ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τίτλος Εργασίας

Ανάπτυξη συστήματος που θα διαχωρίζει αληθείς από ψευδείς ειδήσεις αξιοποιώντας πολλαπλές πηγές

Όνομα φοιτητή

Στεφανία Μπασούκου

Αθήνα, 2018

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τριμελής Εξεταστική Επιτροπή

Βαρλάμης Ηρακλής

Επίκουρος Καθηγητής, Τμήματος Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Τσερπές Κωνσταντίνος
Επίκουρος Καθηγητής, Τμήματος Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Δημητρακόπουλος Γεώργιος
Επίκουρος Καθηγητής, Τμήματος Πληροφορικής και Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Η Στεφανία Μπασούκου

δηλώνω υπεύθυνα ότι:

1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.

2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέποντα καθηγητή Ηρακλή Βαρλάμη για την ολοκλήρωση της πτυχιακής μου. Οι υποδείξεις και οι συμβουλές του με κατεύθυναν , ώστε να αποκτήσω τις απαραίτητες γνώσεις για την συγκεκριμένη εργασία.

Επίσης, θα ήθελα να ευχαριστήσω όλους τους καθηγητές του τμήματος Πληροφορικής και Τηλεματικής για τις γνώσεις και τις ευκαιρίες που μου προσέφεραν.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξη και την εμπιστοσύνη τους προς εμένα.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη στα Ελληνικά

Στις μέρες μας, υπάρχει μια μεγάλη και συνεχόμενη αναπαραγωγή πληροφοριών, λόγω τις εύκολης δημοσίευσης απόψεων και ειδήσεων στο διαδίκτυο. Με αυτόν τον τρόπο ο αναγνώστης δυσκολεύεται και οδηγείται σε λάθος συμπεράσματα με βάση το άρθρο που διαβάζει. Η συγκεκριμένη εργασία πραγματεύεται την κατηγοριοποίηση των ειδήσεων σε αληθείς ή ψευδείς. Έχει ως στόχο να προσδιορίσει τα γνωρίσματα που ενδέχεται να ξεχωρίζουν μια πραγματική από μια πλαστή είδηση και στη συνέχεια να φτιάξει ένα μοντέλο που θα διαχωρίζει τις ειδήσεις. Αρχικά, συγκεντρώθηκαν έρευνες που είχαν ως θέμα την ανίχνευση ψευδών ειδήσεων, στις οποίες παρατηρήθηκαν τα χαρακτηριστικά και οι τεχνικές που χρησιμοποιούν. Οι έρευνες χωρίζονται σε τρεις κατηγορίες, αυτές που υποστηρίζουν την εγκυρότητα ενός κειμένου με βάση τις πηγές του, αυτές που βασίζονται στο κείμενο, και εκείνες που συνδυάζουν και τις δύο τεχνικές. Τα συμπεράσματα ήταν ότι η έρευνα πρέπει να βασιστεί στο κείμενο και τις πηγές ενός άρθρου. Όσον αφορά την υλοποίηση, συγκεντρώθηκαν, με αυτόματο τρόπο, χαρακτηριστικά για 100 αληθινά και 100 ψευδή άρθρα. Το σύνολο των δεδομένων αποθηκεύτηκε σε ένα αρχείο csv , ώστε να επεξεργαστεί για να βγουν τα απαραίτητα συμπεράσματα. Μέσω του εργαλείου knime, έγινε η προ-επεξεργασία των δεδομένων και η κατηγοριοποίηση αυτών. Για τους διάφορους αλγορίθμους, πήραμε το ποσοστό της ακρίβειας που πέτυχαν. Έπειτα, υπάρχει σύγκριση των αλγορίθμων και οι δύο αλγόριθμοι που είχαν την μεγαλύτερη ακρίβεια, σε αυτό το σύνολο δεδομένων, ήταν ο Decision Tree και ο Gradient Boosted Trees με ποσοστό 69,667%. Τέλος, αναλύονται οι προτάσεις για βελτίωση της απόδοσης του μοντέλου που χρησιμοποιήθηκε, οι οποίες αφορούν τους classifiers και το dataset.

Λέξεις κλειδιά: [αληθείς, ψευδείς, ειδήσεις, άρθρο, κατηγοριοποίηση]

Abstract ή Περίληψη στα Αγγλικά

Nowadays, there is a large and continuous reproduction of information, due to publishing easily opinions and news on the internet. In this way, the reader finds it difficult to understand the truth and lead to wrong conclusions based on the article he reads. This particular paper deals with the detection of fake news and classifying them as true or false. It aims to identify the features that may stand out true from a fake news and then make a model that will separate them. Initially, investigations that detect fake news were gathered , in order to observe which features and techniques should be used. Researches are divided into three categories, one that supports the validity of a text based on its sources, those based on the text, and those that combine both techniques. The conclusions were that our research should be based on the text and sources of an article. In terms of implementation, automatically gathered features for 100 true and 100 fake articles. The set of data was stored in an csv file so that it could be processed to get the necessary conclusions. Through the knime tool, data was pre-processed and categorized. For the various algorithms, we got the percentage of accuracy they achieved. Then there is a comparison of the algorithms and the two most accurate algorithms in this dataset were the Decision Tree and the Gradient Boosted Trees with 69,667%. Finally, we analyze the proposals for improving the performance of the model that used, which concern the classifiers and the dataset.

Keywords: [true, fake, news, classification, detection]

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

et al.	Η συντομογραφια et al. βγαίνει από το et alii και σημαίνει 'και άλλοι'.
Π.χ.	Σημαίνει παραδείγματος χάρη.
κλπ	Σημαίνει και τα λοιπά.
RTS	Σημαίνει Rhetorical Structure Theorh.
VSM	Σημαίνει Vector Space Modeling.

1. Εισαγωγή/Σκοπός

1.1 Εισαγωγή

Στις μέρες μας, η πληροφόρηση μεγάλου αριθμού ανθρώπων γίνεται μέσω του διαδικτύου. Ο όγκος και η συχνότητα με την οποία δημοσιεύονται οι ειδήσεις επιτρέπει σε πολλούς να διαδίδουν ψευδείς ειδήσεις, οι οποίες είναι δύσκολο να αναγνωριστούν από τους αναγνώστες. Γι'αυτό δημιουργήθηκε η ανάγκη για την εύρεση αυτοματοποιημένων τρόπων ανίχνευσης παραπλανητικών ειδήσεων.

Για να κατανοήσουμε το υπάρχον σύστημα πληροφόρησης, θα πρέπει να το χωρίσουμε σε τρία στοιχεία: (1) Οι διαφορετικοί τύποι περιεχομένου που δημιουργούνται και δημοσιεύονται (2) Τα κίνητρα αυτών που δημιουργούν αυτό το περιεχόμενο (3) Οι τρόποι με τους οποίους διαδίδεται αυτό το περιεχόμενο.

Το πρώτο στοιχείο, οι διαφορετικοί τύποι περιεχομένου, που αφορούν παραπληροφόρηση είναι η ψευδής σύνδεση, τα ψευδή συμφραζόμενα, το χειραγωγούμενο περιεχόμενο, η σάτιρα ή η παρωδία, το παραπλανητικό περιεχόμενο, το απατηλό περιεχόμενο και το κατασκευασμένο περιεχόμενο. Η ψευδής σύνδεση είναι όταν οι τίτλοι, οι εικόνες ή οι λεζάντες δεν υποστηρίζουν το περιεχόμενο, τα ψευδή συμφραζόμενα είναι όταν γνήσιο περιεχόμενο μοιράζεται με ψευδείς πληροφορίες , και το χειραγωγούμενο περιεχόμενο όταν γνήσιες πληροφορίες ή εικόνες χρησιμοποιούνται για να εξαπατήσουν. Επίσης, στη σάτιρα ή παρωδία δεν υπάρχει πρόθεση να προκληθεί βλάβη αλλά έχει τη δυνατότητα να ξεγελάσει, το παραπλανητικό περιεχόμενο είναι η παραπλανητική χρήση πληροφοριών για να πλαισιωθεί ένα θέμα ή ένα άτομο. Επιπλέον, το απατηλό περιεχόμενο είναι όταν υποδύονται γνήσιες πηγές και το κατασκευασμένο περιεχόμενο όταν νέο περιεχόμενο, που είναι 100% ψευδές, έχει σκοπό να παραπλανήσει και να κάνει κακό.

Αυτές οι παραπλανητικές ειδήσεις δημιουργούνται για κάποιους λόγους. Τα κίνητρα όσων δημιουργούν “ψευδείς ειδήσεις” είναι η κακή δημοσιογραφία, η παρωδία, η πρόκληση, το πάθος, ο κομματισμός, το κέρδος, η πολιτική επιρροή-δύναμη και η προπαγάνδα.

Το παραπλανητικό αυτό περιεχόμενο διαδίδεται με διάφορους τρόπους. Μέρος αυτού δημοσιεύεται άθελά από τους ανθρώπους, στα μέσα κοινωνικής δικτύωσης, όπως κάνοντας κλικ σε retweets και κοινοποιήσεις, χωρίς έλεγχο. Κάποια μεταδίδονται μέσω των δημοσιογράφων που βρίσκονται υπό πίεση, καθώς προσπαθούν να κατανοήσουν και να αναφερθούν σε πληροφορίες που δημοσιεύονται καταιγιστικά στα μέσα κοινωνικής δικτύωσης, σε πραγματικό χρόνο. Τέλος, κάποιες παραπλανητικές ειδήσεις διαδίδονται από χαλαρά συνδεδεμένες ομάδες ανθρώπων που έχουν σκοπό να επηρεάσουν την κοινή γνώμη, είτε trolls, δηλαδή άνθρωποι με σκοπό να προσβάλουν την εικόνα κάποιου με ψεύδη, ή bot συνήθως γράφουν αυτόματα περιεχόμενο για να δείχνουν κινητικότητα ή για να προκαλέσουν κίνηση στους αναγνώστες.

Μέσα από αυτά τα στοιχεία μπορούμε να καταλάβουμε πόσο εύκολα μπορούν να διαδοθούν φήμες και ανακρίβειες, τις οποίες δεν μπορούμε να εντοπίσουμε εύκολα και την ανάγκη για την ανίχνευση αυτών.

1.2 Σκοπός

Αυτή η εργασία έχει ως στόχο να προσδιορίσει τα γνωρίσματα που ενδέχεται να ξεχωρίζουν μια πραγματική από μια πλαστή είδηση και στη συνέχεια να φτιάξει ένα μοντέλο που θα κατηγοριοποιεί ειδήσεις σε ψευδείς ή αληθείς. Τα παραπάνω θα μπορούν να πραγματοποιηθούν βρίσκοντας τα δεδομένα και τις τεχνικές που χρησιμοποιούν παρόμοιες έρευνες και καταλήγοντας στα δικά μας δεδομένα και μεθόδους που μπορούν να μας βοηθήσουν να δημιουργήσουμε το δικό μας μοντέλο.

2. Ερευνητικό υπόβαθρο

Για το σκοπό της έρευνας συλλέξαμε και αξιολογήσαμε ως αληθή και ψευδή ορισμένα άρθρα. Τα άρθρα αυτά χρησιμοποιήθηκαν ως σύνολα εκπαίδευσης και επαλήθευσης των τεχνικών μας μέχρι να καταλήξουμε στις τεχνικές που θα χρησιμοποιήσουμε για να συμπεράνουμε ότι μια είδηση είναι αληθής ή ψευδής. Κάποια άρθρα αναφέρονται στις μεθόδους και άλλα στα δεδομένα. Οι μέθοδοι που παρατηρήσαμε ακολουθούν τρεις διαφορετικές κατευθύνσεις, που βασίζονται είτε στο κείμενο ή στις πηγές ή σε συνδυασμό και των δύο.

2.1 Έλεγχος της αξιοπιστίας της είδησης συνολικά ή της πηγής

2.1.1 Κατεύθυνση Α: με βάση το κείμενο

Η έρευνα των Rubin et al (2016) αναφέρεται στην παραπλάνηση σε μορφή σάτιρας και είναι βασισμένη στον εντοπισμό παραπλάνησης σε επίπεδο κειμένου. Η μέθοδος που προτείνουν οι συγγραφείς προσπαθεί να ανιχνεύσει τις διαφορές άρθρων με σατιρικό και αληθινό περιεχόμενο. Εντοπίζει τις διαφορές σε χαρακτηριστικά σε επίπεδο λέξεων, δηλαδή τη *συχνότητα λέξεων, τις σημασιολογικές κατηγορίες (γενίκευση όρων, χρονικές/χωρικές αναφορές, θετική και αρνητική πολικότητα) και χρήση υπερβολικής/ιδιαίτερης γλώσσας (πχ αισχρολογίες, αργκό)*). Επιπλέον, διαχωρίζει τη σημασιολογική ισχύς ενός κειμένου σάτιρας και πραγματικότητας, παρατηρώντας ότι *η σάτιρα περιέχει ανισορροπίες (χρονικές, εννοιολογικές)*, και εντοπίζει το χιούμορ και τα αντικρουόμενα σενάρια, τον σαρκασμό και την ειρωνεία. Επίσης, οι διαφορές που εντοπίζονται μεταξύ των άρθρων είναι ότι στη σάτιρα

επαναλαμβάνεται ο τίτλος, υπάρχουν νέες ανόμοιες οντότητες, αργκό και προσβλητικές λέξεις, και μεγαλύτερη πολυπλοκότητα προτάσεων (στο μήκος και στην νοηματική πολυπλοκότητα).

2.1.2 Κατεύθυνση Β: με βάση τις πηγές

Στην εργασία τους οι Yimin Chen et al (2015) έχουν ως στόχο δείξουν τις προϋποθέσεις που πρέπει να τηρούν τα δεδομένα και τους τύπους ψευδών ειδήσεων ώστε να εντοπιστούν. Τα δεδομένα πρέπει να ακολουθούν τις παρακάτω προϋποθέσεις:

1. Διαθεσιμότητα αληθινών και ψευδών στιγμιοτύπων. Θα πρέπει να εντοπιστούν μοτίβα στα θετικά και τα αρνητικά σημεία των δεδομένων.

2. Προσβασιμότητα σε κείμενο ψηφιακής μορφής.

3. Επαλήθευση της αλήθειας. Επαλήθευση μιας είδησης από τις πηγές της. Οι αξιόπιστες πηγές ειδήσεων αντιστέκονται σε δοκιμασία χρόνου και έχουν μια φήμη βασισμένη σε ένα σύστημα "ελέγχων και ισορροπιών".

4. Ομογένεια σε μήκη. Ιδιαίτερη προσοχή πρέπει να δοθεί στην απόκτηση παρόμοιου μήκους δεδομένων. Για παράδειγμα, ένα σύντομο tweet με τίτλο, μια συνοπτική παρουσίαση στο Facebook και μια μακρόπνοη μορφή δεν αποτελούν ένα ομοιογενές σύνολο δεδομένων. Η κανονικοποίηση μπορεί να πραγματοποιηθεί για μερικά άνισα κατανεμημένα σύνολα δεδομένων.

5. Ομογένεια σε θέματα γραφής. Το σώμα θα πρέπει να ευθυγραμμίζεται με τα είδη ειδήσεων (π.χ. σπάνιες ειδήσεις, δημοσιεύματα, op-eds) και θέματα (π.χ. επιχειρήσεις, πολιτική, επιστήμη, υγεία), να χρησιμοποιούν παρόμοιους τύπους συγγραφέων (π.χ., επαγγελματικά εκπαιδευμένοι, επικρατέστεροι εναντίον δημοσιογράφων πολιτών, σοβαροί και χιουμοριστικοί).

6. Προκαθορισμένο χρονικό διάστημα. Όλα τα δεδομένα πρέπει να αφορούν συγκεκριμένη χρονική περίοδο.

7. Ο τρόπος χορήγησης των ειδήσεων (π.χ. χιούμορ, ειδησεογραφική αξία, εμπιστοσύνη, παραλογισμός, εντυπωσιασμός). Η γνώση του τρόπου με τον οποίο παρέχονται οι ειδήσεις δημιουργεί ένα πλαίσιο για την ερμηνεία της κατάστασης. Υποπτευόμαστε ότι οι "προκατειλημμένοι από την αλήθεια" αναγνώστες μπορεί, για παράδειγμα, να μεταβούν σε μια "ψευδής" προοπτική όταν διαβάζουν σάτιρα. Χρειάζεται περισσότερη έρευνα για την αλληλεπίδραση του χιούμορ και της εξαπάτησης.

8. Πραγματικές ανησυχίες περιλαμβάνουν τη δημόσια διαθεσιμότητα, την ευκολία απόκτησης, τον κατάλληλο συνολικό όγκο δεδομένων και το απόρρητο των συγγραφέων.

9. Γλώσσα και πολιτισμός. Πρέπει να επιλεχθεί συγκεκριμένη γλώσσα και πολιτισμός για τα δεδομένα που θα βοηθήσουν στην ανίχνευση εξαπάτησης.

Επίσης, τα είδη ψευδών ειδήσεων διαχωρίζονται σε σοβαρές σκευωρίες (Serious fabrications), μεγάλης κλίμακας απάτες (large-scale hoaxes), και χιουμοριστικές (Humorous fakes), όπως σάτιρα ειδήσεων, παρωδία, τηλεπαιχνίδια κλπ. Η μέθοδος που χρησιμοποιείται για να ανιχνευθούν αυτά τα τρία είδη ψευδών ειδήσεων είναι οι πηγές που τα παράγουν. Οι σκευωρίες συνήθως εμφανίζονται στον κίτρινο τύπο, ο οποίος παρουσιάζει ένα ευρύ φάσμα μη επαληθευμένων νέων και χρησιμοποιεί εντυπωσιακά πρωτοσέλιδα ("clickbaits"), υπερβολές, σκανδαλισμό ή αισθησιασμό για να αυξήσει την κυκλοφορία ή τα κέρδη. Η κίτρινη δημοσιογραφία είναι μια κατάλληλη πηγή για ψεύτικες ειδήσεις σε περιπτώσεις προφανών ή εκτεθειμένων πλαστογραφιών, κατασκευών ή υπερβολών και μπορεί να απαιτεί έρευνα. Επίπρόσθετα, η σάτιρα εντοπίζεται σε σελίδες γνωστές ως χιουμοριστικές.

Η έρευνα των Carlos Castillo et al (2011) έχει επικεντρωθεί στην αυτόματη αξιολόγηση της αξιοπιστίας ενός δεδομένου συνόλου των tweets. Αρχικά, χωρίζει τα δεδομένα σε τέσσερα επίπεδα αξιοπιστίας: *almost certainly true, likely to be false, almost certainly false, και "I can't decide".* Οι παράγοντες(τεχνικές) που είναι χρήσιμοι για την αναγνώριση της αξιοπιστίας των πληροφοριών στα μέσα κοινωνικής δικτύωσης:

- οι αντιδράσεις (reactions) (αν χρησιμοποιούν εκφράσεις γνώμης που αντιπροσωπεύουν θετικά ή αρνητικά συναισθήματα σχετικά με το θέμα.),
- το επίπεδο βεβαιότητας των χρηστών για τη διάδοση πληροφοριών (αν αμφισβητούν τις πληροφορίες που τους έχουν δοθεί ή όχι.),
- οι εξωτερικές πηγές(αξιόπιστες- μη αξιόπιστες) που αναφέρονται, δηλαδή αν αναφέρουν ένα συγκεκριμένο URL με τις πληροφορίες που μεταδίδουν και εάν αυτή η πηγή είναι δημοφιλής ή όχι.,
- και τα χαρακτηριστικά των χρηστών που διαδίδουν τις πληροφορίες (τον αριθμό των οπαδών που έχει κάθε χρήστης έχει στην πλατφόρμα).

Για την κατηγοριοποίηση των ειδήσεων οι συγγραφείς χρησιμοποίησαν και αξιολόγησαν διαφορετικά l*earning schemes:* SVM, decision trees, decision rules, Bayes network και ο καλύτερος αλγόριθμος ήταν *ο* J48 decision tree με ακρίβεια 89%.

Τα καλύτερα χαρακτηριστικά για τα δεδομένα:

- topic-based features*:* URL, sentiment-based features,
- user-based features: number of messages, number of friends,
- και propagation-based features: number of re-tweets.

Τέλος, πρότειναν τέσσερα υποσύνολα χαρακτηριστικών των δεδομένων που είναι χρήσιμα για τη διάκριση των ειδήσεων σε αληθείς ή ψευδείς:

- text subset: length, sentiment-based features, URLs, counting elements(eg hashtag, mention),
- network subset: author of messages, number of friends, number of followers,
- propagation subset: re-tweets, total tweets
- και top-element subset: url, hashtag, user mention, author.

Αξίζει να αναφέρουμε την παρατήρηση των συγγραφέων ότι στα μη σημαντικά User-based features περιλαμβάνονται η ηλικία του χρήστη στο μέσο (εδώ πόσο καιρό χρησιμοποιεί το twitter) και οι followers.

2.1.3 Κατεύθυνση Γ: συνδυασμός

Στο άρθρο των Niall et al (2015) χρησιμοποιούνται οι εξής τεχνικές αξιολόγησης: *προσέγγιση γλωσσικού συνθήματος (linguistic approach) και προσέγγιση ανάλυσης δικτύου (network approach).*

Πιο συγκεκριμένα στις μεθόδους γλωσσικής προσέγγισης περιλαμβάνονται οι ακόλουθες:

1. Data representation

-Χρησιμοποιείται η μέθοδος λεγόμενη “bag of words”, η οποία βασίζεται σε λέξεις και γράμματα, εντοπίζοντας συχνότητες μεμονωμένων λέξεων και “n-grams”.

2. Deep syntax

-Χρειαζόμαστε επιπλέον την σύνταξη. Μέσω τις συντακτικής ανάλυσης , οι προτάσεις γίνονται κανόνες φτιάχνοντας ένα δέντρο ανάλυσης (parse tree). Προσθέτοντας αυτή την μέθοδο πετυχαίνουμε ακρίβεια 85-91%.

3. Semantic analysis

-Αυτή η μέθοδος στηρίζεται στον βαθμό της συμβατότητας μεταξύ μιας προσωπικής εμπειρίας (π.χ., μια κριτική ξενοδοχείου) σε σύγκριση με ένα περιεχόμενο «προφίλ» (γενικό προφίλ του θέματος) που προέρχονται από μια συλλογή ανάλογων στοιχείων. Επομένως η εκτίμηση ειλικρίνειας είναι ο βαθμός συμβατότητας μεταξύ μιας ξεχωριστής πτυχής και της περιγραφής μιας γενικότερης άποψης. Η σημασιολογική ανάλυση βασίζεται επίσης στις λεξεις-κλειδιά. Προσθέτοντας αυτή την μέθοδο πετυχαίνουμε ακρίβεια 91%.

4. Rhetorical structure and discourse analysis

-Περιγράφει τις ρητορικές σχέσεις μεταξύ γλωσσικών στοιχείων, δηλαδή βασίζεται στη συνοχή και τη δομή του κειμένου.

5. Classifiers (Support Vector machine και Bayesian model.)

Αντίστοιχα στις μεθόδους ανάλυσης δικτύου (network approach) προτείνονται οι:

1. Linked data: Εδώ γίνεται χρήση άρθρων του δικτύου που συνδέονται μεταξύ τους και αξιοποιείται η σχέση τους. Ο έλεγχος γεγονότος μπορεί να μειωθεί αποτελεσματικά σε ένα απλό πρόβλημα ανάλυσης δικτύου: τον υπολογισμό του απλού συντομότερου μονοπατιού. Τα ερωτήματα εξάγονται με βάση τη σημασιολογική εγγύτητα σε συνάρτηση με την μεταβατική σχέση μεταξύ υποκειμένου και του κατηγορήματος μέσω άλλων κόμβων. Όσο πιο κοντά οι κόμβοι στον γράφο, τόσο μεγαλύτερη είναι η πιθανότητα ότι μια συγκεκριμένη δήλωση αντικείμενο-κατηγόρημα-αντικείμενο είναι αλήθεια. Με αυτή την μέθοδο πετυχαίνουμε ακρίβεια 61-95%.

2. Social network behavior: Ένας τρόπος ανάλυσης κειμένου με βάση το δίκτυο είναι εντοπίζοντας τις πιο σημαντικές λέξεις που συνδέονται με άλλα λόγια στο δίκτυο.

Η έρευνα της Rubin (2017) αναφέρεται στην ανίχνευση παραπλάνησης μέσω των γλωσσικών συνθηματικών. Αρχικά, χωρίζει κάποια γλωσσικά χαρακτηριστικά ως προς τις

παρακάτω δομές ανίχνευσης παραπλάνησης: quantity(word, verb, noun phrase, sentence), complexity(average number of clauses, average sentence length, average word length, average length of noun phrase, pausality), uncertainty (modifiers, modal verbs, uncertainty, other reference), non-immediacy(passive-voice, objectification, generalizing terms, self-reference, group reference), expressivity(emotiveness), diversity(lexical diversity,content word diversity, redundancy), informality(typical error ratio), specificity(spatio-temporal information, perpetual information), και affect(positive affect, negative affect).

Η μέθοδος εντοπισμού παραπλάνησης στοχεύει να εντοπίσει τις διαφορές που έχουν τα παραπλανητικά και τα βασισμένα σε αλήθεια κείμενα, μέσω των παραπάνω χαρακτηριστικών. Με αυτόν τον τρόπο παρατηρείται ότι οι απατεώνες παράγουν περισσότερες συνολικές λέξεις (ποσότητα), περισσότερες λέξεις βασισμένες στο συναίσθημα, περισσότερες αντωνυμίες γενικής γνώσης, χαμηλότερη γνωστική πολυπλοκότητα, λιγότερες λέξεις με αρνητικό συναίσθημα, λέξεις δικαίου και δισταγμού σε σχέση με αυτούς που βασίζονται στην αλήθεια. Αντίθετα, όσοι επιθυμούν να διαδώσουν αξιόπιστες ειδήσεις είναι συντομότεροι και λιγότερο πολύπλοκοι στο λεξιλόγιο και στη δομή.

Επίσης , υπάρχουν έξι προτεινόμενες κατηγορίες ελέγχου για το Twitter. Αρχικά, είναι η αξιοπιστία πηγής, αν ένας λογαριασμός ή url δημοσιεύει αληθής ή σατιρικές ειδήσεις. Επίσης, η προέλευση ταυτότητας (το όνομα, η τοποθεσία και οι επαγγελματικές πληροφορίες) η ποικιλία προέλευσης (αν προέρχεται από διαφορετικές πηγές). Μια επιπλέον κατηγορία είναι ο τόπος προέλευσης και των μαρτύρων, δηλαδή αν υπάρχει κάποιος που άκουσε ή είδε το γεγονός. Τέλος, είναι τα πιστεύω του μηνύματος (υποστήριξη, άρνηση, ερώτηση ή ουδετερότητα) και η διάδοση εκδήλωσης (retweets, mention, hashtags).

Στο άρθρο τους οι Nagura et al (2006) προτείνουν μια μέθοδο για την αξιολόγηση της αξιοπιστίας των ειδήσεων χρησιμοποιώντας τρία στοιχεία: (1) τα κοινά στοιχεία (commonality) του περιεχομένου των ειδήσεων μεταξύ των διαφόρων εκδοτών ειδήσεων (2) την αριθμητική συμφωνία,δηλαδή τις αριθμητικές τιμές που αναφέρονται στα άρθρα και (3) την αντικειμενικότητα που βασίζεται σε υποκειμενικές κερδοσκοπικές φράσεις και πηγές ειδήσεων.

Αρχικά, τα κοινά στοιχεία των άρθρων αναφέρονται στους εκδότες, όσοι περισσότεροι έδωσαν άρθρα με παρόμοιο περιεχόμενο με το αντικείμενο που αξιολογείται, τόσο υψηλότερη ήταν η αξιοπιστία. Το κοινό περιεχόμενο μεταξύ τους εντοπίζεται στη σύγκριση των άρθρων πρόταση-πρόταση για το πόσο μοιάζουν, όσες περισσότερες είναι παρόμοιες τόσο μεγαλύτερο είναι το commonality. Επίσης, για να μετρήσουμε τον βαθμό ομοιότητας, τα άρθρα που συγκρίνουμε πρέπει να είναι από διαφορετικούς εκδότες. Το δεύτερο στοιχείο αξιολόγησης αφορά τις αριθμητικές εκφράσεις, όπως "100 επιβάτες" ή "τρία κομμάτια" που εμφανίζονται σε ειδησεογραφικά δελτία. Όταν οι αριθμητικές εκφράσεις έρχονται σε αντίθεση με εκείνες των άλλων άρθρων από διαφορετικούς εκδότες ειδήσεων, η αξιοπιστία βαθμολογήθηκε χαμηλότερα. Τέλος, η αντικειμενικότητα βασίζεται στην αξιοπιστία των άρθρων, αυτά που περιέχουν υποκειμενική κερδοσκοπία αξιολογούνται διαφορετικά από αυτά που περιέχουν αντικειμενικές πηγές ειδήσεων. Η αντικειμενικότητα εντοπίζεται με δύο τρόπους, είτε με συγκεκριμένες φράσεις είτε από τις πηγές.

Όσον αφορά τις φράσεις, οι οποίες είναι στα αγγλικά, είναι χωρισμένες σε τρεις κατηγορίες ανάλογα με το σκορ αντικειμενικότητας που έχουν. Η πρώτη κατηγορία είναι οι αντικειμενικές, στην οποία ανήκουν οι φράσεις expressing-policy,guarantee-with. Η δεύτερη είναι οι κάπως κερδοσκοπικές και περιλαμβάνει τις εξής: tell, say, report, isn't it?, seem, plan, convincing, expect, prospect, policy, become, look-like, idea, attitude, information, strongly-

possible, highly-possible, motivation, prospect, outlook, assume, affirmation και objective. Η τρίτη είναι οι πολύ κερδοσκοπικές στην οποία ανήκουν οι φράσεις: unclarity, subtlety, hope, possibility, predict, plan, aim και maybe. Ο δεύτερος τρόπος εντοπισμού της αντικειμενικότητας αφορά τις πηγές, για αυτές, το σκορ αντικειμενικότητας αυξάνεται όταν δίνονται στο άρθρο. Οι πηγές μέσα στο άρθρο εντοπίζονται με τις λέξεις “from” και “according to”. Η βαθμολογία αντικειμενικότητας μετρήθηκε με βάση τύπους ειδήσεων και τις συχνότητές τους, τα είδη αυτά είναι τέσσερα: (α) πρακτορεία ειδήσεων και εκδότες, (β) κυβερνητικές υπηρεσίες, (γ) αστυνομία και (δ) τηλεόραση / ραδιόφωνο. Η εμφάνιση ειδησεογραφικών πρακτορείων όπως το "Associated Press" αύξησε την αντικειμενικότητα σε σύγκριση με την εμφάνιση του τηλεόραση/ραδιόφωνο.

Οι Byungkyu Kang et al (2012) παρουσιάζουν τρία μοντέλα αξιολόγησης ψευδών ειδήσεων στο Twitter. Το πρώτο είναι το κοινωνικό μοντέλο (social model) που επικεντρώνεται στην αξιοπιστία σε επίπεδο χρήστη, αξιοποιώντας διάφορες δυναμικές της ροής πληροφοριών στο υποκείμενο κοινωνικό γράφημα για τον υπολογισμό της αξιοπιστίας. Σημαντικό ρόλο έχει η προέλευση της πληροφορίας και η αξιοπιστία ενός χρήστη. Για παράδειγμα, όταν κάποιος χρήστης έχει ανεξήγητα πολλούς ακόλουθους μάλλον αυτοί είναι ψεύτικοι, κάτι το οποίο μπορεί να γίνει αντιληπτό όταν τα προφίλ αυτών δεν έχουν ισχυρή κοινωνική συνδεσιμότητα. Επίσης, αν κάποιος δημοσιεύει συχνά για ένα θέμα και δεν έχει πολλούς ακόλουθους είναι ύποπτο. Αντίθετα, τα retweets θεωρούνται ένδειξη αξιοπιστίας.

Το δεύτερο μοντέλο εφαρμόζει μια στρατηγική βασισμένη στο περιεχόμενο (content-based model) για τον υπολογισμό ενός υψηλότερου βαθμού αξιοπιστίας για μεμονωμένα tweets. Αυτό το μοντέλο εξετάζει το περιεχόμενο του tweet, παρατηρώντας τους αριθμητικούς και δυαδικούς δείκτες. Οι αριθμητικοί είναι: συντελεστής θετικού αισθήματος (αριθμός

θετικών λέξεων (που ταιριάζουν με το λεξικό μας)), παράγοντας αρνητικού αισθήματος (αριθμός αρνητικών λέξεων), πολικότητα συναισθημάτων (άθροισμα των λέξεων συναίσθησης με την εντονότερη βαρύτητα (x2) («πολύ», «υπερβολικά πολύ»)), αριθμός ενισχυτικών: «πολύ», «εξαιρετικά» κλπ., με βάση το λεξικό μας, αριθμός λέξεων-κλειδιών: απλή μέτρηση, με βάση το λεξικό, αριθμός δημοφιλών όρων-ειδικών θεμάτων: απλή μέτρηση, με βάση το λεξικό, αριθμός αρχειοθετημένων χαρακτήρων: απλός Αριθμός, αριθμός Urls, αριθμός θεμάτων (Όλα έχουν τουλάχιστον ένα), αριθμός αναφορών: αριθμός χρηστών που αναφέρθηκαν με '@', μήκος του Tweet (Chars), μήκος του Tweet (λέξεις). Οι δυαδικοί είναι: μόνο διευθύνσεις, όχι κείμενο, ένα retweet, ένα ερωτηματικό: '?' ή αν περιέχει κάποιο από τα Ποιος/Τι/ Πού/Γιατί/Πότε/Πως, το θαυμαστικό, πολλά θαυμαστικά/ερωτηματικά, και θετικά ή αρνητικά emoticon. Το τρίτο μοντέλο που συνδυάζει πτυχές από τα δύο μοντέλα σε μια υβριδική μέθοδο.

Τέλος, αναλύοντας τα δεδομένα καταλήγει σε κάποια συμπεράσματα για την αξιοπιστία, όπως ότι τα πιο μακροσκελή tweet είναι πιο έγκυρα. Επιπλέον, αν ο χρήστης έχει πολύ μεγάλο μέγεθος από ακόλουθους είναι αναξιόπιστος, και αν έχει λίγους που ακολουθεί και ακολουθείται δείχνει ότι είναι νέος χρήστης και κατατάσσεται σε ομάδα χαμηλής αξιοπιστίας, ενώ οι σύνδεσμοι είναι θετικοί στην αξιοπιστία.

Οι Qazvinian et al (2011) αναφέρονται στην αντιμετώπιση του προβλήματος της παραπλάνησης, με την ανίχνευσης φημών σε μικρο-ιστολόγια και διερευνούν την αποτελεσματικότητα των 3 κατηγοριών μεθόδων: με βάση το περιεχόμενο(lexical patterns, part-of-speech), με βάση το δίκτυο, και συγκεκριμένα memes(hastags,URLs) μικρο-ιστολογίων για τον ορθό προσδιορισμό των φημών. Η μέθοδος με βάση το περιεχόμενο απαρτίζεται από τα λεξικά μοτίβα και τα μέρη του λόγου. Στα λεξικά μοτίβα όλες οι λέξεις και τα τμήματα στο

tweet αναπαριστώνται όπως εμφανίζονται και χωρίζονται χρησιμοποιώντας τον χαρακτήρα διαστήματος. Στα μέρη του λόγου, όλες οι λέξεις αντικαθίστανται με τις ετικέτες, ανάλογα με το μέρος του λόγου που είναι. Για να εντοπίσουμε το part-of-speech μιας hashtag την αντιμετωπίζουμε σαν μια λέξη, παραλείποντας το σήμα tag, και στη συνέχεια να προηγείται η ετικέτα TAG /. Επίσης, εισάγουμε μια νέα ετικέτα, URL, για τις διευθύνσεις URL που εμφανίζονται σε ένα tweet. Η δεύτερη μέθοδος βασίζεται στις πηγές, αν η αναφερόμενη πηγή δημοσιεύει φήμες, τότε είναι πιθανό και η είδηση να είναι ψευδής. Τέλος, στην τρίτη μέθοδο συγκρίνουμε τα hashtags που χρησιμοποιούνται από πηγές με παραπλανητικό και μη περιεχόμενο και κρίνουμε τις πηγές μέσα από τα URLs που τις αναφέρουν.

Οι Victoria L. Rubin et al (2015) εστιάζουν στη χρήση του RTS (Rhetorical Structure Theorh) και του VSM(Vector Space Modeling). Η θεωρία της ρητορικής δομής (RST) και το μοντέλο διανυσματικού διαστήματος (VSM) είναι τα δύο θεωρητικά συστατικά που χρησιμοποιούν στην ανάλυσή τους για παραπλανητικές και ειλικρινείς ειδήσεις. Το RST χρησιμοποιείται για την ανάλυση του λόγου των ειδήσεων και το VSM χρησιμοποιείται για την ερμηνεία των χαρακτηριστικών του λόγου σε ένα αφηρημένο μαθηματικό χώρο.

Η ανάλυση της θεωρίας ρητορικής δομής (RST) καταγράφει τη συνοχή μιας ιστορίας από την άποψη των λειτουργικών σχέσεων μεταξύ διαφορετικών σημαντικών μονάδων κειμένου και περιγράφει μια ιεραρχική δομή για κάθε ιστορία. Το αποτέλεσμα είναι ότι κάθε κείμενο που αναλύεται μετατρέπεται σε ένα σύνολο ρητορικών σχέσεων που συνδέονται με έναν ιεραρχικό τρόπο με πιο σημαντικές μονάδες κειμένου να κατευθύνουν αυτό το ιεραρχικό δέντρο. Η θεωρία διαφοροποιεί τα ρητορικά μεμονωμένα μέρη ενός κειμένου, μερικά από τα οποία είναι πιο σημαντικά (πυρήνας) από τα άλλα (δορυφόρος). Τις τελευταίες δεκαετίες, οι εμπειρικές παρατηρήσεις και η προηγούμενη εμπειρική έρευνα επιβεβαίωσαν ότι οι

συγγραφείς τείνουν να δώσουν έμφαση σε ορισμένα μέρη ενός κειμένου για να εκφράσουν την πιο ουσιαστική ιδέα τους. Αυτά τα τμήματα μπορούν να αναγνωριστούν συστηματικά μέσω της ανάλυσης των ρητορικών συνδέσεων ανάμεσα σε όλο και λιγότερο σημαντικά τμήματα ενός κειμένου. Οι σχέσεις RST (π.χ., σκοπός, επεξεργασία, μη-φωνητικό αποτέλεσμα) περιγράφουν πώς τα συνδεδεμένα τμήματα κειμένου συνυπάρχουν μέσα σε μια ιεραρχική δομή δέντρου, η οποία είναι μια ποσοτικά αναπαριστωμένη RST ενός κωδικοποιημένου κειμένου.

Το μοντέλο διανυσματικού διαστήματος (VSM) χρησιμοποιείται για την αναγνώριση των συνόλων σχέσεων ρητορικής δομής. Χρησιμοποιώντας ένα μοντέλο διανυσματικού διαστήματος, κάθε κείμενο ειδήσεων μπορεί να αναπαρασταθεί ως φορείς σε ένα χώρο μεγάλης διάστασης. Στη συνέχεια, κάθε διάσταση του διανυσματικού χώρου είναι ίση με τον αριθμό των ρητορικών σχέσεων σε ένα σύνολο όλων των ειδησεογραφικών εκθέσεων που εξετάζονται. Αυτή η αντιπροσώπευση του κειμένου των ειδήσεων καθιστά το μοντέλο διανυσματικού χώρου πολύ ελκυστικό όσον αφορά την απλότητα και την εφαρμογή του. Οι ειδήσεις εκπροσωπούνται ως φορείς σε ένα μη διακριτικό χώρο. Τα κύρια υποσύνολα του χώρου ειδήσεων είναι δύο ομάδες, παραπλανητικές ειδήσεις και ειλικρινείς ειδήσεις. Το στοιχείο ενός cluster είναι μια ιστορία ειδήσεων και ένα cluster είναι ένα σύνολο στοιχείων που μοιράζονται αρκετή ομοιότητα για να ομαδοποιηθούν, ως παραπλανητικά νέα ή ειλικρινή είδηση. Δηλαδή, κάθε είδηση μπορεί να περιγραφεί από μια σειρά χαρακτηριστικών γνωρισμάτων (ρητορικές σχέσεις, συνυπάρχουσες εμφανίσεις και θέσεις σε ιεραρχική δομή). Μαζί, τα χαρακτηριστικά αυτά καθιστούν κάθε ιστορία ειδήσεων μοναδική και προσδιορίζουν την ιστορία ως μέλος μιας συγκεκριμένης ομάδας. Στην ανάλυσή μας, συγκρίνονται τα διακριτικά χαρακτηριστικά των ειδήσεων και, όταν πληρούται ένα όριο ομοιότητας, τοποθετούνται σε μία από τις δύο ομάδες, παραπλανητικές ή αληθείς. Η ανάλυση

συμπλέγματος ομοιότητας βασίζεται στις αποστάσεις μεταξύ των δειγμάτων στον αρχικό χώρο του φορέα. Τροποποιώντας το πλαίσιο ομαδοποίησης με βάση την ομοιότητα και προσαρμόζοντας τη μεθοδολογία RST-VSM στο πλαίσιο των ειδήσεων, δοκιμάζουμε πόσο καλά μπορεί να εφαρμοστεί το RST-VSM στην επαλήθευση ειδήσεων.

Τέλος, η ίδια ερευνητική εργασία των Victoria L Rubin et al (2015) προτείνει κάποιες πηγές είτε ως αξιόπιστες ή όχι. Προτεινόμενα websites καλά εδραιωμένα σε αξιοπιστία είναι: nrp.org , bbc.com , cbc.ca. Αντίθετα, ιστοσελίδες από ερασιτέχνες δημοσιογράφους είναι: the CNN's iReport.com , thirdreport.com , allvoices.com.

2.2 Έλεγχος της αξιοπιστίας των δεδομένων ενός άρθρου (Fact checking)

Οι Craig Silverman et al (2016) πήραν δείγμα από εννιά ιστοσελίδες, που θεωρούνταν έγκυρες καθώς και αρκετά δημοφιλείς, τρεις της δεξιάς, τρεις της αριστεράς και τρεις ουδέτερες. Η μέθοδος που ακολούθησαν ήταν να βάλουν ανθρώπους να βαθμολογήσουν αρκετά άρθρα από αυτές. Με αυτόν τον τρόπο πήραν δημοσιεύσεις από αυτές και τις βαθμολόγησαν χωρίζοντας τα δεδομένα στις εξής κατηγορίες: Mostly True , Mixture of True and False , Mostly False , No Factual Content. Για κάθε άρθρο βαθμολογούσαν δυο άνθρωποι και αν υπήρχε απόκλιση μεταξύ τους αξιολογούσε και τρίτος. Με αυτήν την διαδικασία έβγαιναν τα ποσοστά αξιοπιστίας για κάθε ιστοσελίδα και ένα σύνολο δεδομένων και χαρακτηριστικών για να αξιολογηθούν νέες ειδήσεις. Οι κατηγορίες δεδομένων που εφαρμόζεται αυτή η έρευνα χωρίζονται στις παρακάτω κατηγορίες: ***account_id*** (numeric): μοναδικός αριθμός, ***post_id*** (numeric): μοναδικός αριθμός, ***Category*** (String): mainstream, left, right, ***Page*** (String): ποια ιστοσελίδα είναι, ***Post URL*** (String), ***Date Published*** (DateTime) , ***Post Type*** (String): video, link, text, photo , ***Rating*** (String): no factual content, mostly true, mixture

of true and false , mostly false, ***Debate*** (String): yes, no, ***share_count*** (numeric), ***reaction_count*** (numeric), ***comment_count*** (numeric). Τέλος, το προτεινόμενο dataset του συγκεκριμένου άρθρου βρίσκεται στο παρακάτω link:

https://github.com/BuzzFeedNews/2016-10-facebook-fact-check/blob/master/data/facebook-fact-check.csv

Η Risdal (2017) αναφέρεται στην ανίχνευση ψευδών ειδήσεων και πιο συγκεκριμένα στα δεδομένα που πρέπει να συλλέξουμε για να μπορούμε να καταλήξουμε στο αν μια είδηση είναι ψεύτικη. Τα δεδομένα που πρέπει να συλλέξουμε από κάθε άρθρο είναι: ***uuid***: *ειναι το μοναδικό αναγνωριστικό για κάθε εγγραφή και είναι* String, ***ord_in_thread***: είναι Numeric, ***author***: *ο συγγραφέας (ή οι συγγραφείς) της ιστορίας και* είναι String, ***published***: *η ημερομηνία που δημοσιεύτηκε το άρθρο και* είναι DateTime, ***title***: *ο τίτλος του άρθρου και* είναι String, ***text***: *το κείμενο του άρθρου και* είναι String, ***language*** *(data from webhose.io): η γλώσσα που είναι γραμμένη το κείμενο, μπορεί να είναι διαφορετική για κάθε άρθρο, όμως τα επιλεγμένα δεδομένα είναι στα αγγλικά , και* είναι String, ***crawled***: *η ημερομηνία η οποία το άρθρο συλλέχθηκε και* είναι DateTime, ***site_url***: είναι String, ***country*** *(data from webhose.io): η χώρα που ανήκει η ιστοσελίδα και* είναι String, ***domain_rank*** *(data from webhose.io):* ένας αριθμός που *(data from webhose.io) και* είναι Numeric, ***thread_title***: επιλέγουμε να βάζουμε τον τίτλο του άρθρου *και* είναι String,

spam_score *(data from webhose.io):* το ποσοστό του σπαμ*(data from webhose.io) και* είναι Numeric, ***main_img_url***: το url *της κεντρικής εικόνας της ιστοσελίδας και* είναι String, ***replies_count***: *ο αριθμός των απαντήσεων και* είναι Numeric, ***participants_count***: *ο αριθμός των συμμετεχόντων της ιστοσελίδας* και είναι Numeric, ***likes***: αριθμός των likes στο facebook και είναι Numeric, ***comments***: αριθμός των *comments* στο facebook *και* είναι Numeric, ***shares***:

αριθμός των shares στο facebook και είναι Numeric, ***type***: *ο τύπος της ιστοσελίδας, για παράδειγμα προκατάληψη, συνωμοσία και άλλα* και είναι String. Τα παραπάνω δεδομένα φανερώνουν όσα στοιχεία πρέπει να εντοπιστούν σε μια είδηση για να καταλήξουμε στο συμπέρασμα ότι είναι παραπλανητική ή αληθής. Το σύνολο δεδομένων περιέχει κείμενο και μεταδεδομένα από 244 ιστότοπους και αντιπροσωπεύει συνολικά 12,999 θέσεις από συγκεκριμένες 30 ημέρες. Τέλος, το προτεινόμενο dataset του συγκεκριμένου άρθρου βρίσκεται στο παρακάτω link:

https://www.kaggle.com/mrisdal/fake-news

Στην ίδια κατηγορία του fact-checking εντάσσεται και η εφαρμογή *Fight hoax (http://fighthoax.com/):*

Η συγκεκριμένη εφαρμογή ελέγχει ένα άρθρο και έπειτα εμφανίζει τα κριτήρια τα οποία οδήγησαν στην απόφαση για το αν αυτό είναι αξιόπιστο ή όχι. Αναλύοντας περισσότερα από 300.000 άρθρα ειδήσεων ο αλγόριθμος fighthoax ταιριάζει τους βασικούς παράγοντες της εμπιστοσύνης των ειδήσεων, για να αποφασίσει αν το άρθρο που μόλις διαβάσατε είναι αξιόπιστο ή όχι. Τα κριτήρια με τα οποία γίνεται ο έλεγχος είναι:

-Fact Checking Organizations: Ένας ανεξάρτητος οργανισμός ελέγχου της αξιοπιστίας των δεδομένων έχει γράψει ένα άρθρο με βάση τι ο αναγνώστης διαβάζει και δεν υπάρχουν συσχετιζόμενα άρθρα.

-Clickbait Title: Αν ο τίτλος συμφωνεί με το περιεχόμενο των άρθρων.

-Language Quality: Είναι η γραμματική και συντακτική ανάλυση. Τα αξιόπιστα άρθρα έχουν αξιοπρεπή γραμματική και σύνταξη, η γραπτή τους γλώσσα είναι εύκολο να γίνει κατανοητή από έναν μέσο αναγνώστη, οι ιδέες και οι εκδηλώσεις ειδήσεων κοινοποιούνται με σαφή τρόπο.

-Persuasive Language: Αν περιέχει το άρθρο παρακινητική γλώσσα.

-Articles Length(μήκος άρθρου): Αν έχει το άρθρο αρκετές πληροφορίες σχετικά με το θέμα.

-Suspicious Website History: Αν είναι η ιστοσελίδα γνωστή για την έκδοση ψευδών ειδήσεων στο παρελθόν.

- Website Biases History: Ιστορικό ιστοσελίδας με βάση:

- προκαταλήψεις(Least Biased, Left/Right Center Bias, Left/Right Bias, Extreme Bias)
- Περιγραφή : προφίλ ιστοσελίδας
- αληθινή αναφορά(πηγές) (very high, high, mixed, low, very low)
- σημειώσεις(π.χ. χώρα, σε ποια μεσα ενημέρωσης υπάρχει)

Αναλυτικότερη περιγραφή στην ιστοσελίδα: mediabiasfactcheck.com/methodology.

-Authors (συγγραφείς): Οι συγγραφείς του άρθρου για ποιες ιστοσελίδες έχουν γράψει.

-Views: Διαφορετικές απόψεις και βοηθητικές πληροφορίες σχετικά με αυτό που διαβάζει ο αναγνώστης.

2.4 Πιθανές προεκτάσεις και σύνοψη

Συνοψίζοντας τις παραπάνω σχετικές έρευνες μπορούμε να συνδυάσουμε τις μεθόδους που χρησιμοποιούν και να βρούμε τις προεκτάσεις που θα μπορούσαν να έχουν. Ο παρακάτω πίνακας παρουσιάζει σύντομα τις μεθόδους αυτές:

Πίνακας που ανιχνεύει την αξιοπιστία μέσω του περιεχομένου:

**Στις στήλες βρίσκονται τα παραπάνω άρθρα με τη σειρά.*

	1	2	3	4	5	6	7	8	9
Σχέσεις νοηματικών ενοτήτων									+
συχνότητα λέξεων-γραμμάτων	+			+					
σημασιολογικές κατηγορίες	+								
χρήση υπερβολικής γλώσσας/υπερβολές	+								
Ανισορροπίες/αντικρουόμενα σενάρια	+								
χιούμορ-ειρωνία	+								
επανάληψη τίτλου	+								
αριθμός θεμάτων							+		
πολυπλοκότητα	+				+				
εντυπωσιακά πρωτοσέλιδα		+							
σύνταξη				+					
σημασιολογική				+					

ανάλυση (προσωπική εμπειρία-προφίλ θέματος)									
λέξεις κλειδιά							+		
συνοχή				+					
δομή κειμένου				+					
μήκος tweet/ ποσότητα λέξεων					+		+		
σημαντικές συνδεόμενες λέξεις				+					
Αβεβαιότητα/όχι αμεσότητα/εκφραστικότητα/ποικιλία/ανεπισημότητα					+				
εξειδίκευση (όροι ειδικών θεμάτων)					+		+		
σύμβολα							+		
επιρροή					+				
τα πιστεύο					+				
κοινά στοιχεία						+			
αριθμητική συμφωνία						+			
Φράσεις αντικειμενικότητας						+			
πολικότητα συναισθημάτων							+		
αριθμός ενισχυτικών λέξεων							+		

μέρη του λόγου								+	
σύγκριση hashtag								+	

Πίνακας 2.4.1

Πίνακας που ανιχνεύει την αξιοπιστία μέσω των πηγών:

**Στις στήλες βρίσκονται τα παραπάνω άρθρα με τη σειρά.*

	1	2	3	4	5	6	7	8	9
πηγές με αντοχή στο χρόνο		+							
πηγές με σύστημα ελέγχου		+							
αντιδράσεις			+						
δημοτικότητα πηγής			+						
στοιχεία χρήστη (followers-following)			+				+		
συνδεόμενα δεδομένα και σχέσεις				+					
τι δημοσιεύει η πηγή (προφίλ πηγής)					+			+	
αριθμός αναφορών								+	
μάρτυρες					+				
διάδοση εκδήλωσης (retweets)					+		+		

τύπος πηγής						+			
ύπαρξη πηγής (URL/tag)			+			+	+	+	
ποικιλία προέλευσης					+				
τόπος προέλευσης					+				
προέλευση ταυτότητας									

Πίνακας 2.4.2

Για τους πίνακες 2.4.1 και 2.4.2, ο αριθμός κάθε στήλης αντιστοιχεί σε ένα άρθρο όπως φαίνεται παρακάτω:

1. Rubin et al (2016)
2. Yimin Chen et al (2015)
3. Carlos Castillo et al (2011)
4. Niall et al (2015)
5. Rubin (2017)
6. Nagura et al (2006)
7. Byungkyu Kang et al (2012)
8. Qazvinian et al (2011)
9. Victoria L. Rubin et al (2015)

Οι παραπάνω μέθοδοι απαρτίζονται από συγκεκριμένα χαρακτηριστικά, είτε κοινά ή διαφορετικά. Θα μπορούσαμε να συνδυάσουμε τα κοινά χαρακτηριστικά, ή αυτά που

θεωρούμε πιο σημαντικά χαρακτηριστικά. Επίσης, πιο αποδοτικό είναι να συνδυάσουμε την ανίχνευση της αξιοπιστίας μέσω των πηγών και του κειμένου. Κάποια χαρακτηριστικά πηγών είναι: τα στοιχεία χρήστη (followers-followings), τι δημοσιεύει, η διάδοση εκδήλωσης (αντιδράσεις-κοινοποιήσεις-σχόλια), και η ύπαρξη της πηγής (URL/tag). Τα στοιχεία κειμένου είναι: η συχνότητα λέξεων-γραμμάτων, η πολυπλοκότητα, η ποσότητα λέξεων ή μήκος tweet και οι εξειδικευμένοι όροι.

Ο παρακάτω πίνακας παρουσιάζει σύντομα τα χαρακτηριστικά των δεδομένων:

	1	2	3
id	+		
Category	+		
Page	+		
Post url	+		
Date published	+		
Post type	+		
rating	+		
debate	+		
Share count	+		
Reaction count	+		
Comment count	+		
ord_in_thread		+	
author		+	
title		+	
text		+	
language		+	
crawled		+	
country		+	
Domain rank		+	
Thread title		+	
Spam score		+	

Main image url		+	
Replies count		+	
Participants count		+	
type		+	
Fact Checking Organizations			+
Articles Length			+
Organizations			+
Website History			+
Views			+

Πίνακας 2.4.3

Για τον πίνακα 2.4.3 χρησιμοποιήθηκαν τα άρθρα τα οποία είχαν προτεινόμενα features ή dataset. Ο αριθμός κάθε στήλης αντιστοιχεί σε ένα άρθρο όπως φαίνεται παρακάτω:

1. *Craig Silverman et al (2016)*
2. Risdal (2017)
3. *Fight hoax*

3. Προτεινόμενη προσέγγιση

3.1 Features

Σύμφωνα με τις παραπάνω μεθόδους και τα στοιχεία των δεδομένων μπορούν να προκύψουν ιδέες για τα γνωρίσματα του συνόλου δεδομένων μας. Τα γνωρίσματα των δεδομένων είναι: id(account-post), date, text, url, author, type, title, page, participants. Αν το άρθρο που εξετάζουμε συνδέεται με το facebook ή κάποιο άλλο μέσο κοινωνικής δικτύωσης τότε μπορούμε να έχουμε followers, followings, reactions, shares/retweets, comments.

Όσον αφορά τις μεθόδους, όπως προκύπτει και παραπάνω, οι προτεινόμενες μέθοδοι που είναι σε επίπεδο πηγών , πρέπει να εντοπίσουν: τα στοιχεία χρήστη (followers-followings), τι δημοσιεύει, τη διάδοση εκδήλωσης (αντιδράσεις-κοινοποιήσεις-σχόλια), και την ύπαρξη της πηγής (URL/tag), και σε επίπεδο κειμένου: τη συχνότητα λέξεων-γραμμάτων, την πολυπλοκότητα, την ποσότητα λέξεων ή μήκος tweet και τους εξειδικευμένοι όροι.

Οι έρευνες, που προαναφέρθηκαν, χρησιμοποίησαν ένα σύνολο δεδομένων και πηγών για να επαληθεύσουν την αξιοπιστία μιας είδησης και αξιολόγησαν κάθε φορά τις επιλογές τους μετρώντας την αποδοτικότητα των εφαρμογών τους σε σύνολα δεδομένων αξιολόγησης. Πιο συγκεκριμένα, οι classifiers που χρησιμοποιήθηκαν στις διάφορες ερευνητικές εργασίες ήταν οι neural nets και decision trees, ο αλγόριθμος βασισμένος σε SVM, και ο logistic regression (ακρίβεια 74%).

Σε επίπεδο ελληνικών δεδομένων, η συγκέντρωση των άρθρων βασίστηκε στις anti-hoax πηγές, όπως το ellinikahoaxes.gr και το www.factchecker.gr, και στο ίδιο το reputation της πηγής. Συγκεντρώθηκαν 100 άρθρα πραγματικά και άλλα τόσα που να είναι fake απο τις εξής

κατηγορίες: κοινωνία, κόσμος, πολιτική, άμυνα, αθλητισμός, πολιτισμός, επιστήμη, οικονομία, πολιτισμός και διατροφή-υγεία.

Στο περιεχόμενο των ελληνικών ειδήσεων παρατηρείται ότι όταν μια ιστορία είναι υπερβολική ή έχει σχέση με μεταφυσικά θέματα συνήθως είναι ψεύτικη. Για παράδειγμα:

- Οι μη λογικές προτάσεις όπως «Με αγώνες Subbuteo θα ολοκληρωθεί το πρωτάθλημα αν συνεχιστούν τα επεισόδια».
- Σελίδες με όχι σοβαρά ονόματα όπως το κουλούρι ή το βατράχι
- Οι προβολές δεν έχουν σχέση με την αξιοπιστία. Πολλές δημοφιλείς σελίδες δημοσιεύουν ψεύδη.
- Υπερβολές και επίκληση στο συναίσθημα.
- Περιεχόμενο με πολλά επίθετα.

Σύμφωνα με τα παραπάνω στοιχεία κατέληξα στα εξής γνωρίσματα των δεδομένων:

- url: Το url του αρχικού άρθρου, αν αυτό προέρχεται από anti-hoax πηγές.
- title: Ο τίτλος του άρθρου.
- Type: Ο τύπος του περιεχομένου του άρθρου, ο οποίος είναι ένας από τις παρακάτω κατηγορίες: κοινωνία, κόσμος, πολιτική, άμυνα, αθλητισμός, πολιτισμός, επιστήμη, οικονομία, πολιτισμός και διατροφή-υγεία
- site: Το site το οποίο δημοσίευσε το άρθρο.
- Source: Οι πηγές που αναφέρονται στο συγκεκριμένο άρθρο, και με βάση αυτών έγινε η συγγραφή του.
- sites with similar articles: Τα urls των άρθρων που έχουν παρόμοιο περιεχόμενο με το αρχικό άρθρο.
- Date: Η ημερομηνία η οποία δημοσιεύτηκε το άρθρο.

- quanity of words: Το πλήθος των λέξεων του άρθρου.
- Text: Το κείμενο του άρθρου.
- Author: Ο συγγραφέας του άρθρου.
- result (class): Τα αποτελέσματα με βάση τα παρακάτω χαρακτηριστικά, δηλαδή αν ένα άρθρο είναι αληθινό ή ψεύτικο. Αποδεκτές τιμές true-false.

3.2 Εξαγωγή features

Η εξαγωγή των χαρακτηριστικών των δεδομένων μπορεί να γίνει με δύο τρόπους είτε manually ή αυτόματα μέσω κώδικα. Πριν την συλλογή όλων των features, θα πρέπει να συγκεντρωθούν τα urls των άρθρων. Γι'αυτό θα πρέπει να γνωρίζουμε, πρώτον τον τύπο του άρθρου, ώστε να μοιραστούν ισόποσα τα δεδομένα ανάλογα με την κατηγορία στην οποία ανήκουν. Δεύτερον, αν το άρθρο είναι βασισμένο σε αληθινά γεγονότα ή είναι ψεύτικο, καθώς θα πρέπει να υπάρχει ένα σύνολο δεδομένων που αποτελείται από 100 αληθινά και 100 ψεύτικα.

Για την εκδοχή όπου τα δεδομένα θα συλλεχθούν χειροκίνητα, θα πρέπει να γίνεται εισαγωγή κάθε url στον browser και να εντοπίζονται τα κατάλληλα στοιχεία.

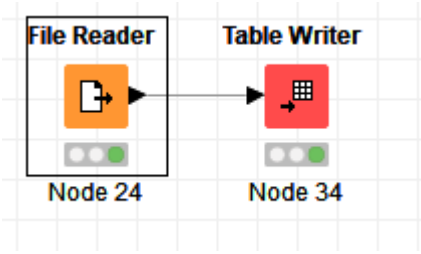
Για τον αυτόματο τρόπο τα urls θα είναι αποθηκευμένα σε ένα .csv αρχείο και οι στήλες: title, site, source, sites with similar articles, date, quanity of words, text, author, θα γεμίζονται αυτόματα. Ο κώδικας για την αυτόματη συλλογή τους θα πρέπει να διαβάζει το αρχείο, να συλλέγει τα παραπάνω στοιχεία για κάθε url και να τα γράφει στις υπόλοιπες στήλες του αρχείου.

3.3 Τεχνική

Αρχικά, δημιουργήθηκε αρχείο της μορφής .csv με τις εξής στήλες:

- url
- title
- type
- site
- source
- sites with similar articles
- date
- quanity of words
- text
- author
- result (class)

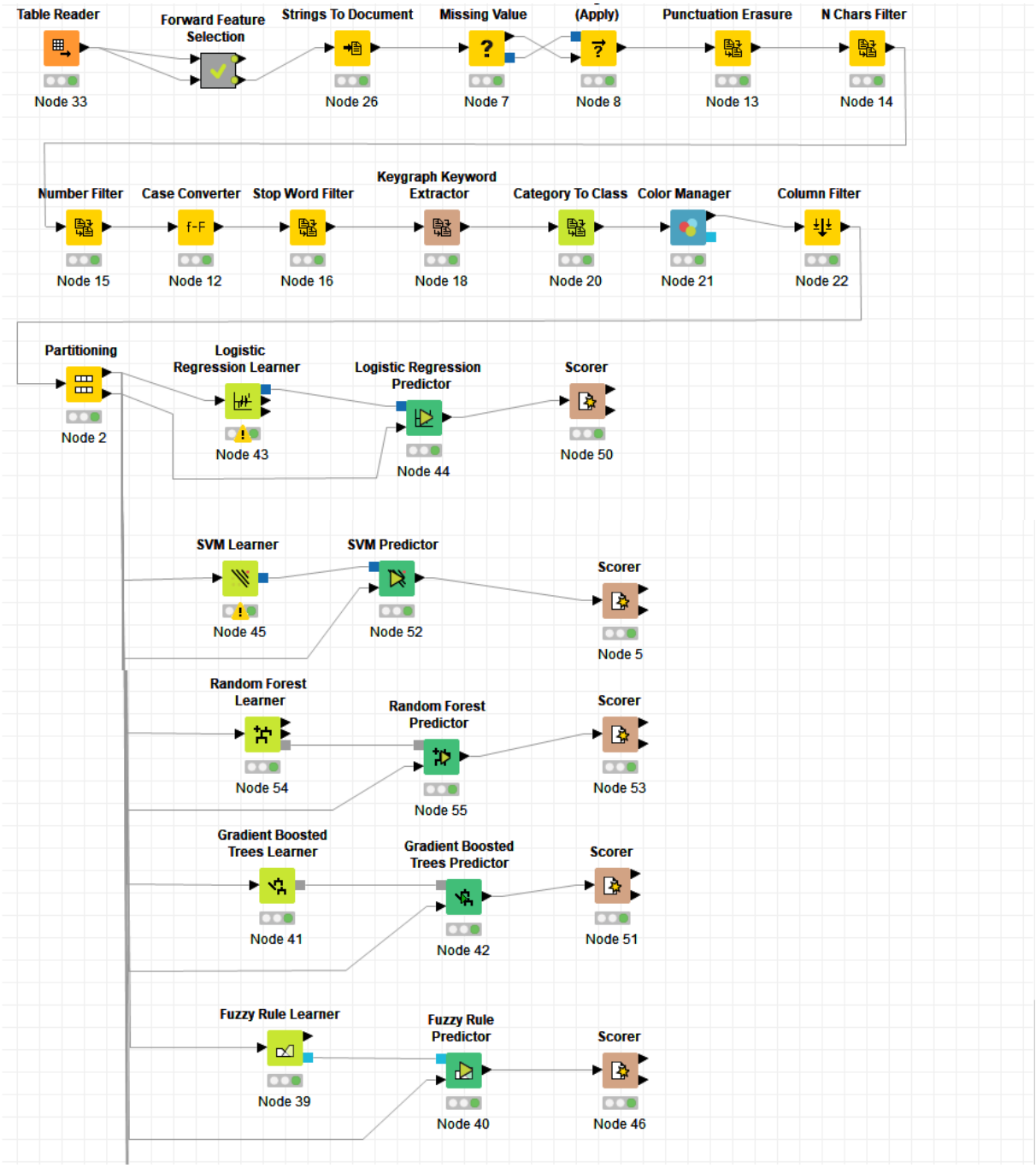
Αφού φορτώθηκε το αρχείο με τα δεδομένα με δείγμα αληθών και ψευδών ειδήσεων, το πρώτο στάδιο του classification είναι η μετατροπή των δεδομένων στο Knime. Μετατροπή του αρχείου από .csv σε .table:

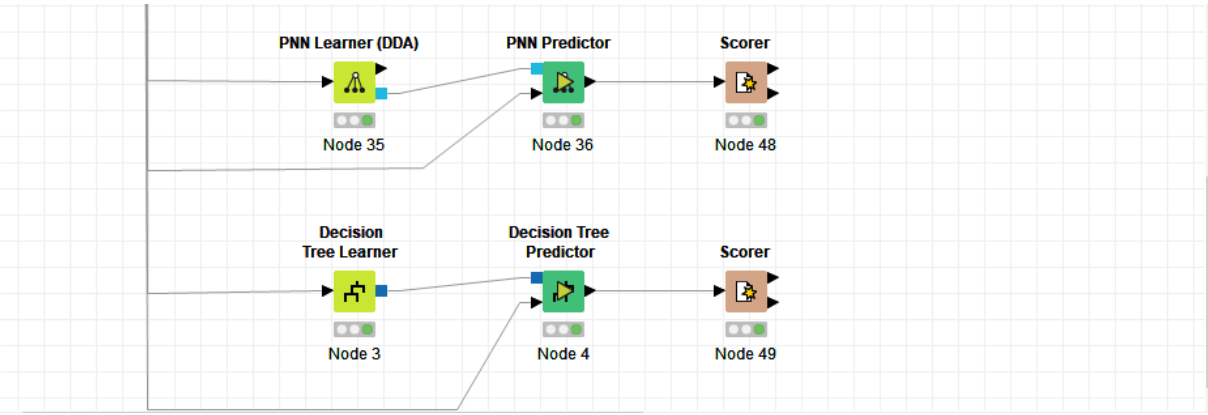


Εικόνα 3.3.1

Για να διαβαστεί το αρχείο, θα ήταν καλό να μετατραπεί σε .table αρχείο ώστε να μπορεί να φορτωθεί στον File Reader σε μορφή αποδεκτή για τους κόμβους που επεξεργάζονται το αρχείο. Όπως φαίνεται παρπάνω, ο File Reader διαβάζει τα δεδομένα από ένα αρχείο και ο Table Reader γράφει έναν πίνακα δεδομένων σε μια εσωτερική μορφή σε ένα αρχείο.

Διαγραμμα ροης πληροφοριας





Σχήμα 3.3.1

Περιγραφή:

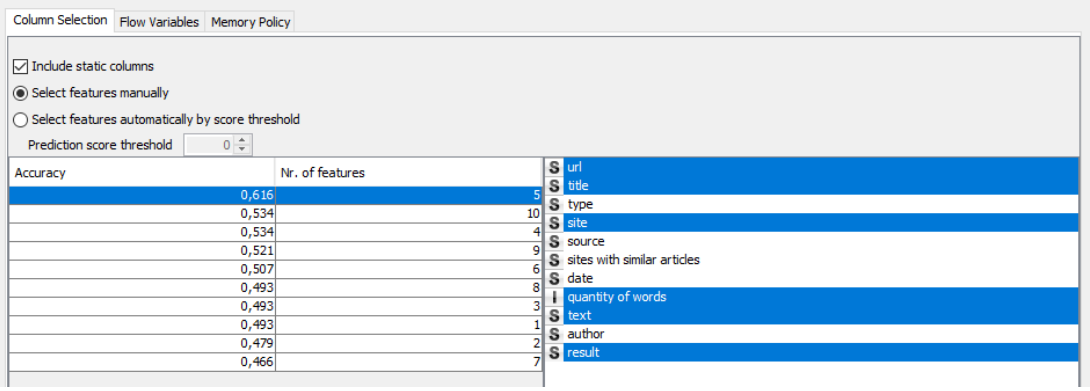
Μετά τη μετατροπή του αρχείου σε .table, ακολουθεί η δημιουργία ενός διαγράμματος ροής που περιλαμβάνει τις παρακάτω διαδικασίες:

1. input data

   Πρώτα, φορτώνονται τα δεδομένα, στην συγκεκριμένη περίπτωση είναι το αρχείο, το οποίο περιέχει το σύνολο των αληθινών και ψευδών ειδήσεων. Το αρχείο φορτώνεται στην μορφή .table.

2. Preprocessing

   Στη συνέχεια, ακολουθεί η προ-επεξεργασία των δεδομένων, η οποία γίνεται μέσω συγκεκριμένων φίλτρων. Το πρώτο φίλτρο είναι το Forward Feature Selection, το οποίο καταλήγει στα επικρατέστερα χαρακτηριστικά ώστε να επιτευχθεί η καλύτερη ακρίβεια. Εφαρμόζει ένα μοντέλο επεξεργασίας χαρακτηριστικών βασισμένο στην επιλογή επικρατέστερων χαρακτηριστικών και δημιουργεί έναν πίνακα που εμφανίζει την ακρίβεια με βάση το συνδυασμό των επιλεγμένων χαρακτηριστικών και ο χρήστης μπορεί να επιλέξει τον καλύτερο συνδυασμό, όπως φαίνεται παρακάτω:



Εικόνα 3.3.2

Παρακάτω φαίνεται ο πίνακας αποτελεσμάτων ακρίβειας ανάλογα με την τα χαρακτηριστικά:

Row ID	Nr. of features	Accuracy	Added feature
1	1	0.53	author
2	2	0.909	title
3	3	0.909	sites with similar articles
4	4	0.924	date
5	5	0.879	quantity of words
6	6	0.924	site
7	7	0.879	text
8	8	0.47	type
9	9	0.712	source
All	10	0.348	url

Εικόνα *3.3.3*

Το φίλτρο Strings To Document μετατρέπει τις καθορισμένες συμβολοσειρές σε έγγραφα. Punctuation Erasure αφαιρεί όλα τα σημεία στίξης των όρων που περιέχονται στα έγγραφα εισόδου. Το N Chars Filter, φιλτράρει τους όρους σύμφωνα με τους χαρακτήρες και το Number Filter, φιλτράρει όλους τους όρους άναλογα με τα ψηφία τους. ToCase Converter μετατρέπει τους αλφαριθμητικούς χαρακτήρες σε πεζά ή Κεφαλαία. Το Stop Word Filter φιλτράρει όλους τους όρους των εγγράφων εισόδου, τα οποία περιέχονται στη λίστα λέξεων διακοπής. Το ReplaceMissingValues και MissingValues(Apply), αυτοί οι κόμβοι βοηθά στη διαχείριση των τιμών που λείπουν

απ’τα κελιά του πίνακα εισόδου. Δίνει τη δυνατότητα να αντικαταστήσει ο χρήστης τις τιμές που λείπουν ανάλογα με τη στήλη που βρίσκονται.

3. keyword extraction

   Έπειτα, υπάρχει το Keyword extraction, το οποίο αναλύει τα έγγραφα και εξάγει τις σχετικές λέξεις-κλειδιά.

4. Category To Class

   Το Category To Class, αυτός ο κόμβος σάς επιτρέπει να προσθέσετε μια στήλη (string) σε κάθε σειρά που περιέχει ένα κελί εγγράφου. Η τιμή της κλάσης είναι η κατηγορία του εγγράφου ως συμβολοσειρά.

5. Color Manager

   Το Color Manager χωρίζει τα δεδομένα τοποθετώντας χρώματα, ανάλογα με την τιμή της επιλεγόμενης στήλης.

6. Column Filter

   Ο κόμβος Column Filter επιτρέπει την επεξεργασία των στηλών από τον πίνακα εισόδου, ενώ μόνο οι υπόλοιπες στήλες μεταφέρονται στον πίνακα εξόδου. Στο Partitioning, ο πίνακας εισόδου χωρίζεται σε δύο κατατμήσεις , π.χ. τα δεδομένα εκπαίδευσης και ελέγχου. Τα δύο μέρη είναι διαθέσιμα στις δύο θύρες εξόδου.

7. classification and scoring

   Τέλος, ακολουθεί η διαδικασία Classification and scoring, που περιλαμβάνει έναν classifier και τον scorer κόμβο. Ο Classifier αναφέρεται στη μαθηματική συνάρτηση, που εφαρμόζεται από έναν αλγόριθμο ταξινόμησης, ο οποίος χαρτογραφεί δεδομένα εισόδου σε μια κατηγορία και ο Scorer συγκρίνει δύο στήλες με τα ζεύγη των τιμών χαρακτηριστικών τους και δείχνει την ακρίβεια του αλγορίθμου.

4. Αξιολόγηση-σύγκριση

4.1 Μέθοδοι-μέτρα αξιολόγησης

Dataset:

Για τη συλλογή του συνόλου δεδομένων, συγκεντρώθηκαν 100 πραγματικά άρθρα και άλλα τόσα ψεύτικα. Τα άρθρα που συλλέχθηκαν, πληρούσαν τις παρακάτω προϋποθέσεις. Αρχικά, ήταν από σελίδες με διαφορετικά επίπεδα αλήθειας, δηλαδή από χιουμοριστικές σελίδες, όπως το Κουλούρι ή Το Βατράχι, μέχρι στρατιωτικά μπλογκς, τα οποία θεωρούνται πιο έμπιστα αλλά μπορείς να βρεις μια ψεύτικη είδηση. Επίσης, τα δεδομένα χωρίστηκαν σε 9 διαφορετικές κατηγορίες, οι οποίες ήταν ισομοιρασμένες στο σύνολο, οι οποίες είναι Κοινωνία, Κόσμος, Αθλητισμός, Πολιτική, Άμυνα, Επιστήμη, Οικονομία, Πολιτισμός, Διατροφή-Υγεία.

Για το σωστό διαχωρισμό των άρθρων σε αληθή ή ψευδή χρησιμοποιήθηκαν τέσσερις τρόποι συλλογής:

1. Συγκέντρωση από σελίδες που ήταν σίγουρα μη έγκυρες, π.χ. χιουμοριστικές όπως το Κουλούρι, ή έγκυρες, όπως το σελίδες υπουργείων.
2. Αναζήτηση άρθρων για τα οποία ήταν γνωστή η εγκυρότητα ή όχι αυτών.
3. Αναζήτηση άρθρων και η αξιολόγησή τους από γνώστες του αντικειμένου που αφορούσε αυτό.
4. Συλλογή βασισμένη σε anti-hoax πηγές, όπως το ellinikahoaxes.gr και factchecker.gr, οι οποίες κατηγοριοποιούν τις ειδήσεις σε αληθείς ή ψευδείς με τα κατάλληλα επιχειρήματα.

Σύμφωνα με τα παραπάνω κριτήρια και καταλήγοντας στα κατάλληλα features (url, title, type, site, source, sites with similar articles, date, quanity of words, text, author, result

(class)) άρχισε η συγκέντρωση του συνόλου δεδομένων. Πρώτα, μαζεύτηκαν τα urls των άρθρων , είτε άμεσα ή έμμεσα, αφου προέρχονταν από anti-hoax πηγες και οι πηγές. Τα urls και οι πηγές αποθηκεύτηκαν σε στήλες σε ένα .csv αρχείο. Έπειτα για κάθε ένα από αυτά αντιστοιχίστηκε στη στήλη results η κατάλληλη τιμή (true-false), ανάλογα με το αν το άρθρο ήταν αληθινό ή ψεύτικο. Οι υπόλοιπες στήλες με τα χαρακτηριστικά ήταν κενές ώστε να συμπληρωθούν με αυτοματοποιημένο τρόπο.

Ο παρακάτω κώδικας υλοποιήθηκε στο εργαλείο IntelliJ για τη συλλογή των δεδομένων για κάθε ένα απο τα url που είχαν συγκεντρωθεί.

Dependencies

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.jsoup</groupId>
```

```
                <artifactId>jsoup</artifactId>
                <version>1.10.2</version>
            </dependency>
            <dependency>
                <groupId>de.l3s.boilerpipe</groupId>
                <artifactId>boilerpipe</artifactId>
                <version>1.1.0</version>
            </dependency>
            <dependency>
                <groupId>com.syncthemall</groupId>
                <artifactId>boilerpipe</artifactId>
                <version>1.2.1</version>
            </dependency>

            <dependency>
                <groupId>com.googlecode.json-simple</groupId>
                <artifactId>json-simple</artifactId>
                <version>1.1.1</version>
            </dependency>

    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-lang3</artifactId>
```

```
        <version>3.7</version>
    </dependency>
```

Περιγραφή:

Η προσθήκη dependencies είναι ένας τρόπος να εισαχθούν συγκεκριμένες βιβλιοθήκες , ώστε να χρησιμοποιηθούν για να εκτελεστούν συγκεκριμένες λειτουργίες. Όπως φαίνεται παραπάνω οι βιβλιοθήκες που προστέθηκαν είναι οι εξής:

1. spring-boot-starter-test και spring-boot-starter-test: για τη χρήση του spring framework.
2. jsoup: αναλύει, εξάγει και χειρίζεται δεδομένα αποθηκευμένα σε έγγραφα HTML.
3. boilerpipe και syncthemall: βοηθούν στο extraction του κειμένου του άρθρου.
4. googlecode.json-simple: για να πάρουμε στοιχεία του κειμένου, όπως η ημερομηνία και ο συγγραφέας.
5. org.apache.commons: για μεθόδους που χρησιμοποιούν τα strings.

ScrapperApplication.java

```
package gr.web.scraper;


import org.apache.commons.lang3.StringUtils;

import org.jsoup.nodes.Document;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.Banner;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;


import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;


import static java.lang.System.exit;


//activates the automated configuration of spring

@SpringBootApplication

public class ScraperApplication implements CommandLineRunner {
```

```
	@Autowired
	private ScraperService scraperService;


	public static void main(String[] args) {
		SpringApplication app = new SpringApplication(ScraperApplication.class);
	//remove banner from logs
		app.setBannerMode(Banner.Mode.OFF);
		app.run(args);
	}


	@Override
	public void run(String... args) throws Exception {


		List<List<String>> inputData = scraperService.getInputData();
		List<List<String>> resultData = new ArrayList<>();



resultData.add(Arrays.asList("URL","Title","Website","Author","PublishedDate","WordCoun
t","SimilarSites","Content"));
		for (List<String> rowData : inputData){
			List<String> resultRow = new ArrayList<>();
			if (!rowData.isEmpty()){
				Document doc =
```

```
scraperService.getDocument(rowData.get(0));

                                resultRow.add(doc.baseUri());
                                resultRow.add(doc.title());
                                resultRow.add(Utilities.getSite(doc.baseUri()));
                                resultRow.add(scraperService.extractAuthor(doc));
                                resultRow.add(scraperService.extractDate(doc.baseUri(),
doc));
                                String                          article                          =
scraperService.getArticleContent(doc.baseUri()).replace("\n"," ");
                                resultRow.add(String.valueOf(Utilities.wordCounter(article)));



resultRow.add(StringUtils.join(scraperService.findSitesWithSimilarArticles(doc.title()), ", "));
                                resultRow.add(article);
                        }
                        resultData.add(resultRow);
                }

                scraperService.writeToFile(resultData);
                exit(0);
        }
}
```

Περιγραφή:

Στην κλάση ScrapperApplicaion.java . Η μέθοδος run() φτιάχνει μια λίστα (resultData) στην οποί αποθηκεύονται τα αποτελέσματα. Σε αυτήν αρχικά προστίθεται το header που θα έχει το .csv αρχείο, δηλαδή τα ονόματα των στηλών που είναι τα χαρακτηριστικά των δεδομένων. Έπειτα, για κάθε url που διαβάζει , εκτελεί τις ενέργειες συλλογής δεδομένων και τις αποθηκεύει στη λίστα. Εκτελεί μεθόδους για να βρει το url, τον τίτλο, το site, τον συγγραφέα, την ημερομηνία που δημοσιεύθηκε, το κείμενο του άρθρου, τον αριθμό των λέξεων του άρθρου και τα sites με παρόμοιο περιεχόμενο. Προσθέτει μια καινούρια εγγραφή στη λίστα για κάθε url, και τα γράφει στο .csv αρχείο.

CSVUtils.java

```
package gr.web.scraper;


import java.io.File;
import java.io.IOException;
import java.io.UncheckedIOException;
import java.io.Writer;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;
```

```
public class CSVUtils {

   private static final char DEFAULT_SEPARATOR = ',';
   private static final char DEFAULT_QUOTE = '"';



   private static String followCVSformat(String value) {

      String result = value;
      if (result.contains("\"")) {
         result = result.replace("\"", "\"\"");
      }
      return result;

   }

       public static void writeLine(Writer w, List<String> values, char separators, char
customQuote) throws IOException {

      boolean first = true;

      //default customQuote is empty
```

```
    if (separators == ' ') {
        separators = DEFAULT_SEPARATOR;
    }

    StringBuilder sb = new StringBuilder();
    for (String value : values) {
        if (!first) {
            sb.append(separators);
        }
        if (customQuote == ' ') {
            sb.append(followCVSformat(value));
        } else {
            sb.append(customQuote).append(followCVSformat(value)).append(customQuote);
        }

        first = false;
    }
    sb.append("\n");
    w.append(sb.toString());

}
```

```
  /**
   * Returns the WHOLE CSV as a List<List<String>>, first entry are headers of the csv
   *
   * @param file
   * @param seperator
   * @return List<List<String>>
   */
  public static List<List<String>> readCSV(File file, String seperator) {
    try {
      return Files.lines(file.toPath(), StandardCharsets.UTF_8)
          .map(line -> Arrays.asList(line.split(seperator)))
          .collect(Collectors.toList());
    } catch (IOException e) {
      throw new UncheckedIOException(e);
    }
  }
}
```

Περιγραφη:

Στην κλάση CSVUtils υπάρχουν όλες οι ενέργειες που γίνονται ώστε να γίνει η επεξεργασία του csv αρχείου. Περιέχει τις εξής μεθόδους:

1. **writeLine():** Γράφει στο αρχείο τα αποτελέσματα μέσα σε διπλά εισαγωγικά (“”).
2. **followCVSformat(String value):** Αντικαθιστά την τιμή "\"" με αυτήν "\"\"".

3. **readCSV(File file, String seperator):** Επιστρέφει ένα CSV σαν λίστα, βάζοντας πρώτα headers.

ScrapperService.java

```
package gr.web.scraper;



import de.l3s.boilerpipe.BoilerpipeProcessingException;

import de.l3s.boilerpipe.document.TextDocument;

import de.l3s.boilerpipe.extractors.CommonExtractors;

import de.l3s.boilerpipe.sax.BoilerpipeSAXInput;

import de.l3s.boilerpipe.sax.HTMLDocument;

import de.l3s.boilerpipe.sax.HTMLFetcher;

import org.apache.commons.lang3.StringUtils;

import org.json.simple.JSONObject;

import org.json.simple.parser.JSONParser;

import org.json.simple.parser.ParseException;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import org.jsoup.nodes.Element;

import org.jsoup.select.Elements;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.stereotype.Service;
```

```
import org.xml.sax.SAXException;


import java.io.*;
import java.net.URL;
import java.net.URLDecoder;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.List;


@Service
public class ScraperService {


    //get the values from aplication.properties
    @Value("${input_csv_path}")
    private String inputCsvPath;


    @Value("${google_search_url}")
    private String googleSearchUrl;


    @Value("${output_csv_path}")
    private String outputCsvPath;


    public List<String> findSitesWithSimilarArticles(String search) throws IOException {
```

```
    List<String> similarSites = new ArrayList<>();
    //URLEncoder.encode(search, StandardCharsets.UTF_8.name()))
                        Document doc = Jsoup.connect(String.format(googleSearchUrl, search)).userAgent("Mozilla/5.0").get();
    Elements links = doc.select(".g> h3.r > a");

    for (Element link : links) {
      String title = link.text();
                    String url = link.absUrl("href"); // Google returns URLs in format "http://www.google.com/url?q=<url>&sa=U&ei=<someKey>".
        url = URLDecoder.decode(url.substring(url.indexOf('=') + 1, url.indexOf('&')), "UTF-8");

      if (!url.startsWith("http")) {
        continue; // Ads/news/etc.
      }
      similarSites.add(url);
    }

    return similarSites;
  }

  public String getTitle(Document doc){
```

```
    String title = doc.title();
    return title;
}



public String getText(Document doc){
    String text = doc.body().text();
    return text;

}

public Document getDocument(String url) throws IOException {
    return Jsoup.connect(url).get();
}

public List<List<String>> getInputData() throws IOException {
    File file = new File(inputCsvPath);
    List<List<String>> inputData = CSVUtils.readCSV(file, ",");

    return inputData;
}

public boolean writeToFile(List<List<String>> inputData)  {
```

```
    try {
      //FileWriter writer = new FileWriter(outputCsvPath,);
          Writer writer = new OutputStreamWriter(new FileOutputStream(outputCsvPath),
StandardCharsets.UTF_8);

      for (List<String> rowData : inputData){
        CSVUtils.writeLine(writer, rowData, '|', '"');
      }
      writer.flush();
      writer.close();
    } catch (IOException e) {
      return false;
    }
    return true;
  }

    public String getArticleContent(String articleUrl) throws IOException, SAXException,
BoilerpipeProcessingException {
    URL url = new URL(articleUrl);
    HTMLDocument htmlDoc = HTMLFetcher.fetch(url);
                                                TextDocument doc = new
BoilerpipeSAXInput(htmlDoc.toInputSource()).getTextDocument();
```

```
        return CommonExtractors.ARTICLE_EXTRACTOR.getText(doc);
    }

    public JSONObject getIdJsonFromDocument(Document document){

        JSONObject result = null;
                                                        String scriptJson =
document.getElementsByAttributeValue("type","application/ld+json").html();

        JSONParser parser = new JSONParser();
        try {
            result = (JSONObject) parser.parse(scriptJson);
        } catch (ParseException e) {
            return null;       }

        return result;
    }

    public String getDateFromMetaTags(Document document){

        Elements metaTags = document.getElementsByTag("meta");
```

```
    String stringDate = null;

    for (Element metaTag : metaTags) {
       String content = metaTag.attr("content");
       String name = metaTag.attr("name");
       String property = metaTag.attr("property");

       if("article.published".equals(name) || "article.published".equals(property) ||
          "bt:pubDate".equals(name)|| "bt:pubDate".equals(property) ||
          "DC.date.issued".equals(name) || "DC.date.issued".equals(property) ||
          "pubdate".equals(name) || "pubdate".equals(property)  ||
                                                "article:published_time".equals(name)     ||
"article:published_time".equals(property) ) {
           stringDate = content;
           break;
       }
    }

    if (stringDate == null){
                                                          Elements       element       =
document.getElementsByAttributeValue("itemprop","datePublished");
       stringDate = element.text();
    }
```

```
    return stringDate;
}



public String getAuthorFromMetaTags(Document document){

    Elements metaTags = document.getElementsByTag("meta");

    String author = null;

    for (Element metaTag : metaTags) {
        String content = metaTag.attr("content");
        String name = metaTag.attr("name");
        String property = metaTag.attr("property");
        if("article:author".equals(name) || "article:author".equals(property) ||
           "author".equals(name) || "author".equals(property) ||
           "article.author".equals(name) || "article.author".equals(property) ){
            author = content;
            break;
        }
    }
```

```
    if (author == null){
        Elements element = document.getElementsByAttributeValue("itemprop","author");
        author = element.text();
    }

    return author;
}

public String extractDate(String url, Document document){

    // get date from URL
    String date = Utilities.extractDateFromUrl(url);

    // get date from script json JSON-LD
    if (StringUtils.isEmpty(date)){
        JSONObject jsonObject = getIdJsonFromDocument(document);
        if (jsonObject != null){
            date = Utilities.getPublishedDate(jsonObject);
        }
    }

    // get date from metatags
    if (StringUtils.isEmpty(date)){
```

```
        date = getDateFromMetaTags(document);
    }

    return date;
}


public String extractAuthor(Document document) {

    // get author from metatags
    String author = getAuthorFromMetaTags(document);

    // get author from script json JSON-LD
    if (StringUtils.isEmpty(author)){
        JSONObject jsonObject = getIdJsonFromDocument(document);
        if (jsonObject != null){
            author = Utilities.getAuthor(jsonObject);
        }
    }

    return author;
}
```

```
}
```

Περιγραφή:

Στην κλάση ScrapperService υπάρχουν όλες οι ενέργειες που γίνονται ώστε να συγκεντρωθούν όλα τα χαρακτηριστικά των δεδομένων για κάθε url που έχει συλεχθεί manually. Υπάρχουν οι εξής μέθοδοι, οι οποίες περιγράφονται παρακάτω:

1. **findSitesWithSimilarArticles(String search):** Περνώντας ως παράμετρο τον τίτλο του άρθρου, η μέθοδος κάνει μια αναζήτηση μέσω google με αυτόν. Τα αποτελέσματα της αναζήτησης αυτής είναι άρθρα που έχουν θέμα παρόμοιο με τον τίτλο. Μέσω του κώδικα συλλέγονται τα urls των άρθρων από τα αποτελέσματα σε μια λίστα και επιστρέφονται.
2. **getTitle(Document doc):** Αυτή η μέθοδος δέχεται ένα αντικείμενο τύπου Document και επιστρέφει τον τίτλο του άρθρου.
3. **getText(Document doc):** Αυτή η μέθοδος δέχεται ένα αντικείμενο τύπου Document και επιστρέφει το κείμενο του άρθρου.
4. **getDocument(String url):** Επιστρέφει το document από την jsoup σύνδεση.
5. **GetInputData():** Διαβάζει ένα αρχείο τύπου csv.
6. **writeToFile(List<List<String>> inputData):** Ανοίγει ένα csv αρχείο και το γράφει γραμμή γραμμή.
7. **getArticleContent(String articleUrl):** Δέχεται ένα url και επισρέφει το κείμενο του αρθρου.
8. **getDateFromMetaTags(Document document):** Παίρνει τα στοιχεία που βρίσκονται στο “meta-tag” και για κάθε στοιχείο ελέγχει αν υπάρχει tag που να είναι σχετικό με την ημερομηνία δημοσίευσης. Αν υπάρχει, βάζει το περιεχόμενο που βρίσκεται στα tags

στην μεταβλητή stringDate, αλλιώς βρίσκει και αποθηκεύει την ημερομηνία που βρίσκει στα attributes.

9. **getAuthorFromMetaTags(Document document):** Παίρνει τα στοιχεία που βρίσκονται στο “meta-tag” και για κάθε στοιχείο ελέγχει αν υπάρχει tag που να είναι σχετικό με τον συγγραφέα. Αν υπάρχει, βάζει το περιεχόμενο που βρίσκεται στα tags στην μεταβλητή author, αλλιώς βρίσκει και αποθηκεύει τον συγγραφέ που βρίσκει στα attributes.
10. **extractDate(String url, Document document):** Η μέθοδος αυτή παίρνει την ημερομηνία που βρίσκεται στο url του άρθρου και την αποθηκεύει σε μια μεταβλητή. Αν η μεταβλητή date δεν έχει πάρει τιμή τότε καλεί την μέθοδο Utilities.getPublishedDate(jsonObject). Τέλος, αν και από αυτήν δεν έχει πάρει τιμή καλεί την getDateFromMetaTags(document) και επιστρέφει την τιμή της.
11. **extractAuthor(Document document):** Επιστρέφει τον συγγραφέα του άρθρου.

application.properties

```
input_csv_path=input.csv
google_search_url=https://www.google.com/search?q=%s
output_csv_path=output.csv
```

Utilities.java

```
package gr.web.scraper;


import org.json.simple.JSONObject;


import java.net.URI;
import java.net.URISyntaxException;
import java.text.SimpleDateFormat;
import java.util.regex.Matcher;
import java.util.regex.Pattern;


public class Utilities {


   public static final String REGEX = "([\\./\\-_]{0,1}(19|20)\\d{2})[\\./\\-_]{0,1}(([0-3]{0,1}[0-
9][\\./\\-_])|(\\w{3,5}[\\./\\-_]))([0-3]{0,1}[0-9][\\./\\-]{0,1})?";



   public static String getSite(String url) throws URISyntaxException {
      URI uri = new URI(url);
      String domain = uri.getHost();


      return domain.startsWith("www.") ? domain.substring(4) : domain;
```

```
}

public static int wordCounter(String text) {

    int wordCount;
    String[] wordArray = text.trim().split("\\s+");
    wordCount = wordArray.length;

    return wordCount;
}

public static String extractDateFromUrl(String url){
    Matcher m = Pattern.compile(REGEX, Pattern.CASE_INSENSITIVE).matcher(url);

    if (m.find()){
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        String dateUrl = m.group(0);
        if (dateUrl.startsWith("/")){
            dateUrl = dateUrl.substring(1).replace("/","-");
        }else {
            dateUrl = dateUrl.replace("/","-");
        }
```

```
        return dateUrl;
      }

      return null;
   }

   public static String getAuthor(JSONObject jsonObject){

      JSONObject object = (JSONObject) jsonObject.get("author");

      return (String) object.get("name");
   }

   public static String getPublishedDate(JSONObject jsonObject){
      return (String) jsonObject.get("datePublished");
   }
}
```

Περιγραφή:

Στην κλάση Utilities υπάρχουν κάποιες επιπλέον ενέργειες που γίνονται ώστε να συγκεντρωθούν όλα τα χαρακτηριστικά των δεδομένων για κάθε url που έχει συλεχθεί manually. Υπάρχουν οι εξής μέθοδοι, οι οποίες περιγράφονται παρακάτω:

1. **getSite(String url):** Περνώντας ως παράμετρο ένα url, η μέθοδος επιλέγει και αποθηκεύει ένα κομμάτι αυτού του url, που είναι το site, και τέλος το επιστρέφει.
2. **wordCounter(String text):** Η συγκεκριμένη μέθοδος δέχεται ένα κείμενο μετράει τις λέξεις αυτού και επιστρέφει τον αριθμό αυτόν.
3. **extractDateFromUrl(String url):** Παίρνει την ημερομηνία από το url και την φτιάχνει στο φορματ που ακολουθεί “yyyy-MM-dd”.
4. **getAuthor(JSONObject jsonObject):** Επιστρέφει τον συγγραφέα.
5. **getPublishedDate(JSONObject jsonObject):** Επιστρέφει την ημερομηνία δημοσίευσης του άρθρου.

4.2 Αποτελέσματα

Μετά την συγκέντρωση των δεδομένων, έγινε χρήση συγκεκριμένων αλγορίθμων, ώστε να βρεθεί ο πιο αποτελεσματικός για τα συγκεκριμένα χαρακτηριστικά των δεδομένων. Οι αλγόριθμοι που χρησιμοποιήθηκαν με την ακρίβεια που πέτυχαν στο συγκεκριμένο dataset φαίνονται παρακάτω:

Classifier	**Accuracy**
PNN(Neutral Nets)	69,505%
NaiveBayes	63,909%
FuzzyRule	52,636%
GradientBoostedTrees	69,667%
Logistic Regression	51,582%
SMO (SVM)	63,909%
k-Nearest Neighbor	69,505%
Decision Tree	69,667%
RandomForest	69,505%

Πίνακας 4.2.1

Test Dataset:

Ο παρακάτω πίνακας δείχνει την ακρίβεια που είχε κάθε classifier για ένα σύνολο δεδομένων που δε χρησιμοποιήθηκε στην εκπαίδευση του classifiers:

Classifier	**Accuracy**
PNN(Neutral Nets)	64,996%
NaiveBayes	67,723%
FuzzyRule	64,996%

GradientBoostedTrees	69,27%
Logistic Regression	69,27%
SMO (SVM)	69,27%
k-Nearest Neighbor	53,058%
Decision Tree	69,27%
RandomForest	69,27%

Πίνακας 4.2.2

Το test dataset έχει την καλύτερη επίδοση στους αλγορίθμους GradientBoostedTrees, Logistic Regression, SVM, Decision Tree και RandomForest.

Παρακάτω φαίνονται οι αλγόριθμοι με την καλύτερη απόδοση και τα measurements που δηλώνουν την αποδοτικότητα τους:

GradientBoostedTrees

	TP	**FP**	**Precision**	**Recall**	**F-measure**
F	466	163	0.741	0.647	0.691
T	474	254	0.651	0.744	0.695
Avg.	470	208.5	0.696	0.6955	0.693

Decision Tree

	TP	**FP**	**Precision**	**Recall**	**F-measure**
F	466	163	0.741	0.647	0.691
T	474	254	0.651	0.744	0.695
Avg.	470	208.5	0.696	0.6955	0.693

Πίνακας 4.2.3

Όπως φαίνεται στους Πίνακες 4.3.1 και 4.3.2 οι αλγόριθμοι με την καλύτερη απόδοση είναι οι GradientBoostedTrees και Decision Tree και αυτός με την χειρότερη είναι ο FuzzyRule.

4.2.1 Λόγοι για υψηλή ακρίβεια – Σύγκριση:

Οι Classifiers που είναι βασισμένοι σε δέντρα χρησιμοποιούνται επιτυχημένα σε πολλά σημεία, όπως αναγνώριση χαρακτήρων, ιατρική διάγνωση ή συστήματα εμπειρογνωμόνων. Το πιο σημαντικό χαρακτηριστικό τους είναι να σπάνε μια περίπλοκη διαδικασία λήψης αποφάσεων σε μια συλλογή απλούστερων αποφάσεων που είναι ευκολότερο να ερμηνευτούν.

5. Συμπεράσματα

Συνοψίζοντας, το πρόβλημα αναπαραγωγής ψευδών ειδήσεων στις μέρες μας γίνεται όλο ένα πιο αισθητό, γι'αυτό έχουν αρχίσει έρευνες για το φιλτράρισμα των ειδήσεων. Οι διάφοροι ερευνητές που έχουν καταλήξει σε κάποια συμπεράσματα για το θέμα χωρίζονται σε τρεις κατηγορίες. Υπάρχουν κάποιοι που υποστηρίζουν την εγκυρότητα ενός κειμένου με βάση τις πηγές αυτού, κάποιοι που βασίζονται στο κείμενο, και άλλοι συνδυάζοντας τις παραπάνω τεχνικές. Παρατηρώντας αυτές τις τρεις τεχνικές, κατέληξα στο συμπέρασμα πως η πιο αποδοτική είναι εκείνη που τα συνδυάζει τις πηγές και το κείμενο.

Το μοντέλο που χρησιμοποίησα αποτελείται από δεδομένα τα οποία αφορούν χαρακτηριστικά του κειμένου και των πηγών. Αρχικά, αφού συγκέντρωσα το σύνολο των δεδομένων και το επεξεργάστηκα , προχώρησα στο classification των δεδομένων. Για την κατηγοριοποίηση, κατέληξα στον Decision Tree και Gradient Boosted Trees οι οποίοι είχαν την καλύτερη απόδοση στα δεδόμενα.

5.1 Βελτιώσεις που αφορούν τους classifiers

Ο Decision Tree είχε την καλύτερη απόδοση στο παραπάνω dataset, γι'αυτό το λόγο οι τρόποι βελτίωσης θα καθοριστούν σύμφωνα με τα χαρακτηριστικά του. Για την βελτίωση της αποδοτικότητας τα παρακάτω πρέπει να γίνουν:

1. Προεπιλογή μεταβλητών: Διαφορετικές δοκιμές χρειάζεται να γίνουν για να επιλεχτούν τα κατάλληλα χαρακτηριστικά. Αυτό θα οδηγήσει σε βελτίωση της απόδοσης καθώς οι ανεπιθύμητες μεταβλητές θα κοπούν.

2. K-Fold cross validation: Το cross validation αυξάνει την ακρίβεια ενός μοντέλου από μόνο του.
3. Υβριδικό μοντέλο: Χρήση του υβριδικού μοντέλου , για παράδειγμα logistic regression μετά από decision trees, για να βελτιωθεί η απόδοση.

Επίσης, ο GradientBoostedTrees είχε την καλύτερη απόδοση στο παραπάνω dataset, οπότε οι βελτιώσεις θα μπορούσαν να εφαρμοστούν με βάση αυτόν. Αυτό που προτείνεται είναι να δημιουργηθεί ένα ισορροπημένο σύνολο δεδομένων ή ένα τυχαίο ισορροπημένο σύνολο δεδομένων, που θα εκπαιδευτεί σύμφωνα με τις προηγούμενες τεχνικές. Επίσης, ο GradientBoostedTrees πετυχαίνει καλύτερες προβλέψεις , γιατι αυξάνει την τυχαιότητα και μειώνει την ομοιότητα στο δείγμα. Μπορούν να εφαρμοστούν πολλά διαφορετικά μοντέλα σε αυτά τα δεδομένα, ώστε να ελεγχθεί η βελτίωση της ακρίβειας. Επομένως, προτείνεται να χρησιμοποιηθούν δεδομένα με διαφορετικά μοντέλα, γιατί υπάρχει μεγάλη πιθανότητα να κάνετε καλύτερα από τα προηγούμενα μοντέλα.

5.2 Βελτιώσεις που αφορούν το dataset

1. Προσθήκη ενός νέου feature το οποίο αφορά αν το άρθρο υπάρχει σε anti-hoax πηγή ή όχι (greekhoaxes-factchecker).
2. Προσθήκη περισσότερων χαρακτηριστικών που αφορούν το κείμενο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Αναφορές:

1. Rubin, V. L., Conroy, N., & Chen, Y. (2015, January). Towards news verification: Deception detection methods for news discourse. In Hawaii International Conference on System Sciences.

2. Rubin, V. L., Conroy, N. J., Chen, Y., & Cornwell, S. (2016). Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. In Proceedings of NAACL-HLT (pp. 7-17).

3. Rubin, V. L., Chen, Y., & Conroy, N. J. (2015). Deception detection for news: three types of fakes. Proceedings of the Association for Information Science and Technology, 52(1), 1-4.

4. Castillo, C., Mendoza, M., & Poblete, B. (2011, March). Information credibility on twitter. In Proceedings of the 20th international conference on World wide web (pp. 675-684). ACM.

5. Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology, 52(1), 1-4.

6. Rubin, V. L. (2017). Deception Detection and Rumor Debunking for Social Media. The SAGE Handbook of Social Media Research Methods, 342.

7. Nagura, R., Seki, Y., Kando, N., & Aono, M. (2006, August). A method of rating the credibility of news documents on the web. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 683-684). ACM

8. Kang, B., O'Donovan, J., & Höllerer, T. (2012, February). Modeling topic specific credibility on twitter. In Proceedings of the 2012 ACM international conference on Intelligent User Interfaces (pp. 179-188). ACM.

9. Qazvinian, V., Rosengren, E., Radev, D. R., & Mei, Q. (2011, July). Rumor has it: Identifying misinformation in microblogs. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 1589-1599). Association for Computational Linguistics.

10. Craig Silverman (BuzzFeed Founding Editor, Canada), Lauren Strapagiel (BuzzFeed Staff) Hamza Shaban (BuzzFeed News Reporter) Ellie Hall (BuzzFeed News Reporter), Jeremy Singer-Vine (BuzzFeed News Reporter) (October, 2016) Hyperpartisan Facebook Pages Are Publishing False And Misleading Information At An Alarming Rate in Kaggle.

11. Megan Risdal (November, 2017), Getting Real About Fake News in BuzzFeed News.