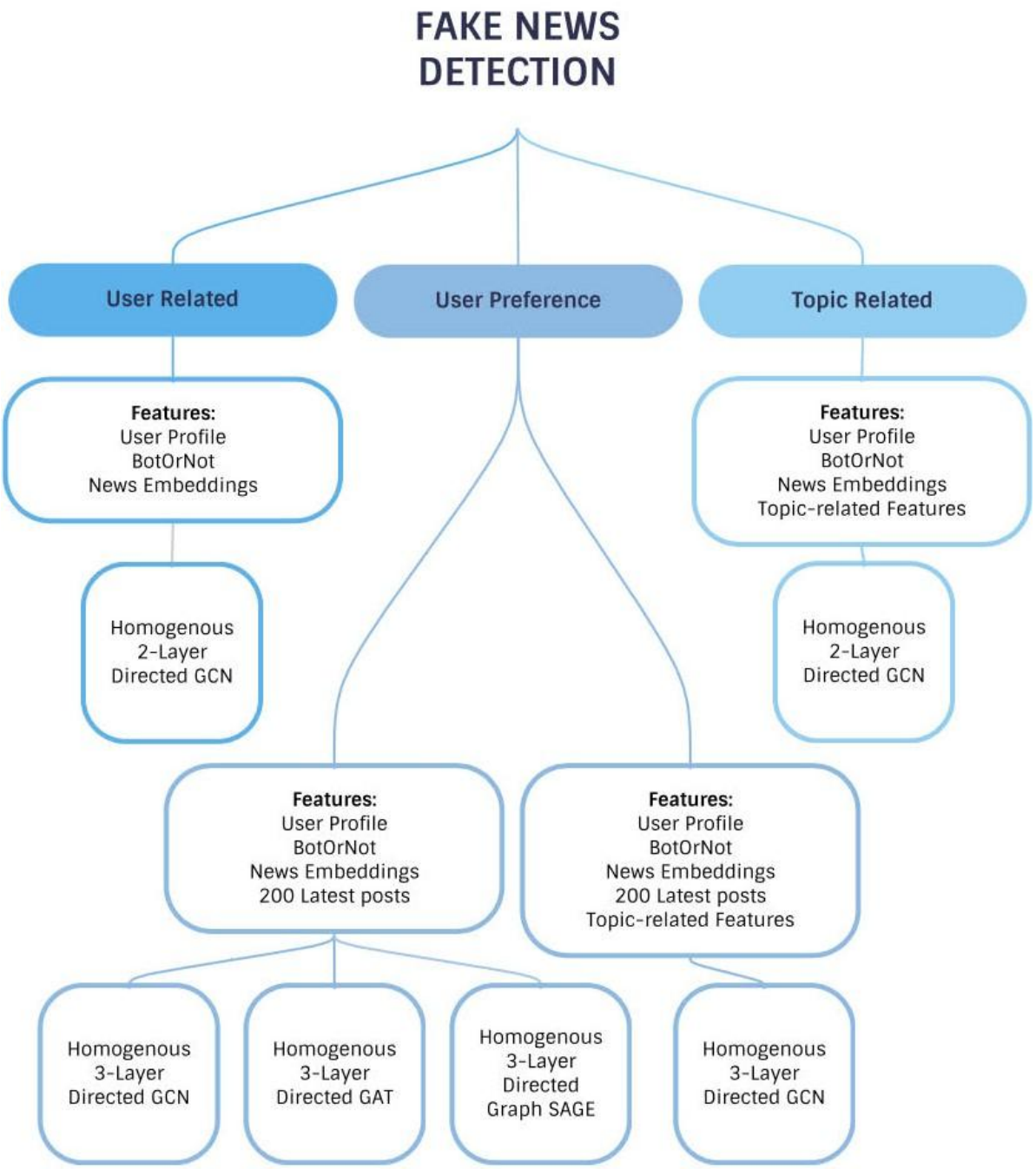
F: $E \rightarrow \{0, 1\}$ τέτοια ώστε,

$$F(A) = \begin{cases} 1, & \text{εάν το } A \text{ είναι πραγματική είδηση} \\ 0, & \text{αλλιώς} \end{cases}$$

Όπου F είναι η συνάρτηση πρόβλεψης που στην περίπτωσή μας εκφράζει το νευρωνικό δίκτυο GCN. Η ανίχνευση ψευδών ειδήσεων αποτελεί πρόβλημα δυαδικής ταξινόμησης

1.4 Διάγραμμα Ροής Εργασίας

Σχηματικά, το διάγραμμα ροής της εργασίας αποδίδεται παρακάτω:



Εικόνα 2 Διάγραμμα ροής της Διπλωματικής Εργασίας.

1.5 Δομή

Στην παρούσα ενότητα θα δοθούν πληροφορίες για κάθε μία από τις παραγράφους της διπλωματικής εργασίας.

2. Περιγραφή *FakeNewsNet*:
Αφορά το σύνολο δεδομένων το οποίο αντλήσαμε για την εκπόνηση της διπλωματικής εργασίας. Συγκεκριμένα, περιγράφεται η διαδικασία δημιουργίας του, από τους ερευνητές του, τους συγγραφείς της δημοσίευσης [53]. Αναλύονται οι λόγοι για τους οποίους επιλέχθηκε το συγκεκριμένο dataset, οπως το πλούσιο περιεχόμενο σε user-related χαρακτηριστικά, όπως και context related χαρακτηριστικά που αφορούν την είδηση. Η μόνη προσθήκη που σημειώθηκε από πλευράς μου στο συγκεκριμένο αποθετήριο ήταν τα botornot χαρακτηριστικά, που δηλώνουν εάν ο χρήστης είναι πραγματικός ή bot. Το χαρακτηριστικό αυτό προστίθεται στα υπόλοιπα user related χαρακτηριστικά.

3. Μηχανική Μάθηση:
Θεωρητική παράγραφος σύντομης παρουσίασης και ανάλυσης του τομέα της Μηχανικής Μάθησης. Ορίζονται οι τύποι προβλημάτων και εργασιών που αφορούν τη συγκεκριμένη εργασία. Στην προκειμένη περίπτωση, διαχειριζόμαστε πρόβλημα ημι-επιβλεπόμενης μάθησης. Επιπλέον, παρουσιάζονται και οι διάφορες τεχνικές και προσεγγίσεις για την επίλυση προβλημάτων βάσει Μηχανικής Μάθησης. Στην εν λόγω εργασία, ασχολούμαστε με τεχνητά νευρωνικά δίκτυα στη μορφή γράφων καθώς και με βαθιά μάθηση. Σύντομη περιγραφή παρατίθεται και για τις υπόλοιπες τεχνικές της Μηχανικής Μάθησης για λόγους πληρότητας.

4. Τεχνητή Νοημοσύνη:
Άλλη μία παράγραφος θεωρίας όπου ορίζεται η έννοια "Τεχνητή Νοημοσύνη". Παράλληλα παρουσιάζονται οι διάφορες τεχνικές επίλυσης προβλημάτων, όπως η αναζήτηση και βελτιστοποίηση, η λογική και τα Τεχνητά Νευρωνικά Δίκτυα. Οι δύο πρώτες μέθοδοι δεν χρησιμοποιούνται στην παρούσα εργασία, ωστόσο παρουσιάζονται συνοπτικά για λόγους πληρότητας. Η εργασία αυτή βασίζεται στα Τεχνητά Νευρωνικά Δίκτυα, Νευρωνικά Δίκτυα για συντομία, ως εργαλεία της Τεχνητή Νοημοσύνης. Για το λόγο αυτό, γίνεται μια πιο ενδελεχής ανάλυση αυτών, στην παράγραφο 4.3, καθώς και των εννοιών που σχετίζονται με αυτά, αφού χρησιμοποιούνται στην υλοποίηση της διπλωματικής εργασίας. Κάθε μία από τις παραγράφους "4.3.8 Backpropagation", "4.3.9 Συνάρτηση Κόστους", "4.3.10 Weight Update", "4.3.13 Stochastic Gradient Descent" είναι σημαντικές καθώς είναι οι τεχνικές που εφαρμόζονται κατά την εκπαίδευση και την αξιολόγηση των GCN μοντέλων της διπλωματικής.

5. Νευρωνικά Δίκτυα σε μορφή Γράφου (GNN):
Εδώ αναλύεται πιο στοχευμένα το είδος του νευρωνικού δικτύου στο οποίο βασίζεται η υλοποίηση της διπλωματικής εργασίας. Αφού αποδοθεί η θεωρία και ο ορισμός των γράφων, η απόδοση των

κοινωνικών δικτύων χρησιμοποιώντας γράφους δίδεται και η μαθηματική ανάλυση του συνελικτικού νευρωνικού δικτύου σε μορφή γράφου.

6. BotOrNot:
Ορισμός των διαδικτυακών bots, καθώς και ο τρόπος εξαγωγής του *BotOrNot* χαρακτηριστικού για κάθε έναν από τους χρήστες που αποτελούν τον εκάστοτε γράφο διάδοσης.

Για την περαιτέρω εξοικείωση με τους όρους της μηχανικής μάθησης, καθώς και τη βαθύτερη κατανόηση των όσων θα αναπτυχθούν στη συνέχεια, παρακάτω παρατίθεται μια, όσο το δυνατόν πιο περιεκτική, κάλυψη των όρων που αναφέρονται στην παρούσα εργασία, αφού πρώτα περιγραφεί το σύνολο δεδομένων από όπου αντλούνται τα δεδομένα εισόδου, η οποία ονομάζεται *FakeNewsNet* [53].

7. Word Embeddings:
Προκειμένου να παραχθούν τα context related χαρακτηριστικά της είδησης, η εξαγωγή, δηλαδή, των feature vectors της εν λόγω είδησης είδησης ακολουθείται η διαδικασία *Machine Learning Natural Language Processing*. Με τον τρόπο αυτό, προκύπτουν τα *word embeddings* του κάθε κειμένου που είναι απολύτως χρήσιμα για την εκπαίδευση του νευρωνικού μας δικτύου. Τα *word embeddings* επιλέγουμε να προκύψουν είτε με την μέθοδο *word2vec* είτε με τη μέθοδο *BERT*. Στις υποπαραγράφους 7.1, 7.2 παρουσιάζονται εκτενώς οι μέθοδοι word2vec και *BERT* αντίστοιχα.

8. Περιγραφή Υλοποίησης:
Περιγράφεται αναλυτικά κάθε μία από τις μεθόδους User-Related, Topic-Related, User-Preference Fake News Detection. Για την User-Related μέθοδο παρουσιάζονται τα σχετικά με το χρήστη χαρακτηριστικά που επιλέχθηκαν. Ομοίως και για το Topic-Related, ενώ στην υποπαράγραφο User Preference παρουσιάζονται επιπλέον η ενδογενής πληροφορία που αφορά τον χρήστη, μέσω των πιο πρόσφατων δημοσιεύσεών του και η εξωγενής πληροφορία που αφορά τα χαρακτηριστικά της είδησης. Στο τέλος της κάθε μεθόδου δίνονται οι πίνακες αποτελεσμάτων accuracy και F1-score για κάθε embedding σε κάθε ένα από τα Politifact και Gossipcop Datasets. Δίνονται επιπλέον και οι υπερπαράμετροι του μοντέλου.

9. Graph SAGE:
Στην παράγραφο αυτή παρουσιάζεται ένα πιο σύνθετο νευρωνικό δίκτυο σε μορφή γράφου. Βασικό του πλεονέκτημα η επαγωγική εύρεση embeddings των κόμβων. Σε αντίθεση με τις συνηθισμένες μεθόδους υπολογισμού embeddings, μέσω πολλαπλασιασμών πινάκων, υπολογίζει τα χαρακτηριστικά των κόμβων, user-preferences, user-related features και textual embeddings στην περίπτωσή μας, προκειμένου να δημιουργήσει μια συνάρτηση embedding που θα γενικεύεται και στους ανεξερεύνητους κόμβους. Στην παράγραφο 9.1 αποδίδεται λεπτομερώς ο αλγόριθμος παραγωγής των embeddings αυτών.

10. Graph Attention Networks:
Ανάλυση του προτεινόμενου μοντέλου για το πρόβλημα ανίχνευσης ψευδών ειδήσεων σε κοινωνικά δίκτυα. Περιγραφή των επιπέδων Attention που προστίθενται στο μοντέλο του νευρωνικού δικτύου και παρουσίαση των πινάκων αποτελεσμάτων accuracy και F1-score για κάθε embedding σε κάθε ένα από τα Politifact και Gossipcop Datasets με τις αντίστοιχες υπερπαραμέτρους.

11. Συμπεράσματα και Μελλοντική Δουλειά:
Ολοκληρώνουμε τη διπλωματική εργασία παρουσιάζοντας τα συμπεράσματα για κάθε ένα από τα μοντέλα που αναλύθηκαν και εξετάστηκαν. Προτείνουμε διάφορες τροποποιήσεις, προσθήκες καθώς και ιδέες που θα μπορούσαν να εφαρμοστούν σε αυτά με σκοπό να βελτιώσουν ακόμα περισσότερο την απόδοσή τους και να ανιχνεύουν με τον καλύτερο δυνατό τρόπο ψευδείς ειδήσεις.

2. Περιγραφή FakeNewsNet

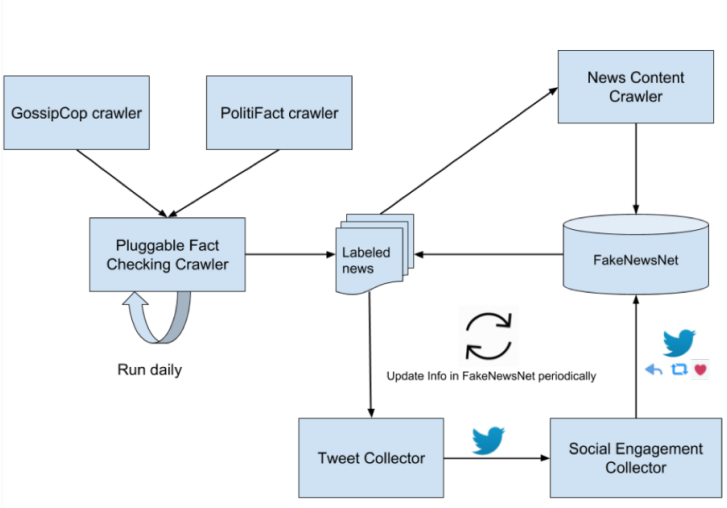
Η ανίχνευση ψευδών ειδήσεων στα κοινωνικά δίκτυα προβάλλει ιδιαίτερες προκλήσεις, καθώς οι ειδήσεις αυτές στοχεύουν στο να παραπλανήσουν τους χρήστες. Συνεπώς, οι κατηγοριοποίησή τους σε αληθείς και ψευδείς δεν μπορεί να επιτευχθεί βασιζόμενοι μόνο στο περιεχόμενό τους. Για το λόγο αυτό, σκόπιμο κρίνεται να εξεταστούν οι κοινωνικές δεσμεύσεις και συμπεριφορές των χρηστών στα social media, συνδυαστικά με το περιεχόμενο των ειδήσεων.
Επιπροσθέτως, αναγκαίο είναι το σύνολο δεδομένων να περιέχει δυναμικές πληροφορίες, προκειμένου να γίνεται αντιληπτό πώς διαδίδονται οι ψευδείς και οι αληθείς ειδήσεις, πώς οι χρήστες αντιδρούν σε αυτές και πώς εξάγονται χρήσιμα μοτίβα ανίχνευσης ψευδών ειδήσεων.
Το σύνολο δεδομένων *FakeNewsNet* αποτελεί το πρώτο dataset που καταφέρνει να συνδυάσει περιεχόμενο ειδήσεων, κοινωνικό περιεχόμενο και δυναμική πληροφόρηση. Αρχικά, το *FakeNewsNet* αντλεί δεδομένα από δύο σύνολα δεδομένων και παρέχει ένα πλούσιο σύνολο χαρακτηριστικών, προσφέροντας πολλές μεθόδους προσέγγισης του προβλήματος ανίχνευσης ψευδών ειδήσεων. Παράλληλα, η δυναμική πληροφόρηση επιτρέπει την παραγωγή των σύνθετων δεσμεύσεων του χρήστη, βάσει των παλαιότερων αναρτήσεών του. Η διαδικασία αυτή περιγράφεται ως *early fake news detection.* Τρίτον, μπορούμε να διερευνήσουμε τη διαδικασία διάδοσης ψευδών ειδήσεων, αναγνωρίζοντας προελεύσεις και τα χαρακτηριστικά των ατόμων που τις διαδίδουν, αναπτύσσοντας έτσι καλύτερες τεχνικές ανίχνευσης ψευδών ειδήσεων.

Platform	PolitiFact		GossipCop	
	Real	Fake	Real	Fake
# Train samples	192	188	9342	3162
# Test samples	49	47	2336	790

Πίνακας 1 Στατιστικά της βάσης δεδομένων *FakeNewsNet* [53]

2.1 Κατασκευή Dataset

Στην παράγραφο αυτή περιγράφεται το πώς μπορούμε να συλλέξουμε περιεχόμενα ειδήσεων (*news contents*) με έγκυρες ετικές αληθείας (*truth labels: fake news, real news*). Στη συνέχεια αναλύεται πώς το περιεχόμενο αυτό εμπλουτίζεται επιπλέον σύμφωνα με το κοινωνικό *background* των χρηστών (*social context information*) και τέλος πώς ανανεώνονται δυναμικά οι πληροφορίες της βάσης με περιοδικό τρόπο. Το διάγραμμα ροής συλλογής και επεξεργασίας των δεδομένων αποδίδεται παρακάτω:



Εικόνα 3 Το διάγραμμα ροής της κατασκευής και επεξεργασίας της βάσης δεδομένων. Αποδίδει τη συλλογή του περιεχομένου των ειδήσεων (*news contents*), το κοινωνικό πλαίσιο (*social context*) και τέλος τη δυναμική πληροφόρηση (*dynamic information*) [53].

2.2 News Content

Προκειμένου να συλλέξουμε έγκυρες ετικέτες για την κάθε είδηση καταφεύγουμε σε δύο *datasets* τα οποία μας προμηθεύουν με περιεχόμενο ειδήσεων, Τα *datasets* αυτά είναι τα *Politifact* και *GossipCop*. Το Politifact είναι ένα site διαχειριζόμενο από την *Tampa Bay Times*, όπου οι δημοσιογράφοι ελέγχουν διαρκώς την εγκυρότητα των πολιτικών άρθρων. Το *Politifact* δημοσιεύει στο *site* του αυτολεξεί το περιεχόμενο των πολιτικών άρθρων καθώς και τα ακριβή αποτελέσματα αξιολόγησης της εγκυρότητάς τους. Το *GossipCop* είναι ένα *website* το οποίο ελέγχει την εγκυρότητα νέων ψυχαγωγικού χαρακτήρα, τα οποία αντλούνται από διάφορα μέσα ψυχαγωγίας. Το *GossipCop* αναλύει κάθε άρθρο ειδήσεων αποδίδοντάς του ένα βαθμό εγκυρότητας από το μηδέν έως και το δέκα. Η αξιολόγηση με μηδέν σηματοδοτεί την απολύτως ψευδή είδηση, ενώ η αξιολόγηση με δέκα, την απολύτως έγκυρη είδηση. Ο *news content crawler* εντοπίζει την γνήσια πηγή ειδήσεων, από τα URLs που παρέχονται στον *fact checking crawler*. Οι πληροφορίες *news content* περιλαμβάνουν διάφορες λεπτομέρειες όπως τίτλος, κείμενο, φωτογραφίες, πληροφορίες για τον συγγραφέα, βοηθητικά *links*. Οι *news content* πληροφορίες συλλέγονται από τα *datasets Politifact & GossipCop*.

2.3 Politifact Crawler

Στο *site* αυτό, οι συντάκτες, οι δημοσιογράφοι και οι ειδικοί στον τομέα αυτό αξιολογούν πολιτικά νέα και παρέχουν τα αποτελέσματα ελέγχου των γεγονότων αυτών. Με αυτό τον τρόπο, οι ειδήσεις λαμβάνουν *true ground label* εάν είναι αληθείς και *false ground label* εάν είναι ψευδείς.

2.4 GossipCop Crawler

Το *GossipCop* παρέχει τα *score* αξιολόγησης της ορθότητας των ειδήσεων σε μια κλίμακα από το μηδέν έως και το δέκα. Σύμφωνα με τις παρατηρήσεις των συγγραφέων του άρθρου [53], περίπου το 90% των ειδήσεών του σημειώνουν *score* χαμηλότερο του 5, καθώς το *dataset* αυτό κυρίως αναδεικνύει τις ψευδείς ειδησεις.

Dataset / Features	PolitiFact		GossipCop	
	Fake	Real	Fake	Real
Total news articles	432	624	6,048	16,817
News articles with text content	353	400	785	16,765
News articles with social engagements	342	314	4,298	2,902
News articles with both social engagements and news content	286	202	675	2,895
News articles with social engagement containing at least 1 reply	236	180	945	752
News articles with social engagement containing at least 1 like	283	219	2,911	845
News articles with social engagement containing at least 1 retweet	282	242	2,249	1,254
No. of tweets with replies	6,686	20,720	3,040	2,546
No. of tweets with likes	18,453	52,082	10,685	2,264
No. of tweets with retweets	13,226	42,059	7,614	5,025
Total no. of tweets	116,005	261,262	71,009	154,383

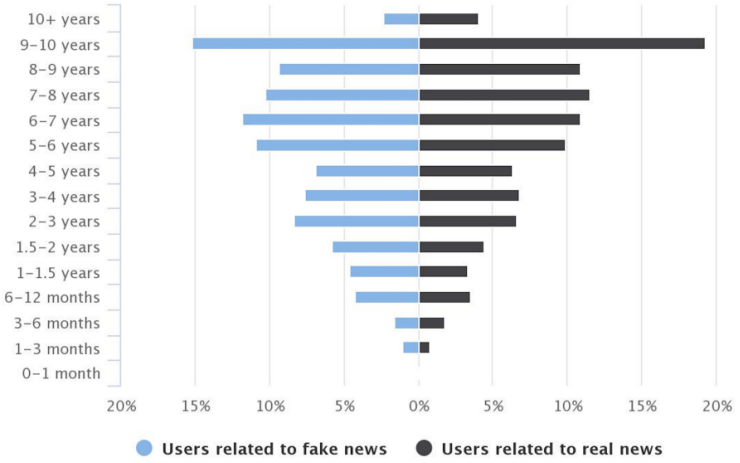
Πίνακας 2 Αναλυτικά Στατιστικά του *FakeNewsNet dataset* [53].

2.5 Social Context

Οι κοινωνικές δεσμεύσεις που σχετίζονται με τις ψευδείς ή τις αληθείς ειδήσεις από τα *fact-checking sites* συλλέγονται χρησιμοποιώντας το *Advanced Search API* του *Twitter*. Λαμβάνεται το σύνολο των δεδομένων από το *Twitter*, αναζητώντας βάσει του τίτλου των ειδήσεων. Με τον τρόπο αυτό, έχοντας εντοπίσει το *post* που απευθύνεται στη συγκεκριμένη είδηση συλλέγουμε τα *second order user behaviors* που περιλαμβάνουν *likes, reposts, answers* των χρηστών στην αρχική είδηση. Οι πληροφορίες αυτές συνθέτουν τις δεσμεύσεις των χρηστών στη διαδικασία διάδοσης των ψευδών ειδήσεων (*users engaging in news dissemination process*). Συγκεκριμένα, συλλέγονται όλες οι πληροφορίες από τα προφίλ των χρηστών (*user profiles*). οι δημοσιεύσεις τους (*user posts*) καθώς και πληροφορίες για τη δομή του κοινωνικού δικτύου (*network structures*).
Το κοινωνικό πλαίσιο (*social context*) αντιπροσωπεύει το πώς πολλαπλασιάζονται οι ειδήσεις με το πέρας του χρόνου, γεγονός που μας παρέχει χρήσιμη βοηθητική πληροφόρηση για να κριθεί ο βαθμός αξιοπιστίας των ειδήσεων. Συγκεκριμένα, υπάρχουν τρεις τομείς *social media context* που θα εξεταστούν. Αυτοί είναι τα *user profiles, user posts* και *network structures.*

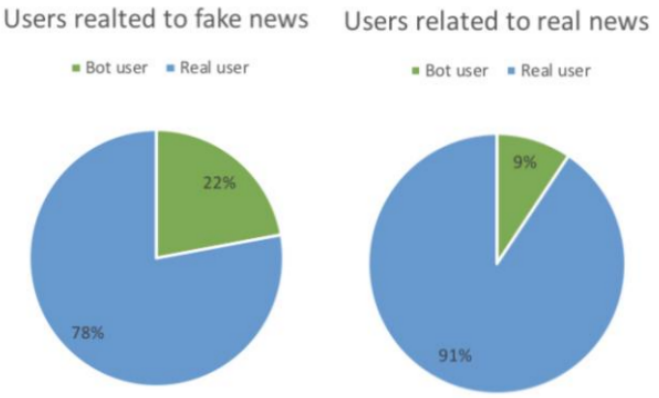
2.6 User Profiles

Έχει αποδειχθεί πως τα προφίλ των χρηστών συνδέονται άμεσα με την ανίχνευση ψευδών ειδήσεων. Βάσει ερευνών, είναι επίσης πιθανό, οι ψευδείς ειδήσεις που κυκλοφορούν στα κοινωνικά δίκτυα, να παράγονται από κακόβουλα λογισμικά, γνωστά ως *cyber bots* ή απλούστερα *bots*. Παρακάτω θα παρουσιαστούν τα χαρακτηριστικά των χρηστών που αντλούνται και εξετάζονται στη βάση δεδομένων *FakeNewsNet*.
Αρχικά, εξετάζεται από τους συγγραφείς της δημοσίευσης [53] αν η ημερομηνία δημιουργίας του λογαριασμού του χρήστη και αυτή της δημοσίευσης της είδησης, διαφέρουν ή όχι. Έπειτα, εξετάζουμε για τα διάφορα χρονικά εύρη εγγραφής των χρηστών στο *Twitter* τα ποσοστά διάδοσης αληθών και ψευδών ειδήσεων και τα αποτελέσματα που προκύπτουν είναι τα ακόλουθα:



Εικόνα 4 Ημερομηνίες δημιουργίας χρηστών στο *Twitter* [53].

Από το παραπάνω διάγραμμα συνάγεται ότι οι χρήστες που διαδίδουν αληθείς ειδήσεις στο *Twitter* είναι περισσότερο καιρο εγγεγραμμένοι στο *Twitter*. Συγκεκριμένα, γύρω στο 19% όλων των χρηστών που διαδίδουν αληθείς ειδήσεις είναι εγγεγραμμένοι εννιά με δέκα χρόνια στο *Twitter*.
Για την έκδοση ενός ακόμη στατιστικού για το σύνολο δεδομένων, οι ερευνητές του επιλέγουν τυχαία 10.000 χρήστες που αναρτούν ειδήσεις στο *Twitter* και με τη μέθοδο του *BotOrNot* αναδεικνύουν πόσοι από αυτούς τους λογαριασμούς αποτελούν κακόβουλο λογισμικό.



Εικόνα 5 Στατιστικά για το πόσοι χρήστες αποτελούν *bots* στη διάδοση ψευδών και αληθών ειδήσεων [53].

Dataset / Features	PolitiFact		GossipCop	
	Fake	Real	Fake	Real
# Users	214,049	700,120	99,765	69,910
# Followers	260,394,468	714,067,617	107,627,957	73,854,066
# Followees	286,205,494	746,110,345	101,790,350	75,030,435
Avg.# followers	1,216.518	1019.922	1078.815	1056.416
Avg.# followees	1,337.102	1065.689	1020.301	1073.243

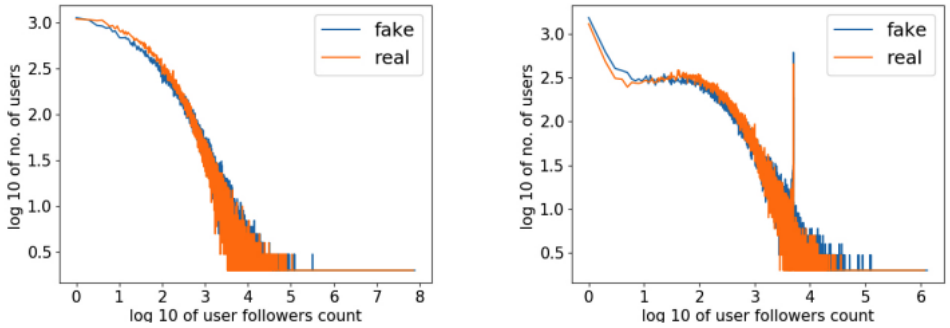
Πίνακας 3 Στατιστικά των Χαρακτηριστικων των Χρηστών του συνόλου δεδομένων *FakeNewsNet* [53].

2.7 User Posts

Οι άνθρωποι, μέσω των δημοσιεύσεών τους, εκφράζουν τα συναισθήματά τους, τις σκέψεις, τις απόψεις και τους προβληματισμούς τους. Τα χαρακτηριστικά αυτά είναι άκρως απαραίτητα για την μελέτη του προβλήματος των *fake news*.

2.8 Network Structure

Οι χρήστες τείνουν να σχηματίζουν διαφορετικά δίκτυα στα *social media* ανάλογα με τα ενδιαφέροντα, τα θέματα και τις πληροφορίες που λαμβάνουν. Οι διαδικασίες διάδοσης ψευδών ειδήσεων συχνά σχηματίζουν έναν *echo chamber cycle* όπου οι χρήστες ζουν απομονωμένοι και δεν εκτίθενται σε πιο έγκυρα και ρεαλιστικά ερεθίσματα περιεχομένου ειδήσεων. Τα χαρακτηριστικά του κοινωνικού δικτύου, όπως αριθμός ακολούθων του χρήστη και αριθμός των χρηστών που ακολουθεί ο χρήστης μπορούν να χρησιμοποιηθούν ως βάση για την εκτίμηση της διάδοσης των ψευδών ειδήσεων στο *Twitter*. Παρατηρώντας τα παρακάτω διαγράμματα για τα δύο αυτά χαρακτηριστικά, συμπεραίνουμε ότι τα *follower* και *followee count* σημειώνουν κατανομή *power law*, η οποία παρατηρείται συχνά στα *social media*. Επιπλέον, υπάρχει μία κορυφή στο διάγραμμα *Followee* count of users και για τις δύο κατηγορίες των χρηστών. Ο λόγος είναι επειδή το *Twitter* εχει θέσει ως περιορισμό οι χρήστες να μην έχουν πάνω από 5000 ακολούθους, όταν ο αριθμός των χρηστών που ακολουθούν είναι μικρότερος από 5000.



Εικόνα 6 Διαγράμματα *follower count of users* και *followee count of users* [53].

2.9 Dynamic Information

Η δυναμική πληροφορηση περιγράφει τη δυναμική ανανέωση των *news content* και *social context*. Καταγράφονται οι χρονικές στιγμές των *user likes, reposts, answers* (*user engagements*) τα οποία μπορούν να χρησιμοποιηθούν ως βάση για την μελέτη της διάδοσης των ειδήσεων στα *social media*. Καθώς τα *fact-checking websites* διαρκώς επεξεργάζονται νέες ειδήσεις, δυναμικά αυτές προστίθενται στο σύνολο δεδομένων *FakeNewsNet*. Έπειτα, για κάθε μία από τις νέες αυτές ειδήσεις, οι συγγραφείς της δημοσίευσης [53] συγκεντρώνουν τα *user engagements* όπως τις πρόσφατες, σχετικές με τις ειδήσεις, αναρτήσεις και τις *second order user behaviors* όπως *replies, likes,* και *retweets*. Επεξηγηματικά, η παραπάνω διαδικασία επιτυγχάνεται τρέχοντας τον *news content crawler* και ανανεώνοντας τον *Tweet collector* καθημερινά διαδικασία που ακολουθήθηκε από τους δημιουργούς του dataset. Η δυναμική πληροφόρηση μας παρέχει χρήσιμη και εξαντλητική πληροφόρηση για το πρόβλημα ανίχνευσης ψευδών ειδήσεων.

Παρακάτω, γίνεται μια σύντομη παρουσίαση και ανάλυση του τομέα της Μηχανικής Μάθησης. Ορίζονται οι τύποι προβλημάτων και εργασιών που αφορούν τη συγκεκριμένη εργασία. Στην προκειμένη περίπτωση, διαχειριζόμαστε πρόβλημα ημι-επιβλεπόμενης μάθησης. Επιπλέον, παρουσιάζονται και οι διάφορες τεχνικές και προσεγγίσεις για την επίλυση προβλημάτων βάσει Μηχανικής Μάθησης. Στην εν λόγω εργασία, ασχολούμαστε με τεχνητά νευρωνικά δίκτυα στη μορφή γράφων καθώς και με βαθιά μάθηση. Σύντομη περιγραφή παρατίθεται και για τις υπόλοιπες τεχνικές της Μηχανικής Μάθησης.

3. Μηχανική Μάθηση

Ο τομέας αυτός αφορά τη δημιουργία αλγορίθμων οι οποίοι αποκτούν γνώση δίχως πρότερο προγραμματισμό βάσει αυστηρών κανόνων. Συγκεκριμένα, οι αλγόριθμοι της μηχανικής μάθησης, εντοπίζουν μοτίβα (*pattern matching*), στα δεδομένα εισόδου και έπειτα, βάσει αυτών, προβαίνουν σε προβλέψεις παράγοντας τις αντίστοιχες εξόδους.
Ο κλάδος της Μηχανικής Μάθησης εξελίχθηκε ερευνώντας τους τομείς του pattern recognition (*αναγνώριση προτύπων*) καθώς και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη, όπου της τελευταίας αποτελεί υποσύνολο η Μηχανική Μάθηση.
Η επιστημονική μελέτη αλγορίθμων και στατιστικών μοντέλων που οι υπολογιστές χρησιμοποιούν για να επιτελέσουν μια συγκεκριμένη εργασία χωρίς να έχουν ρητά προγραμματιστεί, αποτελεί την κεντρική έννοια της Μηχανικής Μάθησης. Το πλεονέκτημα των αλγορίθμων μηχανικής μάθησης, εντοπίζεται στο ότι μόλις ο αλγόριθμος εκπαιδευτεί στο να διαχειρίζεται τα δεδομένα εισόδου, έπειται δουλεύει αυτόματα, δίχως εξαντλητικό κώδικα.
Οι αλγόριθμοι αυτοί εντοπίζονται σε πολλές εφαρμογές οι οποίες χρησιμοποιούνται καθημερινά. Παραδειγματικά, κάθε φορά που χρησιμοποιείται μια μηχανή αναζήτησης, όπως το Google, ο λόγος που επιτελείται η εργασία της τόσο αποδοτικά είναι επειδή εφαρμόζεται ένας αλγόριθμος μάθησης που περιηγεί των χρήστη στις διαδικτυακές σελίδες. Η υλοποίηση σύνθετων μοντέλων καθώς και αλγορίθμων, βάσει των οποίων θα προβεί σε προβλέψεις το μοντέλο βρίσκει εφαρμογή και στον τομέα της ανάλυσης δεδομένων (*data analysis*). Οι επιστήμονες των συγκεκριμένων κλάδων, χάρη στη βελτιστοποίηση των μοντέλων της ανάλυσης δεδομένων, έχουν την δυνατότητα να εξάγουν όσο το δυνατόν πιο έγκυρα συμπεράσματα, για το κάθε πρόβλημα που μελετούν, γεγονός ανεκτίμητης αξίας σε ανθρωπιστικό επίπεδο.

3.1 Είδη Μάθησης

Η πιο συνήθης κατηγοριοποίηση των διεργασιών μηχανικής μάθησης, αφορά τους τομείς: Επιτηρούμενη Μάθηση, Μη επιτηρούμενη Μάθηση, Ενισχυτική Μάθηση. Η κάθε μέθοδος συνοδεύεται με τα πλεονεκτήματα και τα μειονεκτήματά της, ενώ για διαφορετικούς τύπους προβλημάτων επιλέγεται και διαφορετικό είδος μάθησης, όπως αναλύεται παρακάτω.
Οι εργασίες μηχανικής μάθησης συνήθως ταξινομούνται σε τρεις μεγάλες κατηγορίες, ανάλογα με τη φύση του εκπαιδευτικού «σήματος» ή την «ανατροφοδότηση» που είναι διαθέσιμα σε ένα σύστημα εκμάθησης. Αυτές είναι:

1. Επιτηρούμενη Μάθηση (*Supervised Learning*): Το μοντέλο της Μηχανικής Μάθησης εκπαιδεύεται δεχόμενο ως είσοδο δεδομένα με γνωστή κατηγοριοποίηση. Με τον τρόπο αυτό, το μοντέλο μαθαίνει και εκπαιδεύεται σε κάθε κατηγορία δεδομένων και έπειτα προβαίνει στη διαδικασία της αξιολόγησης (*test data*). Η επιτηρούμενη μάθηση χρησιμοποιείται σε προβλήματα ταξινόμησης (*classification*), όπου τα προς αξιολόγηση δεδομένα κατηγοριοποιούνται σε συγκεκριμένες, διακριτές, οντότητες. Ένα τέτοιο πρόβλημα αποτελεί η ταξινόμηση εικόνων σε τρεις κατηγορίες όπως δέντρο, σπίτι, όχημα. Η επιτηρούμενη μάθηση χρησιμοποιείται, επίσης, σε προβλήματα παλινδρόμησης (*regression*) όπου το μοντέλο καλείται να εντοπίσει τη σύνδεση μεταξύ ανεξάρτητης και εξαρτώμενης μεταβλητής και να προβλέψει μια συγκεκριμένη τιμή. Τέτοιο πρόβλημα αποτελεί η πρόβλεψη των κερδών των παρόχων κινητής τηλεφωνίας, δεδομένου του πλήθους πελατών τους.

2. Μη Επιτηρούμενη Μάθηση (*Unsupervised Learning*): Το μοντέλο της Μηχανικής Μάθησης χρησιμοποιεί αλγορίθμους μάθησης προκειμένου να αναλύσει και να ομαδοποιήσει (*cluster*) δεδομένα που δεν διαθέτουν ετικέτα για την κατηγορία στην οποία ανήκουν (*unlabeled data*). Κατά αυτόν τον τρόπο μάθησης, αναζητούνται τα κρυφά μοτίβα στο σύνολο δεδομένων και τα μοντέλα δρουν πάνω στα δεδομένα αυτά, δίχως επίβλεψη. Στη μη επιβλεπόμενη μάθηση, δεν είναι εφικτό να εφαρμοστεί απευθείας classification και regression, διότι δε διαθέτουμε τη σαφή κατηγοριοποίηση των δεδομένων εισόδου. Συνεπώς, το μοντέλο αναζητά τη δομή του συνόλου δεδομένων και προβαίνει σε ομαδοποίηση των δεδομένων βάσει των ομοιοτήτων τους. Ένα πρόβλημα της κατηγορίας αυτής μπορεί να αποτελέσει ένα σύνολο εικόνων από ποδήλατα και μηχανάκια, *unlabeled*, όπου ο αλγόριθμος μη επιβλεπόμενης μάθησης θα ομαδοποιήσει τις εικόνες του συνόλου δεδομένων βάσει των ομοιοτήτων μεταξύ των εικόνων.

3. Ενισχυτική Μάθηση (*Reinforcement Learning*): Το μοντέλο της Μηχανικής Μάθησης, που συχνά καλείται *agent*, καλείται να μάθει σε ένα διαδραστικό περιβάλλον μέσω της διαδικασίας "δοκιμής και σφάλματος" (*trial and error*), λαμβάνοντας ανατροφοδότηση και αξιολόγηση (*feedback*) από τις ίδιες του τις αποφάσεις. Το μοντέλο, δηλαδή, λαμβάνει επιδοκιμασμό ή αποδοκιμασμό ως ενδείξεις ορθής ή λανθασμένης συμπεριφοράς, με σκοπό τη μεγιστοποίηση του συνολικού του κέρδους. Ένας τρόπος να κατανοήσουμε την ενισχυτική μάθηση είναι μέσω των παιχνιδιών. Στο παιχνίδι πάκμαν (*PacMan*) το μοντέλο λαμβάνει επιβράβευση όσο κυκλοφορεί ασφαλές στο πλέγμα του παιχνιδιού, στο διαδραστικό δηλαδή περιβάλλον, και όσο τρώει φρούτα, ενώ λαμβάνει αποδοκιμασία όταν πεθαίνει από τους εχθρούς του.

Υπάρχει ένα ακόμα συχνά εμφανιζόμενο είδος μάθησης ανάμεσα στην επιβλεπόμενη και μη επιβλεπόμενη μάθηση, γνωστό ως ημι-επιβλεπόμενη μάθηση, όπου το μοντέλο πρέπει να διαχειριστεί δεδομένα που διαθέτουν ετικέτα κατηγοριοποίησης παράλληλα με δεδομένα που δε διαθέτουν ετικέτα κατηγοριοποίησης.

3.2 Τεχνικές Μάθησης Μηχανικής Μάθησης

Υπάρχει πληθώρα τεχνικών μάθησης, μοντέλων και αλγορίθμων που χρησιμοποιούνται προκειμένου να επιτελεσθούν οι διεργασίες και η επίλυση προβλημάτων κατά τη Μηχανική Μάθηση. Ορισμένες από αυτές αναπτύσσονται σύντομα ακολούθως.

Με το μοντέλο του δέντρου απόφασης επιτυγχάνεται η αντιστοίχιση παρατηρήσεων-συμπερασμάτων για κάθε τιμή του συνόλου τιμών. Για παράδειγμα, με το μοντέλο ομοιότητας του δέντρου απόφασης, εισάγονται στο μοντέλο μηχανικής μάθησης δυάδες δεδομένων που παρουσιάζουν ομοιότητες καθώς και δυάδες δεδομένων που παρουσιάζουν διαφορετικά χαρακτηριστικά. Με τον τρόπο αυτό, το μοντέλο καλείται να μάθει μια συνάρτηση απόστασεις η οποία θα εκφράζει την ομοιότητα ή ανομοιότητα μεταξύ των δεδομένων, η οποία αποσκοπεί στην πρόβλεψη της ομοιότητας μεταξύ των δυάδων των δειγμάτων.

Μοντέλα βασιζόμενα στον Επαγωγικό Λογικό προγραμματισμό, δέχονται ως είσοδο μια κωδικοποίηση του συνόλου δεδομένων που παρουσιάζονται σαν λογικό αποθετήριο πράξεων και προκύπτει η έξοδος του λογικού προγράμματος που περιλαμβάνει το σύνολο των επιθυμητών παραδειγμάτων δίχως κανένα μη επιθυμητό.

Οι Μηχανές Διανυσμάτων Υποστήριξης (SVMs), μέθοδοι που εφαρμόζονται κατά την επιτηρούμενη μάθηση για *regression* και *classification*, χρησιμοποιούνται κατά κύριο λόγο σε ένα σύνολο δεδομένων εκπαίδευσης, όπου για κάθε παράδειγμα προς επεξεργασία γνωστοποιείται σε ποια από τις δύο κλάσεις ανήκει [7].

Βάσει της μάθησης με κανόνες συσχέτισης, προκύπτει η σχέση μεταξύ των διαφόρων μεταβλητών σε μεγάλα αποθετήρια δεδομένων.

Η Ομαδοποίηση (*clustering*), διαδικασία που αναλύθηκε παραπάνω, αφορά την ομαδοποίηση των διαφόρων παραδειγμάτων του συνόλου δεδομένων σε συγκεκριμένες κατηγορίες βάσει των κοινών τους χαρακτηριστικών. Έτσι, τα δεδομένα εισόδου που έχουν ομαδοποιηθεί σε κοινή ομάδα, διαθέτουν παρόμοια χαρακτηριστικά [8].

Η ενισχυτική μάθηση καλείται να μάθει σε ένα διαδραστικό περιβάλλον μέσω της διαδικασίας "δοκιμής και σφάλματος" (*trial and error*), λαμβάνοντας ανατροφοδότηση και αξιολόγηση (*feedback*) από τις ίδιες του τις αποφάσεις. Το μοντέλο, δηλαδή, λαμβάνει επιδοκιμασμό ή αποδοκιμασμό ως ενδείξεις ορθής ή λανθασμένης συμπεριφοράς, με σκοπό τη μεγιστοποίηση του συνολικού του κέρδους.

Με τα δίκτυα Bayes, που συχνά αναφέρονται ως δίκτυα εμπιστοσύνης, αφορούν μια δομή δεδομένων σε μορφή γράφου, δίχως κύκλο, και αποτελούν το πιθανοτικό μοντέλο της μηχανικής μάθησης αναπαριστώντας τυχαίες ανεξάρτητες μεταβλητές [10].

Παραδειγματικά, με το παραπάνω μοντέλο μπορούν να αποδοθούν πιθανές συνδέσεις μεταξύ εστιατορικών πιάτων και υλικών υλοποίησής τους. Γνωρίζοντας τα υλικά υλοποίησής τους, τα Μπεϋζιανά μοντέλα υπολογίζουν τις πιθανότητες μαγειρέματος των εκάστοτε πιάτων.

Οι γενετικοί αλγόριθμοι βασίζονται στη τεχνική της φυσικής επιλογής. Μιμούμενοι τους έμβιους οργανισμούς έτσι και οι λύσεις του προβλήματος, εφαρμόζοντας τους κατάλληλους τελεστές, μεταβιβάζουν χαρακτηριστικά από την πρωτύτερη στην πιο ύστερη γενιά, με αποτέλεσμα την απαλοιφή μη επιθυμητών λύσεων, ενώ παράλληλα βελτιστοποιούνται οι επιθυμητές λύσεις.

Ολοκληρώνουμε με την περιγραφή του όρου της Βαθιάς Μάθησης, όπου χάρη στην μείωση των τιμών και στα υλικά κατασκευής των GPU, αξιοποιήθηκε η ταχύτητα και η υπολογιστική τους ισχύ και τομείς όπως η όραση υπολογιστών και επεξεργασία της φυσικής γλώσσας άνθισαν.

Στην παρακάτω παράγραφο δίδεται ο ορισμός της έννοιας “Τεχνητή Νοημοσύνη”. Παράλληλα παρουσιάζονται οι διάφορες τεχνικές επίλυσης προβλημάτων, όπως η αναζήτηση και βελτιστοποίηση, η λογική και τα Τεχνητά Νευρωνικά Δίκτυα. Οι δύο πρώτες μέθοδοι δεν χρησιμοποιούνται στην παρούσα εργασία, ωστόσο παρουσιάζονται συνοπτικά για λόγους πληρότητας. Η εργασία αυτή βασίζεται στα Τεχνητά Νευρωνικά Δίκτυα, Νευρωνικά Δίκτυα για συντομία, ως εργαλεία της Τεχνητή Νοημοσύνης. Για το λόγο αυτό, γίνεται μια πιο ενδελεχής ανάλυση αυτών, στην παράγραφο 4.3, καθώς και των εννοιών που σχετίζονται με αυτά, αφού χρησιμοποιούνται στην υλοποίηση της διπλωματικής εργασίας.

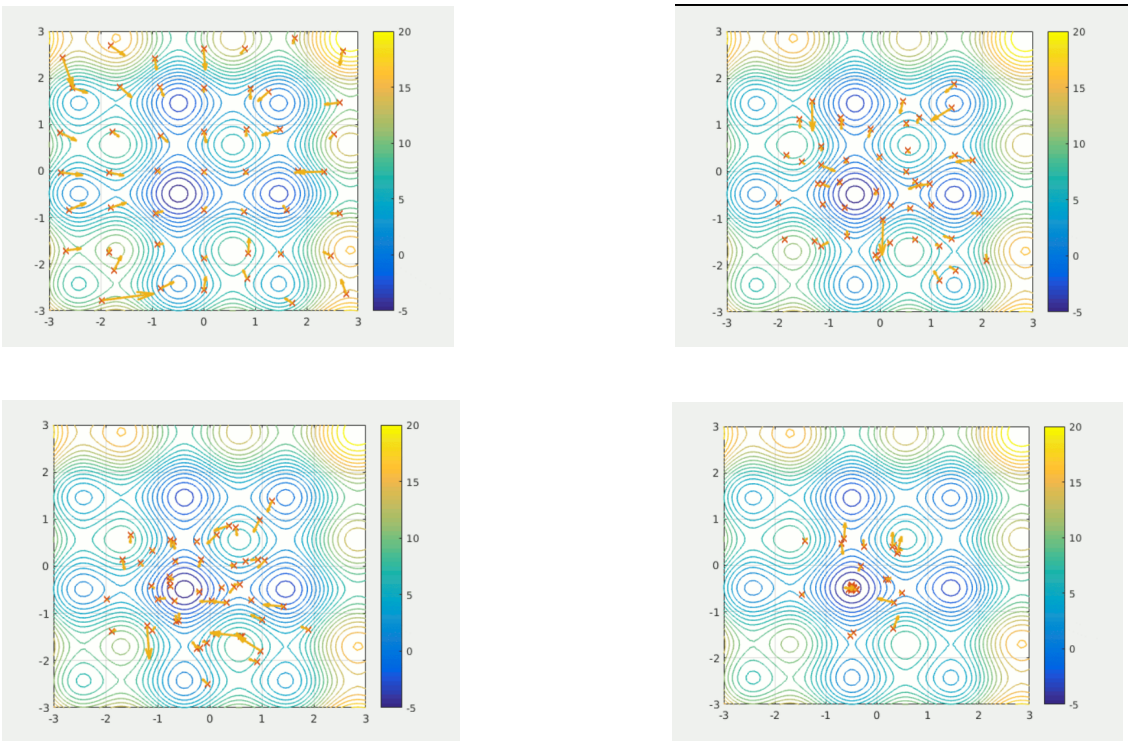
4. Τεχνητή Νοημοσύνη

Η επινόηση και η δημιουργία μοντέλων υπολογισμού που προσπαθούν να αντιγράψουν δείγματα ανθρώπινης νόησης τα οποία διαθέτουν ορισμένη ευφυΐα περιγράφει τον όρο της Τεχνητής Νοημοσύνης. Τα μοντέλα μάθησης πρέπει να είναι ευέλικτα και να προσαρμόζονται, να προβλέπουν, να γενικεύουν και εν τέλει να δίνουν λύση στο πρόβλημα που τους έχει ανατεθεί. Στόχος της τεχνητής νοημοσύνης είναι η προσομοίωση η και η μίμηση της ανθρώπινης νοησης από μηχανές. Η μηχανές αυτές προγραμματίζονται με τρόπο τέτοιο ώστε να μπορούν να “σκέφτονται” όπως οι άνθρωποι και να αναπαράγουν τις ενέργειές τους.

Το βασικό εργαλείο της Τεχνητής νοημοσύνης το οποίο πραγματεύεται το παρόν έργο είναι τα Τεχνητά Νευρωνικά Δίκτυα. Για λόγους πληρότητας παρατίθενται δύο επιπλέον βασικά εργαλεία της τεχνητής νοημοσύνης η “Βελτίωση βάσει Αναζήτησης” και “Λογική”.

4.1 Βελτίωση βάσει Αναζήτησης

Η επίλυση προβλημάτων σε θεωρητικό επίπεδο χρησιμοποιώντας ευφυή αναζήτηση ανάμεσα σε πολλές πιθανές λύσεις, αποτελεί ένα από τα εργαλεία της τεχνητης νοημοσύνης. Η πληθώρα πιθανών λύσεων περιορίζεται βάσει της λογικής. Η εύρεση συντομότερων μονοπατιών τόσο στον τομέα της μηχανικής μάθησης, όσο και της ρομποτικής, με την μετακίνηση των άκρων ρομποτικών χειριστών, επιτυγχάνεται χάρη στη διαδικασία αναζήτησης και βελτιστοποίησης.
Οι τετριμμένες εξαντλητικές αναζητήσεις, σπάνια αποδίδουν σε προβλήματα πραγματικού κόσμου, διότι η διάσταση του χώρου εργασίας, ο αριθμός δηλαδή των χώρων αναζήτησης, γρήγορα μεγεθύνεται σε αστρονομικό αριθμό. Το αποτέλεσμα, άρα, είναι μια αναζήτηση πολύ αργή ή μια αναζήτηση ατέρμονη. Για την αντιμετώπιση του φαινομένου αυτού, χρησιμοποιούνται ευριστικές ή αυτοσχέδιοι κανόνες (rules of thumb), οι οποίες δίνουν προτεραιότητα σε επιλογές που είναι πιο πιθανές να πετύχουν ένα στόχο σε συντομότερο αριθμό βημάτων.
Οι πιο δημοφιλείς αλγόριθμοι εξέλιξης περιλαμβάνουν γενετικούς αλγορίθμους, *gene expression programming* και *genetic programming*. Από την άλλη πλευρά, οι κατανεμημένοι αλγόριθμοι αναζήτησης (*distributed search processes*), βασίζονται σε *swarm intelligence algorithms* με τους πιο ευρέως εφαρμοσμένους από αυτούς να είναι οι *particle swarm optimization* και *ant colony optimization*.



Εικόνα 7 Μέθοδος particle swarm στην αναζήτηση του ολικού ελαχίστου [55].

4.2 Λογική

Κατά την αναπαράσταση της γνώσης και την επίλυση προβλημάτων είναι έντονη η εφαρμογή του τομέα της λογικής. Παραδειγματικά αναφέρονται οι αλγόριθμοι αυτοματοποιημένου σχεδιασμού βάσει της λογικής συνοδευόμενες από τον επαγωγικό λογικό προγραμματισμό ως μέθοδο μάθησης. Η τεχνητή νοημοσύνη χρησιμοποιεί κυρίως τα κάτωθι είδη λογικής [15], [16].

- Προτασιακή λογική:
  Περιλαμβάνει συναρτήσεις αληθείας όπως “ή” και “όχι”

- Λογική πρώτης τάξης:
  Προσθέτει κατηγορήματα και εκφράσεις ποσοδεικτών όπως υπάρχει ($\exists$) και για κάθε ($\forall$). Εκφράζει δεδομένα για αντικείμενα, τις ιδιότητές τους και τις μεταξύ τους σχέσεις.

- Ασαφής Λογική:
  Προσδίδει έναν βαθμό αληθείας, ανάμεσα στο μηδέν και το ένα, προκειμένου να κάνουν ασαφείς προτάσεις όπως “Η Ειρήνη είναι πολύ ψηλή”, προτάσεις δηλαδή που είναι γλωσσολογικά ανακριβείς, προκειμένου να χαρακτηριστούν εντελώς ψευδείς ή αληθείς.

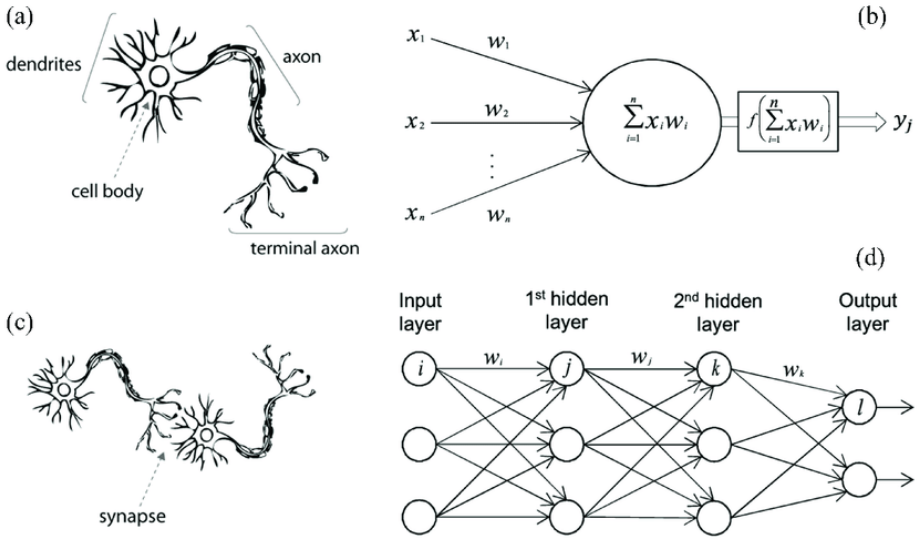
4.3 Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)

Τα τεχνητά νευρωνικά δίκτυα (ΤΝΔ), που συνήθως καλούνται νευρωνικά δίκτυα, αποτελούν υπολογιστικά συστήματα εμπνευσμένα από τα βιολογικά νευρωνικά δίκτυα, τα οποία απαρτίζουν τους ανθρώπινους εγκεφάλους [57].

Ένα ΤΝΔ βασίζεται σε μία συλλογή μονάδων ή, αλλιώς, κόμβων, που ονομάζονται τεχνητοί νευρώνες, οι οποίοι μοντελοποιούν τους νευρώνες ενός βιολογικού εγκεφάλου. Κάθε σύνδεση, όπως η σύναψη στον βιολογικό εγκέφαλο, μεταφέρει ένα σήμα σε άλλους νευρώνες. Το ΤΝΔ λαμβάνει σήματα, έπειτα τα επεξεργάζεται και δημιουργεί τους νευρώνες του σήματος. Το "σήμα" [56] ,[57] σε μια σύνδεση είναι ένας πραγματικός αριθμός και το αποτέλεσμα κάθε νευρώνα υπολογίζεται από κάποια μη γραμμική συνάρτηση, ως το άθροισμα των εισόδων του νευρώνα.

Οι συνδέσεις ονομάζονται ακμές [17]. Οι νευρώνες και οι ακμές τυπικά λαμβάνουν μια τιμή βάρους, η οποία προκύπτει από τη διαδικασία της μάθησης. Η τιμή αυτή, αυξάνει ή μειώνει την ισχύ και τη σημαντικότητα του σήματος σε μία σύνδεση. Οι νευρώνες συνήθως έχουν ένα κατώφλι (*threshold*) το οποίο επιτρέπει την αποστολή του σήματος μόνο εάν το συνολικό σήμα (*aggregated signal*) υπερβαίνει αυτό το κατώφλι.

Συνήθως, οι νευρώνες συγκεντρώνονται και αθροίζονται (*aggregated*) σε κάθε επίπεδο. Ενδεχομένως, κάθε επίπεδο να εφαρμόζει διαφορετικούς μετασχηματισμούς στις εισόδους του. Τα σήματα ταξιδεύουν από το πρώτο επίπεδο, το επίπεδο εισόδου, στο τελευταίο επίπεδο, το επίπεδο εξόδου, πιθανώς έχοντας διασχίσει τα επίπεδα αρκετές φορές.



Εικόνα 8 Ένας βιολογικός νευρώνας εν συγκρίσει με ένα τεχνητό νευρωνικό δίκτυο. (a) Ανθρώπινος Νευρώνας, (b) τεχνητός νευρώνας, (c) Βιολογική Σύναψη, (d) Σύναψη Τεχνητών Νευρωνικών Δικτύων [58].

4.3.1 Εκπαίδευση

Τα ΤΝΔ μαθαίνουν και εκπαιδεύονται επεξεργαζόμενα παραδείγματα, κάθε ένα από τα οποία περιέχει μία γνωστή "είσοδο" και ένα γνωστό "αποτέλεσμα", σχηματίζοντας σχέσεις βαρών πιθανοτήτων ανάμεσα στην "είσοδο" και στο "αποτέλεσμα". Οι σχέσεις βαρών πιθανοτήτων αποθηκεύονται ανάμεσα στη δομή των δεδομένων του δικτύου.
Η εκπαίδευση του νευρωνικού δικτύου από ένα δεδομένο παράδειγμα, συνήθως διεξάγεται καθορίζοντας τη διαφορά ανάμεσα στο επεξεργασμένο αποτέλεσμα εξόδου (*prediction*) και στο πραγματικό αποτέλεσμα εξόδου (*target output*). Η διαφορά αυτή αποτελεί το σφάλμα. Έπειτα, το δίκτυο προσαρμόζει τις σχέσεις βαρών σύμφωνα με τον κανόνα μάθησης και το σφάλμα που προέκυψε.
Οι επιτυχημένες προσαρμογές στις σχέσεις βαρών, συνεισφέρουν στην παραγωγή ενός αποτελέσματος το οποίο είναι παρόμοιο με το πραγματικό αποτέλεσμα (*target output*).
Έπειτα από έναν επαρκή αριθμό προσαρμογών βαρών η εκπαίδευση τερματίζεται χρησιμοποιώντας συγκεκριμένα κριτήρια. Το παραπάνω σύστημα εκπαίδευσης περιγράφει τον *supervised* τρόπο μάθησης. Τα συστήματα αυτά μαθαίνουν να επιτελούν διεργασίες θεωρώντας παραδείγματα και στη γενική περίπτωση δίχως να είναι προγραμματισμένα σε κανόνες συγκεκριμένης διεργασίας. Παραδειγματικά, σε διεργασίες όπως αναγνώριση εικόνων (*image recognition*) μαθαίνουν να αναγνωρίζουν εικόνες που περιέχουν κάποιο συγκεκριμένο ζώο, αναλύοντας παραδείγματα εικόνων που περιέχουν το εν λόγω ζώο. Ας υποθέσουμε ότι το ζώο αυτό είναι ένα πάντα. Οι εικόνες αυτές έχουν *manually* λάβει την ετικέτα πάντα ή όχι πάντα. Η αναγνώριση αυτή επιτυγχάνεται δίχως πρότερη γνώση για τα χαρακτηριστικά των ζώων αυτών. Δεν είναι γνωστές δηλαδή πληροφορίες για το τρίχωμα, το μέγεθος, και τα χαρακτηριστικά του προσώπου τους. Αντιθέτως, τα προσδιοριστικά και αναγνωριστικά χαρακτηριστικά των ζώων αυτών προκύπτουν αυτόματα από τα παραδείγματα που επεξεργάζεται το νευρωνικό.

4.3.2 Μοντέλα

Τα ΤΝΔ ξεκίνησαν ως μια προσπάθεια προσομοίωσης της αρχιτεκτονικής του ανθρώπινου εγκεφάλου, προκειμένου να αποδίδει σε διεργασίες όπου οι συνηθισμένοι αλγόριθμοι τεχνητής νοημοσύνης δεν παρουσίαζαν σημαντική επιτυχία.
Οι νευρώνες ενώνονται μεταξύ τους, με διάφορα μοτίβα, προκειμένου η έξοδος ορισμένων νευρώνων να γίνει η είσοδος κάποιων άλλων. Το συνολικό δίκτυο σχηματίζει έναν κατευθυνόμενο γράφο με βάρη.
Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από μια συλλογή νευρώνων προσομοίωσης. Κάθε νευρώνας είναι ένας κόμβος ο οποίος συνδέεται με άλλους κόμβους μέσω συνδέσμων, ακμών, που θυμίζουν τις βιολογικές αξονικές συνάψεις δενδριτών. Οι δενδρίτες είναι αποφυάδες βιολογικών νευρώνων. Κάθε ακμή έχει ένα βάρος, το οποίο καθορίζει την ισχύ του έχει ένας κόμβος στο να επηρεάζει τον άλλο.

4.3.3 Τεχνητοί Νευρώνες

Οι τεχνητοί νευρώνες λαμβάνουν εισόδους και παράγουν μία έξοδο η οποία μπορεί να σταλεί σε πολλούς άλλους νευρώνες [57].
Ως είσοδοι μπορούν να εισαχθούν οι τιμές των χαρακτηριστικών ενός δείγματος εξωτερικών δεδομένων, τα οποία μπορεί να είναι εικόνες, έγγραφα ή ακόμα έξοδοι άλλων νευρώνων. Οι έξοδοι

των νευρώνων εξόδου του νευρωνικού δικτύου ολοκληρώνουν τη ζητούμενη διαδικασία η οποία για παράδειγμα μπορεί να είναι η αναγνώριση ενός αντικειμένου σε μια εικόνα.
Για την εξαγωγή του αποτελέσματος εξόδου, λαμβάνουμε το άθροισμα βαρών όλων των δεδομένων εισόδου προσθέτοντας τον όρο *bias*. Το άθροισμα αυτό των βαρών, καλείται ενεργοποίηση και μέσω μιας μη γραμμικής συνάρτησης παράγεται το αποτέλεσμα.
Οι νευρώνες οργανώνονται σε πολλά επίπεδα, σύμφωνα με το *deep learning*.
Το επίπεδο που λαμβάνει τα εξωτερικά δεδομένα είναι το επίπεδο εισόδου. Το επίπεδο που παράγει το τελικό αποτέλεσμα είναι το επίπεδο εξόδου. Ανάμεσα σε αυτά τα δύο επίπεδα, παρεμβάλλονται καθόλου ή και περισσότερα κρυφά επίπεδα. Ανάμεσα σε δύο επίπεδα πολλές συνδέσεις και μοτίβα είναι πιθανά.
Δύο επίπεδα καλούνται *fully-connected* όταν κάθε νευρώνας ενός επιπέδου συνδέεται με κάθε νευρώνα του επόμενου επιπέδου. [57]
Δύο επίπεδα καλούνται *pooling* όταν ένα σύνολο νευρώνων του ενός επιπέδου συνδέεται σε έναν νευρώνα του επόμενου επιπέδου. Με τον τρόπο αυτόν, μειώνονται οι νευρώνες του επιπέδου. Οι νευρώνες με συνδέσεις του προηγούμενου είδους σχηματίζουν έναν ακυκλικό γράφο γνωστό ως *feedforward network* [59].
Εναλλακτικά, τα δίκτυα που επιτρέπουν τις συνδέσεις μεταξύ νευρώνων του ιδίου ή προηγούμενων επιπέδων καλούνται δίκτυα ανάδρασης (*recurrent networks*) [60].

4.3.4 Υπερπαράμετροι

Η υπερπαράμετρος είναι μια σταθερή παράμετρος της οποίας η τιμή καθορίζεται πριν την διαδικασία της μάθησης. Παραδείγματα υπερπαραμέτρων είναι ο ρυθμός μάθησης, ο αριθμός των κρυφών επιπέδων καθώς και το μέγεθος των πακέτων δεδομένων που θα λαμβάνεται από τα δεδομένα εισόδου. Είναι σημαντική η βέλτιστη επιλογή παραμέτρων καθώς διαδραματίζουν καθοριστικό ρόλο στην απόδοση του μοντέλου.

4.3.5 Μαθηματική μοντελοποίηση των τεχνητών νευρωνικών δικτύων

Τα τεχνητά νευρωνικά δίκτυα συνδυάζουν βιολογικές αρχές με υψηλή στατιστική, με σκοπό την επίλυση προβλημάτων σε τομείς όπως αναγνώριση μοτίβων (*pattern recognition*). Τα ΤΝΔ υιοθετούν το βασικό μοντέλο των αναλογιών μάθησης συνδεδεμένα μεταξύ τους με ποικίλους τρόπους.

4.3.6 Δομή

4.3.6.1 Νευρώνας

Ένας νευρώνας, j, δεχόμενος μία είσοδο $p_j(t)$ από τους προηγούμενους (προγενέστερους) νευρώνες αποτελείται από τα ακόλουθα χαρακτηριστικά:

- Μια ενεργοποίηση $\alpha_j(t)$, την κατάσταση του νευρώνα, η οποία εξαρτάται από μια διακριτή χρονική παράμετρο.

- Ένα προαιρετικό κατώφλι (*threshold*) θ_j το οποίο μένει σταθερό και τροποποιείται, αν χρειαστεί, κατά τη διαδικασία της μάθησης.

- Μια συνάρτηση ενεργοποίησης f που υπολογίζει την νέα ενεργοποίηση σύμφωνα με τη σχέση:

$$o_j(t) = f_{out}(\alpha_j(t))$$

Ο νευρώνας εισόδου δεν διαθέτει προγενέστερους νευρώνες αλλά εξυπηρετεί ως διεπαφή εισόδου για ολόκληρο το δίκτυο. Παρομοίως, δεν υπάρχει νευρώνας που να διαδέχεται τον νευρώνα εξόδου και γι'αυτό εξυπηρετεί ως διεπαφή εξόδου για ολόκληρο το δίκτυο.

4.3.6.2 Συνάρτηση διάδοσης

Η συναρτηση διάδοσης υπολογίζει την είσοδο $p_j(t)$στο νευρώνα j από την έξοδο $o_i(t)$ σύμφωνα με τη σχέση:

$$p_j(t) = \sum_i o_i(t) w_{ij}$$

4.3.6.3 Bias

Ο όρος w_{0j} λέγεται bias και προστίθεται στην προηγούμενη σχέση

$$p_j(t) = \sum_i o_i(t) w_{ij} + w_{0j}$$

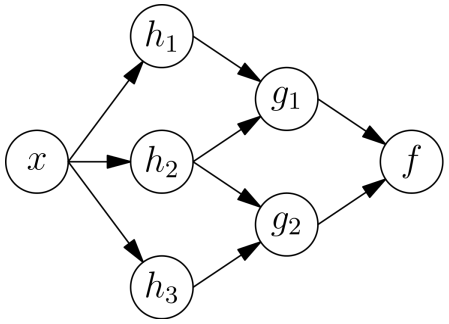
4.3.7 Νευρωνικά Δίκτυα ως Συναρτήσεις

Τα νευρωνικά δίκτυα μπορούν να μελετηθούν ως ο ορισμός μιας συνάρτησης $f: X \to Y$ η οποία λαμβάνει μια είσοδο (παρατήρηση) και παράγει ένα αποτέλεσμα (απόφαση).
Ορισμένες φορές τα μοντέλα είναι άμεσα συνδεδεμένα με έναν συγκεκριμένο κανόνα μάθησης. Με την έκφραση "Μοντέλο ΤΝΔ" ορίζεται μια κλάση τέτοιων συναρτήσεων, όπου τα μέλη της κλάσης αυτής χαρακτηρίζονται από ποικίλες παραμέτρους, όπως βάρη ακμών, αριθμό νευρώνων και επιπέδων.
Μαθηματικά, η συνάρτηση δικτύου f ενός νευρώνα ορίζεται ως η σύνθεση άλλων συναρτήσεων $g_j(x)$, που μπορούν να αποσυντεθούν περαιτέρω σε άλλες συναρτήσεις. Η δομή αυτή του δικτύου μπορεί να αναπαρασταθεί συμβατικά με βέλη που απεικονίζουν τις εξαρτήσεις μεταξύ των συναρτήσεων. Ένας ευρέως χρησιμοποιούμενος τύπος σύνθεσης είναι το μη-γραμμικό αθροισμα βαρών:

$$f(x) = K(\sum_i w_i g_i(x))$$

οπου το Κ είναι μια συνάρτηση ενεργοποίησης που έχει προσδιοριστεί πρωτύτερα όπως η συνάρτηση *softmax* ή *sigmoid*. Το σημαντικό χαρακτηριστικό της συνάρτησης ενεργοποίησης είναι ότι παρέχει μια ομαλή μετάβαση καθώς οι τιμές των εισόδων αλλάζουν. Επεξηγηματικά, μια μικρή

αλλαγή στα δεδομένα εισόδου έχει ως αποτέλεσμα μια μικρή αλλαγή στην έξοδο. Παρακάτω, θεωρούμε μια συλλογή συναρτήσεων g_i που αναπαρίστανται ως διάνυσμα $g_i = (g_1, g_2, \dots, g_n)$.



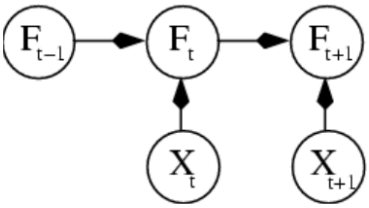
Εικόνα 9 Γράφος εξάρτησης Τεχνητού Νευρωνικού Δικτύου

Η εικόνα αναπαριστά την αποσύνθεση της συνάρτησης f με τις συναρτήσεις μεταξύ των μεταβλητών να αποδίδονται ως βέλη. Η αναπαράσταση αυτή μπορεί να επιτευχθεί με δύο τρόπους.

1. Ο πρώτος τρόπος είναι ο συναρτησιακός τρόπος: Η είσοδος x μετατρέπεται σε ένα τρισδιάστατο διάνυσμα h, το οποίο στην συνέχεια μετατρέπεται σε ένα δισδιάστατο διάνυσμα g το οποίο εν τέλει μετασχηματίζεται στη συνάρτηση f. Η μέθοδος αυτή συχνά χρησιμοποιείται στην τεχνική της βελτιστοποίησης.

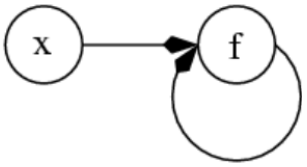
2. Ο δεύτερος τρόπος αφορά τον υπολογισμό πιθανοτήτων. Η τυχαία μεταβλητή $F = f(G)$ εξαρτάται από την τυχαία μεταβλητή $G = g(H)$, η οποία εξαρτάται από την H $= h(G)$, η οποία με τη σειρά της εξαρτάται από την τυχαία μεταβλητή X. Η τακτική αυτή συναντάται περισσότερο στα μοντέλα γράφων.

Οι δύο αυτοί τρόποι είναι αρκετά ισοδύναμοι. Σε κάθε περίπτωση, για τη συγκεκριμένη αυτή αρχιτεκτονική τα συστατικά του κάθε επιπέδου είναι ανεξάρτητα μεταξύ τους. Για παράδειγμα, τα συστατικά του g είναι ανεξάρτητα μεταξύ τους δεδομένων των εισόδων του h. Τα δίκτυα σαν το προηγούμενο συνηθως αποκαλούνται *feedforward*, καθώς οι γράφοι τους είναι κατευθυνόμενοι και μη κυκλικοί.



Εικόνα 10 Δίκτυο feedforward

Δίκτυα με κύκλους ονομάζονται δίκτυα ανάδρασης. Τέτοια δίκτυα συχνά αναπαρίστανται με τον ακόλουθο τρόπο, όπου η f είναι εξαρτώμενη από τον εαυτό της



Εικόνα 11 Δίκτυο ανάδρασης

4.3.8 Backpropagation

Αλγόριθμος

Έστω δίκτυο *N* με *e* συνδέσεις, *m* εισόδους και *n* εξόδους. Παρακάτω θα θεωρούμε τα x_1,x_2, … διανύσματα στο $\mathbb{R}^m$, y_1,y_2, … διανύσματα στο $\mathbb{R}^n$ και w_1,w_2, … διανύσματα στο $\mathbb{R}^e$. Τα διανύσματα αυτά συμβολίζουν την είσοδο, την έξοδο και τα βάρη αντιστοίχως.
Το δίκτυο ανταποκρίνεται σε μια συνάρτηση $y = f_N(w, x)$ η οποία δεδομένης ενός βάρους *w* αντιστοιχεί την είσοδο x στην έξοδο y.
Κατά την επιβλεπόμενη μάθηση, μια ακολουθία παραδειγμάτων εκπαίδευσης (x_1, y_1), … , (x_p, y_p) παράγει μια ακολουθία βαρών w_0,w_1, … , w_p η οποία εκκινεί από ένα τυχαία επιλεγμένο αρχικό βάρος w_0 .
Τα βάρη αυτά υπολογίζονται με την ακόλουθη σειρά:
Αρχικά, υπολογίζουμε το w_i χρησιμοποιώντας μόνο τα (x_i, y_i, w_{i-1}) για *i = 1, … , p*. Η έξοδος του αλγορίθμου είναι τότε το w_p δεδομένης μιας νέας συνάρτησης $x \to f_N(w_p, x)$. Ο υπολογισμός είναι ο ίδιος σε κάθε βήμα. Για το λόγο αυτό, παραθέτουμε μόνο τον υπολογισμό για *i = 1*.
Το w_1 υπολογίζεται από τα (x_1, y_1, w_0) θεωρώντας ένα βάρος w_0 ως μεταβλητή και εφαρμόζοντας *gradient descent* στη συνάρτηση $w \to E(f_N(w_0, x_1), y_1)$ για την εύρεση ενός τοπικού ελαχίστου. Έτσι, προκύπτει το w_1 ως το ελάχιστο βάρος που προέκυψε από το *gradient descent*.
Προκειμένου να αρχικοποιήσουμε τον παραπάνω αλγόριθμο, ορίζουμε την συνάρτηση $E(y, y') = |y - y'|^2$. Ο αλγόριθμος μάθησης χωρίζεται σε δύο μέρη. Τη διάδοση (*propagation*) και την ανανέωση βαρών (*weight update*) [61].

4.3.9 Συνάρτηση Κόστους

Η συνάρτηση κόστους ή αλλιώς συνάρτηση σφάλματος είναι μια σημαντική παράμετρος που καθορίζει πόσο αποδοτικό είναι το μοντέλο στο δεδομένο *dataset*. Συγκεκριμένα υπολογίζει τη διαφορά ανάμεσα στην αναμενόμενη τιμή και την τιμή που προέβλεψε το νευρωνικό δίκτυο.

Στην παρούσα διπλωματική εργασία η συνάρτηση κόστους που χρησιμοποιείται είναι η *binary cross entropy*.

4.3.9.1 Binary Cross Entropy

Στο πρόβλημα δυαδικής ταξινόμησης για το αν η προς εξέταση είδηση πρόκειται για *fake* ή *real news* εξυπηρετεί η χρήση της *Binary Cross Entropy loss function*. Ο τύπος της είναι ο ακόλουθος:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N}\{ y_i log(p(y_i)) + (1 - y_i) log(1 - p(y_i)) \}$$

όπου το *y* ισούται με 1 εάν η είδηση είναι ψευδής και με 0 εάν η είδηση είναι αληθής. Ερμηνεύοντας τον παραπάνω τύπο διαπιστώνουμε ότι για κάθε ψευδή είδηση (*y*=1) προστίθεται ο όρος $log(p(y))$ στο κόστος και στην αντίθετη περίπτωση ο όρος $log(1 - p(y))$.
Ο λόγος για τον οποίον παίρνουμε το λογάριθμο της πιθανότητας ως κόστος εντοπίζεται ακολούθως, Ο υπολογισμός του κόστους έχει να κάνει με την απόσταση της τιμής πρόβλεψης από την πραγματική τιμή. Εάν, λοιπόν, η πιθανότητα που σχετίζεται με την πραγματική κλάση ισούται με ένα, τότε το κόστος πρέπει να είναι μηδενικό. Εάν η πιθανότητα αυτή ήταν πολύ χαμηλή και ίση με 0.01 τότε το κόστος θα έπρεπε να ήταν γιγαντώδες. Προκύπτει, λοιπόν, πως λαμβάνοντας τον αρνητικό λογάριθμο της πιθανότητας εξυπηρετείται ο παραπάνω σκοπός, διότι οι τιμές των λογαρίθμων για στο διάστημα (0,1) των πιθανοτήτων είναι αρνητικές. Με τον αρνητικό λογάριθμο, για πολύ μικρές πιθανότητες λαμβάνουμε, πράγματι, μεγάλο κόστος [11].

4.3.10 Propagation

Η διάδοση περιλαμβάνει τα εξής στάδια:

- Propagation forward στο δίκτυο για την παραγωγή των τιμών εξόδων.
- Υπολογισμός κόστους
- Διάδοση του αποτελέσματος των ενεργοποιήσεων πίσω στο νευρωνικό δίκτυο χρησιμοποιώντας το μοτίβο της εκπαίδευσης για την παραγωγή των *deltas*, δηλαδή της διαφοράς ανάμεσα στην πραγματική τιμή εξόδου και σε αυτήν που προέβλεψε το νευρωνικό. Για την παραγωγή των *deltas* συλλέγονται όλα τα *outputs* και αυτά των κρυφών νευρώνων.

4.3.11 Weight update

Για κάθε βάρος
- Πολλαπλασιάζουμε την έξοδο του βάρους *delta* και της ενεργοποίησης της εισόδου για να βρούμε την παράγωγο του βάρους.
- Αφαιρούμε τον λόγο (*percentage*) της παραγώγου του βάρους από το βάρος.

4.3.12 Ρυθμός Μάθησης

Ο ρυθμός μάθησης είναι ο λόγος (*percentage*) που επηρεάζει την ταχύτητα και την ποιότητα της μάθησης. Όσο μεγαλύτερος είναι ο λόγος, τόσο το γρηγορότερο εκπαιδεύεται ο νευρώνας, αλλά όσο μικρότερος ο λόγος η εκπαίδευση είναι πιο ακριβής. Το πρόσημο της παραγώγου ενός βάρους εκφράζει εάν το σφάλμα ποικίλει ομοίως ή αντίστροφα με το βάρος. Γι'αυτό, το βάρος πρέπει να ανανεώνεται στην αντίθετη κατεύθυνση με το "*descending*" της παραγώγου. Η διαδικασία της εκπαίδευσης επαναλαμβάνεται με νέα πακέτα δεδομένων εισόδου, έως ότου το δίκτυο να αποδίδει ικανοποιητικά.

4.3.13 Stochastic Gradient Descent

Η επαναληπτική διαδικασία *Stochastic Gradient Descent*, που συχνά συμβολίζεται με *SGD*, αποτελεί μια μέθοδο βελτιστοποίησης μιας συνάρτησης κόστους με τις κατάλληλες ιδιότητες διαφορισιμότητας. Μπορεί να θεωρηθεί ως η στοχαστική προσέγγιση της *gradient descent* βελτιστοποίησης, εφόσον αντικαθιστά την πραγματική παράγωγο (που υπολογίζεται από ολόκληρο το σύνολο δεδομένων) με έναν υπολογισμό που προκύπτει από ένα τυχαία επιλεγμένο υποσύνολο των δεδομένων [13]. Ειδικότερα, στα προβλήματα βελτιστοποίησης υψηλών διαστάσεων, η μέθοδος αυτή περιορίζει σημαντικά την υπολογιστή πολυπλοκότητα, αποδίδοντας γρηγορότερες επαναλήψεις με μικρότερο, ωστόσο, ρυθμό σύγκλισης.
Τόσο κατά τους στατιστικούς υπολογισμούς όσο και στον τομέα του *machine learning* θεωρούμε το πρόβλημα ελαχιστοποίησης της συνάρτησης κόστους, η οποία παρίσταται με τη μορφή του αθροίσματος:

$$Q(w) = \frac{1}{n}\sum_{i=1}^{n} Q_i(w)$$

όπου αναζητείται η παράμετρος w που ελαχιστοποιεί το $Q(w)$ [12],[14]. Κάθε συνάρτηση Q_i σχετίζεται με την *i*-οστή παρατήρηση στο σύνολο δεδομένων, κατά την εκπαίδευση.
Για την ελαχιστοποίηση της παραπάνω συνάρτησης, η *standard* ή αλλιώς *batch gradient descent* μέθοδος θα ακολουθούσε τις παρακάτω επαναλήψεις:

$$w := w - \eta \nabla Q(w) = w - \frac{\eta}{n}\sum_{i=1}^{n} \nabla Q_i(w)$$

όπου το η εκφράζει το ρυθμό μάθησης.

Ψευδοκώδικας

Ο ψευδοκώδικας με τον αλγόριθμο *stochastic gradient descent* για την εκπαίδευση ενός νευρωνικού δικτύου τριών επιπέδων, με ένα κρυφό επίπεδο, είναι ο ακόλουθος:

```
initialize network weights (often small random values)
do
   for each training example named ex do
      prediction = neural_net_output(network, ex)  // forward pass
      actual = teacher_output(ex)
      compute error (prediction - actual) at the output units
         compute Δw_h for all weights from hidden layer to output layer  // backward
//pass
         compute Δw_i for all weights from input layer to hidden layer   // backward pass
//continued
      update network weights // input layer not modified by error estimate
until error rate becomes acceptably low
return the network
```

Οι γραμμές που έχουν ως σχόλιο το *backward pass* αρχικοποιούνται χρησιμοποιώντας τον *backpropagation* αλγόριθμο, ο οποίος υπολογίζει την παράγωγο του σφάλματος του δικτύου που εξαρτάται από τα μεταβλητά του βάρη [12].

5. Νευρωνικά Δίκτυα σε μορφή Γράφου (GNN)

Η εκτεταμένη εφαρμογή των νευρωνικών δικτύων σε προβλήματα Machine Learning συνοδευόμενη από την υψηλή απόδοσή τους, έχει ενισχύσει τη έρευνα στους τομείς της αναγνώρισης προτύπων (*pattern recognition*) και της εξόρυξης δεδομένων (*data mining*).
Οι εφαρμογές της βαθιάς μάθησης (Deep Learning) σε αντικείμενα όπως η αναγνώριση αντικειμένων (*object identification*), αναγνώριση φωνής (*voice recognition*) χρησιμοποιώντας τα μοντέλα Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) και autoencoders συνέβαλε σημαντικά στην μελέτη και την ανάπτυξη των νευρωνικών δικτύων.
Δεδομένα με την μορφή κειμένου, εικόνας, βίντεο αναλύονται δίχως κόπο χρησιμοποιώντας βαθιά μάθηση, καθώς αποτελούν δεδομένα σε Ευκλείδεια μορφή.

Εφόσον οι ευκλείδειοι χώροι ορίζονται τοπολογικά από το $\mathbb{R}^n$, για κάποια διάσταση n, δεδομένα σε Ευκλείδεια μορφή ορίζονται ως τα δεδομένα που μπορούν να αναπαρασταθούν σε έναν n-διάστατο γραμμικό χώρο. Οι εικόνες αποτελούν δεδομένα ευκλείδειας μορφής, καθώς χρειαζόμαστε τις συντεταγμένες x,y για να προσδιορίσουμε την θέση του πίξελ στην εικόνα και την τιμή z για την τιμή του χρώματος του πίξελ, θεωρώντας ότι η εικόνα είναι σε grayscale μορφή.
Τι συμβαίνει όμως όταν τα δεδομένα βρίσκονται σε μορφή γράφου, δηλαδή σε μη ευκλείδεια μορφή; Όταν οι σχέσεις μεταξύ των δεδομένων χαρακτηρίζονται από περίπλοκες αλληλεπιδράσεις, τότε χρησιμοποιούνται τα νευρωνικά δίκτυα στη μορφή γράφων. Εκκινούμε την ανάλυση ορίζοντας, αρχικά, την δομή δεδομένων του γράφου.

5.1 Ιστορική Αναδρομή

Τα νευρωνικά δίκτυα που εφαρμόζονται σε γράφους, παρουσιάστηκαν πρώτη φορά από τους Franco Scarselli, Marco Gori [34] ως μια μορφή νευρωνικών δικτύων ανάδρασης. Το μοντέλο αυτό απαιτούσε την επαναλαμβανόμενη εφαρμογή *contraction mapping* ως συναρτήσεις διάδοσης, έως ότου οι αναπαραστάσεις των κόμβων να φτάσουν σε σταθερό σημείο (*stable fixed point*).
Προς βέλτιστη διευκρίνιση των παραπάνω, αναφέρουμε ότι τα *contraction mappings* σε έναν μετρικό χώρο (M, d) είναι μια συνάρτηση, *f,* από το M στον εαυτό του, με την ιδιότητα ότι υπάρχει ένας πραγματικός αριθμός $0 \leq k < 1$ τέτοιος ώστε για κάθε x, y στο M να ισχύει :

$$d(f(x), f(y)) \leq kd(x, y)$$

Το μικρότερο *k* για το οποίο ισχύει η παραπάνω σχέση, λέγεται σταθερά του *Lipschitz*.

Ο περιορισμός των επαναλαμβανόμενων εφαρμογών *contraction maps* αντιμετωπίστηκε από τους Yujia Li, Daniel Tarlow, Marc Brockschmidt [35] οι οποίοι παρουσίασαν μοντέρνες τεχνικές εκπαίδευσης για νευρωνικά δίκτυα ανάδρασης εφαρμοσμένα σε δομές γράφων. Οι David K. Duvenaud, Dougal Maclaurin [36] εισήγαγαν έναν κανόνα διάδοσης, βασισμένο στην συνέλιξη σε γράφους καθώς και μεθόδους για ταξινομήσεις σε επίπεδο γράφων.

Η προσέγγισή τους βασίζεται στην μάθηση πινάκων βάρους και βαθμών κόμβων, οι οποίοι βέβαια δεν μπορούν να σχηματιστούν σε μεγάλους γράφους με υπέρογκες κατανομές κόμβων. Εν αντιθέσει, το μοντέλο που παρουσιάζουμε, χρησιμοποιεί έναν μονάχα πίνακα βαρών ανά επίπεδο ο οποίος περιέχει διάφορους βαθμούς κόμβων μέσω ενός κανονικοποιημένου πίνακα γειτνίασης.
Μια σχετική προσέγγιση για την ταξινόμηση κόμβων με ένα *graph-based neural network* παρουσιάστηκε πρόσφατα από τους *James Atwood* και *Don Towsley* [37].

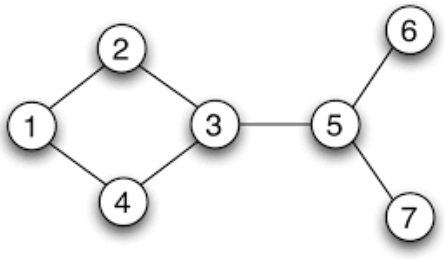
Η χρονική πολυπλοκότητα που σημειώνει το μοντέλο τους είναι της τάξης $O(N^2)$ σημαντικά περιοριστική για πολλές εφαρμογές της πραγματικής ζωής. Σε μια διαφορετική προσέγγιση μοντέλου, οι *Mathias Niepert*, *Mohamed Ahmed*, και *Konstantin Kutzkov* [38] μετατρέπουν τους γράφους τοπικά σε ακολουθίες οι οποίες μπαίνουν ως είσοδο σε ένα μονοδιάστατο νευρωνικό δίκτυο (*1D convolutional neural network*), διαδικασία που απαιτεί ταξινόμηση των κόμβων του γράφου, ως *pre-processing* βήμα.
Η δική μας μέθοδος βασίζεται στις *spectral* συνελικτικά νευρωνικά δίκτυα γράφων, τα οποία εισήγαγαν οι *Joan Bruna* και *Wojciech Zaremba* [39], η οποία στη συνέχεια επεκτάθηκε από τους *Michael Defferrard* και *Xavier Bresson* [40] με την προσθήκη ταχέων τοπικών συνελίξεων. Σε αντίθεση με τις παραπάνω μεθόδους, στο παρόν μοντέλο θεωρούμε την ταξινόμηση κόμβων ανάμεσα σε δίκτυα μεγαλύτερης κλίμακας. Σύμφωνα με αυτό το μοντέλο και με διάφορες παραδοχές που περιγράφονται στην παράγραφο (5.7) το μοντέλο μας επιτυγχάνει βελτιστοποιημένο *scalability* καθώς και υψηλότερη απόδοση στα υψηλής κλίμακας δίκτυα.

Όπως έχει ήδη αναφερθεί πολλές φορές σε αυτήν την εργασία, ένα σημαντικό ποσοστό των λογαριασμών στα κοινωνικά δίκτυα, δεν είναι διαχειριζόμενο από ανθρώπους, αλλά πρόκειται για αυτόματα προγράμματα τα οποία, σύμφωνα με τον αλγόριθμό τους, δημιουργούν περιεχόμενο, αλληλεπιδρούν με τους ανθρώπους και αναρτούν κακόβουλο και αναληθές περιεχόμενο, με σκοπό την παραπλάνηση των χρηστών και τον προσανατολισμό τους σε μια συγκεκριμένη κατεύθυνση.

5.2 Γράφος

Ο απλός μη κατευθυνόμενος γράφος είναι ένα διατεταγμένο ζεύγος G = (V, E), οπου:

- V είναι το σύνολο των κόμβων του γράφου
- $E \subseteq \{\{x, y\} \mid x, y \in V \text{ και } x \neq y\}$ το σύνολο των ακμών, τα οποία είναι μη διατεταγμένα ζεύγη κόμβων

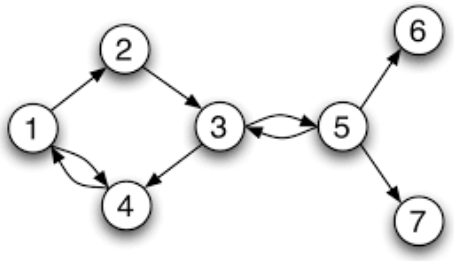


Εικόνα 12 Μη Κατευθυνόμενος Γράφος 7 Ακμών και 7 Κορυφών.

Ο κατευθυνόμενος γράφος είναι ένα διατεταγμένο ζεύγος G = (V, E), οπου:

- V είναι το σύνολο των κόμβων του γράφου

- $E \subseteq \{\{x, y\} \mid x, y \in V^2$ και $x \neq y\}$ το σύνολο των ακμών, τα οποία είναι διατεταγμένα ζεύγη κόμβων.



Εικόνα 13 Κατευθυνόμενος Γράφος 9 Ακμών και 7 Κορυφών.

Ένας γράφος δύναται να χρησιμοποιηθεί για να αναπαραστήσει πληθώρα αντικειμένων, όπως ένα κοινωνικό δίκτυο, το δίκτυο κατοίκων μιας πόλης, χημικά μόρια καθώς επίσης και τις απλές ευκλείδειες δομές δεδομένων, όπως η εικόνα και το κείμενο. Παρακάτω, περιγράφονται εκτενέστερα οι τομείς στους οποίους μπορούν να εισαχθούν τα μοντέλα μορφής γράφων.

5.3 Ανάλυση κοινωνικών δικτύων χρησιμοποιώντας Deep Representation Learning

Η ανάλυση κοινωνικών δικτύων αποτελεί βασικό πρόβλημα στην εξόρυξη δεδομένων. Πρωταρχικό βήμα για την ανάλυση κοινωνικών δικτύων είναι η κωδικοποίηση των δεδομένων του δικτύου σε μία χαμηλής διάστασης αναπαράσταση, η οποία θα αποτελεί τα εμφυτεύματα του δικτύου (*network embeddings*), ούτως ώστε η δομή της τοπολογίας του δικτύου και η πληροφορία των υπόλοιπών του χαρακτηριστικών να διατηρηθεί αποτελεσματικά.
Η διαδικασία μάθησης της αναπαράστασης του δικτύου, διευκολύνει περαιτέρω τη διενέργεια εφαρμογών, όπως η ταξινόμηση των κόμβων του δικτύου σε δύο ή περισσότερες κατηγορίες (*node classification, clustering*), η πρόβλεψη ακμής (*link prediction*), έλεγχο ανωμαλίας (*anomaly detection*). Όπως διαπιστώθηκε, οι εφαρμογές αυτές, που αφορούν βαθιά νευρωνικά δίκτυα, κίνησαν το ενδιαφέρον των ερευνητών τα τελευταία χρόνια. Παρακάτω, θα διεξαχθεί μια σύντομη, και όσο το δυνατόν, περιεκτική αναφορά στις τεχνικές αναπαράστασης των δικτύων, χρησιμοποιώντας μοντέλα νευρωνικών δικτύων.
Αρχικά, θα παρουσιαστούν τα βασικά μοντέλα μάθησης αναπαραστάσεων κόμβων σε ομογενή δίκτυα. Θα ακολουθήσουν, ορισμένες προεκτάσεις των βασικών μοντέλων, περιγράφοντας ορισμένα πιο σύνθετα σενάρια ετερογενών δικτύων, ενώ θα γίνει αναφορά και στην έννοια των εμφυτευμάτων των γράφων (*graph embeddings*).

5.4 Περιγραφή Κοινωνικών Δικτύων

Τα κοινωνικά δίκτυα όπως το Facebook, Twitter, LinkedIn καθιστούν άμεση την επικοινωνία μεταξύ των χρηστών παγκοσμίως. Η ανάλυση των κοινωνικών δικτύων ευνοεί τη συγκέντρωση των ενδιαφερόντων και της γνώμης των χρηστών, τη δημιουργία μονοπατιών βάσει των αλληλεπιδράσεων μεταξύ των χρηστών, ενώ παράλληλα γίνεται έλεγχος για τα διάφορα γεγονότα που λαμβάνουν χώρα στις διαδικτυακές πλατφόρμες. Αν το παραπάνω κοινωνικό δίκτυο

αναπαρασταθεί ως ένας γράφος, τότε, στην περίπτωση αυτή, οι χρήστες θα αποτελούν τους κόμβους και οι μεταξύ τους αλληλεπιδράσεις τις ακμές του γράφου.

5.5 Πρόβλημα feature extraction στα κοινωνικά δίκτυα και διάφορες προσεγγίσεις

Ένα κεντρικό πρόβλημα στην ανάλυση κοινωνικών δικτύων, διατυπώνεται στο ακόλουθο ερώτημα: Πώς θα επιτευχθεί εξαγωγή χρήσιμων χαρακτηριστικών από μη Ευκλείδεια δομημένα δίκτυα, που θα επιτρέψουν την ανάπτυξη μοντέλων πρόβλεψης της μηχανικής μάθησης για συγκεκριμένη ανάλυση;
Αν, λόγου χάριν, είχαμε το πρόβλημα πρότασης νέων φίλων (*new friends recommendation*) σε κάποιον χρήστη ενός κοινωνικού δικτύου, η σημαντικότερη πρόταση είναι πώς θα δημιουργούσαμε τα embeddings όλων των χρηστών σε έναν χώρο χαμηλής διάστασης (*low dimensional space*), ούτως ώστε το πόσο κοντά βρίσκονται οι χρήστες, που αποδίδεται από τον όρο closeness, να μπορεί να μετρηθεί με μετρικές απόστασης.
Παλαιότερα, για την επεξεργασία και τη διαχείριση δομικών πληροφοριών στα δίκτυα, χρησιμοποιούνταν συναρτήσεις πυρήνα, στατιστικά των γράφων όπως βαθμός κόμβων ή συντελεστές ομαδοποίησης και πολλα ακόμη προσεκτικά μηχανικά κατασκευασμένα χαρακτηριστικά. Παρόλα αυτά, οι παραπάνω διαδικασίες αποδείχτηκαν κοστοβόρες χρονικά και οικονομικά, καθιστώντας τες μη αποτελεσματικές για τις εφαρμογές και απαιτήσεις του πραγματικού κόσμου.
Ένας εναλλακτικός τρόπος για την αποφυγή των παραπάνω περιορισμών, προβάλλει η αυτόματη μάθηση χαρακτηριστικών αναπαράστασης, τα οποία συλλέγουν διάφορες πηγές πληροφορίας στα δίκτυα. Στόχος είναι η μάθηση μιας συνάρτησης μετασχηματισμού (*transformation function*) η οποία θα αντιστοιχεί κόμβους, υπογράφους ή ακόμα κι ολόκληρο το δίκτυο ως διανύσματα (*vectors*) σε έναν χώρο χαρακτηριστικών χαμηλής διάστασης (*low-dimensional feature space*). Εκεί, οι χωρικές σχέσεις μεταξύ των διανυσμάτων αντανακλούν τις δομές και τα περιεχόμενα του αρχικού δικτύου. Δεδομένων των παραπάνω διανυσμάτων χαρακτηριστικών (*feature vectors*), τα μοντέλα της μηχανικής μάθησης μπορούν άμεσα να χρησιμοποιηθούν σε εφαρμογές ταξινόμησης, ομαδοποίησης και ανίχνευσης, σε καίρια δηλαδή ζητήματα των σημερινών εποχών.
Παράλληλα με την ουσιαστική βελτίωση της απόδοσης, χάρη στη βαθιά μάθηση, σε εφαρμογές όπως ανίχνευση εικόνων, εξόρυξη κειμένου, και επεξεργασία φυσικής γλώσσας, η ανάπτυξη μεθόδων δικτυακών αναπαραστάσεων χρησιμοποιώντας μοντέλα νευρωνικών δικτύων, έχει λάβει αμέριστη προσοχή τα τελευταία χρόνια [25].

Συμβολισμός όρων και ορισμοί

Στην παράγραφο αυτή θα οριστούν ορισμένες βασικές ορολογίες, συχνά αναφερόμενες παρακάτω, ενώ θα διατυπωθεί με ακρίβεια ο επίσημος ορισμός του προβλήματος μάθησης αναπαράστασης του δικτύου [25].

- Με έντονο κεφαλαίο γράμμα θα συμβολίζονται οι πίνακες (**A**).
- Με έντονο πεζό γράμμα θα συμβολίζονται τα διανύσματα (**α**).
- Με πεζό γράμμα θα συμβολίζονται οι πραγματικοί αριθμοί (α).
- Η καταχώρηση (i,j) στον πίνακα **A,** η i-οστή γραμμή και η j-οστή στήλη του δηλώνονται ως $\mathbf{A}_{ij}$, $\mathbf{A}_{i*}$, $\mathbf{A}_{*j}$ αντίστοιχα.

Ορισμός 1 (Δίκτυο) Έστω G = {V, E, **X, Y**} ένα δίκτυο, όπου ο i-οστός κόμβος δηλώνεται ως $v_i \in V$ και $e_{i,j} \in E$ δηλώνει την ακμή ανάμεσα στον v_i και v_j. **X** είναι τα χαρακτηριστικά των κόμβων και **Y** οι ετικέτες τους, αν διατίθενται. Επιπροσθέτως, δηλώνεται $\mathbf{A} \in \Re^{NxN}$ ως ο σχετικός πίνακας γειτνίασης του G. Το $\mathbf{A}_{ij}$ ισούται με το βάρος του $e_{i,j}$, εάν $\mathbf{A}_{ij} > 0$, το οποίο σημαίνει πως οι δύο κόμβοι ενώνονται μεταξύ τους, αλλιώς $\mathbf{A}_{ij} = 0$. Για τους μη κατευθυνόμενους γράφους, ισχύει $\mathbf{A}_{ij} = \mathbf{A}_{ji}$.

Πολύ συχνά, οι κόμβοι και οι ακμές του γράφου G σχετίζονται με διάφορους τύπους πληροφορίας. Έστω, $\tau_v : V \rightarrow T^v$ μια συνάρτηση αντιστοίχισης τύπων κόμβων και έστω $\tau_e : E \rightarrow T^e$ μια συνάρτηση αντιστοίχισης τύπων ακμών, οπου τα T^v, T^e εκφράζουν τα σύνολα των τύπων των κόμβων και των ακμών αντιστοίχως. Κάθε κόμβος, $v_i \in V$, έχει ένα συγκεκριμένο τύπο, $\tau_v(v_i) \in T^v$. Αντιστοίχως, για κάθε ακμή $e_{i,j}$ $\tau_e(e_{ij}) \in T^e$.

Ορισμός 2 (Ομογενές Δίκτυο) Ομογενές δίκτυο είναι το δίκτυο στο οποίο $|T^e| = |T^v| = 1$. Δηλαδή, όλοι οι κόμβοι και όλες οι ακμές ανήκουν σε έναν τύπο.

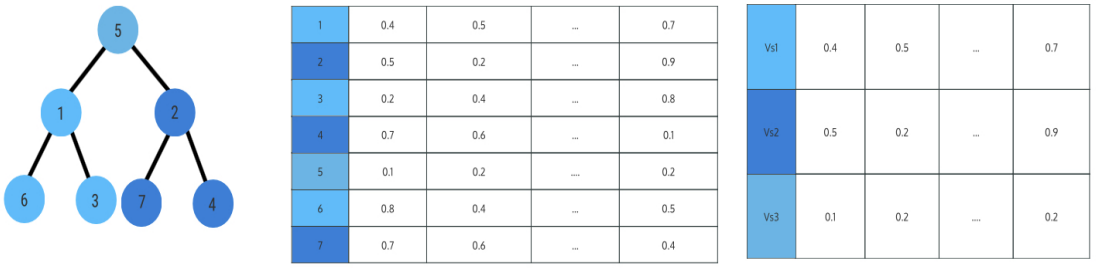
Ορισμός 3 (Ετερογενές Δίκτυο) Ετερογενές δίκτυο είναι το δίκτυο στο οποίο $|T^e| + |T^v| > 2$. Δηλαδή, υπάρχουν τουλάχιστον δύο διαφορετικοί τύποι ακμών ή κόμβων στο δίκτυο.

Δεδομένου ενός δικτύου G, η διαδικασία μάθησης αναπαράστασης δικτύου αφορά την εκπαίδευση της συνάρτησης αντιστοίχισης *f*, η οποία αντιστοιχεί κόμβους, υπογράφους σε έναν λανθάνοντα χώρο (*latent space*). Έστω D η διάσταση του χώρου αυτού. Συνήθως $D << |V|$.

Ορισμός 4 (Μάθηση Αναπαράστασης Κόμβων) Έστω, $z \in R^D$ το διάνυσμα στο latent space που αντιστοιχεί στον κόμβο v. Η μάθηση αναπαράστασης κόμβου, στοχεύει στη δημιουργία συνάρτησης αντιστοίχισης, ούτως ώστε $z = f(v)$. Αναμένεται, κόμβοι με παρόμοιους ρόλους και χαρακτηριστικά, οι οποίοι ορίζονται σύμφωνα με συγκεκριμένους τομείς εφαρμογής, να αντιστοιχίζονται κοντά ο ένας στον άλλον στο latent space.

Ορισμός 5 (Μάθηση Αναπαράστασης Υπογράφου) Έστω g υπογράφος του G. Οι κόμβοι και οι ακμές του g συμβολίζονται με V_s και E_s αντιστοίχως, ενώ $V_s \subset V$ και $E_s \subset E$. Η μάθηση αναπαράστασης υπογράφου, στοχεύει στη δημιουργία συνάρτησης αντιστοίχισης, ούτως ώστε $z = f(g)$, όπου στην περίπτωση αυτή το $z \in R^D$ εκφράζει το latent vector του υπογράφου g.

Στην παρακάτω εικόνα αποδίδονται τα embeddings του δικτύου. Συγκεκριμένα, απεικονίζονται τρεις υπογράφοι στο δίκτυο αυτό, οι οποίοι διακρίνονται χάρη στο διαφορετικό τους χρώμα:
$V_{s1} = \{v_1, v_2, v_3\}$, $V_{s2} = \{v_4\}$, $V_{s3} = \{v_5, v_6, v_7\}$. Δεδομένου ενός δικτύου ως είσοδο, το παρακάτω παράδειγμα παράγει μία αναπαράσταση για κάθε κόμβο, όπως επίσης και μια αναπαράσταση για κάθε υπογράφο.



Εικόνα 14 Παραπάνω αποδίδεται ένα παράδειγμα μάθησης αναπαράστασης κόμβου και μάθησης αναπαράστασης υπογράφου. Οι υπογράφοι αποτυπώνονται με διαφορετικό χρώμα. Το εμφύτευμα κόμβου (*node embedding*) εχει ως στόχο να παράξει μια αναπαράσταση για κάθε κόμβο του γράφου, ενώ το εμφύτευμα υπογράφου (*subgraph embedding*) δημιουργεί μια αναπαράσταση για ολόκληρο τον υπό-γράφο.

Η παρακάτω παράγραφος παραθέτει συμπεράσματα και διαπιστώσεις που παρουσιάζονται στα άρθρα [62], [63]

5.6 Μοντέλα Γράφων Βασισμένα σε Νευρωνικά Δίκτυα

Είναι φανερό πως τα νευρωνικά δίκτυα έχουν ισχυρές δεξιότητες στο να συλλέγουν σύνθετα μοτίβα δεδομένων, σημειώνοντας εξαιρετικές επιδόσεις στον τομέα της όρασης υπολογιστών, στην αναγνώριση ήχου και στην επεξεργασία φυσικής γλώσσας [43]. Τα τελευταία χρόνια, έχουν γίνει προσπάθειες επέκτασης των νευρωνικών δικτύων στη μάθηση αναπαραστάσεων δεδομένων σε μορφή δικτύου. Ανάλογα με την βάση των μοντέλων αυτών κατηγοριοποιούνται σε τρεις κατηγορίες; look-up table based models, autoencoder based models, GCN (*graph convolutional networks*) based models. Το μοντέλο που θα αναλυθεί σχοινοτενώς, είναι το τελευταίο, ο γράφος συνελικτικών δικτύων, εφόσον είναι το πιο αποδοτικό από τα άλλα δύο στην διαχείριση και επεξεργασία κοινωνικών δικτύων, τα οποία αποτελούνται ανα δείγμα από μη σταθερό αριθμό κόμβων και ακμών. Αυτό το μοντέλο επιλέχθηκε, εν τέλει, για την υλοποίηση της διπλωματικής εργασίας. Χάρη στην ικανότητα των GNNs να αποδίδουν και να εκφράζουν επακριβώς σύνθετες δομές δεδομένων, οι γράφοι σημειώνουν σημαντική απήχηση στον τομέα της Μηχανικής Μάθησης. Κάθε κόμβος έχει το δικό του εμφύτευμα (embedding), το οποίο εκφράζει τη θέση του κόμβου στο χώρο των δεδομένων. Τα νευρωνικά δίκτυα στη μορφή γράφων είναι τοπολογίες νευρωνικών δικτύων που εφαρμόζονται σε γράφους. Στόχος της αρχιτεκτονικής των GNN είναι για τον κάθε κόμβο τους να μάθουν ένα εμφύτευμα το οποίο περιέχει πληροφορία για τους γείτονές του. Τα GNN αποτελούν υποσύνολο των τεχνικών βαθιάς μάθησης και απευθύνονται σε δεδομένα μορφής γράφων. Εφαρμόζονται, λοιπόν, στους γράφους και χρησιμοποιούνται σε προβλέψεις που αφορούν κόμβους, ακμές και ολόκληρους γράφους.
Ο λόγος που δεν χρησιμοποιούνται CNN σε δεδομένα μορφής γράφου, εντοπίζεται στο ότι η τοπολογία του γράφου είναι αυθαίρετη και πολύπλοκη, καθώς δεν υπάρχει χωρική τοπικότητα. Επιπροσθέτως, δεν υπάρχει συγκεκριμένη και μοναδική θέση για κάθε κόμβο, γεγονός που περιπλοκεύει τη χρήση τους στην περίπτωση αυτή. Η πολυπλοκότητα στην ανάλυση της δομής του γράφου είναι σημαντική. Οι εικόνες, κατά την επεξεργασία τους μπορούν να οριστούν να έχουν συγκεκριμένα μέγεθος (πχ 243x243x3) και κάθε φορά το νευρωνικό δίκτυο να αναμένει αυτή την

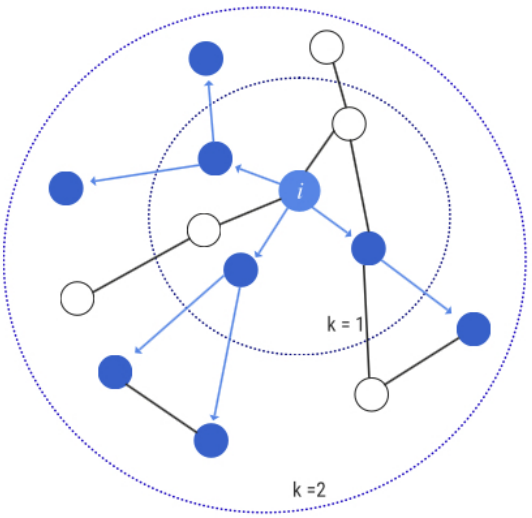
είσοδο για κάθε εικόνα. Ωστόσο, κάθε γράφος μπορεί να έχει διαρκώς μεταβλητό αριθμό κόμβων καθώς επίσης, οι κόμβοι μεταξύ τους διαφέρουν στον αριθμό των γειτόνων τους. Επιπλέον , η υπόθεση που γίνεται σε πολλούς αλγορίθμους της μηχανικής μάθησης, ότι τα δεδομένα είναι ανεξάρτητα μεταξύ τους, δεν ισχύει στην περίπτωση των γράφων καθώς κάθε κόμβος στον γράφο ενώνεται με έναν γείτονα μέσω ακμών ενός ή πολλών ειδών.

Αναλύοντας τον τρόπο εφαρμογής των CNNs μπορούμε να αναλογιστούμε γιατί αυτά αποτυγχάνουν κατά της εφαρμογή τους σε γράφους. Τα CNNs οπτικοποιούν αντικείμενα σε φωτογραφίες, ταξινομούν εικόνες χάρη στις συνελίξεις που εφαρμόζει το κρυφό τους επίπεδο στις εικόνες. Συγκεκριμένα, μετακινείται ο πυρήνας της συνέλιξης, κεντράροντας κάθε φορά σε ένα πίξελ της δισδιάστατης εικόνας εφαρμόζοντας κάποια συνάρτηση στα πίξελ της εικόνας αυτής. Έπειτα, η διαδικασία αυτή επαναλαμβάνεται για πολλά επίπεδα. Ουσιαστικά, με τη συνέλιξη λαμβάνεται ένα μικρό τμήμα της εικόνας, εφαρμόζεται μια συνάρτηση σε αυτό και παράγεται ένα νέο τμήμα, ένα νέο πίξελ. Το κεντρικό πίξελ στο οποίο κέντραρε ο πυρήνας της συνέλιξης, συνδυαστικά με την δική του πληροφορία συλλέγει πληροφορία και από τους γείτονές του παράγοντας μια νέα τιμή. Λόγω της σύνθετης τοπολογία των γράφων και του αυθαίρετου μεγέθους τους δεν υφίσταται χωρική τοπικότητα. Επίσης, επειδή δεν υπάρχει μοναδική διάταξη για τους κόμβους εάν σε αυτούς θέσουμε τις ετικέτες Α, Β, Γ, Δ και έπειτα Β, Α, Γ, Δ τότε τα δεδομένα εισόδου του πίνακα στο δίκτυο θα μεταβληθούν. Οι γράφοι είναι αμετάβλητοι στην διάταξη των κόμβων, γι'αυτό επιθυμούμε να λάβουμε το ίδιο αποτέλεσμα, ανεξάρτητα από την διάταξη των κόμβων.

Αναλύοντας την σπουδαιότητα και την αναγκαιότητα των GNNs στα προβλήματα της μηχανικής μάθησης, προσθέτουμε επιπλέον και την πιο μαθηματική τους υπόσταση.

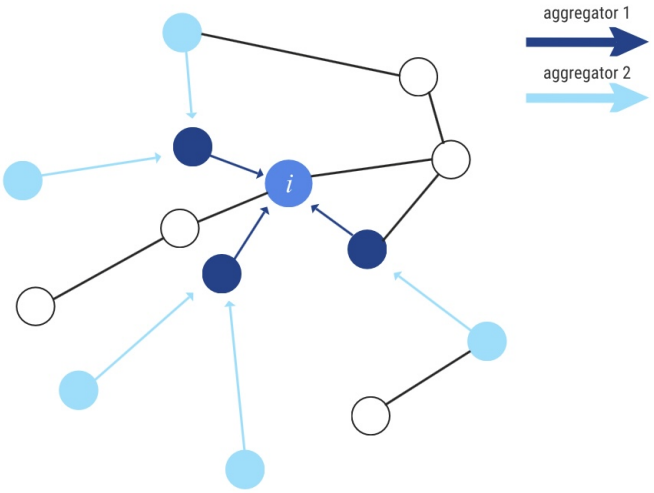
Στην θεωρία των γράφων, εισαγάγεται ο όρος "εμφυτεύματα κόμβων" (*node embeddings*). Περιγράφει την αντιστοίχιση των κόμβων σε ένα d-διάστατο χώρο εμφυτευμάτων, χαμηλότερης διάστασης από τη διάσταση του ίδιου του γράφου, ούτως ώστε τα εμφυτεύματα των παρόμοιων κόμβων να αντιστοιχίζονται κοντά στο χώρο των εμφυτευμάτων. Στόχος είναι να αντιστοιχίσουμε τους κόμβους με τρόπο τέτοιο ώστε η ομοιότητα στον χώρο των εμφυτευμάτων να προσεγγίζει την ομοιότητα μεταξύ των κόμβων του δικτύου. Έστω u,v δύο κόμβοι του γράφου και x_u, x_v τα δύο διανύσματα χαρακτηριστικών που αντιστοιχούν στους κόμβους. Έστω μια συνάρτηση f που αντιστοιχεί τα διανύσματα χαρακτηριστικών x_u, x_v σε διανύσματα z_u, z_v στον χώρο των embeddings. Στόχος είναι η εύρεση της συνάρτησης αυτής. Η συνάρτηση f λοιπόν πρέπει να εφαρμόζεται σε έναν κόμβο και στους γειτονικούς του εξασφαλίζοντας την τοπική πληροφορία, να συλλέγει την επιθυμητή πληροφορία (*aggregating*) και στοιβάζει πολλά επίπεδα.

Η τοπική πληροφορία εξάγεται χρησιμοποιώντας έναν υπολογιστικό γράφο. Στην εικόνα αποδίδεται πώς ο κόμβος i ενώνεται με τους γείτονές του και τους γείτονες των γειτόνων του.



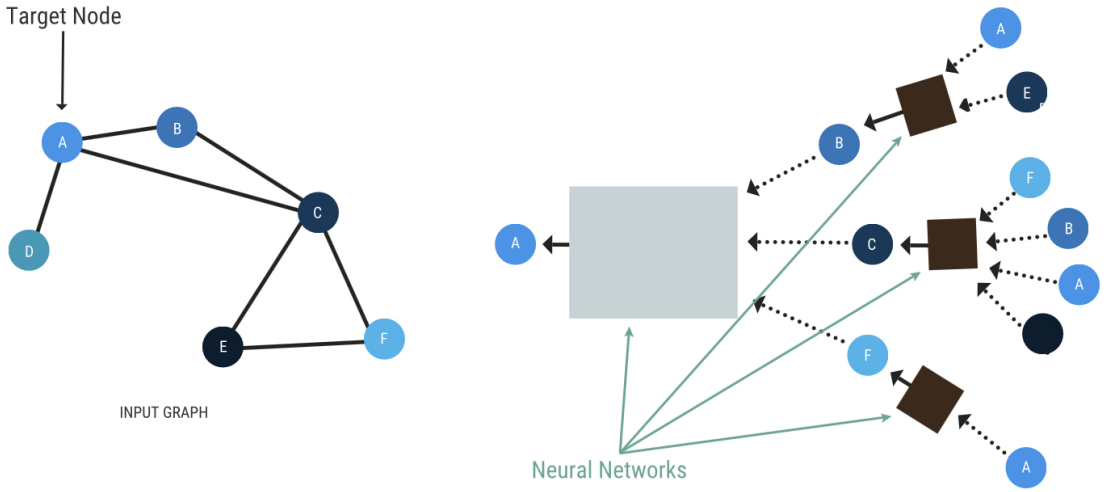
Εικόνα 15 Καθορισμός υπολογιστικού γράφου του κόμβου i

Έχοντας αναπαραστήσει την δομή αυτή, ο κόμβος i συλλέγει την πληροφορία τόσο των δικών του χαρακτηριστικών όσο και των γειτόνων του.

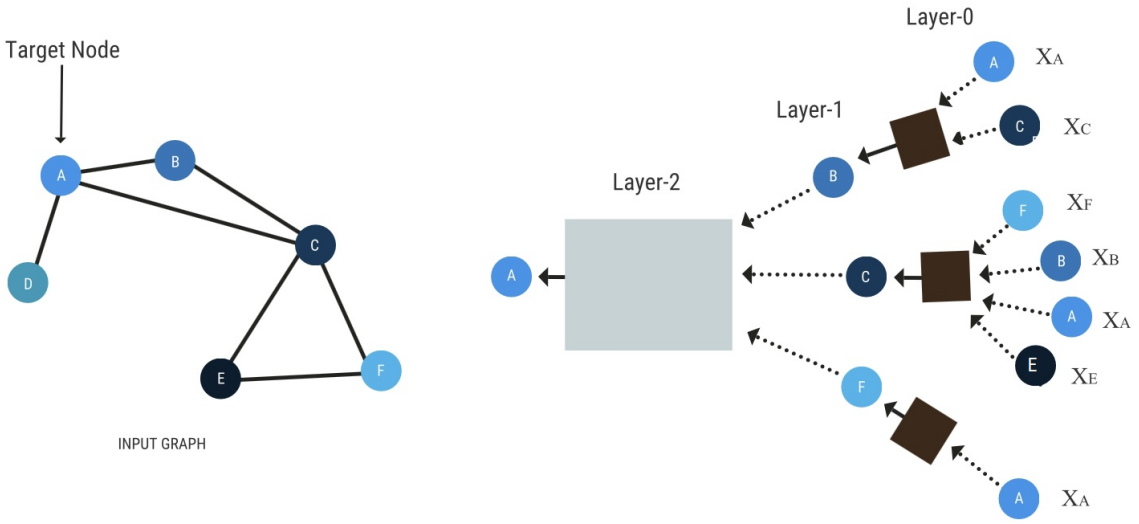


Εικόνα 16 Διάδοση και μεταφορά της πληροφορίας.

Μόλις εξαχθεί η τοπική πληροφορία, αρχίζει η διαδικασία του aggregating χρησιμοποιώντας νευρωνικά δίκτυα.
Τα νευρωνικά δίκτυα αναπαριστώνται με γκρι πλαίσια και βασική προϋπόθεση είναι τα aggregations να μην εξαρτώνται από την σειρά των κόμβων. Για το λόγο αυτό, για το aggregation επιλέγονται συναρτήσεις αθροίσματος, μέσου όρου, μεγίστου.



Εικόνα 17 Ο κανόνας *forward propagation* καθορίζει πώς η πληροφορία εισόδου θα μεταβιβαστεί στην έξοδο του νευρωνικού δικτύου.
Κάθε κόμβος έχει ένα διάνυσμα χαρακτηριστικών. Η είσοδος είναι τα διανύσματα χαρακτηριστικών και το νευρωνικό δίκτυο (γκρι πλαίσιο) λαμβάνει τα χαρακτηριστικά των γειτονικών κόμβων, εφαρμόζει aggregation και τα περνάει στο επόμενο επίπεδο.



Εικόνα 18 Παρατηρούμε ότι ο κόμβος C δέχεται ως είσοδο τα χαρακτηριστικά του κόμβου C, ωστόσο η αναπαράστασή του στο επίπεδο 1 θα είναι μια κρυφή λανθάνουσα αναπαράσταση του κόμβου, όπως ακριβώς και στο επίπεδο 2.

Επομένως, προκειμένου να εφαρμοστεί forward propagation χρειάζονται 3 βήματα.

1. Αρχικοποίηση
   $h_v^0 = X_v$

2. Ορισμός κρυφού επιπέδου για κάθε επίπεδο του δικτύου:

$h_v^k = \sigma(W_k \sum \frac{h_v^{k-1}}{N(v)} + B_k h_v^{k-1})$, όπου k = 1, … , k-1.

Παρατηρούμε πως ο όρος $W_k \sum \frac{h_v^{k-1}}{N(v)}$ εφαρμόζει μέσο όρο σε όλους τους γείτονες του κόμβου v,

Ο όρος $B_k h_v^{k-1}$ περιέχει το εμφύτευμα του προηγούμενου επιπέδου του κόμβου v πολλαπλασιασμένο με τον bias όρο B_k.

Τέλος, η σ είναι η μη γραμμική συνάρτηση ενεργοποίησης που εφαρμόζεται στους δύο όρους.

3. Τελικό επίπεδο $z_v = h_v^k$

   Το εμφύτευμα προκύπτει έπειτα από K aggregations μεταξύ του κόμβου v και των γειτόνων του.

Έπειτα, εκπαιδεύουμε το μοντέλο ορίζοντας μια συνάρτηση κόστους στα embeddings. Το μοντέλο που κρίθηκε καταλληλοτερο μεταξύ των GNNs ηταν το συνελικτικό μοντέλο σε μορφή γράφων (GCN) και συγκεκριμένα, αυτό που παρουσιάστηκε από τους Thomas N. Kipf και Max Welling [25]. Το εν λόγω μοντέλο αναλύεται στην ενότητα Graph Convolutional Networks (GCN). Συνοπτικά, αναφέρεται πως κάθε GCN αποτελείται από τις εξής διαδικασίες. Συνελίξεις στα δεδομένα μορφής γράφου, έπειτα εφαρμόζεται ένα γραμμικό επίπεδο και τέλος μια μη γραμμική συνάρτηση ενεργοποίησης. Τα νευρωνικά δίκτυα σε μορφή γράφων χωρίζονται σε τρεις κατηγορίες. Recurrent Graph Neural Network, Spatial Convolutional Network και Spectral Convolutional Network. Το GCN που επιλέχθηκε για τη συγκεκριμένη εργασία ανήκει στην τρίτη κατηγορία, Spectral Convolutional Network, η οποία περιγράφεται αναλυτικά παρακάτω.

Τα προβλήματα που ένα GNN μπορεί να αντιμετωπίσει χωρίζονται σε τρεις κατηγορίες:

1. Node Classification
   Αφορά την πρόβλεψη των embeddings για κάθε κόμβο του δικτύου.

2. Link Prediction
   Κύριος στόχος είναι ο καθορισμός τη σχέσης μεταξύ δύο κόμβων του γράφου και η απόφανση για το εάν οι δύο αυτοί κόμβοι ενώνονται.

3. Graph Classification
   Αφορά την ταξινόμηση του γράφου σε μία κατηγορία επιλέγοντας ανάμεσα σε δύο ή περισσότερες.

Σε αυτή τη διπλωματική εργασία επιλύεται το πρόβλημα ταξινόμησης κόμβων. Για κάθε ένα κόμβο-χρήστη του γράφου, καθώς και για τον κόμβο κεφαλή που αποτελεί την προς εξέταση είδηση υπολογίζονται τα διανύσματα χαρακτηριστικών (*node embeddings*). Κατά την αξιολόγηση του μοντέλου κρίνεται εάν η εκάστοτε είδηση πρόκειται για αληθινή ή ψευδή.

5.7 Ιστορική Αναδρομή στα GNNs και αναφορά στα μοντέλα που θα εξετασθούν

Για την εκπόνηση της παρούσας διπλωματικής εργασίας, πραγματοποιήθηκε η υλοποίηση τριών διαφορετικών μοντέλων νευρωνικών δικτύων σε μορφή γράφων. Συγκεκριμένα, επιλέχθηκαν τα μοντέλα GCN [25], GraphSAGE [49] και GAT [50]. Προτού μεταβούμε στην αναλυτική περιγραφή των μοντέλων αυτών θα κάνουμε μια ιστορική αναδρομή, όπου θα εξεταστεί ο τρόπος που το ένα μοντέλο διαδέχθηκε το άλλο καθώς και η ανάγκη που ώθησε την ανακάλυψη νέων τεχνικών ανάλυσης δεδομένων.

Τα παραδοσιακά συνελικτικά νευρωνικά δίκτυα (CNNs) εφαρμόστηκαν με τεράστια επιτυχία για την επίλυση προβλημάτων όπως ταξινόμηση εικόνων (*image classification*), σημασιολογική ανάλυση (*semantic analysis*), μετάφραση μηχανών, όπου οι αναπαραστάσεις των δεδομένων έμοιαζαν με δομή πλέγματος (*grid-like data*). Οι αρχιτεκτονικές αυτές χρησιμοποιούσαν αποδοτικά τα τοπικά τους φίλτρα, και παράλληλα με τις εκπαιδευόμενες παραμέτρους τους, εφαρμόζοντας σε όλα τα δεδομένα εισόδου.

Παρόλα αυτά, ορισμένα ενδιαφέροντα προβλήματα περιείχαν δεδομένα δεδομένα που δεν μπορούσαν να αποδοθούν σε μορφή πλέγματος, όπως για παράδειγμα τα κοινωνικά δίκτυα, τα τηλεπικοινωνιακά δίκτυα, τα βιολογικά δίκτυα, τα χημικά μόρια. Τα παραπάνω δεδομένα, περιγράφονται αποτελεσματικότερα με τη μορφή των γράφων.

Ανά τα χρόνια, έχουν παρατηρηθεί πολλές προσπάθειες επέκτασης των νευρωνικών δικτύων προκειμένου να μπορούν να διαχειριστούν αυθαίρετα δομημένους γράφους. Αρχικά, χρησιμοποιήθηκαν αναδρομικά νευρωνικά δίκτυα (Recursive Neural Networks), προκειμένου να επεξεργαστούν δεδομένα σε μορφή γράφων, αποδίδοντάς τα ως κατευθυνόμενους μη κυκλικούς γράφους [19].

Τα νευρωνικά δίκτυα σε μορφή γράφων (GNNs) παρουσιάστηκαν από τους Marco Gori και Franco Scarselli ως μια γενίκευση των RNNs τα οποία μπορούν άμεσα να διαχειριστούν μια πιο γενική κλάση γράφων, όπως για παράδειγμα τους κυκλικούς κατευθυνόμενους ή τους μη κατευθυνόμενους γράφους [20].

Τα GNNs αποτελούνται από μια επαναληπτική διαδικασία, η οποία διαδίδει τις καταστάσεις των κόμβων έως ότου αυτές βρεθούν σε ισορροπία. Το νευρωνικό, αυτό, δίκτυο παράγει ένα αποτέλεσμα για κάθε κόμβο βασιζόμενο στην κατάστασή του. Αυτή η ιδέα εξελίχθηκε από τους Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel το 2016 [21].

Έπειτα, προέκυψε η επιθυμία επέκτασης και γενίκευσης των συνελίξεων στον τομέα των γράφων. Οι υλοποιήσεις αυτές χωρίζονται σε *spectral* και *non-spectral* προσεγγίσεις [42].

Spectral Προσεγγίσεις

Στην πρώτη περίπτωση, οι *spectral* προσεγγίσεις αφορούν *spectral* αναπαραστάσεις των γράφων και εφαρμόζονται επιτυχώς σε προβλήματα ταξινόμησης κόμβων. Στην υλοποίηση των Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun, το 2014, η πράξη της συνέλιξης ορίζεται στο πεδίο *Fourier* υπολογίζοντας τον πίνακα ιδιοτιμών και ιδιοδιανυσμάτων της Λαπλασιανής του γράφου [22].

Οι υπολογισμοί αυτοί ήταν ιδιαίτερα σύνθετοι δίχως *spatially localized* φίλτρα. Προς αντιμετώπιση αυτών των ζητημάτων, οι Mikael Henaff, Joan Bruna, and Yann LeCun, το 2015, παρουσίασαν μια παραμετροποίηση των spectral φίλτρων χρησιμοποιώντας συντελεστές ομαλοποίησης (*smoothing coefficients*) προκειμένου να τα μετατρέψουν σε *spatially localized* φίλτρα [23].

Ένα έτος αργότερα, το 2016, οι Michael Defferrard, Xavier Bresson, and Pierre Vandergheynst προσέγγισαν τα τα *spatially localized* φίλτρα χρησιμοποιώντας το μέσο όρο μιας *Chebyshev*

επέκτασης του Λαπλασιανού γράφου. Με τον τρόπο αυτό, δεν υπήρχε πλέον ανάγκη υπολογισμού των ιδιοδιανυσμάτων της Λαπλασιανής και παραγωγής *spatially localized* φίλτρων [24].
Τέλος, οι Thomas N Kipf and Max Welling, το 2017 [25], απλοποίησαν την προηγούμενη μέθοδο περιορίζοντας τα φίλτρα ούτως ώστε να εφαρμόζονται εντός της γειτονιάς ενός κόμβου (1- *step neighborhood*) μακριά από τον εξεταζόμενο κόμβο. Το μοντέλο αυτό επιλέχθηκε προς ανάλυση και υλοποίηση στην συγκεκριμένη εργασία. Παρουσιάζεται στην παράγραφο " 5.8.2 Spectral Graph Convolutions".
Παρόλα αυτά, σε όλες τις προαναφερόμενες *spectral* προσεγγίσεις, τα φίλτρα που προκύπτουν εξαρτώνται από τον πίνακα βάσης της Λαπλασιανής, ο οποίος, με τη σειρά του, εξαρτάται από την δομή του γράφου. Για το λόγο αυτό, ένα μοντέλο που έχει εκπαιδευτεί σε ένα συγκεκριμένο γράφο, δεν μπορεί απευθείας να εφαρμοστεί σε έναν γράφο διαφορετικής δομής.

Non-Spectral Προσεγγίσεις

Στην δεύτερη περίπτωση, αναλύονται οι *non-spectral προσεγγίσεις*. Οι David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre [26] ορίζουν συνελίξεις απευθείας στον γράφο, οι οποίες εφαρμόζονται σε *groups* από *spatially close* γείτονες.
Μια από τις προκλήσεις της μεθόδου αυτής είναι ο ορισμός ενός τελεστή ο οποίος θα αποδίδει για γειτονιές μεταβλητού μεγέθους, διατηρώντας την ιδιότητα διαμοιρασμού του βάρους (*weight sharing property*) των CNNs. Σε ορισμένες περιπτώσεις αυτό απαιτεί τη μάθηση ενός συγκεκριμένου πίνακα βάρους για κάθε κόμβο, ανάλογα με τον βαθμό του, χρησιμοποιώντας έναν πίνακα μετάβασης για να ορίσουμε την γειτονιά κάθε κόμβου, μαθαίνοντας παράλληλα τα βάρη για κάθε κανάλι εισόδου και βαθμό γειτονιάς ή εξάγοντας και κανονικοποιώντας γειτονιές που έχουν ένα συγκεκριμένο αριθμό κόμβων [27].
Οι Federico Monti, Davide Boscaini παρουσίασαν ένα μοντέλο μίξης CNNs MoNet (*mixture model* CNNs) [28], το 2016, μια τοπική προσέγγιση (*spatial approach*) που παρέχει μία ενοποιημένη γενίκευση της αρχιτεκτονικής των CNNs στους γράφους.
Το 2017 οι William L Hamilton, Rex Y παρουσίασαν το μοντέλο GraphSage [29], το οποίο υλοποιείται και αναλύεται εκτενέστερα στην παρούσα εργασία. Πρόκειται για ένα μοντέλο που υπολογίζει αναπαραστάσεις κόμβων με έναν επαγωγικό τρόπο. Η τεχνική αυτή βασίζεται συλλέγοντας για κάθε κόμβο μια γειτονιά συγκεκριμένου μεγέθους. Έπειτα εφαρμόζεται στην γειτονιά αυτή συγκεκριμένη συνάρτηση *aggregator* (συνάρτηση μέσου όρου σε όλα τα διανύσματα χαρακτηριστικών των γειτονικών κόμβων που έχουν συλλεχθεί ή ακόμα είναι εφικτή η εφαρμογή ενός αναδρομικού νευρωνικού δικτύου στα χαρακτηριστικά αυτά). Οι GraphSage ξεπερνούν σε επιδόσεις όλες τις προαναφερόμενες μεθόδους. Η μέθοδος αυτή αναλύεται στην παράγραφο "9. Graph SAGE".

Οι μηχανισμοί προσοχής, που συνθέτουν την GAT αρχιτεκτονική, χρησιμοποιούνται κατά κύριο λόγο σε πολλά προβλήματα επεξεργασίας δεδομένων με την μορφή ακολουθίας. Σημαντικό προνόμιο των μηχανισμών αυτών είναι ότι εφαρμόζονται σε δεδομένα μεταβλητού μεγέθους, εστιάζοντας στα πιο σημαντικά τμήματα των δεδομένων, προκειμένου να λαμβάνουν αποφάσεις. Όταν ο μηχανισμός *attention* χρησιμοποιείται για τον υπολογισμό της αναπαράστασης μιας μόνο ακολουθίας, συνήθως αναφέρεται ως *self-attention* ή *intra-attention*. Συνδυαστικά με τα RNNs και τα CNNs, ο *self-attention* μηχανισμός αποδίδει εξαίρετα σε *machine reading* και *learning sentence representations*, υπερνικώντας σε επιδόσεις όλα τα μοντέλα των απλών νευρωνικών δικτύων που είχαν παρουσιαστεί έως τότε [30].
Ορμώμενοι από την εξαιρετική απόδοση του μηχανισμού *attention* στα απλά νευρωνικά δίκτυα, παρουσιάζουμε τον μηχανισμό αυτό και στα νευρωνικά δίκτυα με τη δομή γράφων. Κεντρική ιδέα

αποτελεί ο υπολογισμός των κρυφών αναπαραστάσεων κάθε κόμβου του γράφου παρακολουθώντας τους γείτονές του, σύμφωνα με τις ιδιότητες του *self-attention* μηχανισμού.

1. Η λειτουργία του μηχανισμού είναι αποδοτική, εφόσον παραλληλοποιείται σε ζευγάρια γειτόνων.

2. Μπορεί να εφαρμοστεί σε κόμβους γράφων διαφορετικών βαθμών, θέτοντας αυθαίρετα βάρη στους γείτονες.

3. Το μοντέλο είναι άμεσα εφαρμόσιμο σε προβλήματα επαγωγικής μάθησης, συμπεριλαμβάνοντας περιπτώσεις όπου το μοντέλο πρέπει να γενικεύσει σε τελείως ανεξερεύνητους γράφους.

Περισσότερες λεπτομέρειες για την GAT αρχιτεκτονική, το επίπεδο που εφαρμόζει τον μηχανισμό *attention* και παράγει τα νέα χαρακτηριστικά των κόμβων, παρουσιάζονται στην συνέχεια, όπου το μοντέλο *attention* αναλύεται εκτενώς.

5.8 Graph Convolutional Networks (GCN)

Τα ημί-επιβλεπόμενης (semi-supervised) μάθησης συνελικτικά δίκτυα γράφων (GCN), ορίζουν ένα συνελικτικό τελεστή στο δίκτυο, ο οποίος επαναληπτικά συγκεντρώνει τα εμφυτεύματα των γειτόνων ενός κόμβου και τα συγκεντρωμένα αυτά εμφυτεύματα, μαζί με το εμφύτευμα του ίδιου το κόμβου που προέκυψε σε προηγούμενη επανάληψη, παράγουν την νέα αναπαράσταση του κόμβου.
Θεωρούμε το πρόβλημα ταξινόμησης κόμβων σε έναν γράφο [41]. Οι κόμβοι θα αποτελούν τους χρήστες του δικτύου, ενώ ο γράφος το διαδικτυακό κοινωνικό δίκτυο (Twitter). Εμείς γνωρίζουμε για ένα σύνολο των χρηστών εάν διαδίδουν ή όχι ψευδείς ειδήσεις. Το πρόβλημα αυτό μπορεί να αποδοθεί ως σε γράφο βασιζόμενο ημι-επιβλεπόμενης μάθησης, όπου η πληροφορία των κατηγοριών/ετικετών των χρηστών είναι ομαλοποιημένη μέσω κάποιου είδους κανονικοποίησης. Χρησιμοποιώντας Λαπλασιανή κανονικοποίηση στην συνάρτηση σφάλματος του γράφου εχουμε:

$$L = L_o + \lambda L_{reg}, \text{ όπου } L_{reg} = \sum_{i,j} A_{ij} || f(X_i) - f(X_j) ||^2 = f(X)^T \Delta f(X).$$

Όπου, ο όρος L_o εκφράζει την επιβλεπόμενη απώλεια, δηλαδή την απώλεια για τους κόμβους όπου γνωρίζουμε τις ετικέτες, η $f(\cdot)$ εκφράζει παραγωγίσιμη συνάρτηση νευρωνικού δικτύου, ο λ είναι ένας συντελεστής και X ο πίνακας των διανυσμάτων χαρακτηριστικών X_i των κόμβων του γράφου.
Το Δ = D - A ορίζει την μη κανονικοποιημένη Λαπλασιανή ενός μη κατευθυνόμενου γράφου G = (V, E) με N κόμβους $v_i \in V$, ακμές $(v_i, v_j) \in E$, πίνακα γειτνίασης $A \in \mathbb{R}^{NxN}$ με δυαδικές τιμές ή με βάρη συμπληρωμένος και πίνακα βαθμού $D_{ii} = \sum_j A_{ij}$. Η παραπάνω εξίσωση προκύπτει από την θεώρηση ότι οι κόμβοι που συνδέονται σε έναν γράφο είναι πιθανό να είναι ίδιας κατηγορίας (μεταδίδουν ψευδείς ειδήσεις ή όχι). Η θεώρηση αυτή, βέβαια, μπορεί να περιορίζει την μοντελοποίηση του γράφου, εφόσον δεν είναι απαραίτητο να κωδικοποιείται αποκλειστικά βάσει της ομοιότητας των κόμβων. Σημαντικό ρόλο παίζουν και άλλα επιπρόσθετα χαρακτηριστικά.

Ο γράφος κωδικοποιείται άμεσα χρησιμοποιώντας μοντέλο νευρωνικού δικτύου, f (X, A), και εκπαιδεύεται σε έναν επιβλεπόμενο στόχο L_o για όλους τους κόμβους με ετικέτες, για τους οποίους δεν εφαρμόζεται κανονικοποίηση στη συνάρτηση κόστους. Εφαρμόζοντας τη συνάρτηση f(·) στον πίνακα γειτνίασης, επιτρέπεται στο μοντέλο να κατανείμει πληροφορίες για τις παραγώγους της επιβλεπόμενης συνάρτησης κόστους L_o, μαθαίνοντας τις αναπαραστάσεις όλων των κόμβων, με ή χωρίς ετικέτα .

Αρχικά, θα παρουσιαστεί ο κανόνας διάδοσης, σε κάθε επίπεδο, για τα μοντέλα νευρωνικών δικτύων και έπειτα θα περιγραφεί το πώς ένα μοντέλο νευρωνικού δικτύου σε μορφή γράφου, μπορεί να είναι αποδοτικό σε ένα πρόβλημα ταξινόμησης κόμβων του γράφου.

5.8.1 Γρήγορη Προσέγγιση των γράφων μάθησης

Στην ενότητα αυτή παρέχεται το μαθηματικό υπόβαθρο για το μοντέλο νευρωνικού δικτύου σε μορφή γράφου, f (X, A). Θεωρούμε ένα πολύ-επίπεδο Graph Convolutional Network (GCN), με τον ακόλουθο κανόνα διάδοσης σε κάθε επίπεδο:

$$\mathrm{H}^{k+1} = \sigma(\widehat{D}^{-\frac{1}{2}}\widehat{A}\,\widehat{D}^{-\frac{1}{2}}H^{k}W^{k})$$

Ο πίνακας $\widehat{A} = \mathrm{A} + I_N$ είναι ο πίνακας γειτνίασης του μη κατευθυνόμενου γράφου G, με προστεθιμένες τις αναδράσεις του κάθε κόμβου, εάν αυτές υπάρχουν.

Ο I_N είναι ο μοναδιαίος πίνακας, όπως παραπάνω ο $D_{ii} = \sum_j A_{ij}$ είναι ο πίνακας βαθμού του γράφου και W^k είναι ο πίνακας βαρών που χρησιμοποιείται κατά την εκπαίδευση του γράφου και είναι συγκεκριμένος για κάθε επίπεδο του γράφου.

Με σ(·), αποδίδεται η συνάρτηση ενεργοποίησης του γράφου, που μπορεί να είναι η ReLU(·) = max(0, ·)

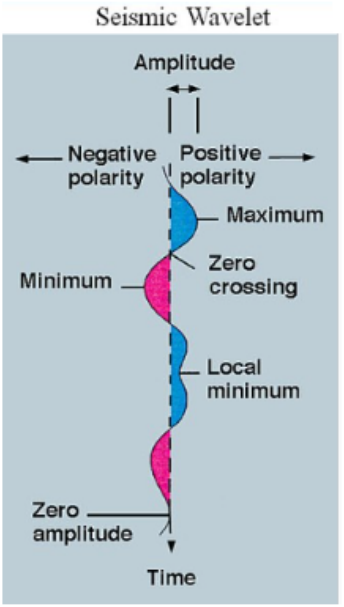
Ο $H^k \in \mathbb{R}^{NxD}$ είναι ο πίνακας των ενεργοποιήσεων στο k-οστό επίπεδο, με $\mathrm{H}^0 = \mathrm{X}$

Υπενθυμίζεται, πως ο πίνακας X αποδίδει τον πίνακα των χαρακτηριστικών του γράφου.

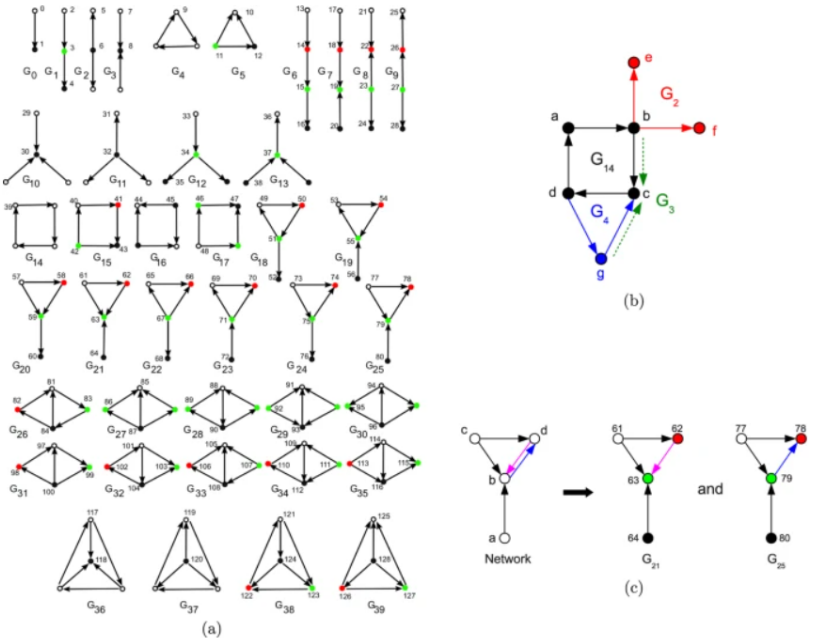
Θα συνεχίσουμε περιγράφοντας πώς εφαρμόζονται τα spectral graph convolutions, ανάλυση που θα ενισχύσει την κατανόηση των συνελικτικών δικτύων των γράφων.

5.8.2 Spectral Graph Convolutions

Με τον όρο “spectral” αποδίδεται η αποσύνθεση ενός σήματος ή εικόνας ή ήχου ή και γράφου σε έναν συνδυασμό, συνήθως άθροισμα, απλών στοιχείων όπως wavelets και graphlets [31].



Εικόνα 19 Το wavelet είναι ένα κύμα-ταλάντωση με πλάτος που εκκινεί στο μηδέν, μειώνεται ή αυξάνεται και επιστρέφει στο μηδέν μία ή περισσότερες φορές. Χαρακτηρίζονται ως "σύντομη ταλάντωση", ενώ, λόγω των ιδιοτήτων τους, χρησιμοποιούνται συχνά στην επεξεργασία σημάτων [32].



Εικόνα 20 Τα graphlets στα μαθηματικά χαρακτηρίζονται ως οι υπογράφοι που προκύπτουν ως ισομορφικές κλάσεις ενός γράφου. Οι υπογράφοι αυτοί προκύπτουν από τον αρχικό γράφο, σε κάθε συχνότητα και προκειμένου να σχηματιστεί ο υπογράφος, πρέπει να συγκεντρωθούν όλες οι ακμές των κόμβων του [64].

Επομένως, ορίζουμε τις spectral συνελίξεις πάνω σε γράφους ως τον πολλαπλασιασμό ενός σήματος $x \in \mathbb{R}^N$ (ένας πραγματικός αριθμός για κάθε κόμβο του γράφου) με ένα φίλτρο $g_\theta = diag(\theta)$, παραμετροποιημένο ώστε $\theta \in \mathbb{R}^N$ στο πεδίο Fourier, ούτως ώστε:

$$g_\theta \cdot x = U g_\theta U^T x \, (1)$$

όπου ο U είναι ο πίνακας των ιδιοδιανυσμάτων των του κανονικοποιημένου Λαπλασιανού γράφου $L = I_N - \widehat{D}^{-\frac{1}{2}} \widehat{A} \widehat{D}^{-\frac{1}{2}} = U \Lambda U^T$, Λ ο διαγώνιος πίνακας των ιδιοτιμών και $U^T x$ ο γραφικός μετασχηματισμός Fourier του x. Μπορούμε να αντιληφθούμε την g_θ ως μια συνάρτηση ιδιοτιμών του L, $g_\theta(\Lambda)$. Επειδή ο υπολογισμός της παραπάνω εξίσωσης μπορεί να είναι επώδυνος σε μεγάλους γράφους, αφού πολλαπλασιάζοντας με το ιδιοδιάνυσμα U η πολυπλοκότητα είναι $O(N^2)$, προσεγγίζουμε τη συνάρτηση $g_\theta(\Lambda)$ με πολυώνυμα Chebyshev εως και K τάξης:

$$g_\theta'(\Lambda) \approx \sum_{k=0}^{K} \theta_k' T_k(\bar{\Lambda}), \, (2)$$

όπου $\bar{\Lambda} = \frac{2}{\lambda_{max}} \Lambda - I_N$, $\theta' \in \mathbb{R}^K$ διάνυσμα συντελεστών Chebyshev, τα πολυώνυμα Chebyshev ορίζονται ως $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, οπου $T_0(x) = 1$ και $T_1(x) = x$

Συνεπώς, η αρχική εξίσωση γράφεται ως:

$$g_\theta' \cdot x \approx \sum_{k=0}^{K} \theta_k' T_k(\bar{L}) x, \text{ όπου } \bar{L} = \frac{2}{\lambda_{max}} L - I_N. \, (3)$$

Η παραπάνω εξίσωση, εκφράζει ένα πολυώνυμο K τάξης και άρα η έκφραση αυτή αφορά μόνο κόμβους που απέχουν το μέγιστο K βήματα μακριά από τον κεντρικό κόμβο. Η πολυπλοκότητα υπολογισμού της παραπάνω εξίσωσης ισούται με O(|E|). Παρατηρούμε, πως η πολυπλοκότητα μειώθηκε σε γραμμική και εξαρτάται από τον αριθμό των ακμών του γράφου. Το παραπάνω K-κανονικοποιημένο πολυώνυμο, θα χρησιμοποιηθεί για τον ορισμό των συνελικτικών νευρωνικών δικτύων σε γράφους (GCN).

5.8.3 Γραμμικό ανά επίπεδο μοντέλο

Ένα νευρωνικό μοντέλο βασισμένο σε γραφικές συνελίξεις κατασκευάζεται στοιβάζοντας πολλά συνελικτικά επίπεδα της μορφής της εξίσωσης (3), κάθε επίπεδο ακολουθείται από ένα σημείο και δεν χαρακτηρίζεται γραμμικό. Έστω, πως τώρα, περιορίζαμε την παράμετρο συνέλιξης των επιπέδων σε K = 1, μια συνάρτηση που είναι γραμμική, με αναφορά στην L [25].
Με τον τρόπο αυτό μπορούμε ακόμα να λάβουμε μια πλούσια κλάση συνελικτικών συναρτήσεων φίλτρων, στοιβάζοντας διάφορα τέτοια επίπεδα, δίχως τον περιορισμό των αυστηρών παραμέτρων των πολυωνύμων Chebyshev. Διαισθητικά, αναμένεται το μοντέλο αυτό να αντιμετωπίσει αποτελεσματικότερα το πρόβλημα της υπερπροσαρμογής (*overfitting*) στις τοπικές δομές γειτονιών των γράφων με πολύ μεγάλες κατανομές βαθμών κόμβων όπως, στην περίπτωσή μας, τα κοινωνικά

δίκτυα. Επιπροσθέτως, για ένα συγκεκριμένο υπολογιστικό όριο, αυτός ο ανα επίπεδο υπολογισμός επιτρέπει τη δημιουργία βαθύτερων μοντέλων, με περισσότερα δηλαδή επίπεδα, διαδικασία που αυξάνει την χωρητικότητα του μοντέλου, ώστε να δύναται να χρησιμοποιηθεί μέχρι και στα πιο ευρεία δίκτυα.

Σε αυτήν την γραμμική μοντελοποίηση των μοντέλων GCN, γίνεται επιπλέον η προσέγγιση $\lambda_{max} \approx 2$, ούτως ώστε οι παράμετροι του νευρωνικού δικτύου να προσαρμόζονται στην αλλαγή της κλίμακας κατά την εκπαίδευση. Έπειτα από τις προσεγγίσεις αυτές, η (3) απλοποιείται σε

$$g_{\theta'} \cdot x \approx \theta_{o'} x + \theta_{1'}(L - I_N)x = \theta_{o'} x - \theta_{1'} D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x \quad (4)$$

με δύο ελεύθερες παραμέτρους $\theta_{o'}$, $\theta_{1'}$. Οι παράμετροι των φίλτρων μπορούν να μοιραστούν σε ολόκληρο τον γράφο. Έχοντας εφαρμόσει διαδοχικά φιλτραρίσματα της μορφής αυτής, μπορούμε έπειτα να συνελίξουμε τους γείτονες της k-τάξης ενός κόμβου, όπου k είναι ο αριθμός των διαδοχικών διεργασιών, ή αλλιώς, ο αριθμός των συνελικτικών επιπέδων του μοντέλου του νευρωνικού δικτύου.

Στην πράξη, είναι πιο επωφελητικό να περιορίσουμε περαιτέρω τον αριθμό των παραμέτρων, για την αντιμετώπιση του overfitting και την ελαχιστοποίηση του αριθμού των πράξεων που συντελούνται (όπως οι πολλαπλασιασμοί πινάκων) σε κάθε επίπεδο. Έτσι, καταλήγουμε με την ακόλουθη έκφραση:

$g_\theta \cdot x \approx \theta\ (I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x$, με μία μοναδική παράμετρο $\theta = \theta_{o'} = -\theta_{1'}$. Σημαντικό να σημειωθεί πως πλέον, ο πίνακας $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ έχει ιδιοτιμές στο εύρος [0, 2]. Επαναλαμβάνοντας την εφαρμογή του τελεστή αυτού μπορεί να οδηγήσει σε αριθμητικές αστάθειες, όπως το φαινόμενο των εκρυγνήμενων ή εξαφανιζόμενων παραγώγων στα βαθιά νευρωνικά μοντέλα. Για την υπερνίκηση του μοντέλου αυτού, εισαγάγουμε το ακόλουθο κόλπο κανονικοποίησης: $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \overline{D}^{-\frac{1}{2}} \overline{A}\, \overline{D}^{-\frac{1}{2}}$, όπου $\overline{A} = A + I_N$ και $\overline{D}_{ii} = \sum_j \overline{A}_{ij}$.

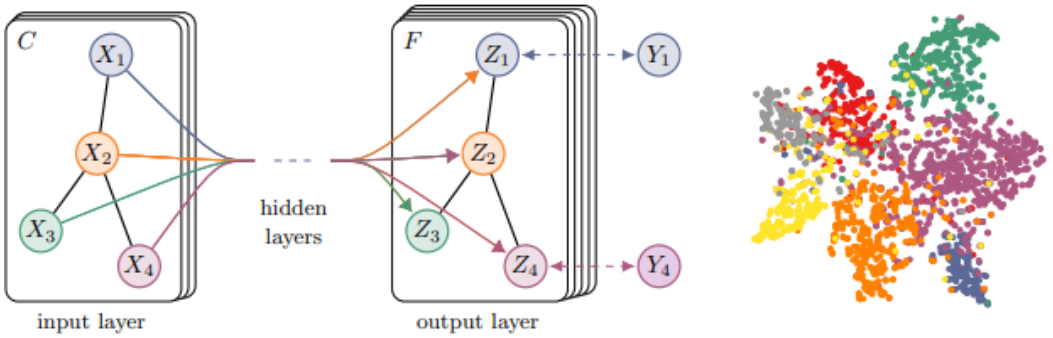
Μπορούμε να επεκτείνουμε τον ορισμό αυτό σε κάθε σήμα $X \in \mathbb{R}^{NxC}$, όπου C τα κανάλια εισόδου (πλέον έχουμε ένα C-διάστατο διάνυσμα χαρακτηριστικών για κάθε κόμβο) και F φίλτρα ή αντιστοιχίσεις χαρακτηριστικών, όπως αποτυπώνεται:

$$Z = \overline{D}^{-\frac{1}{2}} \overline{A}\, \overline{D}^{-\frac{1}{2}} X\Theta$$

οπου $\Theta \in \mathbb{R}^{CxF}$ είναι ένας πίνακας με παραμέτρους φίλτρων και $Z \in \mathbb{R}^{NxF}$ είναι ο πίνακας του συνελιγμένου σήματος. Αυτός ο τελεστής φιλτραρίσματος έχεις πολυπλοκότητα O(|E|FC), εφόσον ο $\overline{A}X$ μπορεί αποτελεσματικά να αρχικοποιηθεί ως το γινόμενο ενός αραιού (sparse) με έναν πυκνό (dense) πίνακα. Προς αποσαφήνιση της παραπάνω διατύπωσης, αναφέρεται πως αραιός χαρακτηρίζεται ένας πίνακας που αποτελείται κυρίως από μηδενικές τιμές, ενώ πυκνός είναι ο πίνακας που οι περισσότερές του τιμές είναι μη μηδενικές.

5.8.4 Ημι-Επιβλεπόμενη Ταξινόμηση Κόμβων

Έχοντας παρουσιάσει το απλό, αλλά συγχρόνως ευέλικτο μοντέλο $f(X, A)$ που συνεισφέρει στην αποδοτική διάδοση πληροφορίας στους γράφους, επιστρέφουμε στο βασικό μας πρόβλημα της ημι-επιβλεπόμενης ταξινόμησης κόμβων. Σύμφωνα με τα όσα ελέχθησαν παραπάνω, είναι εφικτό να απλοποιήσουμε κάποιες υποθέσεις για το μοντέλο $f(X, A)$ τόσο στα δεδομένα X όσο και στον πίνακα γειτνίασης A [25]. Αναμένουμε οι τροποποιήσεις αυτές να είναι ιδιαίτερα επωφελητικές ιδίως σε περιπτώσεις όπου ο A περιέχει πληροφορίες που δεν περιέχονται στα δεδομένα X, όπως στην περίπτωσή μας, ο A περιέχει τις συνδέσεις των χρηστών που έκαναν retweet την αρχική είδηση.
Το μοντέλο που περιγράφηκε παραπάνω, αποδίδεται και σχηματικά παρακάτω:



(a) Graph Convolutional Network (b) Hidden layer activations

Εικόνα 21 : Αριστερά: Παρουσιάζεται σχηματικά το πολυεπίπεδο Συνελικτικό Δίκτυο Γράφου (GCN) για ημι-επιβλεπόμενη μάθηση, με C το αριθμό κανάλια εισόδου και F αντιστοιχίσεις χαρακτηριστικών στο επίπεδο της εξόδου. Οι ακμές αποδίδονται με μαύρο χρώμα. Η δομή του γράφου μοιράζεται στα επίπεδα και τα labels αποδίδονται με Y_i. Δεξιά: t-SNE αναπαράσταση των ενεργοποιήσεων των κρυμμένων επιπέδων, ενός διεπίπεδου GCN για ένα πρόβλημα ταξινόμησης πολλών κλάσεων, όπου διαφορετικό χρώμα και διαφορετική κλάση [25].

Θεωρούμε ένα διεπίπεδο GCN για το πρόβλημα της ταξινόμησης κόμβων σε έναν γράφο με συμμετρικό πίνακα γειτνίασης A (είτε δυαδικό, είτε με βάρη) [25]. Αρχικά, υπολογίζουμε το $\hat{A} = \overline{D}^{-\frac{1}{2}} \overline{A}\, \overline{D}^{-\frac{1}{2}}$ ως βήμα προεπεξεργασίας. Το forward μοντέλο μας λαμβάνει την απλή μορφή:

$$Z = f(X, A) = softmax(\hat{A}\, ReLU(\hat{A} X W^{(0)}) W^{(1)}) \quad (5)$$

Για το κρυφό επίπεδο (*hidden layer*) έστω $W^{(0)} \in \mathbb{R}^{CxH}$ ότι είναι η είσοδος στο κρυφό πίνακα βάρους του του κρυφού επιπέδου με H αντιστοιχίσεις χαρακτηριστικών. $W^{(1)} \in \mathbb{R}^{CxF}$ είναι ο πίνακας βάρους για το κρυφό επίπεδο εξόδου. Η συνάρτηση ενεργοποίησης $softmax(x_i) = \frac{1}{z} exp(x_i)$, οπου $z = \sum_i exp(x_i)$, εφαρμόζεται ανά γραμμή. Το cross-entropy σφάλμα, σε όλα τα δεδομένα με label, είναι το ακόλουθο: $L = -\sum_{l \epsilon y_L} \sum_{f=1}^{F} y_{lf} ln Z_{lf}$,όπου y_L το

σύνολο των κόμβων που έχουν labels. Τα βάρη $W^{(0)}$ και $W^{(1)}$ του νευρωνικού δικτύου εκπαιδεύονται με τη μέθοδο gradient descent. Το στοχαστικό κομμάτι της διαδικασίας, επέρχεται με τη μέθοδο του dropout όπου αφαιρούμε ορισμένους νευρώνες από το δίκτυο προκειμένου αυτό να γενικεύει πιο αποδοτικά.
Για την υλοποίηση της διπλωματικής εργασίας χρησιμοποιήθηκε η αρχιτεκτονική *TensorFlow*, η οποία αξιοποιεί την GPU των υπολογιστών. Η υπολογιστική πολυπλοκότητα είναι ίση με O(|E|CHF), που προκύπτει από την εξίσωση (5), γραμμική ως προς τον αριθμό των κόμβων.

6. BotOrNot

6.1 Τι είναι το Bot;

Το bot είναι μια εφαρμογή λογισμικού η οποία είναι προγραμματισμένη να επιτελεί συγκεκριμένες εργασίες. Τα bots είναι αυτοματοποιημένα, γεγονός που σημαίνει ότι μπορούν να λειτουργήσουν βάσει οδηγιών δίχως να χρειάζονται κάποιον ανθρώπινο χειριστή. Συνήθως, χρησιμοποιούνται σε επαναλαμβανόμενες διαδικασίες, χάρη στην ικανότητά τους να τις επιτελούν σημαντικά ταχύτερα από ότι οι άνθρωποι.
Ορισμένα bots είναι χρήσιμα, όπως αυτά των υπηρεσιών εξυπηρέτησης χρηστών καθώς και των μηχανών αναζήτησης που αντιστοιχούν το περιεχόμενο στην αναζήτηση του χρήστη.
Από την άλλη πλευρά, υπάρχουν τα "κακοπροαίρετα" bots τα οποία προγραμματίζονται με τρόπο τέτοιο ώστε να εισβάλλουν στους λογαριασμούς των χρηστών και να προβαίνουν σε επιβλαβείς για τον χρήστη δραστηριότητες, οπως να διαδίδουν παραπλανητική αλληλογραφία ή ακόμα ψευδείς ειδήσεις στα κοινωνικά δίκτυα, όπως στην περίπτωσή μας. Από αυτά τα bots είναι αναγκαία η προστασία του χρήστη των κοινωνικών δικτύων [65].

6.2 Εξαγωγή Χαρακτηριστικού BotOrNot

Για το λόγο αυτό, ένα επιπλέον χαρακτηριστικό που επιλέγουμε να μάθει το νευρωνικό μας δίκτυο και να εκπαιδευτεί πάνω σε αυτό, είναι το εάν ο χρήστης που διαδίδει την είδηση είναι *bot* ή όχι. Αυτό επιτυγχάνεται βάσει της μεθόδου *BotOrNot* που διαφορετικά ονομάζεται *Botometer* [66].
Χάρη στο *BotOrNot* διαθέσιμο κώδικα βάζουμε ως είσοδο μια λίστα με τα *Twitter usernames* του προς εξέταση χρήστη και το πρόγραμμα έχοντας συλλέξει δεδομένα της πρόσφατης δραστηριότητας του χρήστη, υπολογίζει και επιστρέφει ένα score από το 0 έως και το 1, για το πόσο *bot* θεωρείται ο χρήστης αυτός. Στην εργασία αυτή, οι χρήστες με σκορ 0.5 και άνω θεωρούνται *bots*, ενώ, σε διαφορετική περίπτωση, θεωρούνται πραγματικοί χρήστες του διαδικτύου.

Κατά την επεξεργασία της εκάστοτε είδησης με σκοπό την απόφαση για το εάν αυτή είναι ψευδής ή αληθής ακολουθείται η διαδικασία από το *Machine Learning Natural Language Processing*. Με τον τρόπο αυτό, προκύπτουν τα *word embeddings* του κάθε κειμένου που είναι απολύτως χρήσιμα για την εκπαίδευση του νευρωνικού μας δικτύου. Τα *word embeddings* επιλέγουμε να προκύψουν είτε με την μέθοδο *word2vec* είτε με τη μέθοδο *BERT*. Παρακάτω, παρουσιάζονται πιο εκτενώς οι προαναφερόμενοι όροι ορμώμενοι από τις δημοσιεύσεις [46], [47], [48].

7. Word Embeddings

Τα *word embeddings* είναι μια τεχνική με την οποία η κάθε λέξη σε ένα κείμενο μετατρέπεται σε μια αριθμητική αναπαράσταση, σε ένα διανυσμα (*vector*) [44]. Όταν κάθε λέξη έχει αντιστοιχιστεί με ένα διάνυσμα, το διάνυσμα αυτό μαθαίνεται με έναν τρόπο που ομοιάζει με νευρωνικό δίκτυο. Σκοπός των διανυσμάτων είναι η συγκέντρωση των διάφορων χαρακτηριστικών της εκάστοτε λέξης σε σχέση με το νόημά της και τον ρόλο της στο συνολικό κείμενο. Τα χαρακτηριστικά αυτά έχουν να κάνουν με τη σημασιολογία της λέξης στο κείμενο (*semantic relationship of the word*), τους ορισμούς, το πλαίσιο στο οποίο αναφέρεται.
Με μεγαλύτερη ακρίβεια διατυπωμένο, το *word embedding* είναι ο όρος που χρησιμοποιείται για την αναπαράσταση των λέξεων κατά την ανάλυση κειμένου. Η αναπαράσταση αυτή πρόκειται για ένα διάνυσμα πραγματικών τιμών, το οποίο κωδικοποιεί τη σημασία της λέξης. Κατά συνέπεια, λέξεις με παρόμοιο σημασιολογικό περιεχόμενο βρίσκονται πιο κοντά στον χώρο των διανυσμάτων. Τα *word embeddings* μπορούν να προκύψουν χρησιμοποιώντας ένα σύνολο τεχνικών μοντελοποίησης γλώσσας (*language modeling*) καθώς και τεχνικές μάθησης χαρακτηριστικών (*feature learning*), όπου λέξεις ή φράσεις από το λεξικό αντιστοιχίζονται σε διανύσματα πραγματικών αριθμών (*real number vectors*).
Οι μέθοδοι παραγωγής των διανυσμάτων αυτών περιλαμβάνουν νευρωνικά δίκτυα, μείωση της διαστατικότητας του πίνακα συνύπαρξης της λέξης, πιθανοτικά μοντέλα, εξηγήσιμη γνώση βάσης. Τα *embeddings* των λέξεων και των φράσεων, όταν χρησιμοποιούνται ως είσοδος στα διάφορα προβλήματα μηχανικής μάθησης, ενισχύουν την επίδοση της επεξεργασίας φυσικής γλώσσας κάνοντας τα μοντέλα πιο έγκυρα και αποτελεσματικά.
Ο λόγος που καταφεύγουμε σε διανυσματική αναπαράσταση των λέξεων ενός κειμένου, είναι και χρηστικός, καθώς ο υπολογιστής δεν μπορεί να διαχειριστεί δεδομένα σε ανεπεξέργαστη μορφή κειμένου. Κάποιες επιπλέον δυνατότητες που προσφέρουν τα *word embeddings* είναι η ανίχνευση ομοιότητας μεταξύ των λέξεων.

7.1 Word2vec

Ο αλγόριθμος *word2vec* αποτελεί μια τεχνική επεξεργασίας φυσικής γλώσσας και ο αλγόριθμός του χρησιμοποιεί ένα μοντέλο νευρωνικού δικτύου το οποίο μαθαίνει πώς συσχετίζονται οι λέξεις μέσα σε ένα τεράστιο λεξικό λέξεων [45]. Όταν το μοντέλο αυτό εκπαιδευτεί, είναι σε θέση να ανιχνεύσει συνώνυμες λέξεις ή ακόμα να προτείνει λέξεις για μια ανολοκλήρωτη πρόταση. Όπως προδίδει και το όνομα, τα *word2vec* αναπαριστούν κάθε λέξη με ένα συγκεκριμένο διάνυσμα πραγματικών αριθμών.
Τα *word2vec* είναι ένα σύνολο από μοντέλα που χρησιμοποιούνται για να παράξουν *word embeddings*. Αυτά τα μοντέλα είναι "ρηχά", διεπίπεδα νευρωνικά δίκτυα που εκπαιδεύονται για να ανακατασκευάσουν το γλωσσικό πλαίσιο των λέξεων. Τα *word2vec* λαμβάνουν ως είσοδο ένα μεγάλο λεξικό, που προκύπτει από τις λέξεις καποιου κειμένου και παράγει των διανυσματικό χώρο, τυπικά αρκετών εκατοντάδων διαστάσεων με κάθε λέξη του λεξικού να λαμβάνει μια μοναδική διανυσματική αναπαράσταση.
Η αποτελεσματικότητα της αρχιτεκτονικής *word2vec* εντοπίζεται στο ότι τα διανύσματα των λέξεων τοποθετούνται με τρόπο τέτοιο στον διανυσματικό χώρο, ώστε λέξεις με παρόμοιο σημασιολογικό περιεχόμενο τοποθετούνται κοντά στον διανυσματικό χώρο. Δεδομένου ενός μεγάλου *dataset* τα *word2vec* προβαίνουν σε ισχυρές εκτιμήσεις για την σημασία της λέξης βάσει των εμφανίσεών της στο κείμενο. Έτσι, προκύπτουν και οι σχέσεις της λέξεις με τις υπόλοιπες λέξεις του κειμένου. Για παράδειγμα, λέξεις όπως "King" και "Queen" είναι πολύ παρόμοιες μεταξύ τους. Συγκεκριμένα, όταν προβαίνουμε σε αλγεβρικούς υπολογισμούς μεταξύ των *word embeddings* μπορούμε να βρούμε μια κοντινή προσέγγιση των ομοιοτήτων των λέξεων. Στην περίπτωση των παραπάνω λέξεων, αν από το δισδιάστατο διάνυσμα της λέξης "King" αφαιρέσουμε το δισδιάστατο διάνυσμα

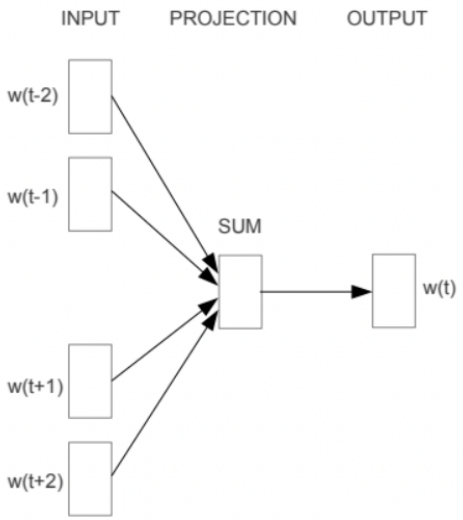
της λέξης “Man” και προσθέσουμε το δισδιάστατο διάνυσμα της λέξης “Woman”, λαμβάνουμε το δισδιάστατο *word embedding* της λέξης “Queen”.

Αρχιτεκτονική word2vec

Υπάρχουν δύο αρχιτεκτονικές στις οποίες οφείλεται η επιτυχία του *word2vec* μοντέλου. Αυτές είναι το skip-gram και η CBOW αρχιτεκτονική [46], [47].

7.1.1 CBOW (Continuous Bag of Words)

Η αρχιτεκτονική αυτή είναι παρόμοια με τα *feedforward* νευρωνικά δίκτυα και προσπαθεί να προβλέψει μια λέξη-στόχο από μια λίστα λέξεων που βρίσκονται στο πλαίσιο της προς πρόβλεψη λέξης. Το πώς επιτυγχάνεται αυτό από το μοντέλο είναι αρκετά απλό. Δεδομένης της φράσης “*Have a great day*” επιλέγουμε τα *context words* να είναι {*“have”, “great”, “day”*} και η προς πρόβλεψη λέξη {“*a*”}. Αυτό που κάνει το μοντέλο είναι να επεξεργάζεται τις διανυσματικές αναπαραστάσεις των *context words* και να προσπαθεί να προβλέψει την λέξη στόχο. Στην γενική περίπτωση, θεωρώντας ένα *context window* μεγέθους 2, σχηματίζουμε τα ζεύγη (*context_window_words, target_word*) τα οποία για το παράδειγμά μας είναι ([it, a], is), ([is, pleasant], a),([a, day], pleasant). Με αυτά τα ζεύγη, το μοντέλο μας προσπαθεί να προβλέψει τα *target words* από τα δεδομένα *context words*.



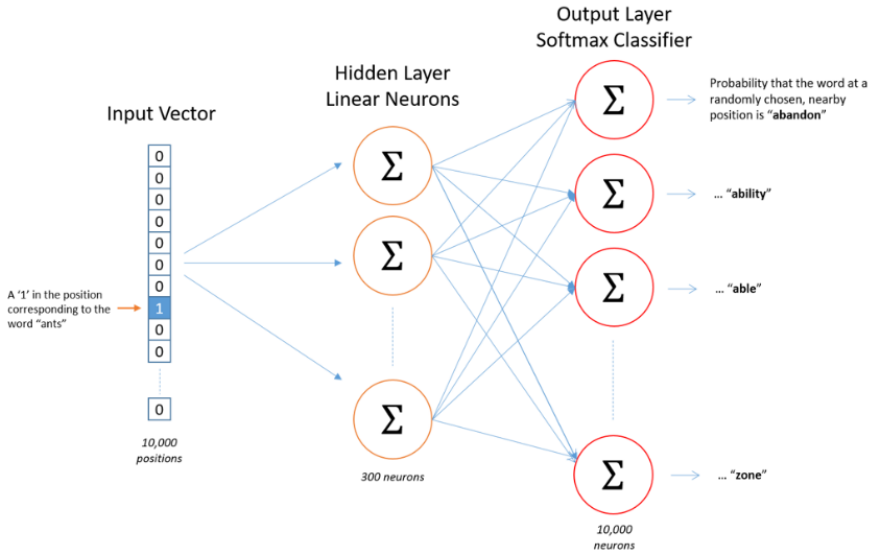
Εικόνα 22 Αρχιτεκτονική μοντέλου CBOW.

7.1.2 Continuous Skip-Gram Model

Σε αντίθεση με το CBOW μοντέλο, στο οποίο οι διανυσματικές αναπαραστάσεις των *context* λέξεων (περιβάλλουσες λέξεις) συνδυάζονται προκειμένου να προβλέψουν την λέξη *target* (κεντρική λέξη), στο *Skip-gram model* η διανυσματική αναπαράσταση της λέξης εισόδου χρησιμοποιείται προκειμένου να προβλέψουμε τις *context* λέξεις. Και σε αυτήν την αρχιτεκτονική χρησιμοποιούνται νευρωνικά δίκτυα για την μάθηση των αναπαραστάσεων των λέξεων. Απαραίτητη προϋπόθεση για κάθε νευρωνικό δίκτυο ή κάθε επιβλεπόμενη μάθηση είναι τα δεδομένα εισόδου να έχουν κάποια

ετικέτα. Στην περίπτωση εξαγωγής *word embeddings* η κάθε λέξη θα πρέπει να είχε ως *label* το *word embedding* της. Στην πραγματικότητα, τα βάρη του κρυφού επιπέδου αποτελούν τα *word vectors* που προσπαθούμε να μάθουμε.

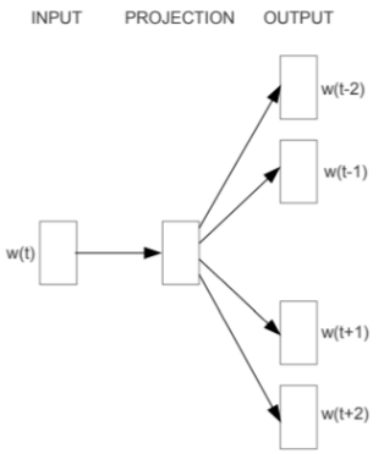
Δεδομένης μιας λέξης, λοιπόν, επιδιώκουμε να προβλέψουμε τις γειτονικές της λέξεις. Έστω ότι έχουμε την πρόταση "*I will have orange juice and eggs for breakfast.*" και το μέγεθος του παραθύρου είναι ίσο με 2, θέτοντας ως *input* λέξη το "*juice*" οι γειτονικές της λέξεις είναι {*have, orange, and, eggs*}. Σχηματίζουμε πάλι τα ζεύγη λέξη εισόδου και λέξη *target* έχουμε:
(*juice, have*), (*juice, orange*), (*juice, and*), (*juice, eggs*). Κατά την εκπαίδευση, το πόσο κοντά νοηματικά ή και μορφολογικά είναι μια λέξη στην *target* λέξη, δεν παίζει κάποιο ρόλο. Ως συνέπεια αυτού, τις λέξεις "{*have, orange, and, eggs*}" θα τις διαχειριστεί με τον ίδιο τρόπο στην εκπαίδευση.



Εικόνα 23 Πρόβλεψη *context words* με είσοδο την λέξη *ant*.

Οι διαστάσεις του διανύσματος εισόδου είναι 1xV οπου V είναι ο αριθμός των λέξεων του λεξικού. Στην εικόνα η λέξη αποδίδεται με *one-hot representation* οπου το διάνυσμα έχει παντού μηδενικά και άσσο στη θέση που ανταποκρίνεται στη λέξη εισόδου. Το κρυφό επίπεδο έχει διάσταση VxE, όπου το E είναι το μέγεθος των *word embeddings* το οποίο πρόκειται για υπεπραράμετρο. Η έξοδος του κρυφού επιπέδου έχει διάσταση 1xE το οποίο περνά ως είσοδος στο *softmax layer*. Οι διαστάσεις του επιπέδου εξόδου ισούνται με 1xV όπου κάθε τιμή στο διάνυσμα είναι η πιθανότητα του *target word* σε αυτή τη θέση.

Η διαδικασία του backpropagation για τα δείγματα της εκπαίδευσης που ανταποκρίνονται σε μία λέξη εισόδου, επιτυγχάνεται με ένα *back pass*. Έτσι, για την λέξη *juice,* ολοκληρώνουμε το *forward pass* και για τις τέσσερις *target* λέξεις {*have, orange, and, eggs*}. Έπειτα υπολογίζουμε τα διανύσματα σφάλματος (*error vectors*), διάστασης 1xV τα οποία αθροίζουμε λαμβάνοντας ένα τελικό 1xV διάνυσμα. Τα βάρη του κρυφού επιπέδου ανανεώνονται βάσει του συγκεντρωτικού αυτού διανύσματος σφάλματος.



Εικόνα 24 Αρχιτεκτονική μοντέλου Skip-gram.

Καταλήγουμε στο ότι:

Το Skip-gram μοντέλο αποδίδει καλύτερα σε μικρό αριθμό δεδομένων εκπαίδευσης, ενώ αναπαριστά ικανοποιητικά σπάνιες λέξεις και φράσεις.

Το CBOW είναι αισθητά πιο ταχύ στην εκπαίδευση από το μοντέλο Skip-gram και πιο αποδοτικό για τις πιο συχνά εμφανιζόμενες λέξεις.

7.2 BERT

Το Bidirectional Encoder Representations from Transformers (BERT) προεκπαιδευμένο γλωσσικό μοντέλο είναι μία τεχνική της μηχανικής μάθησης για την επεξεργασία της φυσικής γλώσσας η οποία αναπτύχθηκε από την *Google*. Το αρχικό, αγγλικής γλώσσας, μοντέλο BERT αποτελείτο από δύο μοντέλα. Το BERT-base που διαθέτει 12 κωδικοποιητές με 12 *bidirectional self-attention heads* και το BERT-large με 24 κωδικοποιητές και 16 bidirectional self-attention heads. Και τα δύο μοντέλα είναι προεκπαιδευμένα από δεδομένα που δεν διαθέτουν ετικέτα από το *dataset BooksCorpus* τα οποία διαθέτουν 800 εκατομμύρια λέξεις και το *dataset* της αγγλικής *wikipedia* που διαθέτει 2.500 εκατομμυρια λέξεις.
Το BEPT χρησιμοποιεί *Transformer*, έναν *attention* μηχανισμό που μαθαίνει τις σχέσεις των λέξεων μέσα στο κείμενο. Στην απλούστερή του μορφή, το μοντέλο διαθέτει δύο μηχανισμούς. Έναν κωδικοποιητή που διαβάζει το κείμενο εισόδου και έναν αποκωδικοποιητή που παράγει το αποτέλεσμα πρόβλεψης. Εφόσον ο στόχος του BERT είναι να παράξει ένα γλωσσικό μοντέλο, επαρκεί ο μηχανισμός κωδικοποίησης. Εν αντιθέσει, με τα λοιπά γλωσσικά μοντέλα που διαβάζουν το κείμενο εισόδου με μορφή ακολουθίας από από αριστερά στα δεξιά ή το αντίστροφο, ο Transformer κωδικοποιητής διαβάζει ολόκληρη την ακολουθία μονομιάς [48]. Για το λόγο αυτό θεωρείται *bidirectional*, εφόσον είναι πιο ακριβές να πούμε ότι τα δεδομένα δεν εισέρχονται με μία συγκεκριμένη κατεύθυνση. Χάρη στο χαρακτηριστικό αυτό, το μοντέλο μαθαίνει το πλαίσιο της κάθε λέξης βασιζόμενη στις περιβάλλουσες από δεξιά και αριστερά λέξεις.
Ως είσοδο δέχεται μία ακολουθία λέξεων οι οποίες πρώτα μετατρέπονται σε διανύσματα και έπειτα επεξεργάζονται από το νευρωνικό δίκτυο. Η έξοδος είναι μια ακολουθία διανυσμάτων μεγέθους Η στην οποία κάθε διάνυσμα αντιστοιχεί σε μία λέξη εισόδου.

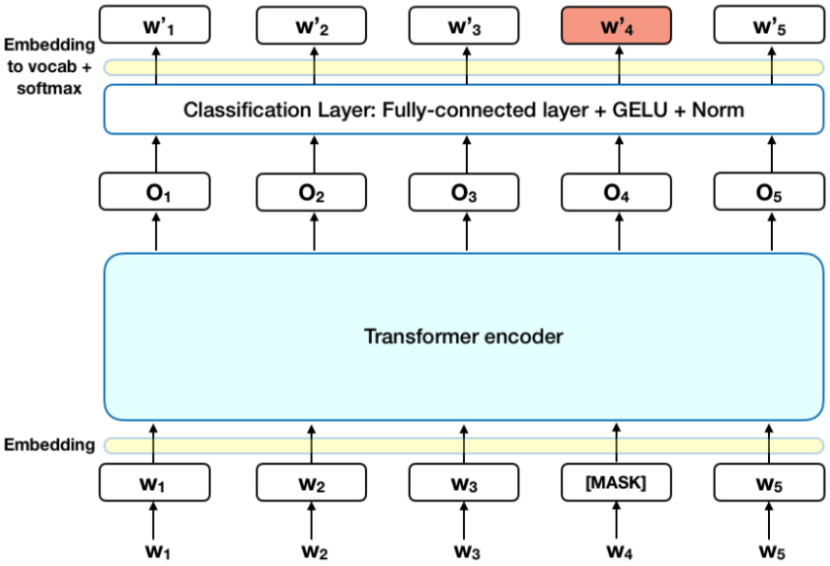
Όταν εκπαιδεύουμε τα γλωσσικά μοντέλα πολλές φορές στόχος είναι η πρόβλεψη της επόμενης λέξης σε μία ακολουθία λέξεων. Στα *bidirectional* γλωσσικά μοντέλα, η εξαγωγή των *context* λέξεων είναι ελαφρώς περιορισμένη. Για την υπερνίκηση του εμποδίου αυτού το BERT χρησιμοποιεί δύο στρατηγικές εκπαίδευσης.

Αρχιτεκτονική

7.2.1 Masked LM (MLM)

Προτού εισαγάγουμε στο BERT τις ακολουθίες λέξεων, το 15% των λεξεων σε κάθε ακολουθία αντικαθίσταται με ένα [MASK] token. Το μοντέλο έπειτα επιχειρεί να προβλέψει την πραγματική τιμή των λέξεων που έχουν αντικατασταθεί με [MASK] token, βασιζόμενο στο *context* άλλων λέξεων της ακολουθίας που δεν έχουν υποστεί [MASK] [48]. Τεχνικά, η πρόβλεψη των λέξεων εξόδου απαιτεί:

1. Προσθήκη ενός επιπέδου ταξινόμησης στην κορυφή του αποτελέσματος κωδικοποίησης.
2. Πολλαπλασιασμός των διανυσμάτων εξόδου με τον πίνακα των *embeddings*. μετατρέποντάς τα στις διαστάσεις του λεξιλογίου.
3. Υπολογισμός της πιθανότητας κάθε λέξης του λεξιλογίου χρησιμοποιώντας τη *softmax* συνάρτηση.



Εικόνα 25 Εκπαίδευση μοντέλου BERT με *Masked LM* [48].

7.2.2 Next Sentence Prediction (NSP)

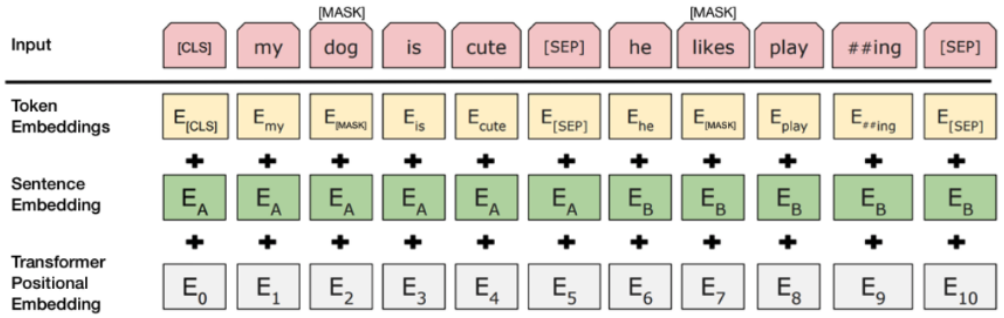
Κατά τη διαδικασία της εκπαίδευσης, το μοντέλο λαμβάνει ζευγάρια προτάσεων ως είσοδο και μαθαίνει να προβλέπει εάν η δεύτερη πρόταση του ζεύγους είναι η πρόταση που ακολουθεί την πρώτη στο αρχικό κείμενο. Κατά την εκπαίδευση, στο 50% των ζευγών εισόδου η δεύτερη πρόταση πράγματι ακολουθεί την πρώτη στο αρχικό κείμενο, ενώ για το υπόλοιπο 50% δεν ισχύει αυτό. Η υπόθεση είναι ότι η τυχαία πρόταση θα αποσυνδεθεί από την πρώτη πρόταση. Προκειμένου το

μοντέλο να διακρίνει τις δύο προτάσεις εισόδου, τις επεξεργαζόμαστε με τον εξής τρόπο προτού μπουν ως είσοδος στο μοντέλο.

1. Ένα [CLS] token προστίθεται στην αρχή της πρώτης πρότασης και ένα [SEP] token στο τέλος της κάθε πρότασης.

2. Το *embedding* της κάθε πρότασης προστίθεται στα *embeddings* της κάθε λέξης της ίδιας πρότασης, όπως αποδίδεται σχηματικά.

3. Ένα *embedding* που προσδιορίζει τη θέση της λέξης στην πρόταση προστίθεται σε κάθε *embedding* της λέξης



Εικόνα 26 Εκπαίδευση μοντέλου BERT με *Next Sentence Prediction (NSP)* [48].

Προκειμένου το μοντέλο BERT να προβλέψει εάν η δεύτερη πρόταση είναι συνδεδεμένη στην πρώτη ακολουθούνται τα επόμενα βήματα:

1. Ολόκληρη η ακολουθία εισόδου περνά στο *Transformer* μοντέλο.

2. Η έξοδος του [CLS] token μετατρέπεται σε ένα 2x1 διάνυσμα, χρησιμοποιώντας ένα απλό επίπεδο ταξινόμησης, το οποίο έχει εκπαιδευτεί με πίνακες βαρών και *biases*.

3. Υπολογίζουμε την πιθανότητα η δεύτερη πρόταση να είναι αυτή που ακολουθεί την πρώτη με τη συνάρτηση *softmax*.

Όταν εκπαιδεύουμε το μοντέλο BERT τα *Masked LM* και *Next Sentence Prediction* εκπαιδεύονται μαζί, με σκοπό την ελαχιστοποίηση του συνδυαστικού σφάλματος που υπολογίζεται από την *loss function* [48].

Έχοντας καλύψει σε ικανοποιητικό βαθμό την απαραίτητη για την κατανόηση της συγκεκριμένης εργασίας θεωρία, αναπτύσσοντας του κύριους όρους της, μεταβαίνουμε στην περιγραφή των βημάτων της υλοποίησης των μεθόδων της ανίχνευσης ψευδών ειδήσεων.

8. Περιγραφή Υλοποίησης

Οι τεχνικές ανίχνευσης που επιλέχθηκαν παρουσιάζονται συγκεντρωτικά παρακάτω:

1. User-Related Fake News Detection με τη χρήση 2-Layer GCNs

2. Topic-Related Fake News Detection με τη χρήση 2-Layer GCNs

3. User Preference Fake News Detection με τη χρήση 3-Layer GCNs και ενισχυμένα διανύσματα χαρακτηριστικών

Για κάθε μία από αυτές σημειώνονται οι μετρικές απόδοσης με τα σχετικά διαγράμματα, πλεονεκτήματα και αδυναμίες. Η κατηγορία που ξεπερνά σε επιδόσεις τόσο τα απλά νευρωνικά δίκτυα όσο και τα νευρωνικά δίκτυα των γράφων, είναι η User Preference Fake News Detection, διαθέτοντας τα πιο σύνθετα χαρακτηριστικά για κάθε κόμβο του δικτύου. Για τον σκοπό αυτό, εξελίσσουμε ακόμα περισσότερο το GCN της κατηγορίας αυτής, προσδίδοντας του μηχανισμό SAmple aggreGatE που συνθέτει το μοντέλο SAGE, όπως και μηχανισμό attention που αποδίδει το μοντέλο GAT.

8.1 User-Related Fake News Detection με τη χρήση GCNs

Η πρώτη προς ανάλυση μέθοδος, με την οποία θα εξετάζονται οι ειδήσεις ως προς την εγκυρότητά τους, εστιάζει στα χαρακτηριστικά του χρήστη. Η ανίχνευση ψευδών ειδήσεων με βάση τα χαρακτηριστικά του χρήστη, πρόκειται για τη πιο συχνή μέθοδο εύρεσης μη έγκυρων ειδήσεων. Στην συγκεκριμένη εργασία, γίνεται μία επέκταση της μεθόδου, από τα πιο απλοϊκά και συνηθισμένα μοντέλα επεξεργασίας κειμένου όπως τα Long Short Term Memory (LSTMs), Convolutional Neural Networks (CNNs), BERT, Multi Layer Perceptron (MLP) στην πιο σύνθετη μορφή, αυτή του γράφου, που δύναται να αναπαραστήσει τις σύνθετες σχέσεις που επικρατούν σε ένα κοινωνικό δίκτυο.
Καθώς η εκτενής περιγραφή των παραπάνω μοντέλων ξεφεύγει από τους στόχους της παρούσας εργασίας, θα γίνει μια σύντομη αναφορά σε κάθε ένα από αυτά για λόγους πληρότητας.

- LSTM: Ειδική μορφή νευρωνικών δικτύων ανάδρασης, ικανά να μαθαίνουν και να μοντελοποιούν χρονικές ακολουθίες και τις μεγάλου εύρους εξαρτήσεις τους με μεγαλύτερη ακρίβεια από άλλους τύπους RNN [67].

- Perceptron: Είναι ένας τεχνητός νευρώνας και το απλούστερο νευρωνικό δίκτυο. Είναι ένας αλγόριθμος δυαδικού ταξινομητή. Αντιστοιχεί την είσοδό του, που συνήθως είναι διάνυσμα πραγματικών τιμών, σε μία τιμή εξόδου 0 ή 1 [69].

- MLP: Δίκτυο που συντίθεται από πολλά επίπεδα Perceptrons [68].

Η ανάγκη για ανίχνευση ψευδών ειδήσεων με πιο σχολαστικό και αναλυτικό τρόπο, μας ωθεί στην εστίαση στα χαρακτηριστικά του χρήστη. Αναζητείται η σύνδεση μεταξύ των ψευδών ειδήσεων και των προφίλ των χρηστών. Τίθεται, λοιπόν, το ερώτημα “Ποια χαρακτηριστικά των χρηστών οδηγούν στη διάδοση των ψευδών ειδήσεων;”.
Σύμφωνα με την δημοσίευση [52], συνολική πρόσβαση στις πληροφορίες του δημόσιου λογαριασμού του κάθε χρήστη επιτράπηκε μέσω αίτησης έκδοσης του Twitter API, από τους ίδιους τους συγγραφείς της δημοσίευσης. Οι συγγραφείς αναλύοντας και επεξεργάζοντας τα χαρακτηριστικά που μας παρέχει το σύνολο δεδομένων, επέλεξαν αυτά τα οποία κατά κύριο λόγο καθορίζουν εάν η είδηση είναι ψευδής ή αληθής με τη μέθοδο “*Feature Importance Analysis*”.

Συγκεκριμένα, τα χαρακτηριστικά που μπορούμε να εξάγουμε για κάθε χρήστη από την βάση δεδομένων μας είναι τα ακόλουθα.

8.1.1 Verified

Εκφράζει εάν ο λογαριασμός είναι επικυρωμένος. Κατα κανόνα, λογαριασμοί που έχουν το διακριτικό αυτό δεν διαδίδουν ψευδείς ειδήσεις.

8.1.2 Location

Το μέρος στο οποίο διαμένει ο χρήστης που έκανε την δημοσίευση. Έρευνες αποδεικνύουν πως η κατανομή της τοποθεσίας διαφέρει για τις ψευδείς και τις αληθείς ειδήσεις. Στη γενική περίπτωση, φαίνεται πως περισσότερες ψευδείς ειδήσεις διαδίδονται στην δυτική πλευρά της Αμερικής παρά στην Ανατολική [52].

8.1.3 Followers Count

Ο αριθμός των ακολούθων του χρήστη. Συνήθως, ο μεγάλος αριθμός ακολούθων προσδίδει εγκυρότητα στους χρήστες.

8.1.4 Friends Count

Παρόμοια με παραπάνω, φίλοι χαρακτηρίζονται οι χρήστες στο Twitter που ακολουθούν ο ένας τον άλλον αμοιβαία.

8.1.5 Statuses Count

Ο αριθμός αναρτήσεων του χρήστη. Το εν λόγω χαρακτηριστικό είναι ενδεικτικό του πόσο ενεργός είναι ο χρήστης.

8.1.6 Favourites Count

Ο αριθμός των αναρτήσεων που οι χρήστες έχουν χαρακτηρίσει ως “αγαπημένες”. Η πρόσβαση σε αυτές τις δημοσιεύσεις χαρακτηρίζουν σημαντικά το προφίλ του χρήστη.

8.1.7 Lists Count

Ο χρήστης δημιουργεί λίστες προκειμένου να οργανώνει και να ιεραρχεί τις δημοσιεύσεις (tweets) που συναντά στο Twitter. Επιλέγοντας λίστες που απαρτίζονται από λογαριασμούς, ο χρήστης ενημερώνεται και δεν χάνει ανάρτηση από τον συγκεκριμένο λογαριασμό.
Επιλέγοντας να δημιουργήσει λίστες βάσει θέματος, ο χρήστης συγκεντρώνει tweets ομοίου θέματος.

8.1.8 Created at

Υπολογίζεται ο αριθμός των μηνών από τότε που ο χρήστης έφτιαξε τον λογαριασμό του.

8.1.9 Number of words in the description

Συνήθως, χρήστες που παραθέτουν εκτενέστερη και πιο λεπτομερή περιγραφή του εαυτού τους, στο πεδίο description του Twitter είναι έγκυροι χρήστες.

8.1.10 Number of words in the screen name

Αριθμός χαρακτήρων στο username του χρήστη.

Επεξεργαζόμαστε τα παραπάνω χαρακτηριστικά για τους χρήστες που έκαναν retweet την προς ανίχνευση είδηση. Ο τόσο μεγάλος αριθμός χαρακτηριστικών δυσχεραίνει τους υπολογιστικούς χρόνους καθώς και την ικανότητα του μοντέλου να γενικεύει, λόγω της υψηλής μεταβλητότητας. Επιπροσθέτως, η τελική αναπαράσταση του μοντέλου μας καταλήγει να είναι πιο απλή. Για τον λόγο αυτό, μέσω των Random Forest και Linear Regressor αποφαινόμαστε για το πόσο σημαντικό είναι το κάθε χαρακτηριστικό για την κατηγοριοποίηση της εκάστοτε είδησης. Η στατιστική τιμή που χρησιμοποιούμε για να αποφανθούμε εάν τα χαρακτηριστικά αυτά επηρεάζουν την έκβαση του αποτελέσματος είναι η *p-value*.

8.2 p-value

Σε κάθε εργασία μοντελοποίησης, υποθέτουμε κάποια συσχέτιση μεταξύ των χαρακτηριστικών και του αποτελέσματος. Η *p-value* εκφράζει την πιθανότητα εξαγωγής αποτελεσμάτων κατά το λιγότερο ασυσχέτιστων από τα αποτελέσματα που πραγματικά προέκυψαν, κάνοντας την υπόθεση ότι η μηδενική υπόθεση είναι ορθή.
Η μηδενική υπόθεση διατυπώνεται ως εξής: Δεν υπάρχει συσχέτιση ανάμεσα στα χαρακτηριστικά και τα παρατηρούμενα αποτελέσματα. Συνεπώς, η *p-value* χρησιμοποιείται προκειμένου να κρίνουμε εάν η κενή υπόθεση ισχύει η απορρίπτεται για το εκάστοτε χαρακτηριστικό. Οι *p-value* τιμές εκφράζονται ως δεκαδικοί, αλλά είναι πιο εύληπτες μετατρέποντας τους σε ποσοστά επί τοις εκατό. Για παράδειγμα, για *p-value* = 0.0294, έχουμε 2.94% πιθανότητα τα αποτελέσματά μας να είναι τυχαία. Από την άλλη, για *p-value* = 91% τα αποτελέσματά μας είναι κατά 91% τυχαία.

Σημαντικότητα της p-value

- Η τιμή *p-value* < 0.05, είναι στατιστικά σημαντική. Εκφράζει σημαντική απόδειξη έναντι της μηδενικής υπόθεσης, καθώς η πιθανότητα η τελευταία να ισχύει είναι μικρότερη του 5%.
- Η τιμή *p-value* ≥ 0.05 δεν είναι στατιστικά σημαντική και εκφράζει μεγάλη πιθανότητα να ισχύει η μηδενική υπόθεση.

8.3 Random Forest

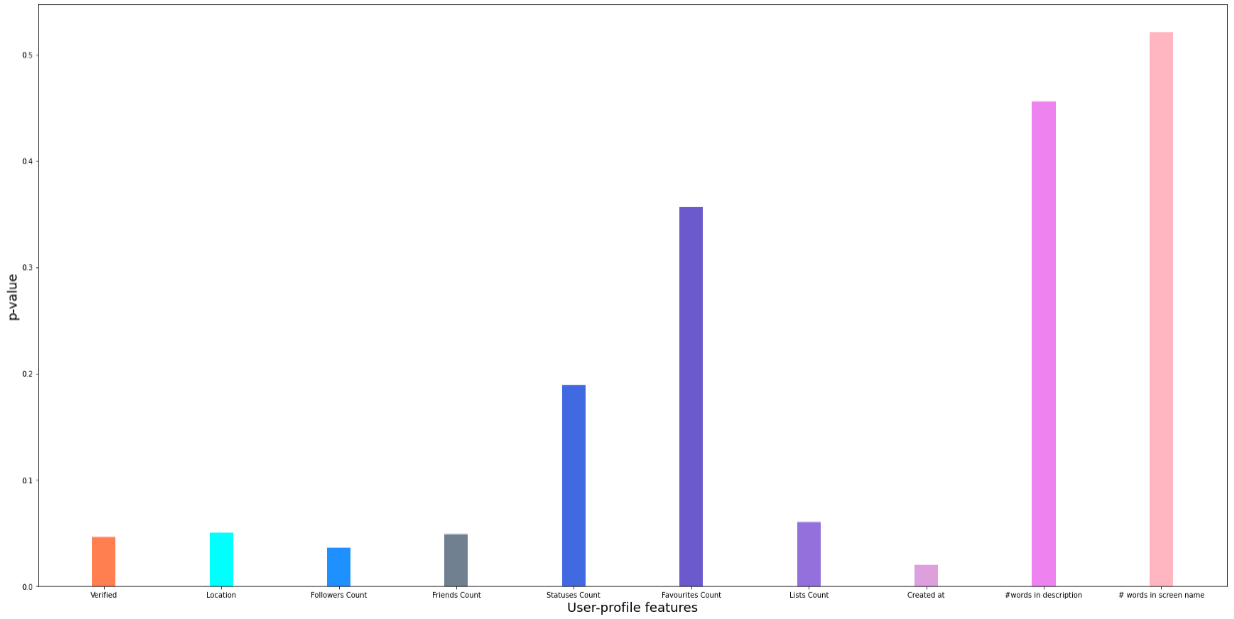
Ο μετα ταξινομητής *Random Forest* προσαρμόζει έναν αριθμό δέντρων αποφάσεων σε διάφορα δείγματα του συνόλου δεδομένων. Έπειτα, υπολογίζει τον μέσο όρο αυτών προκειμένου να ενισχύσει την εγκυρότητα της πρόβλεψης και να ελέγξει την υπερπροσαρμογή.
Τα *random forests* αποτελούνται από 400 με 1200 δέντρα αποφάσεων, κάθε ένα από αυτά κατασκευάζεται από μία τυχαία εξόρυξη παρατηρήσεων από το σύνολο δεδομένων και από μια εξόρυξη τυχαίων χαρακτηριστικών.
Δεν βλέπουν όλα τα δέντρα ολόκληρο το σύνολο των χαρακτηριστικών ή ολόκληρο το σύνολο των χαρακτηριστικών. Το γεγονός αυτό εγγυάται ότι τα δέντρα είναι ασυσχέτιστα και λιγότερο πιθανόν να οδηγηθούν στο *overfitting*.
Κάθε δέντρο είναι μια ακολουθία από "ναι-όχι" ερωτήσεις βασιζόμενες σε ένα μοναδικό χαρακτηριστικό ή σε ένα συνδυασμό αυτών. Σε κάθε ερώτηση, σε κάθε κόμβο, το δέντρο χωρίζει το σύνολο δεδομένων σε δύο *buckets,* κάθε ένας από αυτά περιέχει παρατηρήσεις που είναι πιο σχετικές μεταξύ τους και διαφέρουν περισσότερο με τις παρατηρήσεις του άλλου *bucket*. Γι'αυτό, η σημαντικότητα του κάθε χαρακτηριστικού εξάγεται από το πόσο "*pure*" είναι το κάθε "*bucket*".
Ο ταξινομητής αυτός χρησιμοποιείται συχνά καθώς η στρατηγική του που βασίζεται σε δομή δέντρου αξιολογεί βάσει πόσο βελτιώνεται το *purity* ενός κόμβου. Κόμβοι με την υψηλότερη

μείωση στο *impurity* παρουσιάζονται στην αρχή των δέντρων, ενώ οι κόμβοι με τη μικρότερη μείωση παρουσιάζονται στο τέλος των δέντρων. Αφαιρώντας το δέντρο κάτω από έναν συγκεκριμένο κόμβο και μετά δημιουργούμε ένα υποσύνολο με τα πιο σημαντικά χαρακτηριστικά.

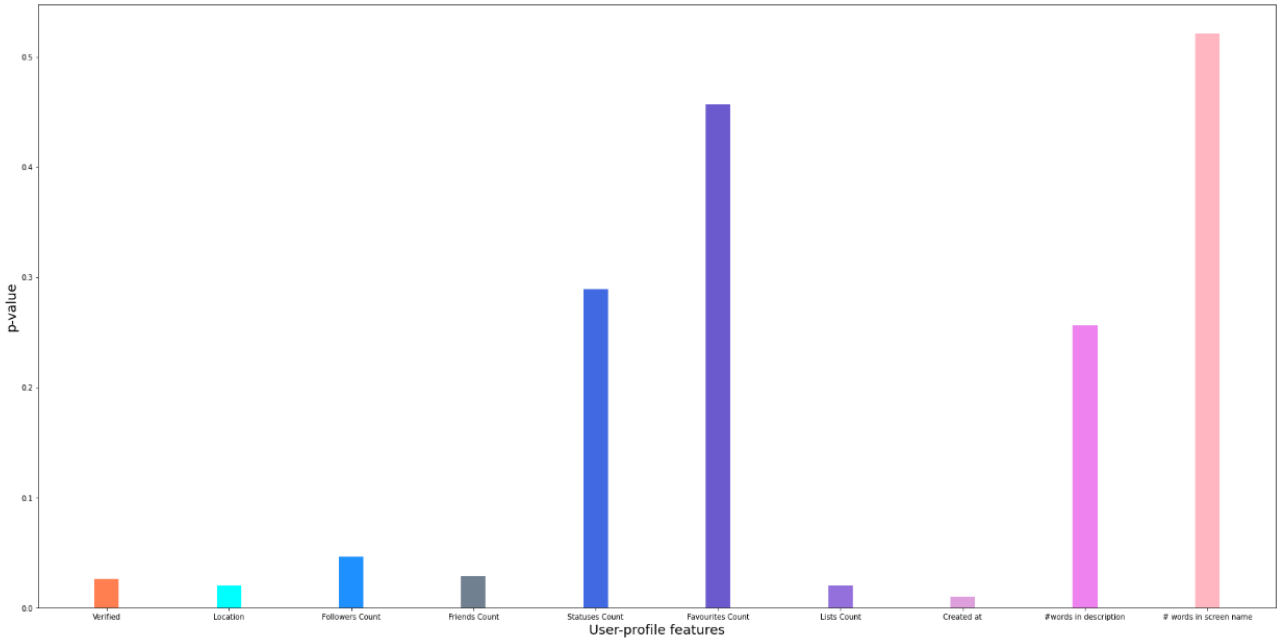
8.4 Linear Regression

Η ανάλυση *Linear Regression* χρησιμοποιείται για να προβλέψει την αξία μιας μεταβλητής βάσει της αξίας μιας άλλης μεταβλητής. Η μεταβλητή που θέλουμε να προβλέψουμε λέγεται εξαρτώμενη μεταβλητή, ενώ η μεταβλητή που χρησιμοποιούμε για να προβλέψουμε την τιμή της άλλης μεταβλητής λέγεται ανεξάρτητη μεταβλητή.
Αυτή η μορφή της ανάλυσης υπολογίζει τους συντελεστές της γραμμικής εξίσωσης, που περιλαμβάνει μία ή περισσότερες ανεξάρτητες μεταβλητές, οι οποίες προβλέπουν πιο αποδοτικά την τιμή της εξαρτώμενης μεταβλητής.
Με το *Linear Regression* προκύπτει μια ευθεία γραμμή ή μια επιφάνεια που ελαχιστοποιεί τις αποκλίσεις μεταξύ των προβλεπόμενων και των πραγματικών τιμών εξόδου.
Το *linear regression* μπορεί να βασιστεί στη μέθοδο "ελαχίστων τετραγώνων", προκειμένου να ευρεθεί η καλύτερη ευθεία για ένα σύνολο ζευγών δεδομένων.

Τα αποτελέσματα που λαμβάνουμε είναι τα ακόλουθα:



Εικόνα 27 *Linear Regressor p-values* για τα *user-profile features* "verified, location, followers count, friends count, statuses count, favourites count, lists count, created at, # words in description, # words in screen name" από αριστερά προς τα δεξιά.



Εικόνα 28 *Random forest p-values* για τα *user-profile features* "verified, location, followers count, friends count, statuses count, favourites count, lists count, created at, # words in description, # words in screen name" από αριστερά προς τα δεξιά.

.

Συνεπώς, καταλήγουμε στα χαρακτηριστικά με τα οποία θα πορευτούμε για την περαιτέρω ανάλυση. Αυτά είναι τα Created at, Verified, Location, Friends Count, Followers Count, Lists Count, τα οποία σημειώνουν *p-value* < *0.05* και για τους δύο ταξινομητές. Επιλέγουμε να απορρίψουμε τα υπόλοιπα χαρακτηριστικά, λόγω μικρού συντελεστή βαρύτητας στην τελική έκβαση, παραχωρώντας έτσι στο μοντέλο τη δυνατότητα να γενικεύει και να επιτυγχάνει υψηλότερο accuracy. Έχοντας εξάγει τα βασικά χαρακτηριστικά, η διαδικασία που ακολουθούμε είναι η ακόλουθη.

Αρχικά, δεδομένης μιας είδησης/ανάρτησης, προσδιορίζονται οι χρήστες οι οποίοι έχουν κοινοποιήσει την εν λόγω είδηση. Η παραπάνω ενέργεια καλείται retweet. Συγκεντρώνοντας τους χρήστες αυτούς, έχουμε πρόσβαση και στα βασικά χαρακτηριστικά τους (*user-related features*).

Για κάθε έναν από αυτούς, το αποθετήριο *FakeNewsNet μας παρέχει τα* Created at, Verified, Location, Friends Count, Followers Count, Lists Count *user-related features*. Για την βελτίωση της απόδοσης εφαρμόζουμε την διαδικασία ελέγχου, εάν οι λογαριασμοί αντιστοιχούν σε πραγματικούς χρήστες ή bots, μέσω της διαδικασίας *BotOrNot*. Για κάθε είδηση που εξετάζεται δημιουργούμε τον γράφο διάδοσής της. Συλλέγουμε όλους τους χρήστες που τον απαρτίζουν σε μία λίστα. Τη λίστα αυτή εισαγουμε ως είσοδο σε μια έτοιμη συνάρτηση από το Github [71].

Με τον τρόπο αυτό γνωρίζουμε ποιοι χρήστες είναι bots.

Προσθέτουμε, λοιπόν, και ένα επιπλέον χαρακτηριστικό σε κάθε χρήστη, που εκφράζει εάν πρόκειται ή όχι για bot. Το χαρακτηριστικό αυτό προστίθεται στα αρχεία χαρακτηριστικών που παρέχονται από το dataset. Τα εφτά *user-related features* καθώς και το *BotOrNot*, κωδικοποιούνται με μοναδικό τρόπο μέσω των *word2vec embeddings*. Συνεπώς, έχουμε ένα διάνυσμα 8 συνολικά χαρακτηριστικών, το οποίο για να διευκολύνει τις πράξεις μας στη συνέχεια, επιλέγουμε να έχει embeddings διάστασης 300 (8x300). Επιλέγεται και ένας επιπλέον τρόπος επεξεργασίας των χαρακτηριστικών του χρήστη. Στην περίπτωση αυτή εξάγουμε τα embeddings μέσω του μοντέλου BERT της Google, ορίζοντας και σε αυτά διάσταση 300.

Με τον ίδιο τρόπο που κωδικοποιήθηκαν τα *user-related* και *BotOrNot features* (word2vec ή BERT embeddings) κωδικοποιείται και η αρχική είδηση/ανάρτηση.
Εν συνεχεία, κατασκευάζεται και ο γράφος διάδοσης σε μορφή δέντρου διάδοσης. Αυτός έχει ως κόμβο ρίζα (root node) του δέντρου την είδηση, ενώ οι υπόλοιποι κόμβοι αναπαριστούν τους χρήστες που κοινοποίησαν την είδηση αυτή. Καθώς η βάση δεδομένων επιλέχθηκε να είναι το Twitter, οι λοιποί κόμβοι του δέντρου αποτελούν τους χρήστες που έκαναν retweet την είδηση.
Χάρη στο GCN το εμφύτευμα που προέρχεται από την αρχική είδηση (*news textual embedding*) καθώς και το εμφύτευμα των χαρακτηριστικών του χρήστη (user-related embedding), που προκύπτει ως διάνυσμα των επτά χαρακτηριστικών (Created at, Verified, Location, Friends Count, Followers Count, Lists Count) και του *BotOrNot*, μπορούν να ληφθούν ως χαρακτηριστικά κόμβων. Δεδομένου του γράφου διάδοσης της είδησης, το GCN προκειμένου να διαμορφώσει το embedding ενός κόμβου, συγκεντρώνει τα χαρακτηριστικά των γειτονικών του κόμβων.Έπειτα, εφαρμόζεται μια *mean pooling readout* συνάρτηση σε όλα τα embeddings των κόμβων, προκειμένου να προκύψει το embedding του συνολικού γράφου διάδοσης. Έτσι, προκύπτει το τελικό graph embedding. Εφόσον το περιεχόμενο των ειδήσεων περιέχει σαφή δείγματα όσον αφορά την αξιοπιστία των ειδήσεων, κρίνεται αναγκαία η ένωση του αρχικού news textual embedding της είδησης και του τελικού graph embedding και έτσι συντίθεται το ολοκληρωμένο news embedding που περιέχει πλέον όλη την πληροφορία. Το ενωμένο πλέον news embedding οδηγείται σε έναν Multi-layer Perceptron (MLP) δύο επιπέδων, με δύο νευρώνες εξόδου, όπου η μία δίνει έξοδο εάν η είδηση είναι αληθής και η έτερη δίνει έξοδο εάν η είδηση είναι ψευδής. Το μοντέλο εκπαιδεύεται χρησιμοποιώντας binary cross-entropy, ενώ η συνάρτηση κόστους ανανεώνεται με SGD.
Το μοντέλο GCN που χρησιμοποιήθηκε για τον γράφο διάδοσης, είναι αυτό που έχει αναλυθεί εκτενώς παραπάνω.

8.5 Δίκτυο Διάδοσης στο Twitter

Το δίκτυο διάδοσης της είδησης στο Twitter περιγράφεται ως ακολούθως.

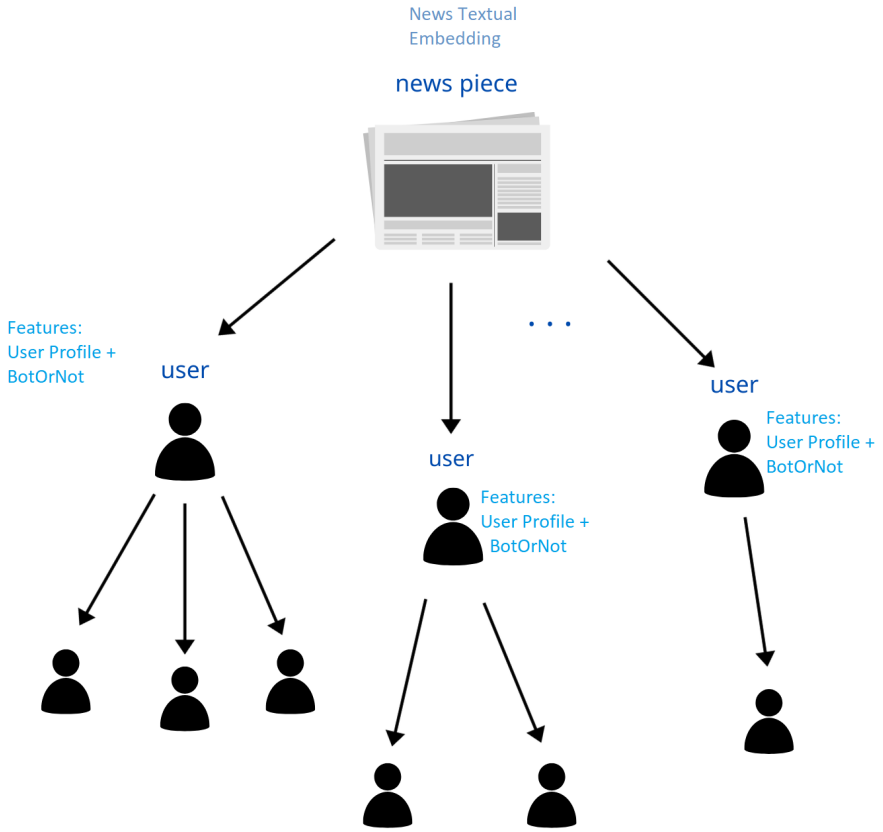
Έστω v_1 η προς ανίχνευση είδηση που εντοπίστηκε στο Twitter και $\{ v_2 , v_3 ,... v_n \}$ είναι η λίστα των χρηστών που έκαναν retweet την v_1, ταξινομημένη με χρονική σειρά. Το μονοπάτι διάδοσης της είδησης προσδιορίζεται σύμφωνα με τους παρακάτω κανόνες:

1. Για κάθε λογαριασμό v_i , εάν ο v_i κάνει retweet την ίδια είδηση, αργότερα από τουλάχιστον ένα από τους λογαριασμούς $\{v_2 , v_3 ,... v_n\}$ τους οποίους μάλιστα ακολουθεί, η διάδοση της είδησης υπολογίζεται από τον λογαριασμό v_j ο οποίος έκανε το πιο ύστερο χρονικά retweet. Κατά συνέπεια, η ανάρτηση του v_i απέχει χρονικά λιγότερο από αυτήν του v_j, συγκριτικά με τους υπόλοιπους χρήστες που είχαν αναρτήσει την ίδια είδηση πριν τον v_i. Ο λόγος που επιλέγεται ο χρήστης με την παλαιότερη, αλλά πιο κοντινή στον v_i ημερομηνία retweet έναντι των υπολοίπων, εντοπίζεται στο ότι η εφαρμογή του Twitter παρουσιάζει τις αναρτήσεις με χρονική σειρά. Πρώτα παρουσιάζονται οι νεότερες και έπειτα οι παλαιότερες αναρτήσεις. Συνεπώς ο χρήστης v_i είναι πιθανότερο να επηρεάστηκε από τον χρήστη v_j , ο οποίος είχε κάνει retweet την είδηση πιο ύστερα από τους υπόλοιπους χρήστες. Κατά συνέπεια, στον γράφο διάδοσης της είδησης, δημιουργείται η ακμή από τον κόμβο v_j στον κόμβο v_i.

2. Εάν ο λογαριασμός v_i δεν ακολουθεί κανέναν εκ των $\{v_2 , v_3 ,... v_n\}$, καθώς ούτε και τον λογαριασμό που ανήρτησε την είδηση v_1 τότε γίνεται η σύμβαση η διάδοση της είδησης να υπολογίζεται από τους λογαριασμούς με τους περισσότερους ακολούθους. Έχει διαπιστωθεί

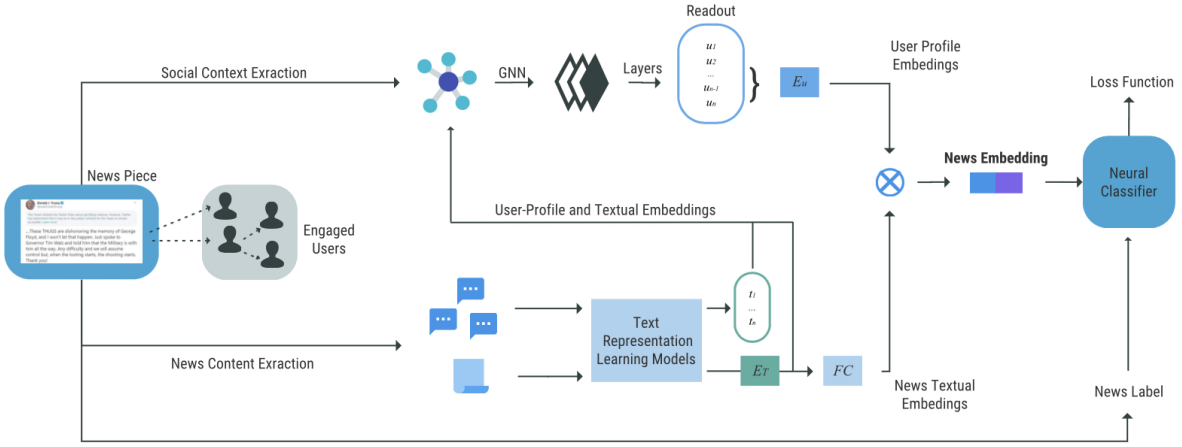
πως περιεχόμενο λογαριασμών με υψηλό αριθμό ακολούθων προτείνεται συχνά στους χρήστες. Κατά συνέπεια, είναι πιθανό κάποιος χρήστης να έρθει σε επαφή με tweet ατόμου με πολλούς ακολούθους και κατ'επέκταση να το επαναρτήσει (*retweet*) και ο ίδιος.

Με βάση τους δύο αυτούς κανόνες, πραγματοποιείται η κατασκευή των γράφων διάδοσης των ειδήσεων στο Twitter. Η εικόνα των γράφων αυτών παρουσιάζεται ακολούθως



Εικόνα 29 Γράφος διάδοσης που προκύπτει από την προς εξέταση είδηση. Όπως αποδίδεται είναι κατευθυνόμενος αποδίδοντας την ροή διάδοσης της είδησης από χρήστη σε χρήστη και μάλιστα ομογενής, αφού οι ακμές εκφράζουν το *retweet* της είδησης από τον έναν χρήστη στον άλλον. Το συνολικό κοινωνικό δίκτυο με όλες τις ειδήσεις αποτελείται από πολλούς αυτόνομους γράφους τέτοιας μορφής. όπως φαίνεται ο κόμβος κεφαλή, η αρχική είδηση, περιέχει τα περιέχει τα χαρακτηριστικά του κειμένου της είδησης, ενώ οι υπόλοιποι κόμβοι που αποτελούν τους χρήστες που έκαναν retweet την είδηση, όπως περιγράφεται στην παράγραφο 8.5, περιέχουν τα user profile καθώς και τα BotOrNot χαρακτηριστικά.

Παρουσιάζεται παρακάτω μια εικόνα που αποδίδει αποτελεσματικότερα όσα περιγράφονται παραπάνω.



Εικόνα 30 User-Related Fake News Detection. Δεδομένης μιας είδησης/ανάρτησης (News Piece) και των χρηστών που την έχουν αναρτήσει, δημιουργείται ο γράφος διάδοσης ειδήσεων με εμφύτευμα την αρχική είδηση και τα οχτώ χαρακτηριστικά που αφορούν τον χρήστη. Τέλος, τα αρχικά embeddings της είδησης και αυτά που προέκυψαν έπειτα από τις συνελίξεις του νευρωνικού δικτύου ενώνονται, ως το τελικό συνολικό embedding. Αυτό οδηγείται στον νευρωνικό ταξινομητή, προκειμένου να γίνει η πρόβλεψη για την είδηση.

Περιγραφή Διαδικασίας Ανίχνευσης *fake news* βάσει *User-Related features*:

Αρχικά, αποδίδονται τα πρώτα βήματα ως η προεργασία που είναι αναγκαία για να προχωρήσουμε στην υλοποίηση. Τα βήματα αυτά είναι κοινά και για τις τρεις μεθόδους ανίχνευσης ψευδών ειδήσεων (*User-Related, Topic-Related, User Preference*) σε κάθε περίπτωση, φυσικά, παραθέτουμε τα αντίστοιχα χαρακτηριστικά.

Προεργασία:

1. Μέσω του *Twitter API*, που εκδώσαμε, έχουμε πρόσβαση στο σύνολο των προς αξιολόγηση ειδήσεων καθώς και σε όλους τους χρήστες που τις δημοσίευσαν, με την διαδικασία *retweet*. Με αυτόν τον τρόπο, διαμορφώνεται ο συνολικός γράφος του δικτύου. Η κάθε είδηση ξεχωριστά, συνδυαστικά με τους χρήστες που την δημοσίευσαν, συνθέτουν τον γράφο διάδοσης που αποτελεί αυτόνομο γράφο του συνολικού δικτύου. Ο γράφος του δικτύου αποδίδεται ως σύνολο αυτόνομων γράφων. Οι γράφοι αυτοί προσδιορίζονται από ένα σύνολο ακμών σε ένα txt αρχείο. Το αρχείο αυτό έχει τη μορφή που αποδίδεται στην παρακάτω εικόνα:

```
1    0, 1
2    0, 2
3    0, 3
4    0, 4
5    0, 5
6    0, 6
7    0, 7
8    0, 8
9    0, 9
10   0, 10
      .
      .
      .
52   0, 52
53   0, 53
54   53, 54
55   53, 55
56   54, 56
57   57, 58
58   57, 59
59   57, 60
60   57, 61
      .
      .
      .
```

Εικόνα 31 Με τον τρόπο αυτό περιγράφονται οι υπόγραφοι διάδοσης του συνολικού δικτύου. Στον κάθε γράφο πρώτα παρουσιάζονται οι ακμές της κεφαλής με τους υπόλοιπους κόμβους. Στην περίπτωση αυτή κεφαλή είναι ο κόμβος 0 που ενώνεται με τους κόμβους 1, 2, … , 10, … 52, 53. Στον ίδιο γράφο, ο κόμβος 53 ενώνεται με τους κόμβους 54 και 55 και ο κόμβος 54 με τον 56. Ο κόμβος 57 αποτελεί κεφαλή νέου απογράφου και ενώνεται με τους κόμβους 58, …, 61, … κ.ο.κ.

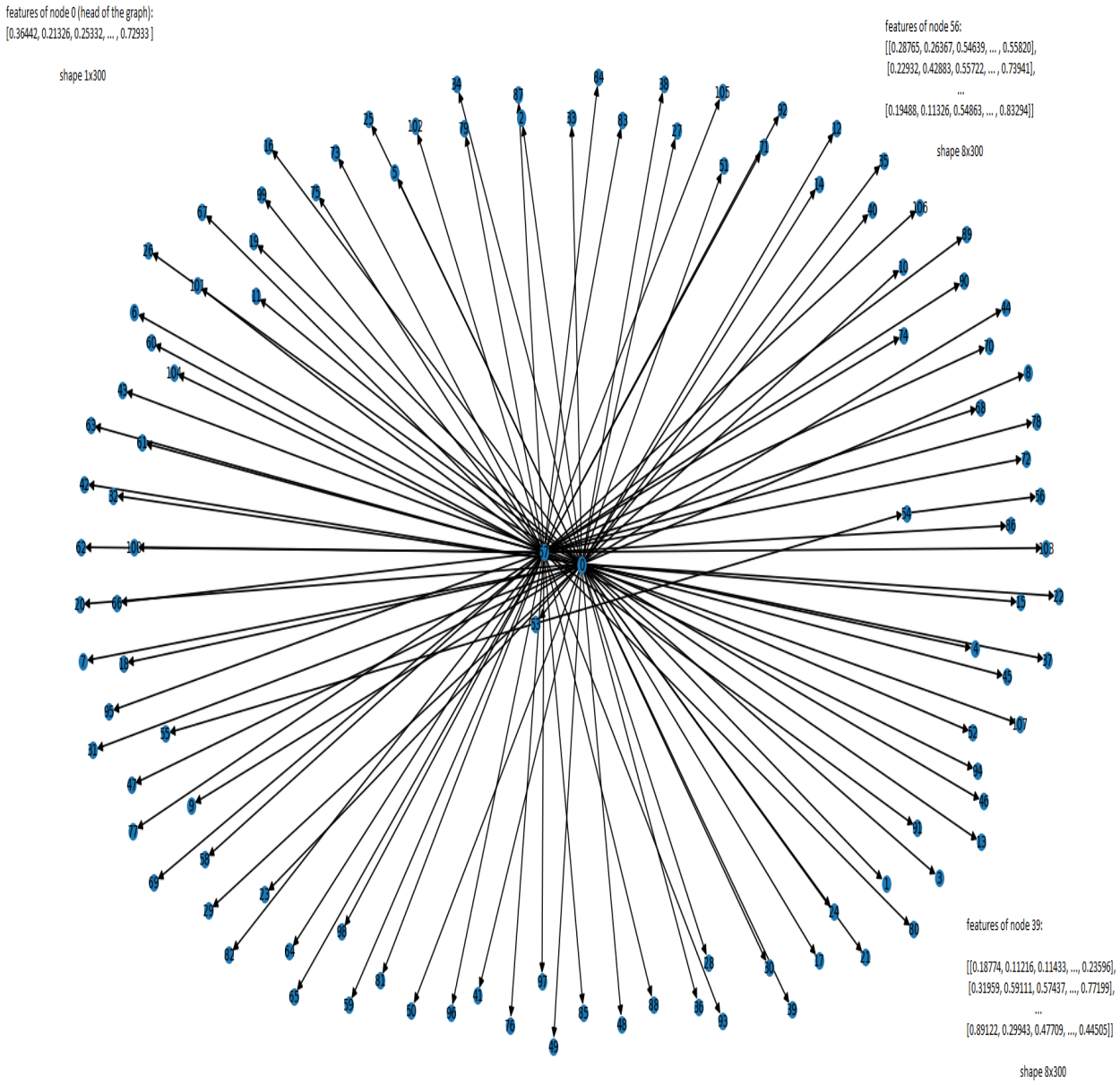
2. Και πάλι χάρη στο *Twitter API* είναι επιτρεπτή η πρόσβαση στα *User-Related features*.

3. Επιλογή των σημαντικότερων *User-Related features* μέσω των *Random Forest* και *Linear Regression*.

4. Εξαγωγή των διανυσμάτων χαρακτηριστικών με τα μοντέλα *word2vec* και *BERT* τόσο για τα *User-Related features* όσο και για την αρχική είδηση καθώς και το BotOrNot χαρακτηριστικό για κάθε χρήστη. Τα συνολικά χαρακτηριστικά αποθηκεύονται σε ένα αρχείο στο οποίο δίνεται κατάλληλο όνομα ώστε να προσδιορίζεται το *dataset (Gossipcop* ή *Politifact)* και η μέθοδος εξαγωγής των χαρακτηριστικών (*BERT* ή *word2vec*). Συγκεκριμένα, για την κεφαλή του κάθε γράφου, εξάγονται τα χαρακτηριστικά της είδησης και για τον κάθε κόμβο-χρήστη του γράφου τα *User-Related features* με την ίδια πάντα μέθοδο (*BERT* ή *word2vec*).

5. Από το dataset παρέχονται το αρχείο με το ποιοι κόμβοι ανήκουν σε κάθε γράφο, καθώς και το αρχείο με τις ετικέτες για κάθε γράφο διάδοσης των ειδήσεων (*fake/real*).

Συμπληρωματικά με την εικόνα 31, η οποία περιγράφει έναν γράφο διάδοσης από το συνολικό δίκτυο, για καλύτερη εποπτεία παραθέτουμε τον γράφο του συγκεκριμένου δικτύου μαζί με τα χαρακτηριστικά ορισμένων κόμβων.



Εικόνα 32 Ο γράφος διάδοσης της εικόνας 31. Κεφαλή είναι ο κόμβος 0, ο οποίος βρίσκεται κεντρικά στο γράφο. Το διάνυσμα χαρακτηριστικών του είναι [0.36442, 021326, 0.25332, ... , 0.72933] διάστασης 1x300. Ο κόμβος αυτός ενώνεται με τους κόμβους 1, 2, … , 10, … 52, 53. Στον ίδιο γράφο, ο κόμβος 53 ενώνεται με τους κόμβους 54 και 55 και ο κόμβος 54 με τον 56. Ο κόμβος 57 αποτελεί κεφαλή νέου απογράφου και ενώνεται με τους κόμβους 58, …, 61, … κ.ο.κ.
Δειγματικά παραθέτουμε και τα διανύσματα χαρακτηριστικών των κόμβων-χρηστών 56 και 39 που έχουν διάσταση 8x300 (7 *user-related features* και *BotOrNot*). Αυτά είναι τα

[[0.28765, 0.26367, 0.54639, ... , 0.55820],
[0.22932, 0.42883, 0.55722, ... , 0.73941],
...
[0.19488, 0.11326, 0.54863, ... , 0.83294]]

και

[[0.18774, 0.11216, 0.11433, ... , 0.23594],
[0.31959, 0.59111, 0.57437, ... , 0.77199],
...
[0.89122, 0.29943, 0.47709, ... , 0.44505]]

για τους κόμβους 56 και 39 αντίστοιχα.

Στη συνέχεια κατασκευάζουμε την κλάση FNNDataset (προκύπτει από τις λέξεις FakeNewsNet Dataset) η οποία θα επιστρέφει τα συνολικά δεδομένα.
Συγκεκριμένα, η κλάση αυτή καλεί την συνάρτηση *read_graph_data* η οποία παίρνει ως όρισμα το είδος των χαρακτηριστικών (*word2vec* ή *BERT*) και το μονοπάτι όπου βρίσκονται τα αρχεία με τα χαρακτηριστικά των χρηστών, των κόμβων κάθε γράφου, των ετικετών κάθε γράφου και το txt που προσδιορίζει τους κόμβους κάθε ακμής. Η συνάρτηση αυτή χρησιμοποιώντας την βιβλιοθήκη PyTorch της python χάρη στην κλάση "Data" κατασκευάζει ένα object, με τη μορφή dictionary, το οποίο περιγράφει έναν ομογενή γράφο. Τα χαρακτηριστικά που επιστρέφει η κλάση αυτή είναι τα

- data.x: Πίνακας χαρακτηριστικών των κόμβων, με διαστάσεις [αριθμός κόμβων, αριθμός χαρακτηριστικών κόμβων]

- data.edge_index: Πίνακας που προσδιορίζει τους κόμβους της κάθε ακμής, με διαστάσεις [2, αριθμός ακμών]

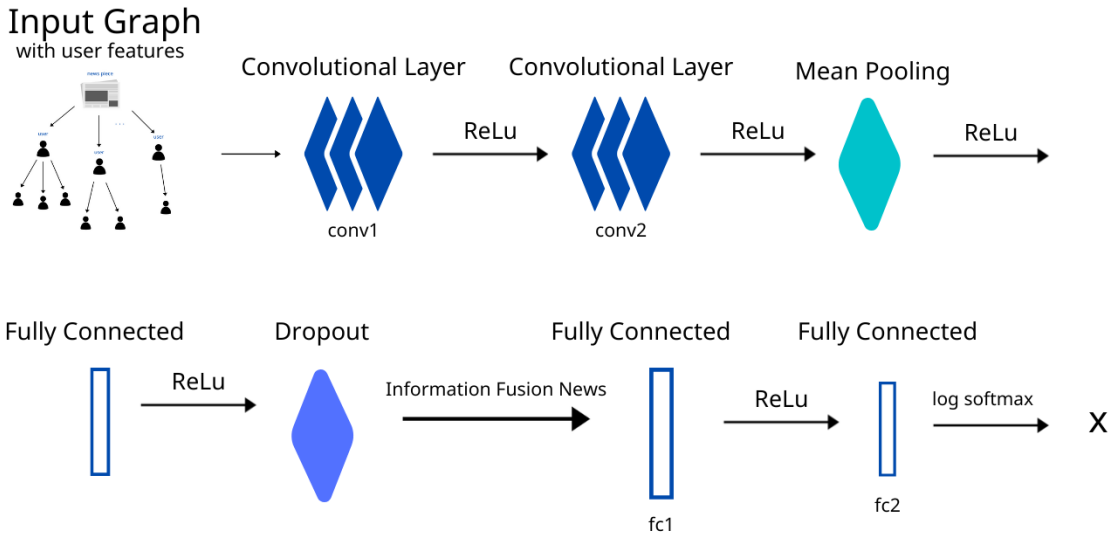
- data.y: Πίνακας με τις ετικέτες για κάθε γράφο, με διαστάσεις [αριθμός γράφων].

Έπειτα, καλείται η συνάρτηση *split* η οποία λαμβάνει ως είσοδο το dictionary data των συνολικών δεδομένων για τα χαρακτηριστικά των κόμβων (data['x']), τους κόμβους κάθε ακμής (data['edge_index']) και τις ετικέτες του κάθε γράφου (data['y']) καθώς και το αρχείο με τους κόμβους κάθε γράφου και επιστρέφει τα συνολικά δεδομένα καθώς και ένα dictionary σε πιο βολική μορφή ομαδοποιημένα κατά παρτίδες (*batches*) που χρειάζονται στην συνέχεια. Η δημιουργία των παρτίδων αυτών είναι κομβική για την ταχύτερη εκπαίδευση του νευρωνικού δικτύου και η διαδικασία ομαδοποίησης των γράφων καλείται *mini-batching*. Συγκεκριμένα, αντί να επεξεργαζόμαστε κάθε έναν γράφο ξεχωριστά και σειριακά, με το mini-batching ομαδοποιούμε ένα σύνολο γράφων σε μια ενοποιημένη αναπαράσταση, χάρη στην οποία κάλλιστα επιτυγχάνεται παράλληλη επεξεργασία των γράφων. Στον τομέα της επεξεργασίας εικόνας και βίντεο, η παραπάνω διαδικασία επιτυγχάνεται επιβάλλοντας τις ίδιες διαστάσεις σε κάθε δείγμα, με τεχνικές *padding* ή *rescaling*. Τα δείγματα αυτά του mini-batching ομαδοποιούνται σε μια επιπλέον διάσταση, της οποίας το μήκος ισούται με τον αριθμό των δειγμάτων που ομαδοποιήθηκαν. Ο αριθμός αυτός καλείται *batch_size.*
Εφόσον οι γράφοι, ως πιο γενικές δομές, μπορούν αποθηκεύσουν ακαθόριστο αριθμό κόμβων και ακμών, οι τεχνικές *padding* και *rescaling* δεν είναι αποδοτικές υπολογιστικά. Η βιβλιοθήκη της PyTorch επιτρέπει την παράλληλη επεξεργασία στα δείγματα του mini-batching.

Οι πίνακες γειτνίασης, A, στοιβάζονται διαγώνια, δημιουργώντας έναν τεράστιο γράφο ο οποίος αποτελείται από πολλούς απομονωμένους γράφους. Τα διανύσματα χαρακτηριστικών καθώς και οι ετικέτα κάθε γράφου του mini-batch περιέχονται στα διανύσματα X και Y αντίστοιχα.

Προχωράμε στην κατασκευή του μοντέλου του Νευρωνικού Δικτύου στη μορφή Γράφου, μέσω της κλάσης Net κατά τον τρόπο που περιγράψαμε στην παράγραφο 5.8. Το GCN αυτό περιλαμβάνει δύο συνελικτικά επίπεδα *conv1*, *conv2* διαστάσεων (αριθμός_χαρακτηριστικών, 2*αριθμός_ακμών) και (2*αριθμός_ακμών, 2*αριθμός_ακμών) αντιστοίχως και τρία fully connected γραμμικά επίπεδα (αριθμός_χαρακτηριστικών, αριθμός_ακμών), (2*αριθμός_ακμών, αριθμός_ακμών), (αριθμός_ακμών, αριθμός_κλάσεων).

Στη συνέχεια μέσω της συνάρτησης *forward* της κλάσης Net περνάμε τα δεδομένα μέσα από το νευρωνικό δίκτυο. Το σχήμα που περιγράφει την παραπάνω διαδικασία αποδίδεται παρακάτω.



Εικόνα 33 Απεικόνιση του νευρωνικού μας δικτύου Τα χαρακτηριστικά των χρηστών (user profile και BotOrNot) εισάγονται ως είσοδος σε δύο συνελικτικά επίπεδα, τα οποία ακολουθούνται από τη μη γραμμική συνάρτηση ReLu. Έπειτα, προκύπτει το συνολικό διάνυσμα των κόμβων-χρηστών του γράφου, user profile embedding, ως μέσος όρος των διανυσμάτων χαρακτηριστικών των κόμβων-χρηστών. Διέρχεται από τη μη γραμμική συνάρτηση ReLu, από ένα Fully Connected επίπεδο και ξανά από τη ReLu. Έπειτα, το επίπεδο dropout αφαιρεί κάποιους νευρώνες από το δίκτυο προκειμένου να επιτευχθεί καλύτερη γενίκευση. Εν συνεχεία, σημειώνεται το information fusion, η ένωση δηλαδή του συνολικού user profile embedding με το news textual embedding. Η ένωση αυτή θα αποτελέσει το συνολικό embedding του γράφου. Το ενοποιημένο πλέον διάνυσμα διέρχεται από ένα Fully Connected Layer, Relu, ακόμη ένα Fully Connected Layer. Η συνάρτηση *log softmax* στην έξοδο του νευρωνικού δικτύου δίνει την πιθανότητα τα δεδομένα εισόδου να είναι ψευδή και την πιθανότητα να είναι αληθή. Με τον τρόπο αυτό, προκύπτει το τελικό διάνυσμα χαρακτηριστικών x.

Το συνολικό διάνυσμα χαρακτηριστικών του γράφου συνενώνεται με το διάνυσμα χαρακτηριστικών του χρήστη, όπως φαίνεται στην εικόνα 30.
Τέλος προβαίνουμε στην εκπαίδευση του νευρωνικού δικτύου καθώς και στην αξιολόγηση αυτού.

Τα παραπάνω μπορούν να γραφούν με την μορφή ψευδοκώδικα ως ακολούθως:

Ψευδοκώδικας:

```
@model.no_grad()  #δεν ανανεώνονται οι παράγωγοι κατά την αξιολόγηση του μοντέλου
def compute_test(test_data):  #συνάρτηση υπολογισμού accuracy στο test set
        model.eval()     #απενεργοποίηση των dropout layers κατά την αξιολόγηση του μοντέλου
        loss_test = 0
        Για κάθε mini_batch στο σύνολο των δεδομένων test_data:
                prediction = model(mini_batch) #περνάμε ως είσοδο στο μοντέλο το σύνολο
                                               #των δεδομένων του mini-batch και λαμβάνουμε
                                               #τις προβλέψεις του
                #υπολογισμός σφάλματος με την λογαριθμική cross entropy συνάρτηση,  binary
                #που επιστρέφει μηδέν ή ένα.
                loss_test += log_cross_entropy(prediction, y)
                accuracy += accuracy_score(prediction, y)*batch_size
        return accuracy/data_size
```

Ο χρήστης θέτει τις παραμέτρους batch_size, learning rate, εποχές, χαρακτηριστικά (BERT word2vec) από το πληκτρολόγιο

```
dataset = FNNDataset() #Κλάση που αναλύθηκε παραπάνω, επιστρέφει τα συνολικά δεδομένα σε mini-batches
ορισμός των train_loader, val_loader, test_loader  #20%, 10% και 70% των συνολικών δεδομένων λόγω
                                                   #ημι-επιβλεπόμενης μάθησης
model = Net()   #Κλάση που επιστρέφει το GCN αναλύθηκε παραπάνω
optimizer = SGD #χρησιμοποιούμε optimizer για την ανανέωση των βαρών

#Ακολουθεί εκπαίδευση του μοντέλου

model.train() #ενεργοποίηση των dropout layers κατά την αξιολόγηση του μοντέλου
Για κάθε εποχή στο σύνολο των εποχών:
        loss_train = 0
        Για κάθε mini_batch στο σύνολο των δεδομένων train_data
                optimizer.zero_grad() # οι παράγωγοι των tensors τίθενται ίσοι με μηδέν
                prediction = model(mini_batch)
                loss = log_cross_entropy(prediction, y)
                loss.backward()
                optimizer.step()
                loss_train += loss
                accuracy_train += accuracy_score(prediction, y)*batch_size
                compute_test(val_loader) #υπολογισμός accuracy του validation set σύμφωνα με την παραπάνω
```

```
                                                    #συνάρτηση
        print(accuracy_train /data_size)
compute_test(test_loader )
```

Διευκρινιστικά σχόλια για τη συνάρτηση train:
Κατά την αρχικοποίηση του optimizer το μοντέλο "γνωρίζει" με ακρίβεια ποιες παραμέτρους πρέπει να ανανεώσει. Συνεπώς, με την εντολή *loss.backward*() οι παράμετροι αυτές ανανεώνονται και αποθηκεύονται από μόνες του στους τένσορες. Αυτό συμβαίνει χάρη στα χαρακτηριστικά "*grad*" και "*requires_grad*" των χαρακτηριστικών εισόδου. Αφού υπολογιστούν οι παράγωγοι για όλους τους τένσορες του μοντέλου, με την εντολή *optimizer_step*(), ο optimizer περνά από όλες τις παραμέτρους (τένσορες) και χρησιμοποιώντας το χαρακτηριστικό τους "grad" ανανεώνει τις τιμές τους.

Παρακάτω, παρατίθενται τα αποτελέσματα της μεθόδου αυτής, συνοδευόμενα από τις παραμέτρους στις οποίες εκπαιδεύτηκε και αξιολογήθηκε το μοντέλο.

	Politifact	Gossipcop
Embeddings	ACC | F1	ACC | F1
word2vev	78.03 | 76.62	79.22 | 79.21
BERT	79.25 | 78.25	80.77 | 80.19

Πίνακας 4 Αποτελέσματα για τις μετρικές *accuracy* και *F1-score* στο κάθε *dataset*, για τα *embeddings word2vec* και *BERT* χρησιμοποιώντας το GCN μοντέλο με *user-profile* χαρακτηριστικά.

Για την καλύτερη κατανόηση των μετρικών που χρησιμοποιήθηκαν αναφέρουμε τα παρακάτω.

Για το δυαδικό πρόβλημα ταξινόμησης ψευδών ειδήσεων, έστω ότι το μοντέλο επιστρέφει "0" εάν η είδηση είναι αληθής και "1" εάν η είδηση είναι ψευδής. Οι παράμετροι TP (True Positive), FN (False Negative), FP (False Positive), και TN (True Negative) ορίζονται ως εξής:

TP = όσα παραδείγματα ανήκουν στην κλάση (εξόδου) 0 και ταξινομήθηκαν στην 0
FN = όσα παραδείγματα ανήκουν στην κλάση (εξόδου) 0, αλλά ταξινομήθηκαν στην 1
FP = όσα παραδείγματα ανήκουν στην κλάση (εξόδου) 1, αλλά ταξινομήθηκαν στην 0
TN = όσα παραδείγματα ανήκουν στην κλάση (εξόδου) 1 και ταξινομήθηκαν στην 1

Συνεπώς, οι μετρικές accuracy, precision, recall και F1-score ορίζονται ως ακολούθως [72]:

$$accuracy = \frac{TP+TN}{TP+FN+TN+FP},\ precision = \frac{TP}{TP+FP},\ recall = \frac{TP}{FN+TP},$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision+recall} \quad [73]$$

Παρατηρούμε ότι τα εμφυτεύματα που επιφέρουν τα καλύτερα αποτελέσματα και στα δύο datasets είναι τα BERT. Πολλές δημοσιεύσεις σε εργασίες Επεξεργασίας Φυσικής Γλώσσας έχουν αποδείξει την υπεροχή των συγκεκριμένων διανυσμάτων έναντι των word2vec [48].

Οι υπερ-παράμετροι του μοντέλου, οι οποίες έχουν προκύψει έπειτα από δοκιμές και fine tuning, είναι οι ακόλουθες:

Dataset	model	feature	epoch	learning rate	embedding size	batch number
Politifact	GCN	word2vec	70	0.001	128	64
Politifact	GCN	BERT	100	0.001	128	128
Gossipcop	GCN	word2vec	60	0.01	128	128
Gossipcop	GCN	BERT	60	0.001	128	128

Πίνακας 5 Υπερ-παράμετροι του GCN μοντέλου για τα *user-profile features*

Ο όρος fine tuning αποδίδει την ακριβή προσαρμογή του μοντέλου στις παραπάνω υπερπαραμέτρους, προκειμένου να αποδίδει βέλτιστα στο test set.
Έχοντας αναλύσει την πρώτη μέθοδο ανίχνευσης ψευδών ειδήσεων ακολουθεί η δεύτερη τεχνική, η οποία επικεντρώνεται στα χαρακτηριστικά που αφορούν την ίδια την είδηση (*topic related features*).

8.6 Topic-Related Fake News Detection με τη χρήση GCNs

Στην δεύτερη μέθοδο ανίχνευσης ψευδών ειδήσεων στο κοινωνικό δίκτυο του Twitter, εστιάζουμε σε συγκεκριμένα χαρακτηριστικά του κειμένου. Συγκεκριμένα, εντοπίζονται το είδος της ανάρτησης, η ημερομηνία δημοσίευσής της και εάν γίνεται κάποια αναφορά στην πηγή από την οποία αντλήθηκε η είδηση. Αναλύουμε, διαδοχικά, τους λόγους που μας ώθησαν στην επιλογή των τριών αυτών χαρακτηριστικών.

8.6.1 Είδος Ανάρτησης

Η βάση δεδομένων μας παραθέτει εάν η συγκεκριμένη είδηση είναι πολιτικό, ιατρικό, εκλογικό, περιβαλλοντολογικό και όποιο άλλο ζήτημα. Η συγκεκριμένη πληροφορία παίζει σημαντικότατο ρόλο στην κατηγοριοποίηση των ειδήσεων ως ψευδών ή αληθών. Συγκεκριμένα, έχει παρατηρηθεί ότι σε ειδήσεις πολιτικού επιπέδου οι χρήστες εκδηλώνουν υψηλό ενδιαφέρον (αυξημένος αριθμός likes, list counts, retweets) [52]. Συνεπώς, το θέμα αυτό δίνει ισχυρό πάτημα σε κακοπροαίρετους χρήστες να αναρτήσουν ανακριβές περιεχόμενο, να φανατίσουν και να παραπλανήσουν τους πιο ανυποψίαστους χρήστες. Επιπλέον, δεν είναι λίγες φορές παρατηρημένο, πως αναγνωρίσιμα πολιτικά και μη πρόσωπα επιλέγουν να δημοσιεύσουν αναληθές καθώς και ασαφές, διφορούμενο περιεχόμενο, στρέφοντας τη κοινή γνώμη με το μέρος τους.

Παράλληλα, με το ξέσπασμα και την έξαρση του κορονοϊού COVID-19 πολλοί ήταν οι χρήστες που δημοσίευσαν πως τα εμβόλια ήταν ακατάλληλα, επικίνδυνα, περιττά για την ίαση του ιού και επικίνδυνα για τον οργανισμό μελλοντικά. Συνεπώς, παρατηρείται πως το είδος της ανάρτησης είναι σημαντικό χαρακτηριστικό για τον χαρακτηρισμό μιας είδησης ως ψευδούς ή αληθούς, εφόσον συγκεκριμένα θέματα παρουσιάζουν μεγαλύτερη σύνδεση με ψευδές περιεχόμενο, από ότι άλλα θέματα.

8.6.2 Ημερομηνία Δημοσίευσης Ανάρτησης

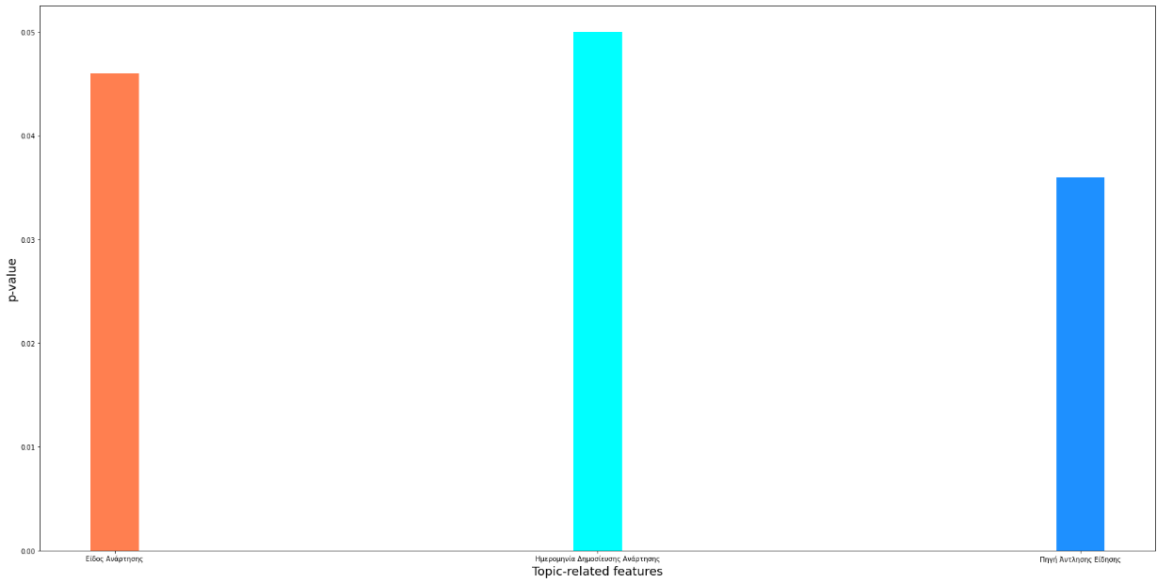
Ο λόγος που δίνεται βάση στο συγκεκριμένο χαρακτηριστικό αποδίδεται στην ακόλουθη παρατήρηση. Σε ημερομηνίες κοντά στις εκλογές, ο κόσμος τείνει να αναρτά και να διαδίδει ψευδές περιεχόμενο, σε βαθμό πολύ υψηλότερο από ότι σε μια οποιαδήποτε άλλη περίοδο [70]. Επιπλέον, όπως αναλύθηκε και παραπάνω στην περίοδο εμφάνισης του κορονοϊού και έπειτα, πέρα από έξαρση κρουσμάτων, παρατηρήθηκε και έξαρση διάδοση ψευδών ειδήσεων. Συνεπώς, η ημερομηνία ανάρτησης της είδησης παίζει σημαντικό ρόλο στην κατηγοριοποίηση των ειδήσεων. Επειδή το dataset αφορά χρήστες της Αμερικής, ελέγχουμε το πόσο απέχει χρονικά η κάθε ανάρτηση από τις εκλογές της Αμερικής των τελευταίων ετών. Συγκεκριμένα, επιλέχθηκαν οι ακόλουθες τρεις ημερομηνίες εκλογών:

6 Νοεμβρίου 2012
8 Νοεμβρίου 2016
3 Νοεμβρίου 2020

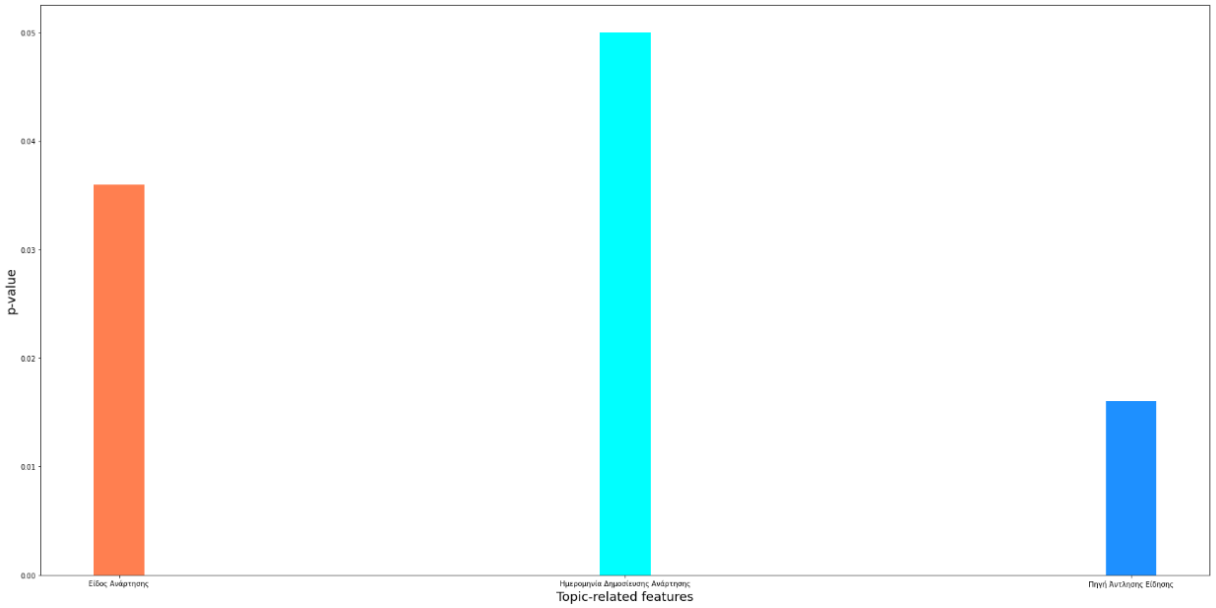
Επιπροσθέτως, επιλέχθηκε και η περίοδος εκδήλωσης του κορονοϊού και των πρώτων κρουσμάτων στις ΗΠΑ, ως το διάστημα με την μεγαλύτερη άγνοια για τον ιό και διάδοση ψευδών γεγονότων. Η χρονική περίοδος αυτή είναι ο Ιανουάριος του 2020.

8.6.3 Πηγή Άντλησης Είδησης

Την πληροφορία αυτή μας την παραχωρεί άμεσα η επιλεγμένη βάση δεδομένων, FakeNewsNet [53]. Πολλά sites, αρθρογράφοι, δημοσιογράφοι, είναι προσανατολισμένοι σε μια πολιτική κατεύθυνση και επιλέγουν να παρουσιάζουν τις ειδήσεις υπό το πρίσμα αυτό. Για το λόγο αυτό, οι ειδήσεις παύουν να είναι αυστηρή παράθεση και περιγραφή πραγματικών γεγονότων. Στη σύγχρονη εποχή , η ενημέρωση εμπεριέχει τις αντιλήψεις, σκέψεις, συναισθήματα και προβληματισμούς των ατόμων που την κοινοποιούν. Από τον βαθμό του σκόπιμου έως και του επιτηδευμένου, είναι αναμενόμενο η πηγή να διαδίδει ανακρίβειες, λαθεμένα γεγονότα και ψέματα.
Παράλληλα, υπάρχουν συγκεκριμένα sites των οποίων ο σκοπός είναι σατιρικός, διαδίδουν ψευδή γεγονότα για χιουμοριστικούς σκοπούς καθώς επίσης εξ ολοκλήρου ψευδείς ειδήσεις, με σκοπό την παραπλάνηση των χρηστών και το στρέψιμό τους σε μια συγκεκριμένη κατεύθυνση.



Εικόνα 34 *Linear Regressor p-values* για τα *topic-related features* "Είδος Ανάρτησης, Ημερομηνία Δημοσίευσης Ανάλυσης, Πηγή Άντλησης Είδησης" από αριστερά προς τα δεξιά.



Εικόνα 35 *Random forest p-values* για τα *topic-related features* "Είδος Ανάρτησης, Ημερομηνία Δημοσίευσης Ανάλυσης, Πηγή Άντλησης Είδησης" από αριστερά προς τα δεξιά.

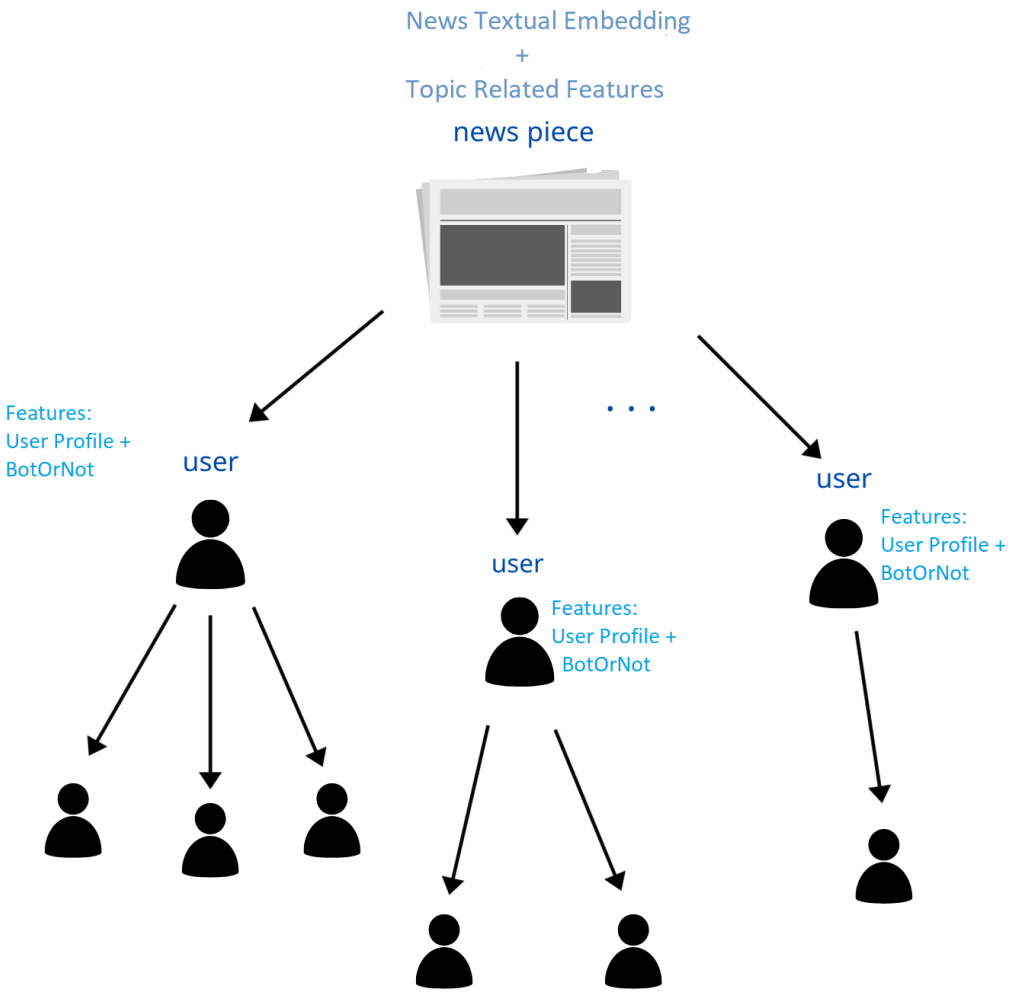
Επεξεργαζόμαστε τα παραπάνω χαρακτηριστικά από την αρχική είδηση. Κατασκευάζουμε αρχικά τα word2vec embeddings των χαρακτηριστικών και έπειτα τα συνενώνουμε με τα word2vec embeddings της είδησης. Η ίδια διαδικασία ακολουθείται στη συνέχεια και για τα embeddings του μοντέλου BERT της Google. Ο γράφος διάδοσης κατασκευάζεται ξανά με root node την αρχική είδηση/ανάρτηση και με υπόλοιπους κόμβους τους χρήστες που έκαναν retweet την εν λόγω είδηση. Η διαδικασία που ακολουθείται στην συνέχεια είναι παρόμοια με αυτή που ακολουθήσαμε παραπάνω.
Συγκεντρώνοντας τους χρήστες που έκαναν retweet την προς εξέταση είδηση, έχουμε πρόσβαση και στα βασικά χαρακτηριστικά τους (*user-related features*).

Συνεπώς,για κάθε χρήστη, έχουμε ένα διάνυσμα 8 συνολικά χαρακτηριστικών (Created at, Verified, Location, Friends Count, Followers Count, Lists Count και *BotOrNot*), το οποίο για να διευκολύνει τις πράξεις μας στη συνέχεια, επιλέγουμε να έχει embeddings διάστασης 300 (8x300). Με τον ίδιο τρόπο που κωδικοποιήθηκαν τα *user-related* και *BotOrNot features* (word2vec ή BERT embeddings) κωδικοποιείται και η αρχική είδηση/ανάρτηση, συνδυαστικά σε αυτήν την περίπτωση με τα χαρακτηριστικά "είδος ανάρτησης", "ημερομηνία δημοσίευσης της ανάρτησης", "πηγή άντλησης είδησης". Τα τρία αυτά πρόσθετα χαρακτηριστικά αναμένουμε να μας αποδώσουν επιπλέον πληροφορία και ακρίβεια.

Εν συνεχεία, κατασκευάζεται και ο γράφος διάδοσης σε μορφή δέντρου διάδοσης. Αυτός έχει ως κόμβο ρίζα (root node) του δέντρου την είδηση, ενώ οι υπόλοιποι κόμβοι αναπαριστούν τους χρήστες που κοινοποίησαν την είδηση αυτή. Καθώς η βάση δεδομένων επιλέχθηκε να είναι το Twitter, οι λοιποί κόμβοι του δέντρου αποτελούν τους χρήστες που έκαναν retweet την είδηση.

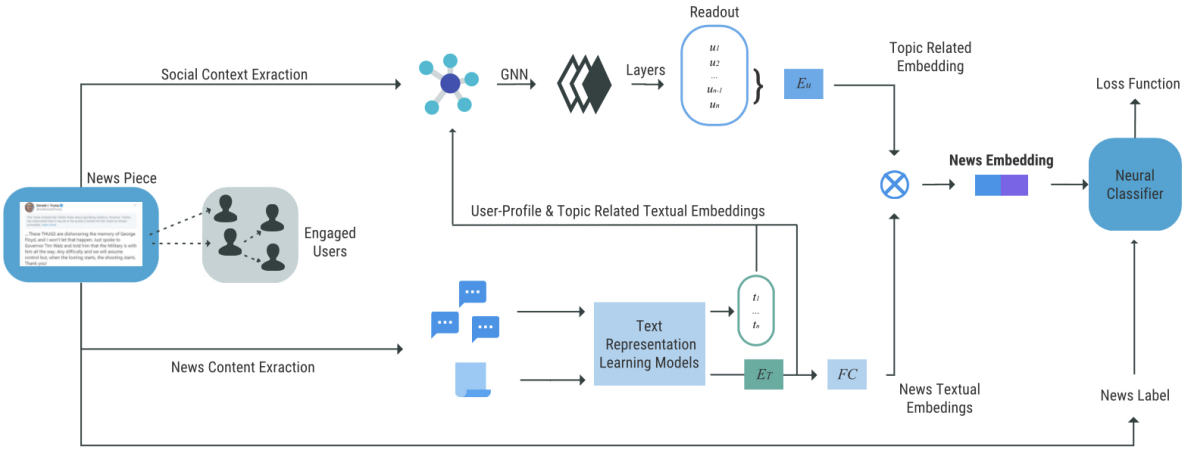
Χάρη στο GCN το εμφύτευμα που προέρχεται από την αρχική είδηση και τα τρία επιπλέον χαρακτηριστικά (*news textual embedding*) καθώς και το εμφύτευμα των χαρακτηριστικών του χρήστη (user-related embedding), που προκύπτει ως διάνυσμα των επτά χαρακτηριστικών (Created at, Verified, Location, Friends Count, Followers Count, Lists Count) και του *BotOrNot*, μπορούν να ληφθούν ως χαρακτηριστικά κόμβων. Δεδομένου του γράφου διάδοσης της είδησης, το GCN προκειμένου να διαμορφώσει το embedding ενός κόμβου, συγκεντρώνει τα χαρακτηριστικά των γειτονικών του κόμβων. Έπειτα, εφαρμόζεται μια *mean pooling readout* συνάρτηση σε όλα τα embeddings των κόμβων, η οποία επιστρέφει ένα embedding ως τον μέσο όρο όλων των διανυσμάτων, προκειμένου να προκύψει το embedding που όταν ενωθεί με το textual embedding της είδησης θα προκύψει το embedding του συνολικού γράφου διάδοσης. Εφόσον το περιεχόμενο των ειδήσεων περιέχει σαφή δείγματα όσον αφορά την αξιοπιστία των ειδήσεων, κρίνεται αναγκαία η ένωση του αρχικού news textual embedding της είδησης και του τελικού graph embedding και έτσι συντίθεται το ολοκληρωμένο news embedding που περιέχει πλέον όλη την πληροφορία. Το ενωμένο πλέον news embedding οδηγείται σε έναν Multi-layer Perceptron (MLP) δύο επιπέδων, με δύο νευρώνες εξόδου, όπου η μία δίνει έξοδο εάν η είδηση είναι αληθής και η έτερη δίνει έξοδο εάν η είδηση είναι ψευδής. Το μοντέλο εκπαιδεύεται χρησιμοποιώντας binary cross-entropy, ενώ η συνάρτηση κόστους ανανεώνεται με SGD.

Το μοντέλο GCN που χρησιμοποιήθηκε για τον γράφο διάδοσης, είναι αυτό που έχει αναλυθεί εκτενώς παραπάνω.



Εικόνα 36 Γράφος διάδοσης που προκύπτει από την προς εξέταση είδηση. Όπως αποδίδεται είναι κατευθυνόμενος αποδίδοντας την ροή διάδοσης της είδησης από χρήστη σε χρήστη και μάλιστα ομογενής, αφού οι ακμές εκφράζουν το *retweet* της είδησης από τον έναν χρήστη στον άλλον. Το συνολικό κοινωνικό δίκτυο με όλες τις ειδήσεις αποτελείται από πολλούς αυτόνομους γράφους. Για το topic related πρόβλημα, προστίθεται στην είδηση και τα τρία επιλεγμένα topic χαρακτηριστικά.

Παρουσιάζεται παρακάτω μια εικόνα που αποδίδει αποτελεσματικότερα όσα περιγράφονται παραπάνω.



Εικόνα 37 Topic-Related Fake News Detection. Δεδομένης μιας είδησης/ανάρτησης (News Piece) και των χρηστών που την έχουν αναρτήσει, δημιουργείται ο γράφος διάδοσης ειδήσεων με εμφύτευμα την αρχική είδηση, τα χαρακτηριστικα "είδος ανάρτησης", "ημερομηνία δημοσίευσης της ανάρτησης", "πηγή άντλησης είδησης" και τα οχτώ χαρακτηριστικά που αφορούν τον χρήστη. Τέλος, τα αρχικά embeddings της είδησης και αυτά που προέκυψαν έπειτα από τις συνελίξεις του νευρωνικού δικτύου ενώνονται, ως το τελικό συνολικό embedding. Αυτό οδηγείται στον νευρωνικό ταξινομητή, προκειμένου να γίνει η πρόβλεψη για την είδηση.

Προεργασία:
Τα βήματα της προεργασίας είναι ακριβώς ίδια με τα προηγούμενα με τις τροποποιήσεις στα βήματα

3. Επιλογή των σημαντικότερων *Topic-Related features* μέσω των *Random Forest* και *Linear Regression*.

4. Εξαγωγή των διανυσμάτων χαρακτηριστικών με τα μοντέλα *word2vec* και *BERT* τόσο για τα *User-Related features* όσο και για την αρχική είδηση καθώς και τα *Topic-Related* features της είδησης. Τα συνολικά χαρακτηριστικά αποθηκεύονται σε ένα αρχείο στο οποίο δίνεται κατάλληλο όνομα ώστε να προσδιορίζεται το *dataset (Gossipcop* ή *Politifact)* και η μέθοδος εξαγωγής των χαρακτηριστικών (*BERT* ή *word2vec*). Συγκεκριμένα, για την κεφαλή του κάθε γράφου, εξάγονται τα χαρακτηριστικά της είδησης μαζί με τα *Topic-Related* και για τον κάθε κόμβο-χρήστη του γράφου τα *User-Related features* με την ίδια πάντα μέθοδο (*BERT* ή *word2vec*).

Επιπλέον, στην κλάση Net συνενώνονται τα Topic-Related και text features που αφορούν την προς εξέταση είδηση, τον κόμβο κεφαλή δηλαδή του γράφου, μαζί με τα συνολικά user-related features των λοιπών κόμβων του γράφου και έτσι συντίθεται το τελικό διάνυσμα χαρακτηριστικών του γράφου.

Κατά τα άλλα, ο ψευδοκώδικας δεν διαφέρει στο παραμικρό από αυτόν των User-Related Fake News Detection.

Ψευδοκώδικας:

```
@model.no_grad()  #δεν ανανεώνονται οι παράγωγοι κατά την αξιολόγηση του μοντέλου
def compute_test(test_data):  #συνάρτηση υπολογισμού accuracy στο test set
        model.eval()     #απενεργοποίηση των dropout layers κατά την αξιολόγηση του μοντέλου
        loss_test = 0
        Για κάθε mini_batch στο σύνολο των δεδομένων test_data:
                prediction = model(mini_batch) #περνάμε ως είσοδο στο μοντέλο το σύνολο
                                               #των δεδομένων του mini-batch και λαμβάνουμε
                                               #τις προβλέψεις του
                #υπολογισμός σφάλματος με την λογαριθμική cross entropy συνάρτηση,  binary
                #που επιστρέφει μηδέν ή ένα.
                loss_test += log_cross_entropy(prediction, y)
                accuracy += accuracy_score(prediction, y)*batch_size
        return accuracy/data_size
```

Ο χρήστης θέτει τις παραμέτρους batch_size, learning rate, εποχές, χαρακτηριστικά (BERT word2vec) από το πληκτρολόγιο

```
dataset = FNNDataset() #Κλάση που αναλύθηκε παραπάνω, επιστρέφει τα συνολικά δεδομένα σε mini-batches
ορισμός των train_loader, val_loader, test_loader  #20%, 10% και 70% των συνολικών δεδομένων λόγω
                                                   #ημι-επιβλεπόμενης μάθησης
model = Net()   #Κλάση που επιστρέφει το GCN αναλύθηκε παραπάνω
optimizer = SGD #χρησιμοποιούμε optimizer για την ανανέωση των βαρών

#Ακολουθεί εκπαίδευση του μοντέλου

model.train() #ενεργοποίηση των dropout layers κατά την αξιολόγηση του μοντέλου
Για κάθε εποχή στο σύνολο των εποχών:
        loss_train = 0
        Για κάθε mini_batch στο σύνολο των δεδομένων train_data
                optimizer.zero_grad() # οι παράγωγοι των tensors τίθενται ίσοι με μηδέν
                prediction = model(mini_batch)
                loss = log_cross_entropy(prediction, y)
                loss.backward()
                optimizer.step()
                loss_train += loss
                accuracy_train += accuracy_score(prediction, y)*batch_size
                compute_test(val_loader) #υπολογισμός accuracy σύμφωνα με την παραπάνω συνάρτηση
        print(accuracy_train /data_size)
compute_test(test_loader )
```

	Politifact	Gossipcop
Embeddings	ACC | F1	ACC | F1
word2vev	79.27 | 79.04	83.09 | 83.24
BERT	79.75 | 79.93	83.20 | 83.39

Πίνακας 6 Αποτελέσματα για τις μετρικές *accuracy* και *F1-score* στο κάθε *dataset*, για τα *embeddings word2vec* και *BERT* χρησιμοποιώντας το GCN μοντέλο με *topic-related* χαρακτηριστικά.

Οι επιδόσεις, αυτή τη φορά, είναι αυξημένες συγκριτικά με τις προηγούμενες, για τα *user-profile features*. Ο λόγος εντοπίζεται στο ότι αυξήθηκε ο αριθμός των χαρακτηριστικών εκπαίδευσης, με τρόπο τέτοιο ώστε το μοντέλο να σημειώνει βελτιωμένη απόδοση, δίχως να κινδυνεύει από *overfitting*. Πλέον διαθέτουμε τα *user-profile features* ενισχυμένα με τα *topic-related*. Τα εμφυτεύματα *BERT* εξακολουθούν να υπερνικούν τα *word2vev* σε επιδόσεις.

Οι παράμετροι του μοντέλου, έπειτα από *fine-tuning* είναι οι ακόλουθες:

dataset	model	feature	epoch	learning rate	embedding size	batch number
Politifact	GCN	word2vec	80	0.001	128	128
Politifact	GCN	BERT	80	0.001	128	64
Gossipcop	GCN	word2vec	60	0.01	128	64
Gossipcop	GCN	BERT	60	0.001	128	64

Πίνακας 7 Υπερ-παράμετροι του GCN μοντέλου για τα *topic-related features.*

Έχοντας αναλύσει διεξοδικά τις βασικές μεθόδους ανάλυσης ψευδών ειδήσεων, προχωράμε στην πιο σύνθετη, ρηξικέλευθη και αποδοτική τεχνικής ανίχνευσης ψευδών ειδήσεων, αυτή του UPFD Framework (*User Preference Fake news Detection*). Η μέθοδος αυτή δημοσιεύθηκε το 2021 και αποτελεί μια καινοτόμα λύση στην ανίχνευση ψευδών ειδήσεων στα κοινωνικά δίκτυα.

Εξέλιξη της μεθόδου αυτής αποτελεί η είσοδος διανυσμάτων ενισχυμένων χαρακτηριστικών στο νευρωνικό δίκτυο. Συγκεκριμένα, τα διανύσματα χαρακτηριστικών, πέρα από την ενδογενή πληροφορία του χρήστη βάσει των διακοσίων πιο πρόσφατων αναρτήσεών του (*endogenous preferences*), και από τα χαρακτηριστικά του κειμένου (*exogenous context*) περιέχουν επιπλέον

1. User-Related πληροφορία που αποτυπώνεται ως τα έξι *features* που αφορούν τον χρήστη (Created at, Verified, Location, Friends Count, Followers Count, Lists Count).

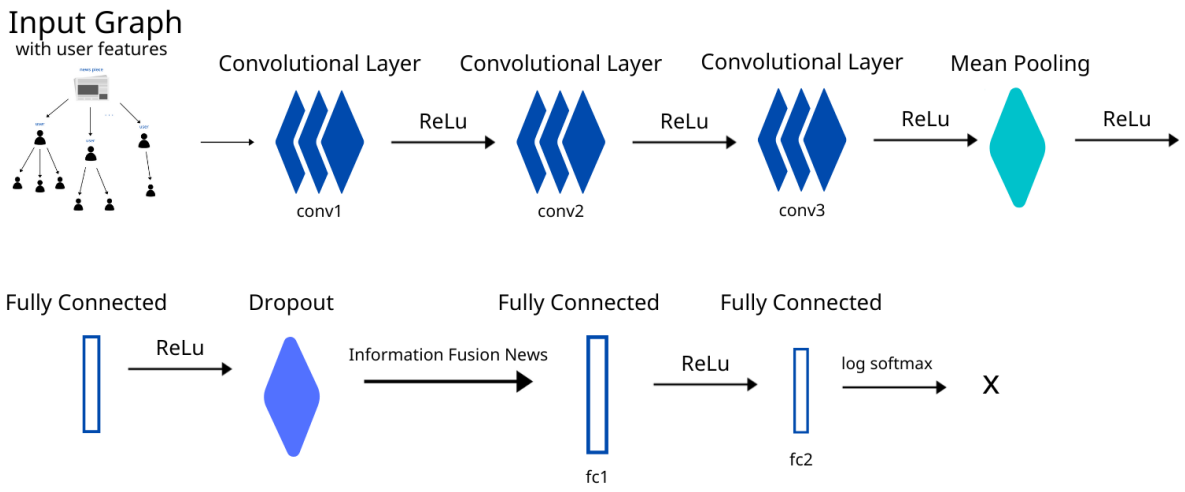
2. Πληροφορία για το εάν ο χρήστης είναι αυτοματοποιημένο λογισμικό ή όχι (*BotOrNot feature*).
3. Topic Related χαρακτηριστικά τα οποία αφορούν το κείμενο της αρχικής είδησης (“είδος ανάρτησης”, “ημερομηνία δημοσίευσης της ανάρτησης”, “πηγή άντλησης είδησης”).

Τα χαρακτηριστικά 1, 2 ομαδοποιούνται και διαμορφώνουν τα χαρακτηριστικά που αφορούν τον χρήστη, ενώ τα χαρακτηριστικά 3 που αφορούν την είδηση, συνενώνονται με τα *exogenous context* χαρακτηριστικά.

Επιπλέον, στο μοντέλο παρέχονται οι κατάλληλες παράμετροι ρυθμού μάθησης, αριθμού εποχών, batch size προκειμένου να προσαρμοστεί με ακρίβεια (*fine tuning*) και να αποδώσει όσο το δυνατόν καλύτερα.

Επίσης, το ίδιο το GCN μοντέλο έχει υποστεί τροποποίηση και εξελιξη με σκοπό τη βέλτιστη ανίχνευση ψευδών ειδήσεων. Προστίθεται σε αυτό ένα τρίτο συνελικτικό επίπεδο που παίρνει ως είσοδο την έξοδο του δεύτερου (conv3 = GCNConv). Επιπρόσθετα, έχουν επιλεχθεί συναρτήσεις σφάλματος και optimizers που αποδίδουν το βέλτιστο για την εν λόγω βάση δεδομένων.

Το σχήμα που περιγράφει το συνελικτικό νευρωνικό δίκτυο είναι το ακόλουθο:



Εικόνα 38 Στο νέο νευρωνικό δίκτυο προστέθηκε ένα τρίτο συνελικτικό επίπεδο, *conv3*.

8.7 User Preference Fake News Detection με τη χρήση GCNs τριών επιπέδων και ενισχυμένα χαρακτηριστικά ως είσοδο

Η παραπληροφόρηση και η διάδοση ψευδών πληροφοριών, μέσω των μέσων κοινωνικής δικτύωσης, αποτελεί φαινόμενο σε έξαρση και χρήζει αντιμετώπισης λόγω των επιβλαβών συνεπειών του τόσο στα άτομα όσο και στο σύνολο της κοινωνίας. Η πλειονότητα των αλγορίθμων ανίχνευσης “ψεύτικων” ειδήσεων (fake news), επικεντρώνεται στην ανάλυση του περιεχομένου ειδήσεων ή του περιβάλλοντος εξωγενούς πλαισίου (exogenous context) που περικλείει μια είδηση. Με αυτόν τον

τρόπο, αναγνωρίζονται τυχόν επιβλαβή νέα. Συνεπώς, ακολουθώντας το μονοπάτι αυτό, πληροφοριες όπως οι ενδογενείς προτιμήσεις του χρήστη, οι οποίες δύνανται να διαδραματίσουν καθοριστικό ρόλο στο εάν ο ίδιος αποφασίσει να διαρρεύσει ψευδείς ειδήσεις, αγνοούνται και το έργο της ανίχνευσης ψεύδους στερείται ακρίβειας.

Σύμφωνα με την επιστημονική θεωρία της επιβεβαίωσης προκατάληψης (*confirmation bias theory*), αποδεικνύεται πως ο εκάστοτε χρήστης είναι πιθανότερο να κοινοποιήσει ανακριβές περιεχόμενο ή μέρος αυτού, όταν το περιεχόμενο αυτό είναι σύμφωνο με τις προσωπικές του πεποιθήσεις, απόψεις και προτιμήσεις. Αυτό αποδίδεται στο ότι γεγονός ότι οι χρήστες του διαδικτύου προτιμούν να διαβάζουν και να αναρτούν δημοσιεύσεις με τις οποίες είναι σύμφωνοι, χωρίς απαραίτητα οι ίδιες να είναι και αληθείς. Κατά συνέπεια, το σύνολο των δημοσιεύσεων του κάθε χρήστη, χαρακτηρίζεται ως ένα είδος δημόσιου αντικατοπτρισμού των αρεσκειών και των αντιλήψεων του. Οι δημοσιεύσεις αυτές αποτελούν πλούσια πηγή άντλησης πληροφοριών, οι οποίες με ορθή διαχείριση μπορούν να βελτιστοποιήσουν την ανίχνευση του εάν ο εκάστοτε χρήστης διέδωσε αναληθές περιεχόμενο στον παγκόσμιο ιστό. Συνεπώς, στην εργασία αυτή, μελετάται το καινοτόμο πρόβλημα αξιοποίησης των προτιμήσεων του χρήστη, δια μέσω των δημοσιεύσεών του. Το εγχείρημα αυτό πραγματοποιείται με την χρήση της δομής *User Preference-aware Fake News Detection* χάρη στην οποία καταγράφονται συγχρόνως διάφορα σήματα, από τις προτιμήσεις του χρήστη, βάσει την μοντελοποίηση του κοινωνικού δικτύου με τη δομή γράφου (κοινοί φίλοι/ακόλουθοι του χρήστη) και βάσει του κοινού περιεχομένου των αναρτήσεών του με αυτά των ψευδών ειδήσεων.

Ανάμεσα στις διάφορες τεχνικές που έχουν αναπτυχθεί μέσα στα χρόνια, ο άμεσος έλεγχος της ορθότητας των ειδήσεων (*fact checking*), βάσει του ελέγχου της εγκυρότητας της πηγής πληροφόρησης, τη σχετικότητα του προσώπου που έκανε την ανάρτηση με το ζήτημα που πραγματεύεται, το εάν το url και το site ανάρτησης είναι έγκυρα ή ακόμα εάν η ημερομηνία ανάρτησης είναι κοντινή με την σημερινή, αποδεικνύεται μέθοδος που απαιτεί πολλές ώρες δουλειάς από τους ειδικούς του τομέα. Για το λόγο αυτό, οι σύγχρονοι επιστήμονες αναζητούν λύσεις στις υπολογιστικές μεθόδους της βαθιάς μάθησης και της εξόρυξης χαρακτηριστικών. Τα μέχρι στιγμής πιο καινοτόμα μοντέλα ανίχνευσης ψευδών ειδήσεων SAFE και FakeBERT χρησιμοποιούν συνελικτικά νευρωνικά δίκτυα (*TextCNN*) και BERT, αντίστοιχα, για την εξόρυξη χαρακτηριστικών κειμένου. Επιπροσθέτως, τα μοντέλα GCNFN και GNN-CL τα οποία βασίζονται σε συνελικτικά δίκτυα γράφων έχουν ως χαρακτηριστικό την κωδικοποίηση του μοτίβου της διάδοσης των νέων στα κοινωνικά δίκτυα μέσω των λογαριασμών των χρηστών. Ωστόσο, και αυτά τα σύγχρονα μοντέλα εστιάζουν στην μοντελοποίηση του περιεχομένου των ειδήσεων καθώς και στο εξωγενές πλαίσιο αλληλεπίδρασης του χρήστη, αγνοώντας για άλλη μια φορά τις εσωτερικές και προσωπικές προτιμήσεις του χρήστη.

Παραπάνω αναφέρθηκε πολλές φορές ο όρος ενδογενής προτίμηση του χρήστη (user endogenous preference), δίχως να δοθεί για αυτόν ένας ξεκάθαρος ορισμός. Προκειμένου, λοιπόν, να επιτευχθεί η αξιοποίηση και η μοντελοποίηση της ενδογενούς προτίμησης του κάθε χρήστη, χρησιμοποιούνται οι παλιές του ως αντιπροσωπευτικά των πεποιθήσεών του. Η παραπάνω μέθοδος έχει αποδειχθεί πως χρησιμεύει στον χαρακτηρισμό μιας ανάρτησης ενός χρήστη ως σάτιρα ακόμα και ως επίθεση σε κάποιο δημόσιο πρόσωπο, πέρα από την ανίχνευση ψευδών ειδήσεων. Συνεπώς, ορίζουμε τις ενδογενείς προτιμήσεις του χρήστη ως τις παλιές του αναρτήσεις, μέσω του λογαριασμού του στα social media.

Λαμβάνοντας υπόψιν τα πλεονεκτήματα και τα μειονεκτήματα που παρουσιάζει η κάθε μέθοδος, από αυτές που αναλύθηκαν παραπάνω, ως πιο αποδοτική λύση προβάλλει η χρήση του εργαλείου

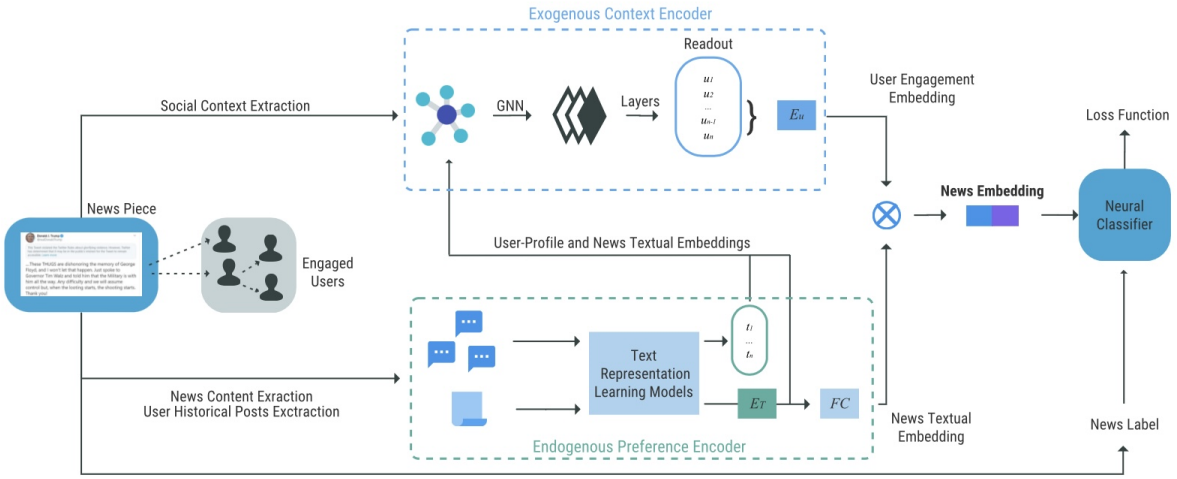
User Preference Fake News Detection (UPFD), χάρη στην οποία μοντελοποιούνται από κοινού τόσο η εσωτερικές προτιμήσεις (endogenous preference) του χρήστη όσο και το εξωτερικό πλαίσιο αυτού (exogenous context). Εκτενέστερα, το UPFD συντίθεται από τα ακόλουθα μέρη:

1. Για την μοντελοποίηση της ενδογενούς προτίμησης (endogenous preferences) κάθε χρήστη, κωδικοποιείται το περιεχόμενο μιας ανάρτησης (είδησης) ταυτόχρονα με παλαιότερες αναρτήσεις του χρήστη χρησιμοποιώντας διάφορες προσεγγίσεις μάθησης αναπαράστασης κειμένου.

2. Για την αποτύπωση του εξωγενούς πλαισίου (exogenous context), κατασκευάζεται ένας γράφος διάδοσης σε δομή δέντρου (tree-structured propagation graph) για κάθε είδηση (ανάρτηση) στα social media. Συγκεκριμένα, η είδηση αποτελεί τον κόμβο ρίζα (root node) του δέντρου, ενώ οι υπόλοιποι κόμβοι αναπαριστούν τους χρήστες που κοινοποίησαν την είδηση αυτή.

3.Τέλος, για την ενοποίηση της εξωγενούς και της ενδογενούς πληροφορίας, εξάγονται οι διανυσματικές αναπαραστάσεις που αφορούν τις ειδήσεις και τους εμπλεκόμενους με αυτές χρήστες και σύμφωνα με αυτά διαμορφώνεται ο γράφος του συνελικτικού δικτύου (Convolutional Graph Network) με κόμβους τους παραπάνω χρήστες. Έτσι ο γράφος αποκτά embeddings που αφορούν τους χρήστες που έχουν αναρτήσει την εκάστοτε είδηση. Τα embeddings αυτά σε συνδυασμό με τα embeddings του κειμένου της ανάρτησης, χρησιμοποιούνται για να εκπαιδεύσουν έναν νευρωνικό ταξινομητή ανίχνευσης ψευδών ειδήσεων.

Τα παραπάνω αποδίδονται σχηματικά στην ακόλουθη εικόνα:



Εικόνα 39: UPFD framework το οποίο λαμβάνει υπόψιν τις προτιμήσεις του χρήστη στην ανίχνευση ψευδών ειδήσεων. Δεδομένης μιας είδησης/ανάρτησης (News Piece) και των χρηστών που την έχουν αναρτήσει (Engaged Users), εξάγεται το εξωγενές πλαίσιο (exogenous context) ως ένας γράφος διάδοσης ειδήσεων, ενώ κωδικοποιείται η ενδογενής πληροφορία βάσει των παλαιών αναρτήσεων των χρηστών. Τέλος, η εξωγενής και η ενδογενής πληροφορία ενώνονται χρησιμοποιώντας για κωδικοποιητές νευρωνικά δίκτυα στην μορφή γράφων (GNN). Το τελικό embedding των ειδήσεων, το οποίο προκύπτει από την ένωση του embedding των σχετικών με την ανάρτηση χρηστών (user engagement embedding) και το κειμενικό embedding (textual embedding), οδηγείται στον νευρωνικό ταξινομητή, προκειμένου να γίνει η πρόβλεψη για την είδηση.

Συγκεντρωτικά, το framework User Preference-aware Fake News Detection μπορεί να περιγραφεί ως ακολούθως:

Αρχικά, δεδομένης μιας είδησης/ανάρτησης γίνεται αναζήτηση στις παλαιότερες αναρτήσεις των χρηστών, οι οποίοι σχετίζονται με την εν λόγω ανάρτηση. Πρόκειται, δηλαδή, για χρήστες που έχουν κοινοποιήσει την είδηση αυτή στα social media. Έμμεσα, λοιπόν, εξάγονται οι ενδογενείς προτιμήσεις των, σχετικών με την είδηση αυτή, χρηστών. Η εξαγωγή των ενδογενών προτιμήσεων επιτυγχάνεται με την κωδικοποίηση των παλαιότερων αναρτήσεων χρησιμοποιώντας τεχνικές μάθησης αναπαραστάσεων κειμένου (text representation learning techniques) όπως word2vec και BERT. Τα κειμενικά δεδομένα κωδικοποιούνται με τον ίδιο τρόπο.
Έπειτα, για την εξαγωγή του εξωγενούς πλαισίου του χρήστη, κατασκευάζεται ο γράφος διάδοσης που έχει ως κόμβο ρίζα (root node) του δέντρου την είδηση, ενώ οι υπόλοιποι κόμβοι αναπαριστούν τους χρήστες που κοινοποίησαν την είδηση αυτή. Καθώς η βάση δεδομένων επιλέχθηκε να είναι το Twitter, οι λοιποί κόμβοι του δέντρου αποτελούν τους χρήστες που έκαναν retweet την είδηση.
Τέλος, ακολουθεί μια σειρά βημάτων που εξασφαλίζουν την ένωση της ενδογενούς και της εξωγενούς πληροφορίας. Συγκεκριμένα, εξασφαλίζονται τα embeddings των χρηστών που είναι σχετικοί με την είδηση (user engagement embeddings) με την χρήση GNN ως κωδικοποιητές του γράφου, όπου τα embeddings των χρηστών και των ειδήσεων (user and news embeddings) που κωδικοποιήθηκαν από τον κωδικοποιητή κειμένου (text encoder), χρησιμοποιούνται ως χαρακτηριστικά των αντίστοιχων κόμβων στον γράφο διάδοσης των ειδήσεων. Τα τελικά news embeddings συντίθενται από την ένωση των user engagement embeddings και τα textual embeddings.

Εν συνεχεία, θα αναλυθεί η διαδικασία κωδικοποίησης της ενδογενούς προτίμησης του χρήστη.

8.7.1 Endogenous Preference Encoding

Η κωδικοποίηση της ενδογενούς προτίμησης του χρήστη αποτελεί διεργασία απλή, δεδομένων των πληροφοριών και των αναρτήσεών στα social media. Πρακτικά, όπως έχει προαναφερθεί, οι παλιές δημοσιεύσεις των χρηστών μοντελοποιούν την προσωπικότητα, τη λογική και τις απόψεις των χρηστών, αποδίδοντας, έμμεσα, την επιθυμητή ενδογενή προτίμηση (endogenous preference).
Για το λόγο αυτό, άλλωστε, επιλέχθηκε η εργασία στη βάση δεδομένων FakeNewsNet, η οποία περιέχει ειδήσεις στη μορφή αναρτήσεων καθώς και τους λογαριασμούς των σχετικών με την είδηση χρήστες που έχουν κάνει retweet την είδηση. Έπειτα, αιτούμενοι οι συγγραφείς του [52] για Twitter Developer Api, αποκτάται η πρόσβαση στις παλιές αναρτήσεις όλων των χρηστών που έκαναν retweet την κάθε είδηση του FakeNewsNet. Συνάγεται, άρα, ότι το σύνολο δεδομένων του Twitter αποτελεί τον τρόπο για την εξόρυξη της επιθυμητής ενδογενούς πληροφορίας των χρηστών.
Για την άντληση πλούσιου ενδογενούς περιεχομένου, συλλεγονται οι διακόσιες πιο πρόσφατες αναρτήσεις (tweets) για κάθε χρήστη, ενώ στο σύνολο γίνεται διαχείριση διακοσίων εκατομμυρίων αναρτήσεων. Στους χρήστες όπου η πρόσβαση δεν είναι εφικτή, λόγω διαγραφής των προφίλ τους, προστίθενται τυχαία διακόσιες αναρτήσεις από άλλους χρήστες. Δυστυχώς, δεν είναι αποτελεσματική η διαγραφή των μη προσβάσιμων χρηστών από το γράφο διάδοσης των ειδήσεων, καθώς θα κλονιστεί και θα καταστραφεί η ροή της είδησης, μειώνοντας την αποτελεσματικότητα της κωδικοποίησης εξωγενούς πλαισίου (exogenous context encoder). Ως προεργασία των εμπλεκόμενων, με την προς επεξεργασία είδηση, χρηστών αφαιρούμε τον χαρακτήρα "@" από τα username τους. Εν συνεχεία, εφαρμόζονται οι μέθοδοι μάθησης αναπαράστασης κειμένου.

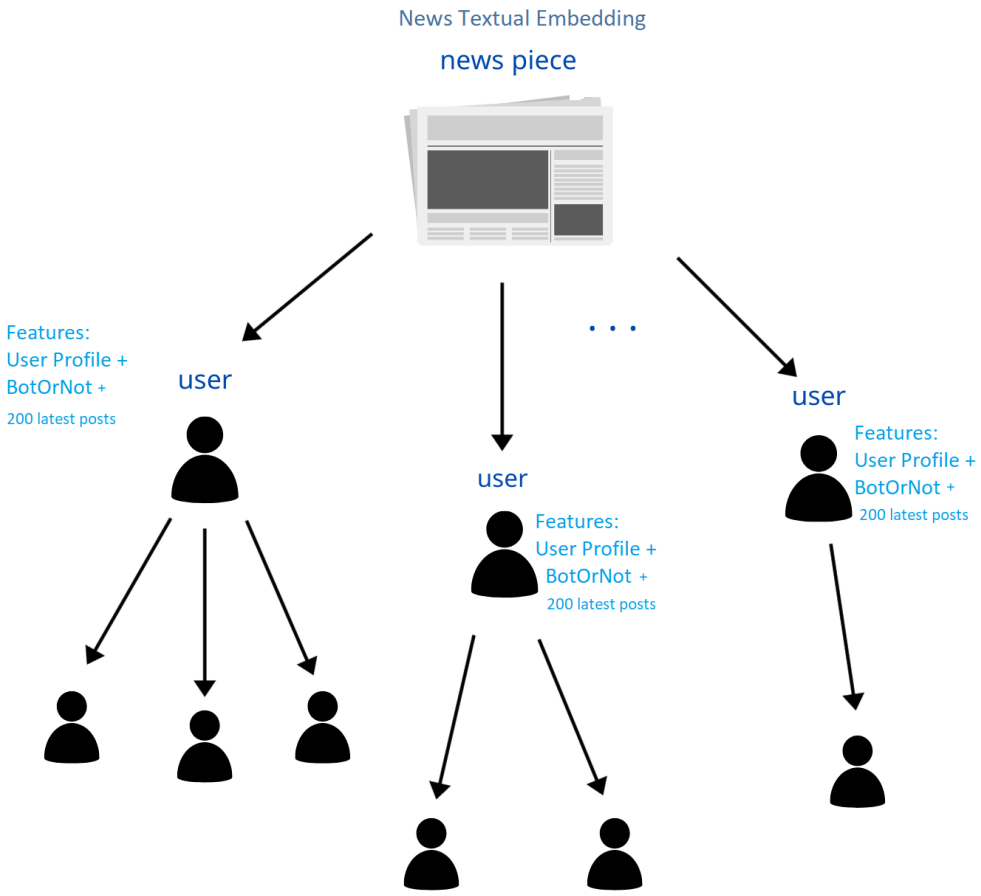
Για την κωδικοποίηση της πληροφορίας από το κείμενο της ανάρτησης, καθώς και την κωδικοποίηση των προτιμήσεων του χρήστη επιλέγονται δύο προσεγγίσεις μάθησης αναπαραστάσεων κειμένου που βασίζονται στην γλωσσικη προεκπαίδευση. Συγκεκριμένα, τα γλωσσικά εμφυτεύματα (*word embeddings*) προεκπαιδεύονται σε ένα μεγάλο λεξικό, πολύ μεγαλύτερο από αυτό που απαρτίζεται από το κείμενο της ανάρτησης, και έτσι δύνανται να κωδικοποιούν περισσότερες σημασιολογικές ομοιότητες ανάμεσα στις διάφορες λέξεις και προτάσεις. Για τις παραπάνω κωδικοποιήσεις επιλέγονται word2vec vectors (680k 300-dimensional), που προεκπαιδεύονται από την spaCy, οπως επίσης και τα προεκπαιδευμένα BERT, εμφυτεύματα που έχουν περιγραφεί εκτενώς στις παραγράφους 7.1, 7.2.

Συνοπτικά, η λειτουργία των μοντέλων μάθησης αναπαραστάσεων κειμένου word2vec και BERT (*text representation learning models*) είναι η ακόλουθη.
Η spaCy είναι μια δωρεάν πρόσβασης και ανοιχτού λογισμικού βιβλιοθήκη της Python, η οποία αφορά την φυσική επεξεργασία γλώσσας (Natural Language Processing). Η spaCy περιλαμβάνει περιλαμβάνει προεκπαιδευμένα διανύσματα για 680 χιλιάδες λέξεις. Έχοντας, λοιπόν, τις διακόσιες τελευταίες αναρτήσεις του χρήστη καθώς και το διάνυσμα για κάθε μία λέξη από τις αναρτήσεις αυτές, βρίσκουμε το μέσο όρο των διανυσμάτων των λέξεων όλων τον αναρτήσεων, ως ένα νέο διάνυσμα που εκφράζει το διάνυσμα της προτίμησης του χρήστη. Με τον τρόπο αυτό, προκύπτει η ενδογενής πληροφορία του χρήστη, ενώ το κειμενικό εμφύτευμα (*textual embedding*) προκύπτει, ομοίως, ως το διάνυσμα μέσος όρος των διανυσμάτων λέξεων του κειμένου της ανάρτησης.
Για το μοντέλο BERT, χρησιμοποιείται συγκεκριμένα ο τύπος BERT-Large για την κωδικοποίηση της είδησης/ανάρτησης και των πληροφοριών του χρήστη. Το περιεχόμενο της είδησης/ανάρτησης κωδικοποιείται χρησιμοποιώντας το μοντέλο BERT που δέχεται ως είσοδο μέγιστο μήκος ακολουθίας συγκεκριμένο αριθμό λέξεων, της επιλογής μας, με μέγιστο τις 512 λέξεις. Εξαιτίας του ορίου αριθμών λέξεων που επιτρέπεται να δεχτεί το μοντέλο, δεν είναι εφικτή η μονομιάς κωδικοποίηση των διακοσίων τελευταίων αναρτήσεων (tweets) του χρήστη ως μία ενιαία ακολουθία. Για το λόγο αυτό, κωδικοποιείται το κάθε tweet ξεχωριστά ως ένα διάνυσμα. Μόλις ολοκληρωθεί η κωδικοποίηση και των διακοσίων διανυσμάτων, προκύπτει το διάνυσμα της ενδογενούς πληροφορίας του χρήστη, ως ο μέσος όρος των παραπάνω διανυσμάτων. Στη γενική περίπτωση, οι προσωπικές αναρτήσεις των χρηστών (tweets) περιέχουν μικρότερο αριθμό λέξεων από μία είδηση. Συνεπώς, το όριο λέξεων που δέχεται το μοντέλο BERT τίθεται στις 16 λέξεις, κατώφλι που επιταχύνει την κωδικοποίηση των tweets.

Με την ολοκλήρωση της περιγραφής της κωδικοποίησης της ενδογενούς πληροφορίας του χρήστη, ακολουθεί η λεπτομερής περιγραφή της κωδικοποίησης της εξωγενούς πληροφορίας.

8.7.2 Exogenous Context Extraction

Δεδομένης μιας είδησης στα social media, το εξωγενές πλαίσιο του χρήστη (*user exogenous context*) συντίθεται από όλους του χρήστες που σχετίζονται με την είδηση αυτή. Με αυτόν τον τρόπο, η πληροφορία των επαναναρτήσεων της είδησης από άλλους χρήστες (retweet), συμβάλλει στην κατασκευή του γράφου διάδοσης της είδησης (*news propagation graph*). Όπως απεικονίστηκε και στην εικόνα 29, πρόκειται για ένα γράφο με δομή δέντρου, όπου ο κόμβος ρίζα αντιπροσωπεύει την είδηση που βρίσκεται στα social media, ενώ οι λοιποί κόμβοι αναπαριστούν τους χρήστες που ανήρτησαν την είδηση αυτή. Εάν κάποιος χρήστης δημοσίευσε την είδηση αυτή από κάποιον άλλον χρήστη, τότε οι δύο κόμβοι-χρήστες συνδέονται μεταξύ τους με τρόπο που περιγράφεται στην παράγραφο 8.5.



Εικόνα 40 Γράφος διάδοσης που προκύπτει από την προς εξέταση είδηση. Όπως αποδίδεται είναι κατευθυνόμενος αποδίδοντας την ροή διάδοσης της είδησης από χρήστη σε χρήστη και μάλιστα ομογενής, αφού οι ακμές εκφράζουν το *retweet* της είδησης από τον έναν χρήστη στον άλλον. Το συνολικό κοινωνικό δίκτυο με όλες τις ειδήσεις αποτελείται από πολλούς αυτόνομους γράφους τέτοιας μορφής. Για την απόδοση της *user preference* πληροφορίας προσθέτουμε στα χαρακτηριστικά του χρήστη και τα *textual embeddings* των διακοσίων πιο πρόσφατων αναρτήσεών του στο *Twitter*.

Έχοντας πλέον αναλύσει τόσο την ενδογενή όσο και την εξωγενή κωδικοποίηση σειρά έχει η ένωση των δύο αυτών πληροφοριών.

8.7.4 Information Fusion

Σύμφωνα με τις ακόλουθες δημοσιεύσεις [51],[52], ενώνοντας τα χαρακτηριστικά του χρήστη με τον γράφο διάδοσης της είδησης, ενισχύεται η απόδοση στην ανίχνευση ψευδών ειδήσεων. Εφόσον, τα νευρωνικά δίκτυα με την μορφή γράφων (GNN) κωδικοποιούν τόσο χαρακτηριστικά κόμβων, όσο και δομές γράφων αποτελούν χρήσιμα εργαλεία για την ενοποίηση της ενδογενούς και της εξωγενούς πληροφορίας. Συγκεκριμένα, εφαρμόζεται ιεραρχική ένωση πληροφορίας. Χάρη στο GNN το εμφύτευμα που προέρχεται από την αρχική είδηση (*news textual embedding*) καθώς και το εμφύτευμα των προτιμήσεων του χρήστη (user preference embedding), που προκύπτει ως διάνυσμα του μέσου όρου των διανυσμάτων των τελευταίων διακοσίων αναρτήσεών του, μπορούν να ληφθούν ως χαρακτηριστικά κόμβων. Δεδομένου του γράφου διάδοσης της είδησης, το GNN προκειμένου να διαμορφώσει το embedding ενός κόμβου, συγκεντρώνει τα χαρακτηριστικά των γειτονικών του κόμβων.Έπειτα, εφαρμόζεται μια *mean pooling readout* συνάρτηση σε όλα τα embeddings των κόμβων, προκειμένου να προκύψει το embedding του συνολικού γράφου διάδοσης. Έτσι, προκύπτει το user engagement embedding. Εφόσον το περιεχόμενο των ειδήσεων περιέχει σαφή δείγματα όσον αφορά την αξιοπιστία των ειδήσεων, κρίνεται αναγκαία η ένωση του news textual embedding της αρχικής είδησης και του user engagement embedding και έτσι συντίθεται το ολοκληρωμένο news embedding που περιέχει πλέον όλη την πληροφορία. Το ενωμένο πλέον news embedding οδηγείται σε έναν Multi-layer Perceptron (MLP) δύο επιπέδων, με δύο νευρώνες εξόδου, όπου η μία δίνει έξοδο εάν η είδηση είναι αληθής και η έτερη δίνει έξοδο εάν η είδηση είναι ψευδής. Το μοντέλο εκπαιδεύεται χρησιμοποιώντας binary cross-entropy, ενώ η συνάρτηση κόστους ανανεώνεται με SGD.

Παρακάτω παρατίθεται ο πίνακας με την απόδοση του μοντέλου, στα δύο *datasets*. Ο πρώτος πίνακας απευθύνεται στα *endogenous preference* και *user-profile features* και ο δεύτερος στα ίδια χαρακτηριστικά μαζί με *topic-related*.

	Politifact	Gossipcop
Embeddings	ACC | F1	ACC | F1
word2vev	80.67 | 80.26	83.18 | 83.27
BERT	84.75 | 84.44	89.88 | 88.69

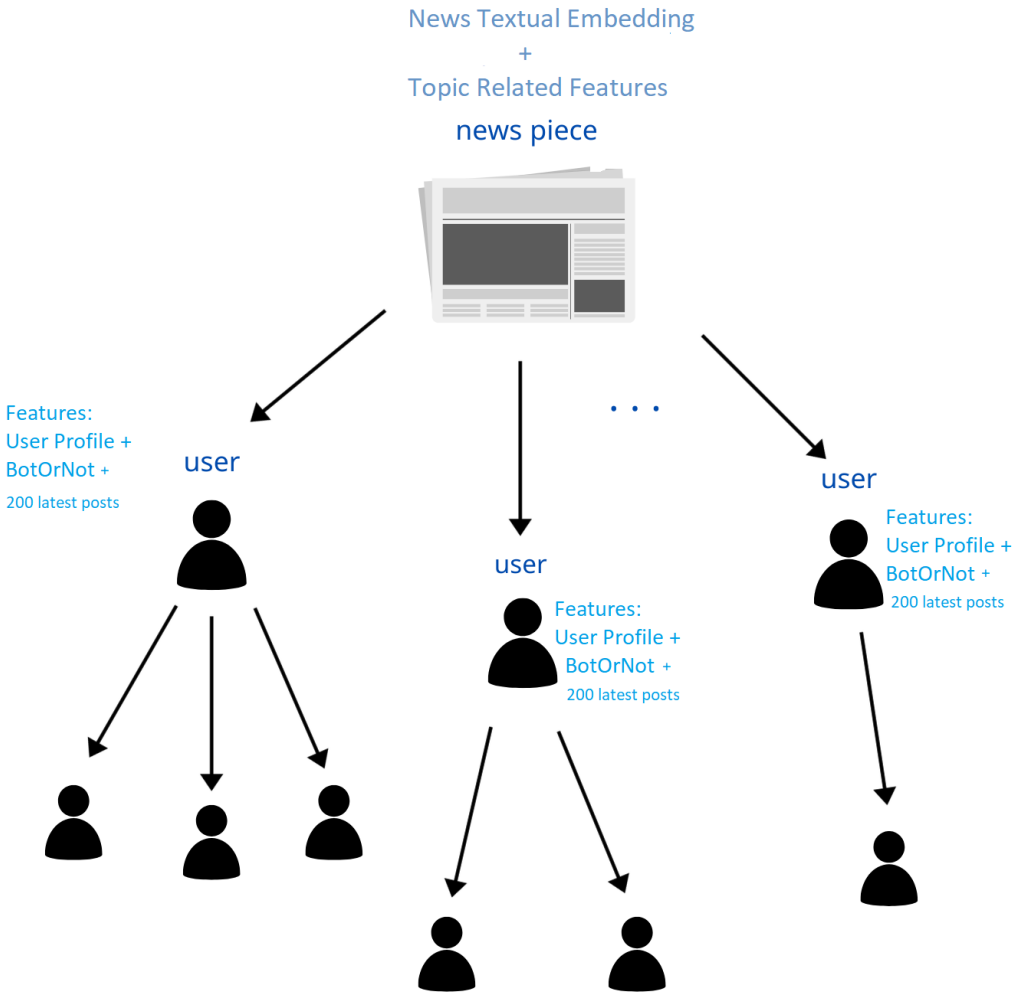
Πίνακας 8 Αποτελέσματα για τις μετρικές *accuracy* και *F1-score* στο κάθε *dataset*, για τα *embeddings word2vec* και *BERT* χρησιμοποιώντας το GCN μοντέλο με *user-related (latest 200 posts)* και *user-profile* χαρακτηριστικά.

Οι παράμετροι του μοντέλου, έπειτα από *fine-tuning* είναι οι ακόλουθες:

dataset	model	feature	epoch	learning rate	embedding size	batch number
Politifact	GCN	word2vec	80	0.001	128	128
Politifact	GCN	BERT	80	0.001	128	64
Gossipcop	GCN	word2vec	100	0.01	128	128
Gossipcop	GCN	BERT	100	0.001	128	64

Πίνακας 9 Υπερ-παράμετροι του GCN μοντέλου για τα *topic-related* και *user-preference features.*

Προσθέτοντας στον κόμβο είδηση και τα Topic Related χαρακτηριστικά, ο γράφος διάδοσης που προκύπτει είναι ο ακόλουθος.



Εικόνα 41 Γράφος διάδοσης που προκύπτει από την προς εξέταση είδηση. Όπως αποδίδεται είναι κατευθυνόμενος αποδίδοντας την ροή διάδοσης της είδησης από χρήστη σε χρήστη και μάλιστα ομογενής, αφού οι ακμές εκφράζουν το *retweet* της είδησης από τον έναν χρήστη στον άλλον. Το συνολικό κοινωνικό δίκτυο με όλες τις ειδήσεις αποτελείται από πολλούς αυτόνομους γράφους, τέτοιας μορφής. Πρόκειται για τον πιο ενισχυμένο γράφο από άποψη χαρακτηριστικών. Ο κόμβος κεφαλή, η προς εξέταση είδηση, περιέχει τα χαρακτηριστικά του κειμένου και τα τρία επιλεγμένα topic-related χαρακτηριστικά. Οι υπόλοιποι κόμβοι, οι χρήστες που έκαναν retweet την είδηση, σύμφωνα με τους κανόνες στην 8.5, περιέχουν τα χαρακτηριστικά user profile, BotOrNot και τα text embeddings των 200 πιο πρόσφατων αναρτήσεών τους.

	Politifact	Gossipcop
Embeddings	ACC | F1	ACC | F1
word2vev	80.26 | 79.80	83.19 | 83.54
BERT	82.34 | 82.23	85.17 | 85.04

Πίνακας 10 Αποτελέσματα για τις μετρικές *accuracy* και *F1-score* στο κάθε *dataset*, για τα *embeddings word2vec* και *BERT* χρησιμοποιώντας το GCN μοντέλο με *user-related (latest 200 posts)*, *user-profile* και *topic-related* χαρακτηριστικά.

Οι παράμετροι του μοντέλου, έπειτα από *fine-tuning* είναι οι ακόλουθες:

dataset	model	feature	epoch	learning rate	embedding size	batch number
Politifact	GCN	word2vec	80	0.001	128	128
Politifact	GCN	BERT	100	0.01	128	128
Gossipcop	GCN	word2vec	100	0.01	128	128
Gossipcop	GCN	BERT	100	0.001	128	128

Πίνακας 11 Υπερ-παράμετροι του GCN μοντέλου για τα *user-related (latest 200 posts)*, *user-profile* και *topic-related features*.

Συνεπώς το UPFD *framework* αποτελεί μία καινοτόμα προσέγγιση για το πρόβλημα της ανίχνευσης ψευδών ειδήσεων στα κοινωνικά δίκτυα. Το GCN μοντέλο που παρουσιάστηκε, χρησιμοποιεί έναν αποδοτικό κανόνα διάδοσης, ανά επίπεδο, βασισμένο στην πρώτης τάξης ($K = 1$, διάδοση κατά τους πρώτους άμεσους γείτονες) προσέγγιση των *spectral convolutions* στους γράφους. Τα πειράματά μας στη βάση δεδομένων, απέδειξαν ότι το προτεινόμενο GCN μοντέλο είναι ικανό να κωδικοποιήσει τόσο τη δομή του γράφου, όσο και τα χαρακτηριστικά των κόμβων, ξεπερνώντας σε επιδόσεις τα υπόλοιπα νευρωνικά δίκτυα, όντας παράλληλα υπολογιστικά οικονομικό.
Όπως φανερώθηκε από τους πίνακες αποτελεσμάτων, όταν το μοντέλο εκπαιδευτεί με ενισχυμένα χαρακτηριστικά, δηλαδή με *user-related (latest 200 posts)*, *user-profile* μαζί με το BotOrNot feature και *topic-related* η επίδοσή του δυσχεραίνεται. Ενώ του έχουμε προσδώσει πολλή πληροφορία και διαισθητικά αναμένεται να αποδώση βελτιστοποιημένα, το μοντέλο καταλήγει να υπερπροσαρμόζεται στα δεδομένα και να στερείται την ικανότητά του να γενικεύει. Τέλος, για άλλη μια φορά, τα εμφυτεύματα *BERT* είναι αυτά που εξασφαλίζουν την καλύτερη επίδοση.

Όπως έγινε αντιληπτό από τα παραπάνω αποτελέσματα, η μέθοδος UPFD είναι αυτή που ξεπερνάει τις baseline τεχνικές όταν συνοδεύεται από τα χαρακτηριστικά *user-profile* μαζί με το *BotOrNot* feature και τα χαρακτηριστικά του κειμένου, όπως επίσης και τις σύγχρονες μεθόδους των GNNs

που επεκτάθηκαν στα τρία συνελικτικά επίπεδα. Για το σκοπό αυτό, ενισχύουμε το προηγούμενο GCN μοντέλο, μελετώντας δύο νέα GCN μοντέλα, για την ανίχνευση των ψευδών ειδήσεων, βασισμένα στα *user-preferences* και *user-profile* χαρακτηριστικά του UPFD framework.

9. Graph SAGE

Το μοντέλο GraphSAGE χρησιμοποιείται στην επαγωγική εύρεση embeddings των κόμβων. Σε αντίθεση με τις συνηθισμένες μεθόδους υπολογισμού embeddings, μέσω πολλαπλασιασμών πινάκων, χρησιμοποιούμε τα χαρακτηριστικά των κόμβων, user-preferences, user-related features και textual embeddings στην περίπτωσή μας, προκειμένου να δημιουργήσουμε μια συνάρτηση embedding που θα γενικεύεται και στους ανεξερεύνητους κόμβους.
Ενσωματώνοντας τα χαρακτηριστικά των κόμβων στον αλγόριθμο μάθησης, συγχρόνως γίνεται γνωστή η τοπολογική δομή της γειτονιάς κάθε κόμβου, όπως επίσης και η κατανομή των χαρακτηριστικών των κόμβων σε κάθε γειτονιά. Η μέθοδος αυτή μπορεί να χρησιμοποιηθεί τόσο στην περίπτωσή μας, όπου οι κόμβοι διαθέτουν υψηλό αριθμό χαρακτηριστικών, καθώς επίσης και σε περιπτώσεις όπου οι κόμβοι δε διαθέτουν χαρακτηριστικά. Στην περίπτωση αυτή, χρησιμοποιούμε τον βαθμό κάθε κόμβου.
Στην περίπτωση των SAGE γράφων, αντί να εκπαιδεύουμε ένα συγκεκριμένο διάνυσμα embedding για κάθε κόμβο, εκπαιδεύουμε ένα σύνολο aggregator συναρτήσεων, οι οποίες μαθαίνουν να εφαρμόζουν aggregation σε πληροφορίες που βρίσκονται σε διαφορετικό βάθος στον γράφο. Παραδειγματικά, έχοντας έναν κόμβο στόχο, η πρώτη aggregation function εφαρμόζεται στους άμεσους γείτονες του κόμβου, η δεύτερη aggregation function στους γείτονες του δεύτερου επιπέδου κ.ο.κ. Έχοντας ολοκληρώσει κατά αυτόν τον τρόπο την εκπαίδευση του γράφου, κατά την αξιολόγησή του, χρησιμοποιούμε το εκπαιδευμένο σύστημα για να παράγουμε embeddings για γράφους που δεν έχουν διερευνηθεί, βάσει των aggregate συναρτήσεων που προέκυψαν από την εκπαίδευση. Η συνάρτηση κόστους του μοντέλου αυτού είναι επιλεγμένη με τρόπο τέτοιο ώστε να επιτρέπει στο μοντέλο SAGE να εκπαιδεύεται δίχως επίβλεψη συγκεκριμένου task.
Επαναλαμβάνεται, πως η ιδιαιτερότητα της GraphSAGE τακτικής εντοπίζεται στον τρόπο με τον οποίον μαθαίνει να εφαρμόζει aggregation στις πληροφορίες των χαρακτηριστικών της γειτονιάς του εκάστοτε κόμβου. Αρχικά, περιγράφεται ο αλγόριθμος παραγωγής των embeddings του γράφου GraphSAGE (*forward propagation*) θεωρώντας τις παραμέτρους του μοντέλου γνωστές. Έπειτα, αναλύεται το πώς αυτές οι παράμετροι μπορούν να προκύψουν χρησιμοποιώντας τις τεχνικές stochastic gradient descent.

9.1 Αλγόριθμος Παραγωγής των Embeddings

Στην παράγραφο αυτή αναλύεται ο αλγόριθμος forward propagation, βάσει του οποίου παράγονται τα embeddings των κόμβων, θεωρώντας ότι το μοντέλο έχει ήδη εκπαιδευτεί και ότι οι παράμετροι του είναι καθορισμένοι. Συγκεκριμένα, γίνεται η υπόθεση ότι έχουμε μάθει τις παραμέτρους από K συναρτήσεις aggregation, οι οποίες θα συμβολίζονται με $AGGREGATE_k$, $\forall k \in \{1,\dots,K\}$. Οι συναρτήσεις αυτές εφαρμόζουν aggregate στις πληροφορίες των γειτονικών κόμβων κάθε κόμβου. Επίσης, γνωστοί θεωρούνται και οι πίνακες βαρών W^k, $\forall k \in \{1,\dots,K\}$ οι οποίοι χρησιμοποιούνται για να διαδίδουν πληροφορία ανάμεσα στα διάφορα στρώματα του μοντέλου ή στα διαφορετικά βάθη αναζήτησης του γράφου.
Παρακάτω, παρουσιάζεται ο αλγόριθμος παραγωγής εμφυτευμάτων:

GraphSAGE embedding generation (i.e., forward propagation) algorithm

Input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth K; weight matrices $\mathbf{W}^k, \forall k \in \{1, ..., K\}$; non-linearity σ; differentiable aggregator functions $\text{AGGREGATE}_k, \forall k \in \{1, ..., K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$

Output : Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

$\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$;
for $k = 1...K$ **do**
  for $v \in \mathcal{V}$ **do**
    $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$;
    $\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\right)$
  end
  $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$
end
$\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$

Όπως φαίνεται από τον παραπάνω αλγόριθμο, σε κάθε επανάληψη ή βάθος αναζήτησης οι κόμβοι συγκεντρώνουν πληροφορίες από τους γείτονές τους και όσο οι επαναλήψεις αυξάνουν οι κόμβοι συλλέγουν όλο και περισσότερα δεδομένα, εξερευνώντας ολοένα και περισσότερους κόμβους.
Δεδομένου, λοιπόν του γράφου $G = (V, E)$ και των χαρακτηριστικών του κάθε κόμβου x_v, $\forall v \in V$, σε κάθε βήμα της εξωτερικής επανάληψης του αλγορίθμου, όπου το k εκφράζει το τρέχον βήμα ή το βάθος της αναζήτησης στον γράφο, και το h^k την αναπαράσταση του κόμβου στο τρέχον βήμα, αρχικά κάθε κόμβος $v \in V$ συλλέγει τις αναπαραστάσεις των κόμβων της άμεσης γειτονιάς του, $\{h_u^{k-1}, \forall u \in N(v)\}$, σε ένα διάνυσμα $h_{\mathrm{N}(u)}^{k-1}$. Το βήμα αυτής της συλλογής χαρακτηριστικών (*aggregation step*) εξαρτάται από τις αναπαραστάσεις των κόμβων που συλλέχθηκαν στο προηγούμενο βήμα, k - 1, ενώ για $k = 0$ οι αναπαραστάσεις των κόμβων ταυτίζονται με τα χαρακτηριστικά εισόδου τους.
Έχοντας εφαρμόσει aggregation στα γειτονικά διανύσματα χαρακτηριστικών, ο GraphSAGE συνενώνει (*concatenates*) την τρέχουσα αναπαράσταση του κόμβου, h_v^{k-1} μαζί με συνολικό διάνυσμα της γειτονιάς του $h_{\mathrm{N}(v)}^{k-1}$ και το τελικό διάνυσμα περνάει σε ένα *fully connected* στρώμα, μέσω μιας μη γραμμικής συνάρτησης ενεργοποίησης, σ, η οποία μετασχηματίζει την αναπαράσταση, προκειμένου να χρησιμοποιηθεί στο επόμενο βήμα του αλγορίθμου, $h_v^k, \forall v \in V$.
Οι τελικές αναπαραστάσεις των κόμβων σε βάθος K ταυτίζονται με $z_v \equiv h_v^K, \forall v \in V$.

9.2 Aggregators

Παρακάτω, εξετάζουμε τα διάφορα είδη συναρτήσεων *aggregation*.

9.2.1 Mean Aggregator

Η συνάρτηση aggregation μπορεί να είναι αυτή του μέσου όρου, mean aggregator, όπου λαμβάνουμε τον μέσο όρο των διανυσμάτων $\{h_u^{k-1}, \forall u \in N(v)\}$. Ο τελεστής του μέσου όρου,

υπολογίζει embeddings παρόμοια με το GCN μοντέλο. Συγκεκριμένα, μπορούμε να λάβουμε μια επαγωγική μορφή του GCN μοντέλου, απλά αντικαθιστώντας τις γραμμές 4 και 5 του παραπάνω αλγορίθμου, με την ακόλουθη:

$$h_v^k \leftarrow \sigma(W \cdot MEAN(\{h_v^{k-1}\} \cup \{h_u^{k-1}, \forall u \in N(v)\}).$$

Ο τροποποιημένος αυτός *mean-based aggregator* καλείται συνελικτικός, *convolutional*, καθώς είναι μια απλοϊκή γραμμική προσέγγιση μιας τοπικής spectral convolution. Μια σημαντική διαφοροποίηση ανάμεσα στον *mean-based aggregator* και τους υπόλοιπους *aggregators* είναι ότι δεν εφαρμόζει την συνένωση της γραμμής 5 του παραπάνω αλγορίθμου. Ο εν λόγω *aggregator* συνενώνει την προηγούμενου επιπέδου αναπαράσταση του κόμβου, h_v^{k-1}, με τον ενοποιημένο διάνυσμα $h_{\mathrm{N}(v)}^k$. Αυτή η ένωση μεταφράζεται ως "*skip connection*" μεταξύ των διαφόρων επιπέδων και σημειώνει σημαντικά αποτελέσματα στην απόδοση.

9.2.2 LSTM Aggregator

Πιο σύνθετη συνάρτηση, βασισμένη στην αρχιτεκτονική των LSTMs. Συγκρινόμενη με τον mean aggregator πλεονεκτεί στο ότι έχει μεγαλύτερη δυνατότητα έκφρασης (*expressive capability*). Ωστόσο, κρίνεται αναγκαίο να σημειωθεί πως τα LSTMs δεν είναι εκ φύσεως συμμετρικά και αμετάβλητα με τις μεταθέσεις, εφόσον επεξεργάζονται τα δεδομένα τους σε μια ακολουθία. Προσαρμόζουμε τα LSTMs προκειμένου να λειτουργήσουν σε ένα μη ακολουθιακά διατεταγμένο σύνολο, εφαρμόζοντας τα LSTMs σε μία τυχαία μετάθεση των γειτόνων των κόμβων.

9.2.3 Pooling Aggregator

Ο τελικός *aggregator* είναι και εκπαιδεύσιμος και συμμετρικός. Στην προσέγγιση αυτή, κάθε διάνυσμα των γειτονικών κόμβων περνάει ξεχωριστά και ανεξάρτητα σε ένα *fully-connected* νευρωνικό δίκτυο. Σύμφωνα με αυτό τον μετασχηματισμό, ο τελεστής *max-pooling* εφαρμόζεται στο σύνολο των γειτόνων προκειμένου να συλλέξει την συνολική πληροφορία.

$$AGGREGATE_k^{pool} = max(\{\sigma(W_{pool} h_{u_i}^k + b), \forall u_i \in N(v)\}),$$

οπου το max συμβολίζει το τελεστή μεγίστου και το σ τη μη γραμμική συνάρτηση ενεργοποίησης. Η συνάρτηση που εφαρμόζεται πριν το *max pooling* μπορεί να είναι από ένα από απλό νευρωνικό δίκτυο λίγων επιπέδων έως ένα βαθύ πολυεπίπεδο perceptron. Εφαρμόζοντας τον *max-pooling* τελεστή σε κάθε ένα από τα υπολογισμένα χαρακτηριστικά, το μοντέλο συλλέγει αποτελεσματικά τα διάφορα χαρακτηριστικά της γειτονιάς των κόμβων. Στη θέση της συνάρτησης max μπορεί να ενταχθεί οποιαδήποτε άλλη συνάρτηση συμμετρική συνάρτηση διανυσμάτων.
Για την εκπαίδευση του μοντέλου GraphSAGE στην βάση δεδομένων μας καθώς και για την αξιολόγησή του, χρησιμοποιήθηκε ο *convolutional mean-based aggregator*.

9.3 Μαθαίνοντας τις παραμέτρους του *GraphSAGE*

Προκειμένου να μάθουμε χρήσιμες αναπαραστάσεις πρόβλεψης σε ένα πλήρως μη επιβλεπόμενο περιβάλλον, χρησιμοποιούμε μία *graph-based loss function* στο αποτέλεσμα των αναπαραστάσεων,

z_u, $\forall\, u \in V$, και προσαρμόζουμε-συντονίζουμε (*tune*) τον πίνακα βαρών W^k $\forall\, k \in \{1, \dots, K\}$ και τις παραμέτρους των συγκεντρωτικών (*aggregate*) συναρτήσεων μέσω στοχαστικού *gradient descent*. Χάρη στην *graph-based loss function* οι διπλανοί κόμβοι έχουν παρόμοιες αναπαραστάσεις, ενώ οι αναπαραστάσεις των μη όμοιων κόμβων διαφέρουν σημαντικά:

$$J_G(z_u) = -\log(\sigma(z_u^T z_v)) - QE_{v_n \sim P_n(v)} \log(\sigma(-z_u^T z_{v_n})), \quad (1)$$

οπου *v* είναι ένας κόμβος που συνυπάρχει δίπλα στον u σε ένα καθορισμένου μήκους τυχαίο περίπατο, σ είναι η σιγμοειδής συνάρτηση, P_n είναι μία κατανομή αρνητικών δειγμάτων και το Q καθορίζει τον αριθμό των αρνητικών δειγμάτων. Τονίζεται, πως σε αντίθεση με τις προηγούμενες προσεγγίσεις, οι αναπαραστάσεις z_u που περνιούνται στη συνάρτηση σφάλματος παράγονται από τα χαρακτηριστικά που περιέχονται στην τοπική γειτονία κάθε κόμβου και όχι εκπαιδεύοντας ένα μοναδικό *embedding* για κάθε κόμβο.
Αυτό το μη-επιβλεπόμενο περιβάλλον μιμείται καταστάσεις όπου τα χαρακτηριστικά των κόμβων παρέχονται προκειμένου να επιλύσουν *downstream tasks* σε *machine learning* αλγορίθμους, ως μια υπηρεσία ή στατικά *repositories*. Σε περίπτωση που αναπαραστάσεις χρησιμοποιούνται σε ένα συγκεκριμένο *downstream task* η μη επιβλεπόμενη εξίσωση (1) μπορεί να αντικατασταθεί με μία *task-specific* συνάρτηση (πχ. *cross-entropy loss*).

Τα *downstream tasks*, στο πλαίσιο της μη επιβλεπόμενης μάθησης, αφορούν την εκπαίδευση ενός μοντέλου σε ένα γενικό *dataset*, και όχι σε αυτό το *dataset* όπου επιθυμούμε να λυθεί το πρόβλημά μας, προκειμένου το μοντέλο να αποκτήσει ορισμένα γενικά χαρακτηριστικά που ενισχύουν την ικανότητα γενίκευσης του. Έπειτα, εφαρμοζουμε *fine-tuning* στο προεκπαιδευμένο μοντέλο για την εύρεση των βέλτιστων παραμέτρων στο *dataset* που αφορά το προς επίλυση πρόβλημα. Έπειτα, το προς επίλυση *task* ονομάζεται *downstream task.*

Οι επιδόσεις του μοντέλου φαίνονται παρακάτω:

	Politifact	Gossipcop
Embeddings	ACC | F1	ACC | F1
word2vev	84.56 | 84.43	85.48 | 84.88
BERT	85.87 | 85.93	88.43 | 88.06

Πίνακας 12 Αποτελέσματα για τις μετρικές *accuracy* και *F1-score* στο κάθε *dataset*, για τα *embeddings word2vec* και *BERT* χρησιμοποιώντας το GraphSAGE μοντέλο με *user-related (latest 200 posts)* και *user-profile* χαρακτηριστικά.

Οι παράμετροι του μοντέλου, είναι οι βέλτιστες για την απόδοσή του και παρουσιάζονται παρακάτω:

dataset	model	feature	epoch	learning rate	embedding size	batch number
Politifact	GraphSAGE	word2vec	45	0.01	128	128
Politifact	GraphSAGE	BERT	30	0.01	128	128
Gossipcop	GraphSAGE	word2vec	80	0.01	128	128
Gossipcop	GraphSAGE	BERT	80	0.001	128	128

Πίνακας 13 Υπερ-παράμετροι του GraphSAGE μοντέλου για τα *user-related (latest 200 posts)* και *user-profile features.*

Όπως ήταν αναμενόμενο, το μοντέλο convolutional GraphSAGE υπερβαίνει τις επιδόσεις των μέχρι στιγμής προτεινόμενων μοντέλων. Τα embeddings δημιουργήθηκαν με αποδοτικό τρόπο από τους μη εξερευνημένους κόμβους, χάρη στη μέθοδο της επαγωγικής μάθησης. Τα πιο υψηλά αποτελέσματα, έχουν ως συνέπεια την αύξηση του χρόνου που απαιτείται προκειμένου το μοντέλο να κάνει προβλέψεις, λόγω της δειγματοληψίας των γειτόνων των κόμβων και της μάθησης της δομής του γράφου. Πιθανές επεκτάσεις που θα μπορούσαν να δοθούν και βελτιστοποιήσουν περαιτέρω το μοντέλο είναι να δέχεται κατευθυνόμενους γράφους, όπως επίσης οι συναρτήσεις που χρησιμοποιούνται για τη συλλογή των γειτόνων να είναι μη κανονικής κατανομής.

10. Graph Attention Networks

Η ανάλυση εκκινεί περιγράφοντας ένα μόνο επίπεδο του γράφου με μηχανισμό attention, ως το μόνο επίπεδο που χρησιμοποιείται για την αξιολόγηση των δεδομένων μας μέσα στην αρχιτεκτονική GAT.
Ως είσοδος στο επίπεδό μας αυτό είναι ένα σύνολο από χαρακτηριστικά κόμβων,

$$h = \{\overline{h_1}, \overline{h_2}, \ldots, \overline{h_N}\}, \overline{h_i} \in \mathbb{R}^F,$$

όπου N ο αριθμός των κόμβων, F είναι ο αριθμός των χαρακτηριστικών του κάθε κόμβου. Το επίπεδο με το μηχανισμό του *attention*, παράγει ένα νέο σύνολο χαρακτηριστικών κόμβων, με πιθανώς διαφορετικό αριθμό χαρακτηριστικών F′, $h' = \{\overline{h'_1}, \overline{h'_2}, \ldots, \overline{h'_N}\}, \overline{h'_i} \in \mathbb{R}^{F'}$ ως έξοδο.
Προκειμένου να αποκτήσουμε αποδοτική δύναμη έκφρασης, ώστε να μετατρέψουμε τα χαρακτηριστικά εισόδου σε χαρακτηριστικά υψηλότερου επιπέδου, απαιτείται τουλάχιστον ένας γραμμικός μετασχηματισμός μάθησης. Ως αρχικό βήμα, ένας γραμμικός μετασχηματισμός, παραμετροποιημένος από ένα πίνακα βαρών, $W \in \mathbb{R}^{F'xF}$, εφαρμόζεται σε κάθε κόμβο. Έπειτα, εφαρμόζεται *self-attention* στους κόμβους. Ο μηχανισμός $a: \mathbb{R}^{F'} x \mathbb{R}^{F} \rightarrow \mathbb{R}$, υπολογίζει τους συντελεστές *attention* $e_{ij} = a(W\overline{h_i}, W\overline{h_j})$.
Η παραπάνω σχέση υπογραμμίζει τη σημαντικότητα των χαρακτηριστικών του κόμβου j για τον κόμβο i. Στην γενικότητα των περιπτώσεων, το μοντέλο επιτρέπει σε κάθε κόμβο να παρακολουθεί κάθε άλλο κόμβο, παραμερίζοντας την δομή του γράφου. Στην περίπτωσή μας, λαμβάνουμε υπόψιν τη δομή του γράφου εφαρμόζοντας *masked attention*, υπολογίζουμε, δηλαδή, τους e_{ij} μόνο για τους

κόμβους $j \in N_i$, οπου N_i η γειτονιά του κόμβου i στον γράφο. Για την υλοποίηση του μοντέλου, ως γειτονιά του εκάστοτε κόμβου επιλέγονται οι πρώτης τάξης γείτονες. Για να είναι οι συντελεστές ευκόλως συγκρίσιμοι με ανάμεσα στους διάφορους κόμβους, τους κανονικοποιούμε για κάθε επιλογή του j χρησιμοποιώντας την συνάρτηση *softmax*.

$$a_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})}$$

Κατά την υλοποίηση, ο μηχανισμός *attention* α επιλέχθηκε να είναι ένα μονοεπίπεδο *feedforward* νευρωνικό δίκτυο παραμετροποιημένο από ένα διάνυσμα βάρους $\overline{\alpha} \in \mathbb{R}^{2F'}$ και εφαρμόζοντας LeakyReLU, με negative input slope α = 0.2. Συντελεστές, πλέον, υπολογίζονται ως ακολούθως:

$$a_{ij} = \frac{exp(LeakyReLU(\overline{a}^T [W\overline{h}_i \,||\, W\overline{h}_j]))}{\sum_{k \in N_i} exp(LeakyReLU(\overline{a}^T [W\overline{h}_i \,||\, W\overline{h}_j]))} \quad (10.1)$$

Όπου με T συμβολίζουμε τον ανάστροφο πίνακα και με || την πράξη της συνένωσης, *concatenation*. Οι κανονικοποιημένοι συντελεστές *attention* χρησιμοποιούνται για τον υπολογισμό ενός γραμμικού συνδυασμού των χαρακτηριστικών που ανταποκρίνονται σε αυτούς και παρουσιάζονται ως τα τελικά χαρακτηριστικά εξόδου για κάθε κόμβο:

$$\overline{h'}_i = \sigma\left(\sum_{j \in N_i} a_{ij} W\overline{h}_j\right) \quad (10.2)$$

Προκειμένου να σταθεροποιήσουμε τη διαδικασία μάθησης του *self-attention* μηχανισμού, επεκτείνουμε τον μηχανισμό προκειμένου, ο *multi-head* μηχανισμός να είναι αποδοτικός. Συγκεκριμένα, *K* ανεξάρτητοι μηχανισμοί προσοχής εκτελούν την προηγούμενη εξίσωση και έπειτα τα χαρακτηριστικά τους συνενώνονται, διαμορφώνοντας την επόμενη αναπαράσταση των χαρακτηριστικών εξόδου:

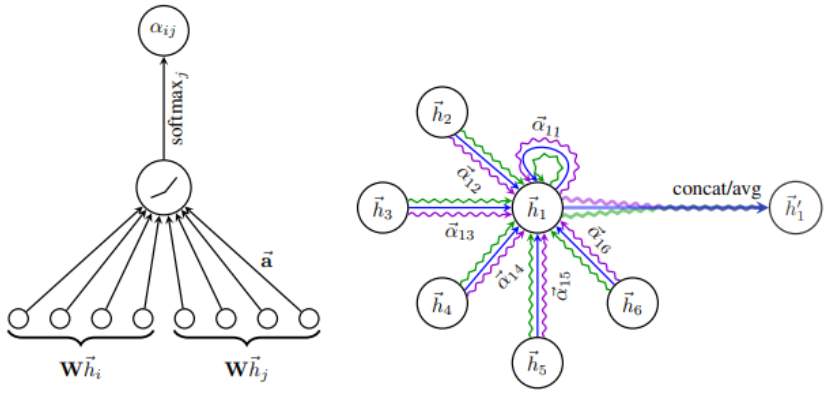
$$\overline{h'}_i = ||_{k=1}^{K} \left[\sigma\left(\sum_{j \in N_i} a_{ij}^{\,k} W^k \overline{h}_j\right)\right] \quad (10.3)$$

οπου με || συμβολίζεται η πράξη της συνένωσης, $a_{ij}^{\,k}$ είναι οι κανονικοποιημένοι συντελεστές *attention* υπολογισμένοι στην υπολογισμένοι από τον *k* - οστό μηχανισμό προσοχής α^k και W^k είναι ο αντίστοιχος πίνακας βαρών γραμμικού μετασχηματισμού της εισόδου. Σημειώνεται, πως το τελικό επιστρεφόμενο διάνυσμα h' θα αποτελείται από $K \cdot F'$ χαρακτηριστικά, αντί για F', για κάθε κόμβο.

Συγκεκριμένα, εάν εφαρμόσουμε *multi-head attention* στο τελικό επίπεδο του νευρωνικού δικτύου, το επίπεδο πρόβλεψης, η πράξη της συνένωσης δεν θα είχε νόημα, αντιθέτως, εφαρμόζοντας εφαρμόζοντας τον τελεστή μέσου όρου και τέλος μια μη γραμμική συνάρτηση σ (*softmax* ή *logistic softmax*) προκύπτει:

$$\overline{h'}_i = \sigma\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in N_i} a_{ij}^{\,k} W^k \overline{h}_j\right) \quad (10.4)$$

Η διαδικασία *aggregation* ενός *multi-head attentional* επιπέδου του γράφου αποδίδεται στο δεξί μέρος της εικόνας 34.



Εικόνα 42 Αριστερά: Ο μηχανισμός προσοχής $a(W\overline{h_i}, W\overline{h_j})$ παραμετροποιημένος από ένα διάνυσμα βάρους $\overline{\alpha} \in \mathbb{R}^{2F'}$, εφαρμόζοντας LeakyReLU συνάρτηση ενεργοποίησης. Δεξιά: Μια αναπαράσταση του *multi-head attention* μηχανισμού (με $K = 3$ *heads*) εφαρμοσμένος στον κόμβο 1 και στην γειτονιά του. Το διαφορετικό σχήμα και χρώμα που αποδίδεται στα βέλη εκφράζει την ανεξαρτητους υπολογισμούς του *attention*. Τα συνολικά χαρακτηριστικά (*aggregated features*) από κάθε *head* συνενώνονται ή εφαρμόζεται μέσος όρος σε αυτά για να σχηματίσουν το $\overline{h'}_1$ [30] .

Οι επιδόσεις του μοντέλου σε κάθε ένα από τα *datasets* για κάθε είδους *embeddings* για τα *user-preferences* και *user-profile* χαρακτηριστικά, είναι οι ακόλουθες:

	Politifact	Gossipcop
Embeddings	ACC | F1	ACC | F1
word2vev	86.63 | 86.78	90.54 | 90.36
BERT	87.77 | 87.32	92.55 | 92.63

Πίνακας 14 Αποτελέσματα για τις μετρικές *accuracy* και *F1-score* στο κάθε *dataset*, για τα *embeddings word2vec* και *BERT* χρησιμοποιώντας το GAT μοντέλο με *user-related (latest 200 posts)* και *user-profile* χαρακτηριστικά.

Οι παράμετροι του μοντέλου, είναι οι βέλτιστες για την απόδοσή του και παρουσιάζονται παρακάτω:

dataset	model	feature	epoch	learning rate	embedding size	batch number
Politifact	GAT	word2vec	30	0.01	128	128
Politifact	GAT	BERT	50	0.001	128	128
Gossipcop	GAT	word2vec	30	0.001	128	128
Gossipcop	GAT	BERT	80	0.001	128	128

Πίνακας 15 Υπερ-παράμετροι του GAT μοντέλου για τα *user-related (latest 200 posts)* και *user-profile features.*

Στη συνέχεια θα παραθέσουμε τις επιδόσεις των baseline μεθόδων και παράλληλα των μοντέλων που αναλύθηκαν στην παρούσα εργασία, για το FakeNewsNet Dataset.

Baseline Methods	Accuracy
SVM	0.580
Logistic Regression	0.642
Naive Bayes	0.617
CNN	0.629
SAF/S	0.633
RST	0.607
LIWC	0.769
text-CNN	0.653

Πίνακας 16 Παρουσίαση *accuracy* για τις *baseline* μεθόδους στο *Politifact dataset*.

Baseline Methods	Accuracy
RST	0.531
LIWC	0.736
text-CNN	0.739

Πίνακας 17 Παρουσίαση *accuracy* για τις *baseline* μεθόδους στο *Gossipcop dataset*.

model	features	embeddings	Accuracy	F1
GCN	user-profile	word2vec	78.03	76.62
GCN	topic-related	word2vec	79.27	79.04
GCN	user-preference & user-profile	word2vec	80.67	80.26
GCN	user-preference & user-profile & topic-related	word2vec	80.26	79.80
GraphSAGE	user-preference & user-profile	word2vec	84.56	84.43
GAT	user-preference & user profile	word2vec	86.63	86.78
GCN	user-profile	BERT	79.25	78.25
GCN	topic-related	BERT	82.75	82.93
GCN	user-preference & user-profile	BERT	84.75	84.44
GCN	user-preference & user-profile & topic-related	BERT	82.34	82.23
GraphSAGE	user-preference & user-profile	BERT	85.87	85.93
GAT	user-preference & user-profile	BERT	87.77	87.32

Πίνακας 18 Συνολικά αποτελέσματα για το *Politifact dataset*.

model	features	embeddings	Accuracy	F1
GCN	user-profile	word2vec	79.22	79.21
GCN	topic-related	word2vec	83.09	83.24
GCN	user-preference & user-profile	word2vec	83.18	83.27
GCN	user-preference & user-profile & topic-related	word2vec	83.19	83.54
GraphSAGE	user-preference & user-profile	word2vec	85.48	84.88

GAT	user-preference & user profile	word2vec	90.54	90.36
GCN	user-profile	BERT	80.77	80.19
GCN	topic-related	BERT	83.20	83.39
GCN	user-preference & user-profile	BERT	89.88	88.69
GCN	user-preference & user-profile & topic-related	BERT	85.17	85.04
GraphSAGE	user-preference & user-profile	BERT	88.43	88.06
GAT	user-preference & user-profile	BERT	92.55	92.63

Πίνακας 19 Συνολικά αποτελέσματα για το *Gossipcop dataset*.

Τα Graph Attention Networks (GATs) ως σύγχρονα συνελικτικά νευρωνικά δίκτυα που εφαρμόζονται σε δεδομένα μορφής γράφου, αποτελούμενα από *masked self-attentional layers* σημειώνουν υψηλότατες επιδόσεις στον τομέα ανίχνευσης ψευδών ειδήσεων . Το επίπεδο του μηχανισμού προσοχής που υλοποιείται στα δίκτυα αυτά, είναι υπολογιστικά αποδοτικό, αφού χάρη σε αυτό αποφεύγονται οι πολύπλοκες πράξεις με πίνακες, συνίσταται η εφαρμογή του σε κόμβους διαφορετικού είδους και με μεταβλητό αριθμό γειτόνων, ενώ δεν απαιτείται η γνώση ολόκληρου του γράφου εκ των προτέρων. Χάρη στις ιδιότητες αυτές, αποφεύγει πολλές θεωρητικές απλοποιήσεις και παραδοχές, που είχαν εφαρμοστεί στο spectral convolutional GCN και αποδίδει καλύτερα από κάθε μοντέλο που έχει προταθεί στην παρούσα διπλωματική. Συνεπώς, θα αποτελέσει το μοντέλου που προτείνεται για την ανίχνευση ψευδών ειδήσεων στο κοινωνικό δίκτυο του Twitter, εργαζόμενοι στην βάση δεδομένων των *FakeNewsNet*.

Και σε αυτήν την περίπτωση, υπάρχουν ωστόσο κάποιες επεκτάσεις, που δύναται να λάβει το μοντέλο προκειμένου να ενισχυθεί η απόδοσή του. Η αύξηση του μέγιστου batch size, εξέλιξη του μοντέλου ώστε να ταξινομεί ολόκληρους γράφους και να αντιλαμβάνεται το είδος σύνδεσης που προσφέρουν οι ακμές, αναλογιζόμενο τα χαρακτηριστικά και τις σχέσεις των κόμβων, αποτελούν κάποιες από τις προκλήσεις του μέλλοντος για την περαιτέρω βελτιστοποίηση των Graph Attentional Networks.

11. Συμπεράσματα και Μελλοντική Δουλειά

Το UPFD *framework* αποτελεί μία καινοτόμα προσέγγιση για το πρόβλημα της ανίχνευσης ψευδών ειδήσεων στα κοινωνικά δίκτυα. Το GCN μοντέλο που παρουσιάστηκε, χρησιμοποιεί έναν αποδοτικό κανόνα διάδοσης, ανά επίπεδο, βασισμένο στην πρώτης τάξης ($K = 1$, διάδοση κατά τους πρώτους άμεσους γείτονες) προσέγγιση των *spectral convolutions* στους γράφους. Τα πειράματά μας στη βάση δεδομένων, απέδειξαν ότι το προτεινόμενο GCN μοντέλο είναι ικανό να

κωδικοποιήσει τόσο τη δομή του γράφου, όσο και τα χαρακτηριστικά των κόμβων, ξεπερνώντας σε επιδόσεις τα υπόλοιπα νευρωνικά δίκτυα, όντας παράλληλα υπολογιστικά οικονομικό.

Όπως έγινε αντιληπτό από τα παραπάνω αποτελέσματα, η μέθοδος UPFD είναι αυτή που ξεπερνάει τις baseline τεχνικές, όπως επίσης και τις σύγχρονες μεθόδους των GNNs. Για το σκοπό αυτό, ενισχύουμε το προηγούμενο GCN μοντέλο, μελετώντας δύο νέα GCN μοντέλα, για την ανίχνευση των ψευδών ειδήσεων, βασισμένα στα *user-preferences* και *user-profile* χαρακτηριστικά του UPFD framework.

Όπως ήταν αναμενόμενο, το μοντέλο convolutional GraphSAGE υπερβαίνει τις επιδόσεις του προηγούμενου μοντέλου. Τα embeddings δημιουργήθηκαν με αποδοτικό τρόπο από τους μη εξερευνημένους κόμβους, χάρη στη μέθοδο της επαγωγικής μάθησης. Τα πιο υψηλά αποτελέσματα, έχουν ως συνέπεια την αύξηση του χρόνου που απαιτείται προκειμένου το μοντέλο να κάνει προβλέψεις, λόγω της δειγματοληψίας των γειτόνων των κόμβων και της μάθησης της δομής του γράφου. Πιθανές επεκτάσεις που θα μπορούσαν να δοθούν και βελτιστοποιήσουν περαιτέρω το μοντέλο είναι να δέχεται κατευθυνόμενους γράφους, όπως επίσης οι συναρτήσεις που χρησιμοποιούνται για τη συλλογή των γειτόνων να είναι μη κανονικής κατανομής.

Τα Graph Attention Networks (GATs) ως σύγχρονα συνελικτικά νευρωνικά δίκτυα που εφαρμόζονται σε δεδομένα μορφής γράφου, αποτελούμενα από *masked self-attentional layers* σημειώνουν υψηλότατες επιδόσεις στον τομέα ανίχνευσης ψευδών ειδήσεων . Το επίπεδο του μηχανισμού προσοχής που υλοποιείται στα δίκτυα αυτά, είναι υπολογιστικά αποδοτικό, αφού χάρη σε αυτό αποφεύγονται οι πολύπλοκες πράξεις με πίνακες, συνίσταται η εφαρμογή του σε κόμβους διαφορετικού είδους και με μεταβλητό αριθμό γειτόνων, ενώ δεν απαιτείται η γνώση ολόκληρου του γράφου εκ των προτέρων. Χάρη στις ιδιότητες αυτές, αποφεύγει πολλές θεωρητικές απλοποιήσεις και παραδοχές, που είχαν εφαρμοστεί στο spectral convolutional GCN και αποδίδει καλύτερα από κάθε μοντέλο που έχει προταθεί στην παρούσα διπλωματική. Συνεπώς, θα αποτελέσει το μοντέλου που προτείνεται για την ανίχνευση ψευδών ειδήσεων στο κοινωνικό δίκτυο του Twitter, εργαζόμενοι στην βάση δεδομένων των *FakeNewsNet*.

Και σε αυτήν την περίπτωση, υπάρχουν ωστόσο κάποιες επεκτάσεις, που δύναται να λάβει το μοντέλο προκειμένου να ενισχυθεί η απόδοσή του. Η αύξηση του μέγιστου batch size, εξέλιξη του μοντέλου ώστε να ταξινομεί ολόκληρους γράφους και να αντιλαμβάνεται το είδος σύνδεσης που προσφέρουν οι ακμές, αναλογιζόμενοι τα χαρακτηριστικά και τις σχέσεις των κόμβων, αποτελούν κάποιες από τις προκλήσεις του μέλλοντος για περαιτέρω βελτιστοποίηση των Graph Attention Networks.

12. Κώδικας Υλοποίησης, Εκπαίδευσης και Αξιολόγησης Νευρωνικού Δικτύου

```python
import argparse
import time
from tqdm import tqdm
import copy as cp
import torch.nn.functional as F
from torch.utils.data import random_split
from torch_geometric.data import DataLoader, DataListLoader
from torch_geometric.nn import DataParallel
from torch.nn import Linear
from torch_geometric.nn import global_mean_pool, GATConv

from utils.data_loader import *
from sklearn.metrics import f1_score, accuracy_score, recall_score, precision_score, roc_auc_score, average_precision_score



def eval_deep(log, loader):
    """
    Evaluating the classification performance given mini-batch data
    """

    # get the empirical batch_size for each mini-batch
    data_size = len(loader.dataset.indices)
    batch_size = loader.batch_size
    if data_size % batch_size == 0:
        size_list = [batch_size] * (data_size//batch_size)
    else:
        size_list = [batch_size] * (data_size // batch_size) + [data_size % batch_size]

    assert len(log) == len(size_list)

    accuracy, f1_macro, f1_micro, precision, recall = 0, 0, 0, 0, 0

    prob_log, label_log = [], []

    for batch, size in zip(log, size_list):
        pred_y, y = batch[0].data.cpu().numpy().argmax(axis=1), batch[1].data.cpu().numpy().tolist()
        prob_log.extend(batch[0].data.cpu().numpy()[:, 1].tolist())
        label_log.extend(y)

        accuracy += accuracy_score(y, pred_y) * size
        f1_macro += f1_score(y, pred_y, average='macro') * size
        f1_micro += f1_score(y, pred_y, average='micro') * size
        precision += precision_score(y, pred_y, zero_division=0) * size
        recall += recall_score(y, pred_y, zero_division=0) * size

    auc = roc_auc_score(label_log, prob_log)
    ap = average_precision_score(label_log, prob_log)

    return accuracy/data_size, f1_macro/data_size, f1_micro/data_size, precision/data_size, recall/data_size, auc, ap



class Net(torch.nn.Module):
    def __init__(self, information_fusion=True):
        super(Net, self).__init__()

        self.num_classes = args.num_classes
        self.num_hidden = args.num_hidden
        self.num_features = dataset.num_features
        self.information_fusion = information_fusion

        self.conv1 = GCNConv(self.num_features, self.num_hidden * 2)
        self.conv2 = GCNConv(self.num_hidden * 2, self.num_hidden * 2)

        self.fc1 = Linear(self.num_hidden * 2, self.num_hidden)

        if self.information_fusion:
            self.fc0 = Linear(self.num_features, self.num_hidden)
            self.fc1 = Linear(self.num_hidden * 2, self.num_hidden)

        self.fc2 = Linear(self.num_hidden, self.num_classes)

```

```
    def forward(self, data):
        x, edge_index, batch = data.x, data.edge_index, data.batch

        x = F.relu(self.conv1(x, edge_index))
        x = F.relu(self.conv2(x, edge_index))
        x = F.relu(global_mean_pool(x, batch))
        x = F.relu(self.fc1(x))
        x = F.dropout(x, p=0.5, training=self.training)

        if self.information_fusion:
            news = torch.stack([data.x[(data.batch == idx).nonzero().squeeze()[0]] for idx in range(data.num_graphs)])
            news = F.relu(self.fc0(news))
            x = torch.cat([x, news], dim=1)
            x = F.relu(self.fc1(x))

        x = F.log_softmax(self.fc2(x), dim=-1)

        return x

@torch.no_grad()
def compute_test(loader, verbose=False):
    model.eval()
    loss_test = 0.0
    out_log = []
    for data in loader:
        if not args.multi_gpu:
            data = data.to(args.device)
        pred = model(data)
        if args.multi_gpu:
            y = torch.cat([d.y.unsqueeze(0) for d in data]).squeeze().to(pred.device)
        else:
            y = data.y
        if verbose:
            print(F.softmax(pred, dim=1).cpu().numpy())
        out_log.append([F.softmax(pred, dim=1), y])
        loss_test += F.nll_loss(pred, y).item()
```

```
    return eval_deep(out_log, loader), loss_test


parser = argparse.ArgumentParser()

# original model parameters
parser.add_argument('--seed', type=int, default=777, help='random seed')
parser.add_argument('--device', type=str, default='cuda:0', help='specify cuda devices')

# hyper-parameters
parser.add_argument('--dataset', type=str, default='politifact', help='[politifact, gossipcop]')
parser.add_argument('--batch_size', type=int, default=128, help='batch size')
parser.add_argument('--lr', type=float, default=0.001, help='learning rate')
parser.add_argument('--weight_decay', type=float, default=0.01, help='weight decay')
parser.add_argument('--num_hidden', type=int, default=128, help='hidden size')
parser.add_argument('--epochs', type=int, default=60, help='maximum number of epochs')
parser.add_argument('--information_fusion', type=bool, default=False, help='whether concatenate news embedding and graph embedding')
parser.add_argument('--multi_gpu', type=bool, default=False, help='multi-gpu mode')
parser.add_argument('--feature', type=str, default='spacy', help='feature type, [profile, spacy, bert, content]')

args = parser.parse_args()
torch.manual_seed(args.seed)
if torch.cuda.is_available():
    torch.cuda.manual_seed(args.seed)

dataset = FNNDataset(root='data', feature=args.feature, empty=False, name=args.dataset, transform=ToUndirected())

args.num_classes = dataset.num_classes
args.num_features = dataset.num_features

print(args)

num_training = int(len(dataset) * 0.2)
num_val = int(len(dataset) * 0.1)
num_test = len(dataset) - (num_training + num_val)
training_set, validation_set, test_set = random_split(dataset, [num_training, num_val, num_test])

```

```
148 if args.multi_gpu:
149     loader = DataListLoader
150 else:
151     loader = DataLoader
152 
153 train_loader = loader(training_set, batch_size=args.batch_size, shuffle=True)
154 val_loader = loader(validation_set, batch_size=args.batch_size, shuffle=False)
155 test_loader = loader(test_set, batch_size=args.batch_size, shuffle=False)
156 
157 model = Net(information_fusion=args.information_fusion).to(args.device)
158 if args.multi_gpu:
159     model = DataParallel(model)
160 model = model.to(args.device)
161 optimizer = torch.optim.SGD(model.parameters(), lr=args.lr, weight_decay=args.weight_decay)
162 
163 
164 if __name__ == '__main__':
165     # Model training
166     t = time.time()
167     model.train()
168     for epoch in tqdm(range(args.epochs)):
169         out_log = []
170         loss_train = 0.0
171         for i, data in enumerate(train_loader):
172             optimizer.zero_grad()
173             if not args.multi_gpu:
174                 data = data.to(args.device)
175             out = model(data)
176             if args.multi_gpu:
177                 y = torch.cat([d.y.unsqueeze(0) for d in data]).squeeze().to(out.device)
178             else:
179                 y = data.y
180             loss = F.nll_loss(out, y)
181             loss.backward()
182             optimizer.step()
183             loss_train += loss.item()
184             out_log.append([F.softmax(out, dim=1), y])
```

```
185         acc_train, _, _, _, recall_train, auc_train, _ = eval_deep(out_log, train_loader)
186         [acc_val, _, _, _, recall_val, auc_val, _], loss_val = compute_test(val_loader)
187         print(f'loss_train: {loss_train:.4f}, acc_train: {acc_train:.4f},'
188               f' recall_train: {recall_train:.4f}, auc_train: {auc_train:.4f},'
189               f' loss_val: {loss_val:.4f}, acc_val: {acc_val:.4f},'
190               f' recall_val: {recall_val:.4f}, auc_val: {auc_val:.4f}')
191 
192     [acc, f1_macro, f1_micro, precision, recall, auc, ap], test_loss = compute_test(test_loader, verbose=False)
193     print(f'Test set results: acc: {acc:.4f}, f1_macro: {f1_macro:.4f}, f1_micro: {f1_micro:.4f}, '
194           f'precision: {precision:.4f}, recall: {recall:.4f}, auc: {auc:.4f}, ap: {ap:.4f}')
```

13. Βιβλιογραφία

[1]https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/.

[2] https://reutersinstitute.politics.ox.ac.uk/digital-news-report/2021/greece.

[3]https://www.pewresearch.org/journalism/2021/09/20/news-consumption-across-social-media-in-2021/.

[4] https://www.reuters.com/article/us-usa-cyber-twitter-idUSKCN1GK2QQ.

[5] https://www.statista.com/statistics/657111/fake-news-sharing-online/#statisticContainer.

[6]Bottou, Léon, and Olivier Bousquet. "The tradeoffs of large scale learning." *Advances in neural information processing systems* 20 (2007).

[7] Mitchell, Tom M., and Tom M. Mitchell. *Machine learning*. Vol. 1. No. 9. New York: McGraw-hill, 1997.

[8] Bishop, Christopher M., and Nasser M. Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. No. 4. New York: springer, 2006.

[9] Zhou, Victor. "Machine learning for beginners: An introduction to neural networks." *Towards Data Science* (2019).

[10] Duda, Richard O., and Peter E. Hart. "Pattern recognition and scene analysis." (1973).

[11]https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a. Accessed 23 August 2022.

[12] Bottou, Léon, and Olivier Bousquet. "The tradeoffs of large scale learning." *Advances in neural information processing systems* 20 (2012).

[13] Ferguson, Thomas S. "An inconsistent maximum likelihood estimate." *Journal of the American Statistical Association* 77.380 (1982): 831-834.

[14] Bilmes, Jeff, et al. "Using PHiPAC to speed error back-propagation learning." *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 5. IEEE, 1997.

[15] Luger, George F. *Artificial intelligence: structures and strategies for complex problem solving*. Pearson education, 2005.

[16] Nilsson, Nils J., and Nils Johan Nilsson. *Artificial intelligence: a new synthesis*. Morgan Kaufmann, 1998.

[17] Russell, Stuart J. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[18] Alvarado, Matías. "Computational intelligence: a logical approach." *Computación y Sistemas* 2.002 (1999).

[19] Frasconi, Paolo, Marco Gori, and Alessandro Sperduti. "A general framework for adaptive processing of data structures." *IEEE transactions on Neural Networks* 9.5 (1998): 768-786.

[20] Gori, Marco, Gabriele Monfardini, and Franco Scarselli. "A new model for learning in graph domains." *Proceedings. 2005 IEEE international joint conference on neural networks*. Vol. 2. No. 2005. 2005.

[21] Yujia, Li, et al. "Gated graph sequence neural networks." *International Conference on Learning Representations*. 2016.

[22] Spectral networks and locally connected networks on graphs. International Conference on Learning Representations

[23] Henaff, Mikael, Joan Bruna, and Yann LeCun. "Deep convolutional networks on graph-structured data." *arXiv preprint arXiv:1506.05163* (2015).

[24] Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering." *Advances in neural information processing systems* 29 (2016).

[25] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907* (2016).

[26] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan´ Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In Advances in neural information processing systems, pp. 2224–2232, 2015

[27] Niepert, Mathias, Mohamed Ahmed, and Konstantin Kutzkov. "Learning convolutional neural networks for graphs." *International conference on machine learning*. PMLR, 2016.

[28] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M ` Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. arXiv preprint arXiv:1611.08402, 2016.

[29] Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." *Advances in neural information processing systems* 30 (2017).

[30] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

[31]https://towardsdatascience.com/spectral-graph-convolution-explained-and-implemented-step-by-step-2e495b57f801

[32] Rappaport, Theodore S. *Wireless communications: principles and practice*. Vol. 2. New Jersey: prentice hall PTR, 1996.

[33] Gori, Marco, Gabriele Monfardini, and Franco Scarselli. "A new model for learning in graph domains." *Proceedings. 2005 IEEE international joint conference on neural networks*. Vol. 2. No. 2005. 2005.

[34] Scarselli, Franco, et al. "The graph neural network model." *IEEE transactions on neural networks* 20.1 (2008): 61-80.

[35] Yujia, Li, et al. "Gated graph sequence neural networks." *International Conference on Learning Representations*. 2016.

[36] Duvenaud, David K., et al. "Convolutional networks on graphs for learning molecular fingerprints." *Advances in neural information processing systems* 28 (2015), pp. 2224–2232, 2015.

[37]Atwood, James, and Don Towsley. "Diffusion-convolutional neural networks." *Advances in neural information processing systems* 29 (2016).

[38] Niepert, Mathias, Mohamed Ahmed, and Konstantin Kutzkov. "Learning convolutional neural networks for graphs." *International conference on machine learning*. PMLR, 2016.

[39] Bruna, Joan, et al. "Spectral networks and locally connected networks on graphs." *arXiv preprint arXiv:1312.6203* (2013).

[40] Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering." *Advances in neural information processing systems* 29 (2016).

[41]https://towardsdatascience.com/spectral-graph-convolution-explained-and-implemented-step-by-step-2e495b57f801.

[42]https://towardsdatascience.com/understanding-graph-convolutional-networks-for-node-classification-a2bfdb7aba7b.

[43] Sanchez-Lengeling, Benjamin, et al. "A gentle introduction to graph neural networks." *Distill* 6.9 (2021): e33.

[44] Jurafsky, Daniel, and James H. Martin. "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.".

[45] Sahlgren, Magnus. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Diss. Institutionen för lingvistik, 2006.

[46] https://towardsdatascience.com/word2vec-explained-49c52b4ccb71.

[47] https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314.

[48] Devlin, B. E. R. T., et al. "pre-training of deep bidirectional transformers for language understanding, arXiv." *arXiv preprint arXiv:1810.04805* (2018).

[49] Velickovic, Petar, et al. "Graph attention networks." *stat* 1050 (2017): 20.

[50] Iraklis Varlamis, Dimitrios Michail, Foteini Glykou and Panagiotis Tsantilas, et al. "A Survey on the Use of Graph Convolutional Networks for Combating Fake News." *Future Internet* 14.3 (2022): 70.

[51] Dou, Y., Shu, K., Xia, C., Yu, P. S., & Sun, L. (2021, July). User preference-aware fake news detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2051-2055).

[52] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2018). Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*.

[53]https://www.researchgate.net/publication/344717762_Machine_Learning_Algorithms_-A_Review

[54] https://pypi.org/project/pyswarms/

[55] Grossi, Enzo, and Massimo Buscema. "Introduction to artificial neural networks." *European journal of gastroenterology & hepatology* 19.12 (2007): 1046-1054.

[56] Maind, Sonali B., and Priyanka Wankar. "Research paper on basic of artificial neural network." *International Journal on Recent and Innovation Trends in Computing and Communication* 2.1 (2014): 96-100.

[57]https://www.researchgate.net/publication/339446790_Using_a_Data_Driven_Approach_to_Predict_Waves_Generated_by_Gravity_Driven_Mass_Flows

[58] https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/

[59] Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." *Physica D: Nonlinear Phenomena* 404 (2020): 132306.

[60] Wang, Hanzhi, et al. "Approximate graph propagation." Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021.

[61] https://distill.pub/2021/gnn-intro/

[62] https://distill.pub/2021/understanding-gnns/

[63] Aziz, F., Akbar, M. S., Jawad, M., Malik, A. H., Uddin, M. I., & Gkoutos, G. V. (2021). Graph characterisation using graphlet-based entropies. *Pattern Recognition Letters*, *147*, 100-107.

[64] https://www.cloudflare.com/learning/bots/what-is-a-bot/

[65] Davis, C. A., Varol, O., Ferrara, E., Flammini, A., & Menczer, F. (2016, April). Botornot: A system to evaluate social bots. In *Proceedings of the 25th international conference companion on world wide web* (pp. 273-274).

[66] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[67]Popescu, M. C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. (2009). Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, *8*(7), 579-588.

[68] https://www.w3schools.com/ai/ai_perceptrons.asp

[69] https://www.ucf.edu/news/how-fake-news-affects-u-s-elections/

[70] https://github.com/IUNetSci/botometer-python

[71] https://eclass.upatras.gr/modules/document/file.php

[72] https://www.educative.io/answers/what-is-the-f1-score