Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Ανίχνευση κακόβουλων αρχείων PHP με τη χρήση ευφυών τεχνικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΡΑΦΑΗΛ Ν. ΣΚΟΥΛΟΣ

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Συνεπιβλέπων : Γεώργιος Αλεξανδρίδης
Ε.ΔΙ.Π. Ε.Μ.Π.

Αθήνα, Μάιος 2018

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Ανίχνευση κακόβουλων αρχείων PHP με τη χρήση ευφυών τεχνικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΡΑΦΑΗΛ Ν. ΣΚΟΥΛΟΣ

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Συνεπιβλέπων : Γεώργιος Αλεξανδρίδης
Ε.ΔΙ.Π. Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29η Μαΐου 2018.

.....	.....	.....
Ανδρέας-Γεώργιος Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Παναγιώτης Τσανάκας Καθηγητής Ε.Μ.Π.	Γεώργιος Στάμου Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Μάιος 2018

Ραφαήλ Ν. Σκουλός

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

Περίληψη

Ο κακόβουλος κώδικας είναι οποιοσδήποτε κώδικας προστίθεται, αλλάζει ή καταργείται από ένα σύστημα λογισμικού για να προξενήσει σκόπιμα βλάβη ή να υπονομεύσει την επιθυμητή λειτουργία του συστήματος. Στην εποχή μας όπου το Διαδίκτυο είναι μέρος της καθημερινότητας μας, η μόλυνση διακομιστών Παγκόσμιου Ιστού με κακόβουλο κώδικα είναι πολύ συχνό φαινόμενο με αρνητικές συνέπειες τόσο για τον ιδιοκτήτη του όσο και για τους χρήστες του.

Για τον εντοπισμό τέτοιου κώδικα έχουν αναπτυχθεί πολλά εργαλεία λογισμικού στο εμπόριο τα οποία έχουν υψηλά ποσοστά επιτυχίας. Το πρόβλημα όμως με τα εργαλεία αυτά είναι ότι αποτυγχάνουν να αναγνωρίσουν κακόβουλο κώδικα τον οποίο συναντάμε για πρώτη φορά και τον οποίο οι δημιουργοί του έχουν σκόπιμα φτιάξει με τον τρόπο αυτό χρησιμοποιώντας διάφορες τεχνικές ”θόλωσης” (obfuscation) ώστε να μην αναγνωρίζεται από τέτοια προϊόντα.

Ο σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος το οποίο θα αναγνωρίζει αν ένα αρχείο είναι μολυσμένο ή όχι και θα είναι σε θέση να αναγνωρίζει και τα αρχεία που έχουν υποστεί θόλωση εκτός από όλα τα υπόλοιπα. Στο πλαίσιο αυτό συλλέξαμε μολυσμένα και μη μολυσμένα αρχεία, εξάγαμε τρία σύνολα χαρακτηριστικών από αυτά και έπειτα χρησιμοποιήσαμε τα χαρακτηριστικά ώστε με την χρήση μηχανικής μάθησης να φτιάξουμε μοντέλα που είναι σε θέση να προβλέψουν αν ένα αρχείο είναι μολυσμένο ή όχι. Τέλος αξιολογήσαμε τα αποτελέσματα κάθε αλγορίθμου και κάθε συνόλου χαρακτηριστικών και τα συγκρίναμε μεταξύ τους. Ένα σημαντικό πρόβλημα που αντιμετωπίσαμε ήταν η ανισορροπία κλάσεων στα δεδομένα εισόδου.

Συγκεκριμένα, τα χαρακτηριστικά που εξάγαμε αφορούν την λεξικογραφική ανάλυση του κειμένου και τη συχνότητα εμφάνισης των συναρτήσεων της γλώσσας. Οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήσαμε είναι τα Δέντρα Αποφάσεων, οι Μηχανές Διανυσμάτων Υποστήριξης και η Στοχαστική Κατάβαση Κλίσης. Οι κυριότερες μετρικές αξιολόγησης που χρησιμοποιήθηκαν είναι η Ακρίβεια, η Ανάκληση και η μετρική F1 στην κλάση μειοψηφίας καθώς και ο Γεωμετρικός Μέσος. Τέλος το πρόβλημα της ανισορροπίας κλάσεων στα δεδομένα εισόδου το αντιμετωπίσαμε με τη χρήση μάθησης με ευαισθησία κόστους.

Τέλος, αναλύονται τα αποτελέσματα και τα συμπεράσματα που προέκυψαν από τα πειράματα του εκπονήσαμε και δίνονται και μελλοντικές κατευθύνσεις έρευνας.

Λέξεις κλειδιά

Μηχανική Μάθηση, Δέντρα Αποφάσεων, Μηχανές Διανυσμάτων Υποστήριξης, Στοχαστική Κατάβαση Κλίσης, Δεδομένα, Εκπαίδευση, Μοντέλο, Κακόβουλος Κώδικας, PHP, Ανάκληση, Ακρίβεια, Μετρική F1, Γεωμετρικός Μέσος, Ανισορροπία Κλάσεων

Abstract

Malicious code is any code added, modified or removed by a software system to deliberately damage or compromise the system's functionality. In our time, where the Internet is part of our everyday life, infecting web servers with malicious code is a very common phenomenon with negative consequences for both the owner and its users.

To identify such code, there have been developed many software solutions that have high success rates. The problem with these solutions is that they fail to recognize malicious code that is encountered for the first time and whose creators have deliberately obfuscated it so as not be recognizable by this type of software.

The purpose of this thesis is to develop a system that will recognize whether a file is malicious or benign, and will be able to recognize obfuscated files in addition to other files. In this context, we collected malicious and benign files, extracted three sets of attributes from them, and then we used these features to create models by using machine learning techniques that are able to predict whether a file is infected or not. Finally, we evaluated the results of each algorithm and set of attributes and compared them to each other. One important problem we encountered was the class imbalance in the input data.

In particular, the features were related to the lexical analysis of the text and the frequency of occurrence of the programming language functions. The machine learning algorithms we used were Decision Trees, Support Vector Machines, and Stochastic Gradient Descent. The main metrics used are Precision, Recall and F1 Measure in the minority class as well as the Geometric Mean. Finally, we dealt with the problem of the class imbalance in the input data by using cost-sensitive learning.

Finally, we analyze the results and conclusions of our experiments and provide future research directions.

Key words

Machine Learning, Decision Trees, Support Vector Machines, Stochastic Gradient Descent, Data, Training, Model, Malicious Code, PHP, Metric, Recall, Precision, F1 Measure, Geometric Mean, Class Imbalance

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε στο πλαίσιο του προπτυχιακού προγράμματος σπουδών της Σχολής Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου και σηματοδοτεί την ολοκλήρωση των σπουδών μου ενώ συγχρόνως αποτελεί το ερέθισμα για περαιτέρω έρευνα στο συγκεκριμένο αντικείμενο.

Προτού όμως αναφερθώ στη περιγραφή της εργασίας και στα αποτελέσματα που προέκυψαν, θα ήθελα να ευχαριστήσω θερμά τους ανθρώπους οι οποίοι μέσω της συνεργασίας μας, συνέβαλαν σημαντικά στην ολοκλήρωση αυτής της εργασίας.

Αρχικά θα ήθελα να απευθύνω τις ευχαριστίες μου στον επιβλέποντα κ. Ανδρέα-Γεώργιο Σταφυλοπάτη, Καθηγητή Ε.Μ.Π, ο οποίος μου προσέφερε τη δυνατότητα να εκπονήσω την διπλωματική μου σε ένα αντικείμενο ιδιαίτερα ελκυστικό και ενδιαφέρον για μένα και να διευρύνω τις επιστημονικές μου γνώσεις. Παράλληλα θα ήθελα να ευχαριστήσω τους κ.κ. Παναγιώτη Τσανάκα, Καθηγητή Ε.Μ.Π και Γεώργιο Στάμου, Αναπληρωτή Καθηγητή Ε.Μ.Π για την τιμή που μου έκαναν να είναι μέλη της επιτροπής εξέτασης της διπλωματικής εργασίας.

Επίσης οφείλω ιδιαίτερες ευχαριστίες στον κ. Γεώργιο Αλεξανδρίδη, Ε.ΔΙ.Π Ε.Μ.Π. για το χρόνο που αφιέρωσε και την θεμελιώδη του συνεισφορά στην εκπόνηση της συγκεκριμένης εργασίας. Η στήριξη του, επιστημονική και πνευματική, καθώς και η καθοδήγηση του σε όλη τη διάρκεια της πορείας αυτής συνέβαλαν τα μέγιστα στην επίτευξη ενός πολύ σημαντικού για εμένα στόχου. Η προθυμία του να με βοηθήσει μέσω της εμπειρίας και των γνώσεων του σε οποιαδήποτε δυσκολία συνάντησα στάθηκαν καθοριστικές και η συνεργασία μας θεωρώ πως ήταν άκρως επιτυχημένη και εποικοδομητική.

Τέλος, με εξίσου μεγάλη θέρμη θέλω να αναφερθώ και να ευχαριστήσω την οικογένεια μου , η οποία με στήριξε όλα αυτά τα χρόνια σε όλες τις δύσκολες στιγμές, καθώς και τους φίλους και τους συμφοιτητές μου, οι οποίοι στάθηκαν δίπλα μου σε όλη τη διάρκεια της ακαδημαϊκής μου πορείας, ο καθένας με τον δικό του ξεχωριστό τρόπο.

Ραφαήλ Ν. Σκουλός,

Αθήνα, 29η Μαΐου 2018

Περιεχόμενα

Κατάλογος πινάκων

Κατάλογος σχημάτων

Κατάλογος Αλγορίθμων

Κεφάλαιο 1

Εισαγωγή

1.1 Περιγραφή του προβλήματος

1.1.1 Κακόβουλος κώδικας PHP

Ο κακόβουλος κώδικας είναι οποιοσδήποτε κώδικας προστίθεται, αλλάζει ή καταργείται από ένα σύστημα λογισμικού για να προξενήσει σκόπιμα βλάβη ή να υπονομεύσει την επιθυμητή λειτουργία του. Τέτοιος κώδικας έχει χρησιμοποιηθεί πολλές φορές για να θέσει σε κίνδυνο συστήματα υπολογιστών, για να καταστρέψει τις πληροφορίες τους και να τις καταστήσει άχρηστες. Επίσης έχει χρησιμοποιηθεί για τη συλλογή πληροφοριών, όπως κωδικών πρόσβασης και αριθμούς πιστωτικών καρτών, καθώς και για τη διανομή πληροφοριών, όπως πορνογραφία, όλα χωρίς τη γνώση του χρήστη του συστήματος. Δεδομένου ότι περισσότεροι αρχάριοι χρήστες αποκτούν εξελιγμένους υπολογιστές με υψηλής ταχύτητας σύνδεσης στο διαδίκτυο, η δυνατότητα για περαιτέρω κατάχρηση είναι μεγάλη.

Με πολλούς τρόπους, η *PHP* (PHP: Hypertext Preprocessor)[1] έχει αποτελέσει μια πλατφόρμα-στόχο για κακόβουλες επιθέσεις. Για τους προγραμματιστές και τους τεχνικούς, η PHP είναι εύκολη στην εκμάθηση. Σχεδόν όλοι οι *Διακομιστές του Παγκόσμιου Ιστού* (Web Servers) εκτελούν PHP κώδικα, οπότε υπάρχουν τεράστιοι αριθμοί δυνητικών στόχων. Και όπως πολλές μορφές εκτελέσιμου κακόβουλου λογισμικού, τα πιο επιτυχημένα κακόβουλα αρχεία PHP επιτρέπουν στους χειριστές τους να ελέγχουν και να χειρίζονται διακομιστές ιστού για δικό τους όφελος και κέρδος.

Η διαδικασία «μόλυνσης» ενός διακομιστή ιστού με κακόβουλο PHP κώδικα απαιτεί την είσοδο στον διακομιστή ιστού χρησιμοποιώντας πλαστά πιστοποιητικά, τοποθέτηση των αρχείων σε *διαδρομές* (paths) προσβάσιμες από τους έξω χρήστες και αποσύνδεση. Αυτό μπορεί να επιτευχθεί χειροκίνητα, δηλαδή προσβάλλοντας έναν διακομιστή κάθε φορά, αλλά γίνεται πιο συχνά χρησιμοποιώντας αυτοματοποιημένες διαδικασίες που επιχειρούν να εισέλθουν σε μεγάλο αριθμό διακομιστών.

Οι πιο απλοϊκές μορφές κακόβουλων αρχείων PHP απλά ανακατευθύνουν τους επισκέπτες της ιστοσελίδας σε διαφορετική ιστοσελίδα. Ενώ η συμπεριφορά αυτή χαρακτηρίζεται σίγουρα ως κακόβουλη, δεν είναι ιδιαίτερα δυναμική ή ακόμα και ιδιαίτερα ενδιαφέρουσα. Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι μορφές αρχείων που προσφέρουν απομακρυσμένη πρόσβαση στο σύστημα αρχείων του διακομιστή, καθώς και τα PHP αρχεία που, όταν εκτελούνται, αναγκάζουν τους διακομιστές να ενταχθούν στα *botnets* [Esla12]. Ένα botnet είναι ένας αριθμός συσκευών συνδεδεμένων στο Διαδίκτυο, εκ των οποίων το καθένα εκτελεί ένα ή περισσότερα *bots*. Τα bots είναι προγράμματα υπολογιστή που εκτελούν αυτοματοποιημένες εργασίες στο Διαδίκτυο. Τα botnets μπορούν να χρησιμοποιηθούν για την εκτέλεση επιθέσεων *κατανεμημένης άρνησης υπηρεσίας* (Distributed Denial-of-Service - DDoS), κλοπή δεδομένων, αποστολή ανεπιθύμητων μηνυμάτων και επιτρέπουν στον εισβολέα να έχει πρόσβαση στη συσκευή και στη σύνδεσή της. Η ίδια η λέξη «botnet» προέρχεται από τον συνδυασμό των λέξεων «robot» και «network» και συνήθως έχει αρνητική χροιά.

1.1.2 Λειτουργικότητες κακόβουλων αρχείων PHP

Ο πιο κοινός *πελάτης botnet* (botnet client) είναι ένα αρχείο το οποίο όταν εκτελείται, δηλαδή όταν κάποιος μεταβεί στη σελίδα σε ένα Διακομιστή Ιστού όπου βρίσκεται το αρχείο, ξεκινά μια διαδικα-

[1] http://php.net/

σία σύνδεσης με ένα *διακομιστή συνομιλίας IRC* (Internet Relay Chat), που ο ιδιοκτήτης του αρχείου έχει ρυθμίσει να συνδέεται. Το αρχείο αυτό συνοδεύεται από πλήρεις οδηγίες, οι οποίες περιλαμβάνουν την ικανότητα εκτέλεσης αυθαίρετων εντολών, την εισαγωγή PHP αρχείων ή εντολών, και την επιβολή επιθέσεων σε άλλους διακομιστές. Επίσης, περιέχει συνήθως κώδικα «connect back», ο οποίος όταν εκτελείται επιτρέπει στον χειριστή του bot να συνδεθεί από απόσταση στον πληττόμενο διακομιστή, παρακάμπτοντας τα τυπικά *τείχη προστασίας* (firewalls) [webr11].

Υπάρχουν ακόμα λειτουργικότητες για την αποστολή ανεπιθύμητων μηνυμάτων όσο το δυνατόν γρηγορότερα. Κάποιες από αυτές δίνουν στον αποστολέα τη δυνατότητα να προκαθορίσει τις κεφαλίδες και το σώμα του μηνύματος *ηλεκτρονικού ταχυδρομείου* (e-mail) και μέσω ενός αρχείου κειμένου με διευθύνσεις ηλεκτρονικού ταχυδρομείου, να στείλει στις διευθύνσεις αυτές την ανεπιθύμητη αλληλογραφία.

Μια άλλη λειτουργικότητα είναι αυτή που επιχειρεί να χρησιμοποιήσει οποιαδήποτε εντολή αντιγραφής αρχείου που ο διακομιστής είναι σε θέση να εκτελέσει, για να ανακτήσει τα ωφέλιμα αρχεία από απομακρυσμένους διακομιστές ή να δημιουργήσει διπλότυπα αρχεία του κακόβουλου κώδικα σε πολλαπλές τοποθεσίες του επηρεαζόμενου διακομιστή (λειτουργία *worm*).

Τέλος μια λειτουργικότητα που συναντάται συχνά είναι αυτή του *απομακρυσμένου φλοιού* (remote shell), όπου παρέχεται μια πλήρως απομακρυσμένη πρόσβαση στο διακομιστή, τουλάχιστον στο πλαίσιο των δικαιωμάτων της διαδικασίας στην οποία εκτελείται το σενάριο. Αυτά τα σενάρια είναι δυνητικά τα πιο επικίνδυνα, επειδή δίνουν σε έναν απομακρυσμένο εισβολέα το ίδιο επίπεδο ελέγχου πάνω στο διακομιστή, όμοιο με αυτό ενός διαχειριστή που έχει φυσική πρόσβαση στον εξοπλισμό.

1.1.3 Τρόποι ανίχνευσης κακόβουλου κώδικα

Έχουν αναπτυχθεί πολλά εργαλεία λογισμικού, τα οποία χρησιμοποιούνται για την ανίχνευση ήδη γνωστών τύπων κακόβουλου κώδικα και παρουσιάζουν μεγάλα ποσοστά επιτυχίας. Τα εργαλεία αυτά ψάχνουν στον κώδικα για ήδη γνωστά μοτίβα κακόβουλου κώδικα ή λειτουργούν βάσει κανόνων για να θεωρήσουν ένα αρχείο μολυσμένο ή όχι. Οι δημιουργοί των μολυσμένων αρχείων, πολλές φορές γνωρίζοντας τον τρόπο λειτουργίας των εργαλείων ανίχνευσης, αποκρύπτουν τα χαρακτηριστικά αυτά του αρχείου που το κατατάσσουν ως κακόβουλο μέσω του της τεχνικής της *θόλωσης* (obfuscation) [webr11]. Για παράδειγμα, συχνά τέτοια αρχεία περιέχουν αναφορές σε γνωστές εταιρείες για να παραπλανήσουν το σύστημα ανίχνευσης. Σκοπός μας είναι να φτιάξουμε ένα σύστημα που θα είναι σε θέση να κατατάσσει τα αρχεία αυτά με μεγάλη ακρίβεια ως μολυσμένα ή μη μολυσμένα.

1.2 Μηχανική Μάθηση

Η *Μηχανική Μάθηση* (Machine Learning) είναι ένα υποπεδίο της επιστήμης των υπολογιστών, το οποίο προέκυψε από την τομή της Επιστήμης των Υπολογιστών με την Στατιστική [Mitc06]. Το 1959, ο Arthur Samuel ορίζει τη μηχανική μάθηση ως «Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί» [Simo13, p. 89]. Η χρήση του και η έρευνα στο πεδίο αυτό έχει γνωρίσει μεγάλη ανάπτυξη τα τελευταία 30 χρόνια και χρησιμοποιείται σε τομείς όπως η αναζήτηση στο διαδίκτυο, στο φιλτράρισμα ανεπιθύμητης αλληλογραφίας, στα συστήματα συστάσεων, στις συναλλαγές μετοχών, κ. ά. Στην ουσία, η μηχανική μάθηση ασχολείται με την κατασκευή αλγορίθμων οι οποίοι «μαθαίνουν» μέσω των δεδομένων και στη συνέχεια είναι σε θέση να κάνουν προβλέψεις για αυτά. Η μέθοδος αυτή είναι σε θέση να λύσει προβλήματα που ο κλασικός προγραμματισμός αδυνατεί, με μεγαλύτερο όμως κόστος σε υπολογιστικούς πόρους και χρόνο. Επίσης με την αύξηση των δεδομένων που έχει στην διάθεση του ένας τέτοιος αλγόριθμος, θα είναι σε θέση να επιλύει δυσκολότερα προβλήματα με μεγαλύτερη ακρίβεια [Domi12].

Υπάρχουν τρεις μεγάλες κατηγορίες αλγορίθμων μηχανικής μάθησης, ανάλογα με τον τρόπο που επεξεργάζονται τα δεδομένα εισόδου, την ανατροφοδότηση που υπάρχει στο σύστημα εκμάθησης και τα παραγόμενα αποτελέσματα [JRus95]:

1. *Επιβλεπόμενη μάθηση* (Supervised learning): Οι αλγόριθμοι δέχονται ως είσοδο τα δεδομένα μαζί με τις ετικέτες τους και καλούνται να μάθουν έναν γενικό κανόνα προκειμένου να βρίσκουν τις αντιστοιχήσεις μεταξύ δεδομένων και ετικετών.

2. *Μη-επιβλεπόμενη μάθηση* (Unsupervised learning): Οι αλγόριθμος δέχονται ως είσοδο δεδομένα χωρίς ετικέτες και καλούνται να βρουν την δομή τους. Η μη-επιβλεπόμενη μάθηση μπορεί να ανακαλύπτει κρυμμένα μοτίβα σε δεδομένα ή μπορεί να αποτελέσει μέσο για την εύρεση χαρακτηριστικών, που στη συνέχεια χρησιμοποιούνται στη διαδικασία της μάθησης.

3. *Ενισχυτική μάθηση* (Reinforced learning): Αλγόριθμοι που αλληλεπιδρούν με ένα δυναμικό περιβάλλον, για την επίτευξη ενός συγκεκριμένου στόχου. Στην περίπτωση αυτή δεν υπάρχει πληροφορία που να υποδεικνύει πόσο κοντά ή μακρυά έχει φτάσει η εκπαίδευση από τον στόχο. Παράδειγμα ενισχυτικής μάθησης θα μπορούσε να είναι η αυτόνομη οδήγηση ενός οχήματος.

Στο πλαίσιο αυτής της διπλωματικής εργασίας θα ασχοληθούμε με προβλήματα επιβλεπόμενης μάθησης.

1.3 Κυριότερες Προκλήσεις

Το ζήτημα το οποίο θα απασχολήσει την παρούσα εργασία είναι η ανίχνευση νέων τύπων κακόβουλου κώδικα PHP. Για το σκοπό αυτό, θα χρησιμοποιηθούν τεχνικές μηχανικής μάθησης. Θα φτιαχτεί δηλαδή ένα σύστημα το οποίο θα είναι σε θέση να αναγνωρίζει με μεγάλη ακρίβεια αν ένα PHP αρχείο είναι μολυσμένο ή όχι, δίχως να αναζητά σε αυτό ήδη γνωστά μοτίβα μολυσμένου κώδικα. Αντ' αυτού θέλουμε να είναι σε θέση να αναγνωρίζει νέους τύπους κακόβουλου κώδικα, χωρίς να «μπερδεύεται» όταν ένας ήδη γνωστός τύπος κακόβουλου κώδικα αλλάζει για να το εμποδίζει να το αναγνωρίσει, όπως συμβαίνει συνήθως στη πράξη.

Για να το πετύχουμε αυτό, κατασκευάσαμε ένα σύστημα που θα περιέχει έναν δυαδικό ταξινομητή που θα αποφασίζει αν ένα αρχείο είναι μολυσμένο ή όχι. Τα χαρακτηριστικά που θα περιέχουν τα διανύσματα εισόδου πρέπει αν είναι χαρακτηριστικά που είναι παρόντα σε μολυσμένα αρχεία που έχουν υποστεί θόλωση. Βέβαια υπάρχει η περίπτωση κάποια μη μολυσμένα αρχεία έχουν υποστεί θόλωση ενώ κάποια μολυσμένα όχι. Σκοπός μας είναι να ταξινομήσουμε σχεδόν όλα τα μολυσμένα αρχεία σωστά και πολύ λίγα μη μολυσμένα αρχεία λάθος. Για να το πετύχουμε αυτό, πειραματιστήκαμε με διάφορους αλγορίθμους μηχανικής μάθησης και δοκιμάσαμε διαφόρους τύπους χαρακτηριστικών.

Ένα σημαντικό εμπόδιο που αντιμετωπίσαμε ήταν η ανισορροπία μεταξύ του αριθμού των μολυσμένων και των μη μολυσμένων αρχείων που περιέχουν τα δεδομένα μας. Βέβαια η ανισορροπία αυτή δεν είναι επιτηδευμένη, καθώς αυτή υπάρχει και στα δεδομένα που παρατηρούνται σε πραγματικά συστήματα. Για την αντιμετώπιση της πειραματιστήκαμε με διαφόρους μεθόδους που προτείνονται στη βιβλιογραφία.

1.4 Δομή της εργασίας

Τα επόμενα κεφάλαια της διπλωματικής εργασίας έχουν την ακόλουθη μορφή. Στο Κεφάλαιο 2 περιγράφεται η προεπεξεργασία των δεδομένων. Πιο συγκεκριμένα αναλύεται ο τρόπος που γίνεται η λεξικογραφική ανάλυση μέσω της κατασκευής ενός λεκτικού αναλυτή και πως από τα αποτελέσματα της εξάγουμε τα λεξικογραφικά χαρακτηριστικά που θα χρησιμοποιήσουμε σαν είσοδο στους αλγορίθμους μηχανικής μάθησης. Επίσης αναλύεται και το δεύτερο σύνολο χαρακτηριστικών που θα χρησιμοποιήσουμε, το οποίο σχετίζεται με την συχνότητα χρησιμοποίησης των διαφόρων συναρτήσεων της γλώσσας προγραμματισμού PHP.

Στο Κεφάλαιο 3 περιγράφεται ο τρόπος λειτουργίας των αλγορίθμων μηχανικής μάθησης με επιβλεπόμενη μάθηση που εξετάσαμε. Συγκεκριμένα οι αλγόριθμοι αυτοί είναι τα *Δέντρα Αποφάσεων* (Decision Trees - ΔΑ), οι *Μηχανές Διανυσμάτων Υποστήριξης* (Support Vector Machines - ΜΔΥ) και

η *Στοχαστική Κατάβαση Κλίσης* (Stochastic Gradient Descent - ΣΚΚ). Για κάθε ένα από τους αλγορίθμους αυτούς αναλύεται ο τρόπος λειτουργίας του, τα ιδιαίτερα χαρακτηριστικά του και αναφέρονται τα πλεονεκτήματα και τα μειονεκτήματα του.

Στο Κεφάλαιο 4 περιγράφεται η πειραματική διαδικασία που ακολουθήθηκε καθώς και τα αποτελέσματα που προέκυψαν από αυτή. Συγκεκριμένα αναλύεται το σύνολο των δεδομένων που χρησιμοποιήθηκε στην πειραματική διαδικασία, το σύστημα που κατασκευάσαμε, οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση των αποτελεσμάτων, η μέθοδος επικύρωσης των αποτελεσμάτων, και τέλος παρουσιάζονται και σχολιάζονται τα αποτελέσματα που προέκυψαν. Επίσης αναλύεται το πρόβλημα της ανισορροπίας κλάσεων καθώς και οι τρόποι αντιμετώπισης του.

Τέλος, στο Κεφάλαιο 5 παρουσιάζονται τα συμπεράσματα που προέκυψαν από την ανάλυση των αποτελεσμάτων και αναφέρονται κάποιες μελλοντικές κατευθύνσεις για την περαιτέρω διερεύνηση του θέματος που πραγματεύεται η παρούσα διπλωματική εργασία.

Κεφάλαιο 2

Προεπεξεργασία των δεδομένων

2.1 Λεξικογραφική ανάλυση

Μια βασική κατηγορία χαρακτηριστικών που μπορούν να εξαχθούν από πηγαίο κώδικα αρχείων είναι αυτά που προέρχονται από τη λεξικογραφική ανάλυση του, η οποία πραγματοποιείται με την χρήση ενός *λεκτικού αναλυτή* (lexical analyzer). Κατά τη λεξικογραφική ανάλυση, ο πηγαίος κώδικας ο οποίος έχει τη μορφή μιας ακολουθίας από χαρακτήρες μετατρέπεται σε μια ακολουθία από *λεκτικές μονάδες* (tokens). Μια λεκτική μονάδα είναι μια ακολουθία από χαρακτήρες που έχουν καταχωρηθεί ανάλογα με τους κανόνες της γλώσσας ως κάποιο σύμβολο (π.χ., αριθμός, μεταβλητή, τελεστής). Ο λεκτικός αναλυτής κατηγοριοποιεί τις μονάδες ανάλογα με τον τύπο συμβόλου τους, χρησιμοποιώντας τον πίνακα συμβόλων της εκάστοτε γλώσσας προγραμματισμού [Πα02, pp. 70-74]. Δεν υπάρχει κάποιος κανόνας που να προσδιορίζει ποιες είναι οι λεκτικές μονάδες στη γενική περίπτωση. Συνήθως, λεκτικές μονάδες θεωρούνται οι λέξεις κλειδιά, τα ονόματα, οι σταθερές, οι τελεστές και οι διαχωριστές [Πα02, pp. 69-70].

Οι λεκτικές μονάδες μιας γλώσσας προγραμματισμού κατατάσσονται σε κατηγορίες, κάθε μια από τις οποίες μπορεί να περιγραφεί από μια κανονική έκφραση και έτσι να αναγνωρισθεί από ένα πεπερασμένο αυτόματο [Πα02, pp. 69-70]. Επομένως το πρόβλημα της αναγνώρισης των λεκτικών μονάδων φαίνεται καταρχήν να συμπίπτει με το πρόβλημα της σχεδίασης τόσων πεπερασμένων αυτομάτων όσες και οι κατηγορίες των προς αναγνώριση λεκτικών μονάδων [Πα02, pp. 70-74].

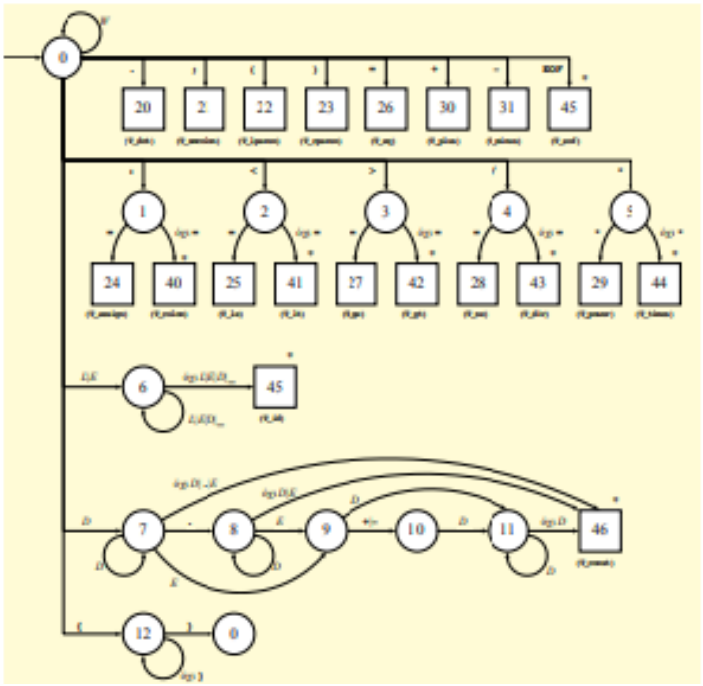
Όμως το πρόβλημα στην πραγματικότητα είναι λίγο πιο πολύπλοκο. Οι συμβολοσειρές που απαρτίζουν τις λεκτικές μονάδες δεν είναι ξεχωριστές αλλά αποτελούν τμήματα της συμβολοσειράς εισόδου. Έτσι κατά την λεξικογραφική ανάλυση πρέπει να αναγνωρισθεί μια λεκτική μονάδα, η οποία είναι πρόθεμα του εναπομείναντος τμήματος της συμβολοσειράς εισόδου, χωρίς όμως να είναι γνωστό το τέλος της. Το γεγονός δηλαδή ότι το τέλος των λεκτικών μονάδων είναι άγνωστο δημιουργεί πρόβλημα στη αναγνώρισή τους [Πα02, pp. 70-74].

Το τέλος μια λεκτικής μονάδας αναγνωρίζεται στην πράξη με την ανασκόπηση επιπλέον χαρακτήρων. Στην περίπτωση αυτή απαιτείται να ενημερώνεται ο δείκτης του τρέχοντος χαρακτήρα στη συμβολοσειρά εισόδου, ούτως ώστε όταν αναγνωρισθεί μια λεκτική μονάδα, αυτός να δείχνει στην αρχή της επόμενης. Αυτό επιτυγχάνεται με την κατάλληλη οπισθοδρόμηση του δείκτη, που σημαίνει ότι κατά την εξαγωγή των λεκτικών μονάδων ορισμένοι χαρακτήρες στη συμβολοσειρά εισόδου διαβάζονται περισσότερες από μια φορές [Πα02, pp. 77-76].

Το συνολικό πρόβλημα της λεξικογραφικής ανάλυσης λύνεται με την χρησιμοποίηση κατάλληλα τροποποιημένων *Ντετερμινιστικών Πεπερασμένων Αυτομάτων* (Deterministic Finite Automata - ΝΠΑ), τα οποία:

1. Διαβάζουν ενδεχομένως περισσότερους χαρακτήρες από όσους έχει μια λεκτική μονάδα.
2. Οπισθοδρομούν αν αυτό χρειαστεί.
3. Διαθέτουν έξοδο, στην οποία προκύπτει κάθε φορά η λεκτική μονάδα που αναγνωρίζεται.

Για την παράσταση των τροποποιημένων ΝΠΑ χρησιμοποιείται ένα διάγραμμα ειδικής μορφής που ονομάζεται *διάγραμμα κατάστασης* (state diagram), όπως αυτό που φαίνεται στο παράδειγμα του



Σχήμα 2.1: Παράδειγμα διαγράμματος καταστάσεων

Σχήματος 2.1. Τα διαγράμματα κατάστασης, από τα οποία προκύπτει άμεσα ο γράφος μετάβασης των αντίστοιχων απλών *Μη-ντετερμινιστικών Πεπερασμένων Αυτομάτων* (Non-deterministic Finite Automata - ΜΠΑ), είναι πολύ χρήσιμα γιατί απεικονίζουν εύγλωττα τις λειτουργίες που επιτελούνται στη λεξικογραφική ανάλυση. Η άφιξη σε μια τελική κατάσταση ενός διαγράμματος κατάστασης ισοδυναμεί με αναγνώριση κάποιας λεκτικής μονάδας. Προκαλεί την οπισθοδρόμηση, αν αυτό είναι απαραίτητο, καθώς και την έξοδο του κατάλληλου κωδικού [Πα02, pp. 76-84].

Τα βήματα για την υλοποίηση ενός λεκτικού αναλυτή είναι τα εξής [Πα02, pp. 76-84]:

1. Καταγραφή και ταξινόμηση των χαρακτήρων: Στο βήμα αυτό γίνεται η καταγραφή των χαρακτήρων που αποτελούν το αλφάβητο της αρχικής γλώσσας. Για τον σκοπό αυτό καταγράφονται οι λεκτικές μονάδες της αρχικής γλώσσας και καταγράφεται το σύνολο των χαρακτήρων που τις απαρτίζουν. Πολλές φορές χρησιμοποιείται ως αλφάβητο της αρχικής γλώσσας ένα υπερσύνολο (π.χ. όλοι οι χαρακτήρες ASCII). Επίσης, είναι δυνατόν να προστεθεί στο αλφάβητο της αρχικής γλώσσας και ένας επιπλέον χαρακτήρας που αντιστοιχεί στο τέλος της συμβολοσειράς εισόδου και συμβολίζεται συνήθως με EOF. Συχνά οι χαρακτήρες ταξινομούνται σε ομάδες που έχουν την ίδια λειτουργική θέση, καθώς με τον τρόπο αυτό μειώνεται το μέγεθος του αλφαβήτου εισόδου Σ των αυτομάτων που αναγνωρίζουν τις λεκτικές μονάδες και διευκολύνεται η περιγραφή τους με την χρήση κανονικών εκφράσεων. Η αντιστοίχιση των χαρακτήρων σε ομάδες μπορεί να περιγραφεί και να υλοποιηθεί ως μια συνάρτηση από το αλφάβητο της αρχικής γλώσσας στο σύνολο Σ.

2. Καταγραφή και ταξινόμηση των λεκτικών ομάδων: Στο βήμα αυτό γίνεται η καταγραφή των λεκτικών μονάδων της γλώσσας προγραμματισμού που εξετάζουμε. Στη καταγραφή αυτή οι λεκτικές μονάδες ταξινομούνται σε ομάδες που παρουσιάζουν λειτουργικές ομοιότητες (π.χ. ονόματα, αριθμητικές σταθερές). Επίσης κάποιες μορφές λεκτικών μονάδων (συνήθως λέξεις κλειδιά) χρειάζεται να εξαιρεθούν από γενικότερες ομάδες. Σε κάθε λεκτική μονάδα ή ομάδα λεκτικών μονάδων αντιστοιχεί ένας μοναδικός κωδικός που αποτελείται από τα εξής μέρη:

a. Έναν κωδικό αριθμό που συνήθως είναι φυσικός αριθμός αλλά για να διευκολύνεται η ανάγνωση του παριστάνεται με ένα κατάλληλο συμβολικό όνομα (π.χ. $T_{\text{plus}}, T_{\text{id}}$).

b. Την ακολουθία των χαρακτηριστικών που αντιστοιχεί στη λεκτική μονάδα. Η συμβολοσειρά αυτή συνήθως ονομάζεται *λέξημα* (lexeme) και χρησιμεύει για το διαχωρισμό

Θέματα που χρήζουν προσοχής σε αυτό το βήμα είναι επίσης τα εξής:

a. Ο τρόπος διαχωρισμού των λεκτικών μονάδων, που συνήθως είναι ένας ή περισσότεροι κενοί χαρακτήρες.

b. Τα σχόλια, δηλαδή τμήματα του κώδικα τα οποία αγνοούνται. Οι περισσότερες γλώσσες προγραμματισμού υποστηρίζουν σχόλια δύο ειδών: (i) τμηματικά σχόλια τα οποία περιβάλλονται από ειδικές ακολουθίες χαρακτήρων και (ii) σχόλια μιας γραμμής, τα οποία αρχίζουν με μια ειδική ακολουθία χαρακτήρων και εκτείνονται μέχρι το τέλος της τρέχουσας γραμμής.

3. Ενδιάμεση μνήμη και ανάνηψη από σφάλματα: Όπως αναφέρθηκε παραπάνω, ο λεκτικός αναλυτής πρέπει να έχει την δυνατότητα να οπισθοδρομεί. Αυτό είναι αναγκαίο κυρίως στην περίπτωση που το τέλος μιας λεκτικής μονάδας δεν μπορεί να αναγνωριστεί παρά μόνο αν διαβαστούν χαρακτήρες που ακολουθούν. Για παράδειγμα το τέλος μιας ακέραιας σταθεράς που περιγράφεται από τη κανονική έκφραση D^{+-} δεν μπορεί να εντοπισθεί παρά μόνο αν διαβαστεί ένας χαρακτήρες ο οποίος δεν είναι ψηφίο. Για την υλοποίηση της οπισθοδρόμησης, είναι αναγκαίο ο λεκτικός αναλυτής να διαβάζει την είσοδό του από μια *ενδιάμεση μνήμη* (buffer). Στη γενική περίπτωση που απαιτείται οπισθοδρόμηση περισσότερων του ενός χαρακτήρων, είναι απαραίτητο να καθοριστούν οι προδιαγραφές της ενδιάμεσης μνήμης, όπως η δομή, το μέγεθος και η οργάνωσή της. Η οργάνωση της ενδιάμεσης μνήμης εξαρτάται και από την πολιτική που ακολουθεί ο λεκτικός αναλυτής ως προς την *ανάνηψη από σφάλματα* (error recovery).

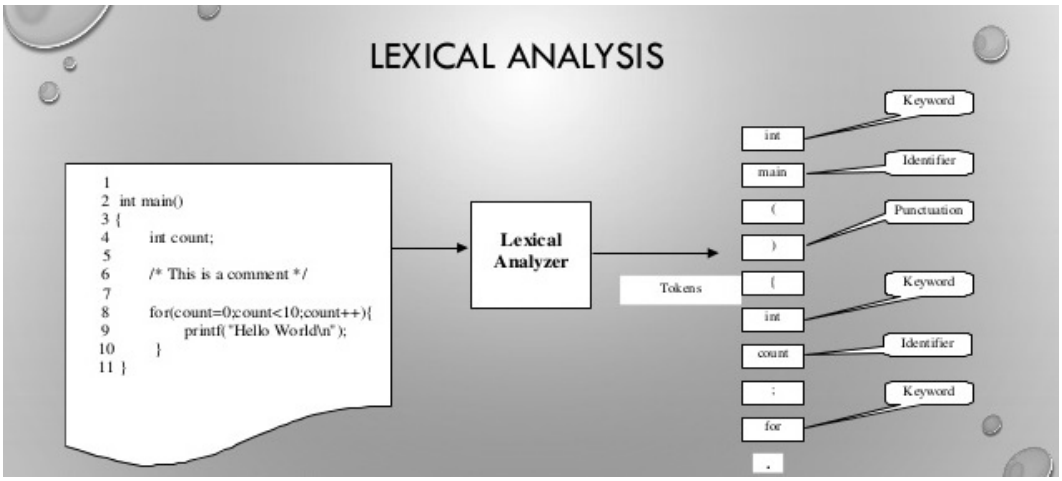
4. Σχεδίαση του διαγράμματος μετάβασης: Στο βήμα αυτό για κάθε ομάδα λεκτικών μονάδων κατασκευάζεται ένα ΝΠΑ, το οποίο στη συνέχεια μετατρέπεται σε διάγραμμα καταστάσεων. Ύστερα, όλα τα διαγράμματα καταστάσεων που αντιστοιχούν σε όλες τις λεκτικές μονάδες της αρχικής γλώσσας ενοποιούνται σε ένα γενικό διάγραμμα καταστάσεων, που περιγράφει τη λειτουργία του λεκτικού αναλυτή. Η ενοποίηση είναι πιθανόν να απαιτεί την τροποποίηση των επιμέρους διαγραμμάτων κατάστασης, έτσι ώστε το τελικό διάγραμμα καταστάσεων να είναι ντετερμινιστικό. Οι κενοί χαρακτήρες, οι άκυροι χαρακτήρες και τα σχόλια μπορούν να συμπεριληφθούν στο τελικό διάγραμμα καταστάσεων ως λεκτικές μονάδες που δεν έχουν έξοδο, αλλά επιστρέφουν στην αρχική κατάσταση.

5. Υλοποίηση του λεκτικού αναλυτή: Στο βήμα αυτό έχουμε τις εξής δύο επιλογές για την υλοποίηση του λεκτικού αναλυτή:

a. Περιγράφουμε με κώδικα τις μεταβάσεις του διαγράμματος καταστάσεων για κάθε κατάσταση χωριστά, ξεκινώντας από την αρχική κατάσταση. Ο τρόπος αυτός είναι περισσότερο εμπειρικός.

b. Χρησιμοποιούμε έναν πίνακα δ, που μας δίνει την επόμενη μετάβαση του λεκτικού αναλυτή. Ο πίνακας αυτός έχει μια γραμμή για κάθε μη-τελική κατάσταση του διαγράμματος καταστάσεων και μια στήλη για κάθε ομάδα χαρακτήρων εισόδου. Επιπλέον διατηρούμε τους χαρακτήρες της υπό αναγνώριση συμβολοσειράς σε μια ενδιάμεση μνήμη που λέγεται λέξημα. Αν, για παράδειγμα, αναγνωρισθεί το όνομα, τότε στο λέξημα θα βρίσκονται οι χαρακτήρες του ονόματος και έτσι θα μπορούμε να διακρίνουμε αν πρόκειται για λέξη κλειδί.

Ως αποτέλεσμα της λεξικογραφικής ανάλυσης, έχουμε όλες τις λεκτικές μονάδες του πηγαίου κώδικα του αρχείου καθώς και τις κατηγορίες που αυτές ανήκουν. Από εκεί μπορούν να εξαχθούν διάφορα χαρακτηριστικά που να περιγράφουν σε μεγάλο βαθμό το αρχείο. Αυτά μπορεί να περιλαμβάνουν το μέγεθος του αρχείου (μέγεθος σε γραμμές, μέγεθος σε χαρακτήρες, κ.α.), την φύση των λεκτικών μονάδων του αρχείου (ποσοστό μικρότερων από 10 χαρακτήρες, ποσοστό Unicode συμβολών στις αλφαριθμητικές μεταβλητές κ.α.), τη κατηγορία των λεκτικών μονάδων (αριθμοί, μεταβλητές κ.α.) και πολλά άλλα μεγέθη.



Σχήμα 2.2: Διάγραμμα λειτουργίας λεκτικού αναλυτή

Στο Σχήμα 2.2 απεικονίζεται το διάγραμμα λειτουργίας ενός λεκτικού αναλυτή. Στη γενική περίπτωση, ένας λεκτικός αναλυτής δεν επεξεργάζεται συνδυασμούς λεκτικών μονάδων. Αυτό αποτελεί έργο του συντακτικού αναλυτή.

2.2 Εξαγωγή χαρακτηριστικών

Το πρώτο στάδιο της προεπεξεργασίας των δεδομένων είναι η *εξαγωγή χαρακτηριστικών* (feature extraction) από τα δεδομένα εισόδου. Αυτό συμβαίνει διότι οι περισσότεροι αλγόριθμοι μηχανικής μάθησης δεν μπορούν να πάρουν ως είσοδο ένα απλό κείμενο, αλλά αντ' αυτού περιμένουν ως είσοδο διανύσματα σταθερού μήκους με αριθμητικές τιμές.

Σε αυτό το στάδιο τα δεδομένα εισόδου επεξεργάζονται ώστε να εξαχθούν από αυτά κάποια χαρακτηριστικά. Η επιλογή των χαρακτηριστικών γίνεται ώστε να αντικατοπτρίζουν τις διαφορές των δεδομένων μεταξύ των δυο κλάσεων, καθώς ο τρόπος κωδικοποίησης τους μπορεί να έχει τεράστιο αντίκτυπο στην ικανότητα ενός ταξινομητή να κατασκευάσει ένα αντιπροσωπευτικό μοντέλο. Επιπλέον πρέπει να αποφεύγεται η εξαγωγή περιττών χαρακτηριστικών, μιας και αυτά αυξάνουν σημαντικά τον χρόνο για την εκπαίδευση του μοντέλου και επιπλέον προσθέτουν «θόρυβο», ο οποίος μπορεί να επηρεάσει αρνητικά την ακρίβεια των αποτελεσμάτων του συστήματος. Έτσι ένα σημαντικό κομμάτι στο χτίσιμο ενός ταξινομητή είναι η επιλογή και ο τρόπος αναπαράστασης των χαρακτηριστικών που θα πάρει ως είσοδο. Παρόλο που συχνά επιτυγχάνεται καλή απόδοση με την χρησιμοποίηση απλών χαρακτηριστικών, συνήθως το όφελος από την προσεκτική επιλογή των χαρακτηριστικών βασισμένη σε βαθειά κατανόηση του προβλήματος είναι μεγάλο.

Η αρχική επιλογή γίνεται συνήθως συνδυάζοντας χαρακτηριστικά που υπάρχουν στη βιβλιογραφία και έχουν αποδεχτεί αποτελεσματικά σε προβλήματα αντίστοιχης ή παρόμοιας φύσης, καθώς και κάποια τα οποία αντιλαμβάνεται ως χρήσιμα αυτός που καλείται να λύσει το πρόβλημα, μετά από παρατήρηση και μελέτη των δεδομένων εισόδου. Επίσης, μια συχνή τακτική που χρησιμοποιείται είναι να επιλέγονται αρχικά πάρα πολλά χαρακτηριστικά και στη συνέχεια να διαλέγονται αυτά τα οποία έχουν την μεγαλύτερη επίδραση στο αποτέλεσμα. Και στις δύο περιπτώσεις, τα αρχικά αυτά χαρακτηριστικά αναπροσαρμόζονται, αξιολογώντας τα αποτελέσματα του αλγορίθμου και την επίδραση του κάθε χαρακτηριστικού σε αυτά, ώστε τελικά να επιλεγούν αυτά που περιγράφουν καλύτερα το πρόβλημα.

Ωστόσο υπάρχουν συχνά περιορισμοί στον αριθμό των χαρακτηριστικών που πρέπει να δοθούν σε ένα αλγόριθμο ταξινόμησης. Συγκεκριμένα, αν δοθούν σε αυτόν πάρα πολλά χαρακτηριστικά, τότε ο αλγόριθμος έχει μεγαλύτερη πιθανότητα να εμφανίσει το φαινόμενο της *υπερπροσαρμογής* (overfitting), δηλαδή ο αλγόριθμός να κάνει πολύ καλές προβλέψεις για τα δεδομένα εκπαίδευσης και ελέγχου, αλλά να μην γενικεύει καθόλου καλά σε νέα και άγνωστα δεδομένα [Bram16]. Αυτό

σημαίνει ότι ο αλγόριθμος έχει «μάθει» πολύ καλά τα συγκεκριμένα δεδομένα που του δίνουμε στο στάδιο της εκπαίδευσης και του ελέγχου, αλλά η γενίκευση του δεν είναι η επιθυμητή καθώς είναι πια προσαρμοσμένος να απαντά σωστά μόνο στα δεδομένα που του παρουσιάσαμε. Το φαινόμενο της υπερεκπαίδευσης συναντάται συχνά όταν το σύνολο που χρησιμοποιείται για την εκπαίδευση του μοντέλου περιέχει λίγα δεδομένα.

Εκτός των παραπάνω χαρακτηριστικών που προκύπτουν από την λεξικογραφική ανάλυση του πηγαίου κώδικα των αρχείων, υπάρχουν πολλά άλλα είδη χαρακτηριστικών που μπορούμε να εξάγουμε. Ένα σημαντικό είδος χαρακτηριστικών είναι η συχνότητα εμφάνισης κάποιων λεκτικών μονάδων οι οποίες αποτελούν λέξεις-κλειδιά για την εκάστοτε γλώσσα προγραμματισμού.

Το πρώτο από τα δύο σύνολα χαρακτηριστικών που χρησιμοποιήσαμε προέκυψε πραγματοποιώντας λεξικογραφική ανάλυση των αρχείων. Η εξαγωγή των χαρακτηριστικών αυτών έγκειται σε μεγάλο βαθμό στη κατανόηση και εξέταση γενικότερα των PHP αρχείων καθώς και στην αναγνώριση των διαφορών τους, ανάλογα με την κλάση που ανήκουν (καλοήθη ή μολυσμένα). Στην συνέχεια απαιτείται η κωδικοποίηση των διαφορών αυτών ως χαρακτηριστικά που θα επιτρέψουν στον ταξινομητή να κατατάξει το κάθε αρχείο στη σωστή κατηγορία.

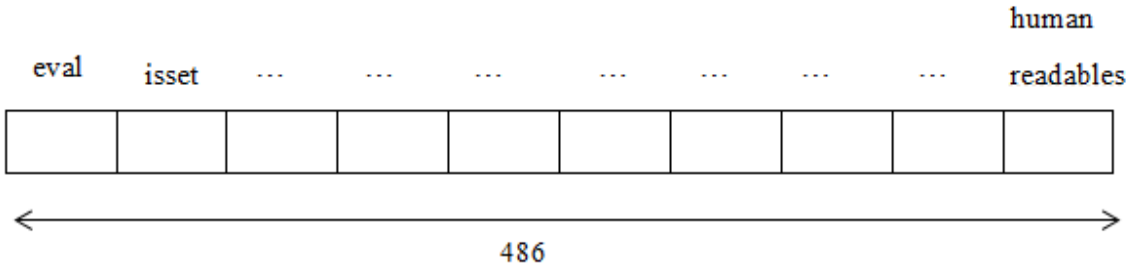
Η λεξικογραφική ανάλυση είναι πολύ σημαντική στην εξαγωγή χαρακτηριστικών σε περιπτώσεις όπως αυτή που εξετάζουμε, όπου τα δεδομένα που έχουμε στη διάθεση μας είναι αρχεία πηγαίου κώδικα, δηλαδή μια μορφή κειμένου. Ωστόσο το γεγονός ότι πρόκειται για μια γλώσσα προγραμματισμού και όχι κάποια φυσική γλώσσα δημιουργεί μεγάλες διαφορές στον τύπο των χαρακτηριστικών που περιγράφει καλύτερα τα δεδομένα μας. Για παράδειγμα, τα n-grams τα οποία χρησιμοποιούνται σε πληθώρα περιπτώσεων στην εξαγωγή χαρακτηριστικών από κείμενα γραμμένα σε φυσική γλώσσα δεν θα έχουν ιδιαίτερο νόημα στην περίπτωση μας, καθώς ο πηγαίος κώδικας περιλαμβάνει την επανάληψη πολλών λέξεων που αποτελούν λέξεις κλειδιά και στερείται κάποιας κατανομής των λεκτικών του μονάδων αντίστοιχης της φυσικής γλώσσας.

Στην περίπτωση μας παρατηρήσαμε ότι η τεχνική της θόλωσης που χρησιμοποιείται στα μολυσμένα αρχεία, χρησιμοποιεί συνήθως περίεργη κωδικοποίηση για τις αλφαριθμητικές και τις αριθμητικές μεταβλητές. Άρα αυτά περιέχουν μεγάλο πλήθος δεκαεξαδικών ή οκταδικών αριθμών. Συνέπεια αυτού είναι και η αύξηση του μεγέθους των μεταβλητών και η μείωση των κενών χαρακτήρων στο αρχείο. Επίσης παρατηρήσαμε μείωση των σχολίων στον κώδικα καθώς και μείωση των λέξεων που είναι αναγνώσιμες από τον άνθρωπο [Lika09]. Με βάση αυτές τις παρατηρήσεις, το πρώτο σύνολο χαρακτηριστικών που εξάγαμε φαίνεται στον Πίνακα 2.1

Για την εξαγωγή των παραπάνω χαρακτηριστικών χρησιμοποιήσαμε τη βιβλιοθήκη της Python, *phply*[1], η οποία αποτελεί έναν *αναλυτή* (parser) για τη γλώσσα προγραμματισμού PHP.

Το δεύτερο σύνολο χαρακτηριστικών που χρησιμοποιήσαμε αφορούσε τη χρησιμοποίηση των προκαθορισμένων συναρτήσεων της γλώσσας από κάθε PHP αρχείο. Παρατηρήσαμε ότι όπως και στο [Lika09], ο αριθμός των εμφανίσεων κάποιων συγκεκριμένων συναρτήσεων της γλώσσας PHP διέφερε αρκετά μεταξύ των αρχείων των δυο κλάσεων. Για παράδειγμα συναρτήσεις όπως οι $eval$, $base64\_decode$, $gzinflate$ κ.α. εμφανίζονται πολύ συχνότερα σε μολυσμένα αρχεία από ότι σε καλοήθη. Έτσι αφού βρήκαμε όλες τις προκαθορισμένες συναρτήσεις που καλούνται στο σύνολο των αρχείων μας, φτιάξαμε για κάθε αρχείο ένα διάνυσμα όπου κάθε στοιχείο του αντιστοιχεί σε μία από αυτές τις συναρτήσεις και η τιμή αντιπροσωπεύει τον αριθμό των εμφανίσεων αυτής της συνάρτησης στο εκάστοτε αρχείο. Επιπλέον στο διάνυσμα αυτό προσθέσαμε και ένα από τα χαρακτηριστικά του πρώτου συνόλου που αναφέραμε, αυτό του «ποσοστού λέξεων αναγνώσιμων από άνθρωπο» (Πίνακας 2.1).

[1] https://github.com/viraptor/phply



Σχήμα 2.3: Παράδειγμα διανύσματος εισόδου για χαρακτηριστικά σχετικά με τον αριθμό εμφανίσεων των συναρτήσεων

Πίνακας 2.1: Πρώτο σύνολο λεξικογραφικών χαρακτηριστικών

Α/Α	**Χαρακτηριστικό**	**Περιγραφή**
1	Μήκος σε χαρακτήρες	Το μήκος ολόκληρου του αρχείου σε χαρακτήρες
2	Μέσος πλήθος χαρακτήρων ανά γραμμή	Ο μέσος αριθμός χαρακτήρων σε κάθε γραμμή
3	Πλήθος γραμμών	Το πλήθος των εμφανίσεων του χαρακτήρα αλλαγής γραμμής στο αρχείο
4	Πλήθος αλφαριθμητικών μεταβλητών (strings)	Το πλήθος των αλφαριθμητικών μεταβλητών του αρχείου
5	Πλήθος συμβόλων Unicode	Το πλήθος των χαρακτήρων σε μορφή Unicode στο αρχείο
6	Πλήθος δεκαεξαδικών ή οκταδικών αριθμών	Το πλήθος των αριθμών που αναπαρίστανται σε δεκαεξαδική ή σε οκταδική μορφή
7	Ποσοστό λέξεων αναγνώσιμων από άνθρωπο [Lika09]	Το ποσοστό των λέξεων οι οποίες είναι αναγνώσιμες από τον άνθρωπο. Μια λέξη θεωρείται ότι ανήκει στην κατηγορία αυτή τηρούνται οι παρακάτω προϋποθέσεις: i. Τουλάχιστον το 70% των χαρακτήρων της είναι αλφαβητικό ii. Το ποσοστό των χαρακτήρων της λέξης οι οποίοι είναι φωνήεντα είναι μεταξύ 20% και 60% iii. Έχει μήκος το πολύ 15 χαρακτήρες. iv. Δεν περιέχει περισσότερους από δύο ίδιους χαρακτήρες στη σειρά.
8	Ποσοστό κενών διαστημάτων (whitespace)	Το ποσοστό του αρχείου το οποίο αποτελείται από κενά διαστήματα
9	Πλήθος των κλήσεων σε μεθόδους ή συναρτήσεις	Ο αριθμός των μεθόδων ή συναρτήσεων που καλούνται από το αρχείο
10	Μέσο μήκος αλφαριθμητικών μεταβλητών	Ο μέσος αριθμός χαρακτήρων ανά αλφαριθμητική μεταβλητή στο αρχείο
11	Μέσο μήκος ορισμάτων	Το μέσο μήκος των ορισμάτων σε μία συνάρτηση ή μέθοδο, σε χαρακτήρες
12	Πλήθος σχολίων	Το πλήθος των σχολίων στο αρχείο
13	Μέσο πλήθος σχολίων ανά γραμμή	Το πλήθος των σχολίων προς τον συνολικό αριθμό των γραμμών του αρχείου
14	Πλήθος λέξεων	Ο αριθμός των λέξεων στο αρχείο όπου οι λέξεις διαχωρίζονται μεταξύ τους από κενά διαστήματα και σύμβολα της γλώσσας PHP (π.χ. αριθμητικοί τελεστές)
15	Ποσοστό λέξεων έξω από σχόλια	Το ποσοστό των λέξεων στο αρχείο οι οποίες δεν είναι σχολιασμένες

Κεφάλαιο 3

Ευφυείς Τεχνικές

Σε αυτό το κεφάλαιο γίνεται η περιγραφή των ευφυών τεχνικών που χρησιμοποιήθηκαν κατά την εκπόνηση των πειραμάτων.

3.1 Δέντρα Αποφάσεων

Τα ΔΑ είναι μια επιβλεπόμενη μέθοδος μάθησης, η οποία χρησιμοποιείται τόσο για ταξινόμηση όσο και για παλινδρόμηση [Mitc97, Chapter 3]. Σκοπός της είναι η δημιουργία ενός μοντέλου το οποίο θα προβλέπει την τιμή μίας μεταβλητής μέσω απλών κανόνων απόφασης, οι οποίοι εξάγονται από τα χαρακτηριστικά των δεδομένων εκπαίδευσης. Στην ταξινόμηση προβλέπεται η κατηγορία που ανήκει η μεταβλητή ενώ στην παλινδρόμηση προβλέπεται η τιμή της μεταβλητής. Στην ταξινόμηση οι πιθανές κατηγορίες μπορεί να είναι είτε δύο, οπότε και έχουμε δυαδικό ταξινομητή, είτε παραπάνω όπου και έχουμε ταξινομητή πολλαπλών κατηγοριών. Οι πιθανές κατηγορίες που μπορεί να ανήκει το στιγμιότυπο συμβολίζονται με έναν αριθμό, ο οποίος ορίζεται στην διατύπωση του προβλήματος και παραμένει σταθερός. Στη συνέχεια θα περιγραφεί η λειτουργία του ΔΑ ως δυαδικού ταξινομητή, καθώς σε αυτή τη μορφή χρησιμοποιήθηκε στην παρούσα εργασία.

3.1.1 Περιγραφή σχηματισμού δέντρου

Υπάρχουν πάρα πολλά δέντρα τα οποία μπορούν να σχηματιστούν από ένα δεδομένο σύνολο από χαρακτηριστικά. Πολλά από αυτά είναι περισσότερο ακριβή από άλλα, ωστόσο η διαδικασία εύρεσης του βέλτιστου είναι υπολογιστικά αδύνατη, λόγω του εκθετικού μεγέθους των πιθανών δέντρων. Ωστόσο, υπάρχουν αρκετοί αλγόριθμοι που σχηματίζουν ένα αρκετά ακριβές δέντρο αποφάσεων σε λογικό χρόνο. Οι αλγόριθμοι αυτοί συνήθως ξεκινούν από τη ρίζα του δέντρου και πάνε προς τα φύλλα και χρησιμοποιούν κάποια μετρική για τον υπολογισμό του χαρακτηριστικού που θα χρησιμοποιηθεί στον διαχωρισμό των δεδομένων. Οι μετρικές αυτές συνήθως μετρούν την ομοιογένεια των κλάσεων-στόχων μέσα στα υποσύνολα και εφαρμόζονται σε κάθε υποψήφιο υποσύνολο [Safa91]. Οι μετρικές που χρησιμοποιούνται συνήθως είναι οι δύο παρακάτω:

Gini impurity: Αποτελεί ένα μέτρο του πόσο συχνά ένα τυχαία επιλεγμένο στοιχείο από το σύνολο θα κατηγοριοποιούνταν λανθασμένα, εάν κατηγοριοποιούνταν τυχαία με βάση την κατανομή των κατηγοριών στο υποσύνολο. Ο υπολογισμός του για ένα σύνολο χαρακτηριστικών δίνεται στην Εξίσωση 3.1:

$$I_G(p) = \sum_{i=1}^{J} p_i(1 - p_i) \tag{3.1}$$

όπου $J \in [0, 1, \ldots, K-1]$ οι τιμές των κλάσεων και p_i το ποσοστό των στιγμιότυπων που έχουν κατηγοριοποιηθεί στην κλάση i μέσα στο σύνολο.

Πληροφοριακή Απολαβή (Information Gain): Αποτελεί την αναμενόμενη ποσότητα πληροφοριών που θα χρειαστεί για να αποφασιστεί σε ποια κατηγορία θα ταξινομηθεί ένα νέο στιγμιότυπο

δεδομένου ότι έχει φτάσει στον κόμβο αυτό. Ο υπολογισμός του βασίζεται στον υπολογισμό της *εντροπίας* (entropy). Η εντροπία προκύπτει από την Εξίσωση 3.2:

$$H(T) = -\sum_{i=1}^{J} p_i \log_2 p_i \tag{3.2}$$

όπου $p_1, p_2, \ldots$ είναι πιθανότητες που αθροίζονται στο 1 και αναπαριστούν την παρουσία κάθε κλάσης στο κόμβο-απόγονο, ο οποίος είναι αποτέλεσμα του διαχωρισμού. Κατόπιν, η πληροφοριακή απολαβή υπολογίζεται από την Εξίσωση 3.3:

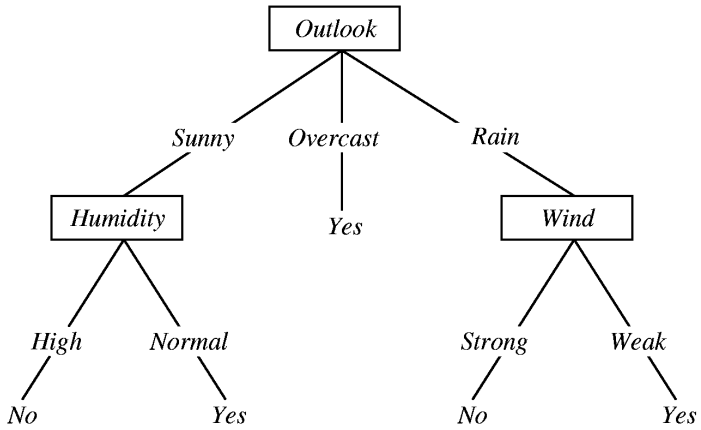
$$IG(T, a) = H(T) - H(T|a) \tag{3.3}$$

η οποία ορίζει την πληροφοριακή απολαβή ως τη διαφορά της σταθμισμένης εντροπίας των κόμβων-απογόνων από την εντροπία του κόμβου-γονέα.

Τέλος εάν δυο δέντρα είναι εξίσου αποτελεσματικά σε ένα σύνολο δεδομένων εκμάθησης, τότε επιλέγεται εκείνο με τα λιγότερα φύλλα.

3.1.2 Περιγραφή λειτουργίας

Το ΔΑ είναι ένα ταξινομητής $h : X \to Y$, ο οποίος προβλέπει την κλάση $y \in Y$ που ανήκει ένα στιγμιότυπο $x \in X$, διασχίζοντας ένα δέντρο [Mitc97, Chapter 3]. Η διάσχιση ξεκινάει από την ρίζα του δέντρου και καταλήγει σε ένα φύλλο αυτού, το οποίο περιέχει μια τιμή η οποία συμβολίζει κάποια από τις πιθανές κλάσεις. Σε κάθε κόμβο που διασχίζεται από το μονοπάτι ρίζα-φύλλο, η απόφαση για το ποίος κόμβος (παιδί του κόμβου που εξετάζεται αυτή τη στιγμή) θα διασχιστεί στη συνέχεια λαμβάνεται εξετάζοντας ένα υποσύνολο (ή ολόκληρο του σύνολο) των τιμών των χαρακτηριστικών του στιγμιότυπου εισόδου x. Έτσι ανάλογα με την τιμή αυτών αποφασίζεται ποιο από τα δύο παιδιά του παρόντος κόμβου θα διασχιστεί στη συνέχεια και η διαδικασία αυτή επαναλαμβάνεται με το ίδιο τρόπο, θεωρώντας ως ρίζα του δέντρου τον κόμβο που επιλέχθηκε. Η διαδικασία αυτή τερματίζεται όταν επιλεγεί κόμβος ο οποίος αποτελεί φύλλο του δέντρου. Τότε ανάλογα με την τιμή του φύλλου αυτού επιλέγεται και η κλάση που θα κατατάξει ο ταξινομητής το στιγμιότυπο εισόδου.



Σχήμα 3.1: Παράδειγμα ενός Δέντρου Απόφασης

Στην Σχήμα 3.1 φαίνεται ένα παράδειγμα ΔΑ για τα δεδομένα του Πίνακα 3.1, τα οποία περιγράφουν την απόφαση ενός ατόμου να παίξει τένις ή όχι, ανάλογα με κάποιες καιρικές συνθήκες.

3.1.3 Προβλήματα

Ένα συχνό πρόβλημα που συναντάται στα ΔΑ, όπως και σε οποιοδήποτε άλλο ταξινομητή, είναι η υπερπροσαρμογή. Ένα ΔΑ λέγεται ότι έχει υπερπροσαρμοστεί στα δεδομένα εκμάθησης , εάν υπάρχει h' τέτοιο, ώστε να δίνει μεγαλύτερο σφάλμα πρόβλεψης, σε σχέση με το h, όταν εφαρμόζεται στα

Πίνακας 3.1: Δεδομένα εκμάθησης για το Δέντρο Αποφάσεων του Σχήματος 3.1

Outlook	Humidity	Wind	Play Tennis
Sunny	High	Strong	No
Sunny	Normal	Weak	Yes
Sunny	Normal	Strong	Yes
Sunny	High	Strong	No
Overcast	High	Weak	Yes
Overcast	Normal	Strong	Yes
Overcast	High	Weak	Yes
Rain	Normal	Weak	Yes
Rain	Normal	Weak	No
Rain	Normal	Strong	No

δεδομένα εκπαίδευσης και μικρότερο όταν εφαρμόζεται στα δεδομένα ελέγχου [Bram16]. Οι βασικές τεχνικές που χρησιμοποιούνται για την αποφυγή της υπερπροσαρμογής στα ΔΑ είναι ο *πρόωρος τερματισμός* (early stopping) της διαδικασίας εκπαίδευσης (με κάποιο κριτήριο πριν το σημείο της υπερπροσαρμογής) και το *κλάδεμα* (pruning) του ΔΑ.

3.1.4 Πλεονεκτήματα

Τα βασικά πλεονεκτήματα των ΔΑ είναι τα παρακάτω [Gare13]:

- Είναι εύκολο να κατανοηθούν και να ερμηνευτούν από τον άνθρωπο καθώς είναι πιο κοντά στον ανθρώπινο τρόπο νόησης.
- Χρειάζεται ελάχιστη προεπεξεργασία των δεδομένων, καθώς τα ΔΑ είναι σε θέση να διαχειριστούν διαφορές κλίμακας και δεν επηρεάζονται από τιμές υπερβολικά υψηλές, υπερβολικά χαμηλές και κενές.
- Έχει καλή απόδοση για μεγάλα σύνολα δεδομένων.
- Το αποτέλεσμα της πρόβλεψης είναι εύκολο να ερμηνευτούν λόγω της μορφής του μοντέλου (δέντρο).

3.2 Μηχανές Διανυσμάτων Υποστήριξης

Στη μηχανική μάθηση, οι ΜΔΥ είναι επιβλεπόμενα μοντέλα μάθησης που χρησιμοποιούν αλγορίθμους μάθησης, οι οποίοι αναλύουν τα δεδομένα που χρησιμοποιούνται για την ταξινόμηση. Λαμβάνοντας ένα σύνολο δεδομένων εκπαίδευσης, το καθένα από τα οποία ανήκει σε μια από τις δύο κατηγορίες, ένας αλγόριθμος εκπαίδευσης ΜΔΥ δημιουργεί ένα μοντέλο που κατατάσσει τα νέα παραδείγματα στην μία ή την άλλη κατηγορία, καθιστώντας τον έναν μη-πιθανοτικό δυαδικό γραμμικό ταξινομητή [Hear98].

3.2.1 Περιγραφή τρόπου λειτουργίας

Μια ΜΔΥ κατασκευάζει ένα υπερεπίπεδο ή σύνολο υπερεπιπέδων σε έναν χώρο πολλών ή άπειρων διαστάσεων, που μπορεί να χρησιμοποιηθεί για ταξινόμηση, παλινδρόμηση ή άλλες εργασίες [Hear98]. Διαισθητικά, ένας καλός διαχωρισμός επιτυγχάνεται από το υπερεπίπεδο που έχει τη μεγαλύτερη απόσταση από το πλησιέστερο σημείο των δεδομένων εκπαίδευσης οποιασδήποτε κλάσης (λεγόμενο *λειτουργικό περιθώριο*), αφού γενικά όσο μεγαλύτερο είναι το περιθώριο τόσο χαμηλότερο είναι το σφάλμα γενίκευσης του ταξινομητή.

Ένα μοντέλο SVM είναι αναπαράσταση των δεδομένων εισόδου ως σημεία στο χώρο, χαρτογραφημένα έτσι ώστε να χωρίζονται τα παραδείγματα των ξεχωριστών κατηγοριών από ένα σαφές χάσμα που είναι όσο το δυνατόν ευρύτερο. Με τον τρόπο αυτό τα νέα παραδείγματα κατατάσσονται σε μια από τις πιθανές κατηγορίες με βάση το σε ποια πλευρά του χάσματος ανήκει η αναπαράσταση τους στο χώρο.

3.2.2 Γραμμική κατηγοριοποίηση (Hard margin)

Έστω ότι έχουμε ένα σύνολο n παραδειγμάτων εκπαίδευσης της μορφής:

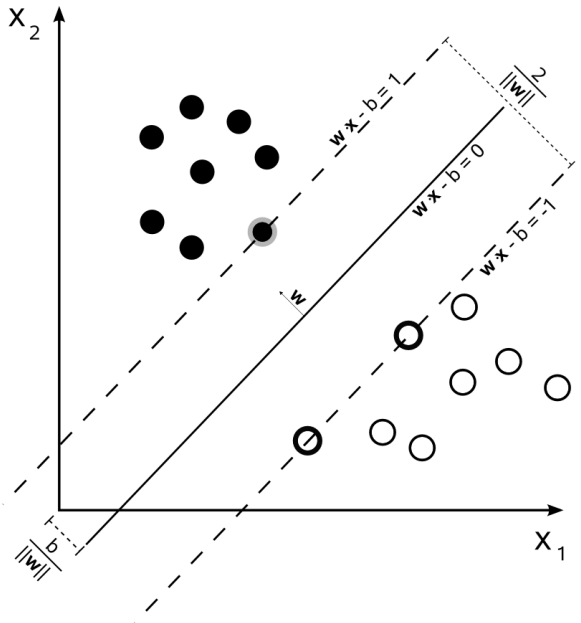
$$(\overrightarrow{x_1}, y_1), (\overrightarrow{x_2}, y_2), \ldots, (\overrightarrow{x_n}, y_n)$$

όπου το y_i έχει την τιμή είτε 1 είτε -1, ανάλογα με την κλάση στην οποία κατατάσσεται το διάνυσμα εισόδου $\overrightarrow{x_i}$. Σκοπός μας είναι να εντοπίσουμε το υπερεπίπεδο με το μεγαλύτερο περιθώριο, το οποίο χωρίζει τα $\overrightarrow{x_i}$ για τα οποία το y_i έχει την τιμή 1, από αυτά για τα οποία το y_i έχει την τιμή -1, δηλαδή η απόσταση μεταξύ του υπερεπιπέδου και του κοντινότερου σημείου $\overrightarrow{x_i}$ από κάθε μία κλάση να είναι η μέγιστη.

Ένα υπερεπίπεδο μπορεί να διατυπωθεί ως ένα σύνολο σημείων $\overrightarrow{x}$, το οποίο ικανοποιεί την Εξίσωση 3.4,

$$\overrightarrow{w}.\overrightarrow{x} - b = 0 \tag{3.4}$$

όπου $\overrightarrow{w}$ είναι το κανονικό διάνυσμα ως προς το υπερεπίπεδο. Η παράμετρος $\frac{b}{||\overrightarrow{w}||}$ καθορίζει την απόσταση του υπερεπιπέδου από το πλησιέστερο σημείο κατά τη διεύθυνση του κανονικού διανύσματος.



Σχήμα 3.2: Παράδειγμα γραμμικής κατηγοριοποίησης Μηχανών Διανυσμάτων Υποστήριξης

Εάν τα δεδομένα εκπαίδευσης είναι γραμμικά διαχωρίσιμα (Σχήμα 3.2), τότε μπορούμε να επιλέξουμε δύο παράλληλα υπερεπίπεδα τα οποία διαχωρίζουν τις κλάσεις των δεδομένων, έτσι ώστε η απόσταση μεταξύ τους να είναι η μέγιστη δυνατή. Η περιοχή μεταξύ των υπερεπιπέδων ονομάζεται *περιθώριο* (margin) και το υπερεπίπεδο με το μέγιστο περιθώριο είναι το υπερεπίπεδο το οποίο βρίσκεται στο μέσο αυτών των δύο. Με τα κατάλληλα δεδομένα, η κατασκευή των δύο αυτών υπερεπιπέδων περιγράφεται από τις Εξισώσεις 3.5-3.6:

$$\overrightarrow{w}.\overrightarrow{x} - b = \quad 1 \tag{3.5}$$
$$\overrightarrow{w}.\overrightarrow{x} - b = -1 \tag{3.6}$$

Γεωμετρικά, η απόσταση μεταξύ των δύο αυτών υπερεπιπέδων είναι $\frac{2}{||\overrightarrow{w}||}$, οπότε για να μεγιστοποιήσουμε την μεταξύ τους απόσταση πρέπει να ελαχιστοποιήσουμε την τιμή του $\overrightarrow{w}$. Επίσης για να αποφύγουμε τον κίνδυνο κάποια σημεία να ανήκουν στο περιθώριο μεταξύ των υπερεπιπέδων θέτουμε τους περιορισμούς των Ανισώσεων 3.7-3.8, οι οποίοι εξασφαλίζουν ότι κάθε δεδομένο εκπαίδευσης θα βρίσκεται στη σωστή πλευρά του περιθωρίου.

$$\overrightarrow{w}.\overrightarrow{x_i} - b \geq 1, \quad \text{αν } y_i = 1 \tag{3.7}$$

$$\overrightarrow{w}.\overrightarrow{x_i} - b \leq 1, \quad \text{αν } y_i = -1 \tag{3.8}$$

Οι παραπάνω Ανισώσεις μπορούν να γραφούν υπό τη μορφή μιας ως εξής:

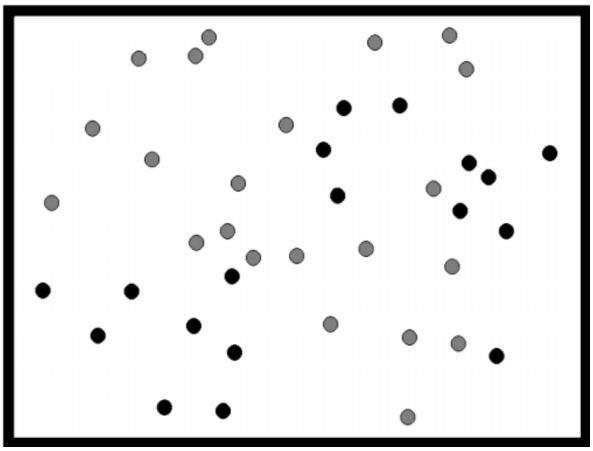
$$y_i(\overrightarrow{w}.\overrightarrow{x_i} - b) \geq 1, \forall i \in [1, n] \tag{3.9}$$

Έτσι το τελικό πρόβλημα βελτιστοποίησης που προκύπτει είναι να βρεθεί η ελάχιστη τιμή του $||\overrightarrow{w}||$, για το οποίο ισχύει η Ανίσωση 3.9.

Ένα πολύ σημαντικό συμπέρασμα που προκύπτει από την γεωμετρική περιγραφή είναι ότι το υπερεπίπεδο με το μέγιστο περιθώριο καθορίζεται ολοκληρωτικά από αυτά τα x_i, τα οποία βρίσκονται πλησιέστερα σε αυτό. Αυτά καλούνται διανύσματα υποστήριξης.

3.2.3 Μη γραμμική κατηγοριοποίηση (Soft margin)

Προηγουμένως υποθέσαμε ότι τα δεδομένα μας ήταν γραμμικά διαχωρίσιμα και με τη βοήθεια των ΜΔΥ περιγράψαμε πως μπορούμε να πετύχουμε το βέλτιστο δυνατό αποτέλεσμα. Ένα φαινόμενο όμως που συναντάται συχνά είναι τα δεδομένα που έχουμε να μην είναι γραμμικά διαχωρίσιμα και οι δύο κλάσεις να εμφανίζονται ανακατεμένες, όπως στο παράδειγμα του Σχήματος 3.3



Σχήμα 3.3: Μη γραμμικά διαχωρίσιμα δεδομένα για Μηχανές Διανυσμάτων Υποστήριξης

Για να ξεπεράσουμε το πρόβλημα αυτό εισάγουμε μια νέα μεταβλητή στο πρόβλημα βελτιστοποίησης που διατυπώσαμε στο προηγούμενο κεφάλαιο. Με την κατάλληλη επιλογή της μεταβλητής αυτής ξεπερνάμε σε μεγάλο βαθμό το πρόβλημα της μη-γραμμικής διαχωρισιμότητας. Έτσι, εισάγοντας τις *μεταβλητές χαλαρότητας* (slack variables) $\xi_i \geq 0$ στην Ανίσωση 3.9, προκύπτει η Ανίσωση 3.10

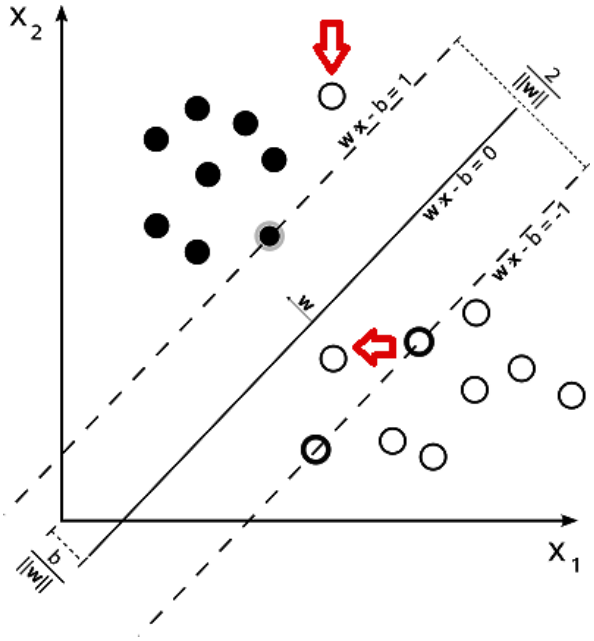
$$y_i(\overrightarrow{w}.\overrightarrow{x_i} - b) \geq 1 - \xi_i, \forall i \in [1, n] \tag{3.10}$$

Συνεπώς αν ισχύει $\xi_i > 1$, τότε το δεδομένο ταξινομείται λάθος, δηλαδή στην άλλη κλάση. Οπότε το σύνολο μεταβλητών χαλαρότητας είναι το άνω όριο του πλήθους των δεδομένων που μπορούν να ταξινομηθούν λάθος, δηλαδή να είναι όλα λάθος με $\xi_i > 1$. Άρα το πλήθος των λάθος ταξινομημένων στιγμιότυπων είναι $\sum_{i=1}^{n} \xi_i$ και πρέπει στο μέγεθος $||\overrightarrow{w}||$ που προσπαθούμε να ελαχιστοποιήσουμε πριν,

να προσθέσουμε το ανάλογο του αθροίσματος αυτού. Το ελάχιστο υπολογίζεται σύμφωνα με τους περιορισμούς των μεταβλητών χαλαρότητας ξ_i (Εξίσωση 3.11)

$$||\overrightarrow{w}|| + C\sum_{i=1}^{n}\xi_i \tag{3.11}$$

Την τιμή της σταθεράς C την καθορίζουμε εμείς και ουσιαστικά είναι το βάρους του κόστους των λανθασμένων ταξινομήσεων. Αν η σταθερά C τείνει στο 0, τότε δεν δίνουμε βαρύτητα στις μεταβλητές χαλαρότητας, ενώ όταν η τιμή της είναι μεγάλη τότε εστιάζουμε στη σωστή ταξινόμηση των δεδομένων αποφεύγοντας τις αστοχίες (Σχήμα 3.4).



Σχήμα 3.4: Μη-γραμμική κατηγοριοποίηση από Μηχανές Διανυσμάτων Υποστήριξης

3.2.4 Συναρτήσεις πυρήνα

Μέχρι στιγμής, με την χρήση των ΜΔΥ έχουμε αντιμετωπίσει προβλήματα στα οποία μπορούν να βρεθούν με τα διανύσματα υποστήριξης διαχωριστικές γραμμές (ή υπερεπίπεδα) με σκοπό το βέλτιστο διαχωρισμό δύο κλάσεων όταν αυτές μπορούν να διαχωριστούν γραμμικά (hard margin) ή ακόμη και όταν δεν είναι αυτό εφικτό (soft margin). Στην τελευταία όμως περίπτωση, όπου ανάλογα με την τιμή που δίνει ο χρήστης στο C, ελέγχει και τη σωστή ταξινόμηση των δεδομένων, συχνά υπάρχει αποτυχία διαχωρισμού αρκετών στιγμιότυπων.

Ο βέλτιστος διαχωρισμός μπορεί να επιτευχθεί σε μη γραμμικά υπερεπίπεδα χρησιμοποιώντας μία μη γραμμική διανυσματική συνάρτηση μετασχηματισμού (Εξίσωση 3.12)

$$\phi(x) = (\phi(x_1), \phi(x_2), \ldots, \phi(x_n))^\top \tag{3.12}$$

Υποθέτουμε ότι τα μετασχηματισμένα πλέον στιγμιότυπα $\phi(x_1), \phi(x_2), \ldots, \phi(x_n)$ μπορούν να διαχωριστούν γραμμικά. Έτσι καταλήγουμε πάλι στην προηγούμενη ενότητα των γραμμικά διαχωρίσιμων δεδομένων, μόνο που το κάθε στιγμιότυπο αντί για $\overrightarrow{x_i}$ θα είναι πλέον $\phi(\overrightarrow{x_i})$. Συνεπώς, σε αυτή την περίπτωση, το βέλτιστο διαχωριστικό υπερεπίπεδο περιγράφεται από την Εξίσωση 3.13

$$g(\overrightarrow{x}) = \overrightarrow{w}\phi(x) - w_0 = \sum_{i=1}^{n}\lambda_i.d_i.\phi(\overrightarrow{x_i}).\phi(\overrightarrow{x}) - w_0 \tag{3.13}$$

Η συνάρτηση $k(x,y) = \sum_i \phi(x_i).\phi(y_i)$ ονομάζεται *συνάρτηση πυρήνα* (kernel function) [Bose92]. Η κύρια ιδέα είναι ότι τα δεδομένα μας στο πραγματικό κόσμο δεν είναι σχεδόν ποτέ γραμμικώς διαχωρίσιμα, οπότε τα προβάλουμε μέσω μιας συνάρτησης πυρήνα ϕ σε ένα χώρο πιο μεγάλης διάστασης, όπου ίσως να πετύχουμε ένα γραμμικό διαχωρισμό με λιγότερα σφάλματα.

Θεώρημα Mercer

Υπάρχουν πολλές συναρτήσεις μετασχηματισμού μεγάλων ή απείρων διαστάσεων οι οποίες μπορούν να εφαρμοστούν σε συναρτήσεις πυρήνα. Σύμφωνα με το θεώρημα του Mercer [Bose92], μια συνάρτηση k που μας δίνει το εσωτερικό γινόμενο μιας απεικόνισης ϕ, δηλαδή $k(x,y) = \sum_i \phi(x_i).\phi(y_i)$ καλείται συνάρτηση πυρήνα αν και μόνο αν για κάθε συνάρτηση $g(x)$ τέτοια ώστε $\int g(x)^2 dx \in \mathcal{R}$ ισχύει (Ανίσωση 3.14)

$$\int k(x,y)g(x)g(y)dxdy \geq 0 \tag{3.14}$$

Οι συναρτήσεις πυρήνα που χρησιμοποιούνται πιο συχνά είναι οι εξής [Bose92]:

- Πυρήνας RBF: $k(x,y) = \exp(-\gamma||x-y||^2)$, με $\gamma = \frac{1}{2\sigma^2}$
- Σιγμοειδής πυρήνας: $k(x,y) = \tanh(\kappa.x.y + c)$, για $c < 0$ και ορισμένα $\kappa > 0$
- Πολυωνυμικός πυρήνας: $k(x,y) = (x.y+1)^d$
- Γραμμικός πυρήνας: $k(x,y) = x.y$

Όσον αφορά τις διάφορες παραμέτρους των συναρτήσεων πυρήνα, αυτές καθορίζονται από τον χρήστη. Αυτό γίνεται συνήθως πειραματικά, όπου ο χρηστής μετά από δοκιμές επιλέγει τον συνδυασμό παραμέτρων εκείνο που του δίνει το βέλτιστο αποτέλεσμα ταξινόμησης.

3.2.5 Πλεονεκτήματα

Τα πλεονεκτήματα αυτής της μεθόδου μηχανικής μάθησης είναι τα εξής:

1. Είναι αποδοτική σε χώρους υψηλών διαστάσεων, ακόμη και σε περιπτώσεις που ο αριθμός των διαστάσεων είναι μεγαλύτερος του αριθμού των παραδειγμάτων εκπαίδευσης.
2. Χρησιμοποιεί ένα υποσύνολο των σημείων εκπαίδευσης για την τελική *συνάρτηση απόφασης* (decision function), οπότε είναι αποδοτική και όσον αφορά τη μνήμη.
3. Είναι αρκετά ευέλικτη αφού, ανάλογα με το πρόβλημα, μπορούν να οριστούν εξειδικευμένες συναρτήσεις πυρήνα.

3.2.6 Μειονεκτήματα

Τα μειονεκτήματα αυτής της μεθόδου μηχανικής μάθησης είναι τα εξής:

1. Εάν ο αριθμός των διαστάσεων είναι σημαντικά μεγαλύτερος από αυτόν των παραδειγμάτων εκπαίδευση,ς υπάρχει ο κίνδυνος της υπερπροσαρμογής
2. Απαιτεί σημαντικό χρόνο για μεγάλο αριθμό δεδομένων.

3.3 Στοχαστική Κατάβαση Κλίσης

Η ΣΚΚ είναι μια στοχαστική προσέγγιση της βελτιστοποίησης κατάβασης κλίσης και μια επαναληπτική μέθοδος για την ελαχιστοποίηση ή την μεγιστοποίηση μιας αντικειμενικής συνάρτησης, η οποία εκφράζεται ως άθροισμα διαφορίσιμων συναρτήσεων [Mitc97, Section 4.4.3.3]. Άρα βασική της λειτουργία είναι η εύρεση της ελάχιστης ή της μέγιστης τιμής μέσω επαναλήψεων.

3.3.1 Περιγραφή λειτουργίας

Η μέθοδος της ΣΚΚ προσπαθεί να ελαχιστοποιήσει μία αντικειμενική συνάρτηση η οποία έχει την μορφή αθροισμάτων (Εξίσωση 3.15)

$$Q(w) = \frac{1}{n}\sum_{i=1}^{n} Q_i(w) \tag{3.15}$$

Το ζητούμενο είναι να βρεθεί η παράμετρος w η οποία ελαχιστοποιεί τη συνάρτηση $Q(w)$. Κάθε στοιχείο Q_i του αθροίσματος σχετίζεται με την i-οστή παρατήρηση του συνόλου των παραδειγμάτων εκπαίδευσης.

Η τιμή του $Q(w)$ για ένα παράδειγμα εκπαίδευσης υπολογίζεται από τον επαναληπτικό τύπο $w := w - \eta\nabla Q_i(w)$, όπου η είναι ο *ρυθμός μάθησης* (learning rate), ο οποίος ελέγχει το μέγεθος κάθε βήματος (τον ρυθμό δηλαδή που αλλάζει το w) [Rude16].

Όσο ο αλγόριθμος διασχίζει τα παραδείγματα εκπαίδευσης, ενημερώνει την τιμή του w με τον παραπάνω τρόπο για καθένα από τα παραδείγματα. Είναι πιθανόν τα παραδείγματα αυτά να προσπελαστούν αρκετές φορές ώσπου ο αλγόριθμος να συγκλίνει, οπότε και τερματίζει. Εάν αυτό συμβεί, τότε σε κάθε προσπέλαση τα παραδείγματα ανακατεύονται (άρα αλλάζει η σειρά εμφάνισης τους) έτσι ώστε να αποφευχθούν οι κύκλοι. Επίσης πολλές φορές προσαρμόζεται και ο ρυθμός μάθησης η, έτσι ώστε ο αλγόριθμος να συγκλίνει ταχύτερα.

Η λειτουργία του αλγορίθμου ΣΚΚ περιγράφεται υπό τη μορφή ψευδοκώδικα στον Αλγόριθμο 1

Αλγόριθμος 1 Λειτουργία αλγορίθμου Στοχαστικής Κατάβασης Κλίσης

1: Επέλεξε μια αρχική τιμή για το διάνυσμα w και τον ρυθμό μάθησης η
2: **Επανέλαβε** έως ότου επιτευχθεί ένα προσεγγιστικό ελάχιστο
3: Ανακάτεψε τυχαία τα παραδείγματα εκπαίδευσης
4: **Για** $i = 1, 2, \ldots n$
5: $w \leftarrow w - \eta\nabla Q_i(w)$
6: **Τέλος Για**
7: **Τέλος Επανέλαβε**

Όσον αφορά την σύγκλιση του αλγορίθμου, όταν ο ρυθμός μάθησης η μειώνεται με τον κατάλληλο βήμα, ο αλγόριθμος ΣΚΚ συγκλίνει σχεδόν σίγουρα σε ολικό ελάχιστο όταν η αντικειμενική συνάρτηση είναι κυρτή ή ψευδοκυρτή. Σε αντίθετη περίπτωση συγκλίνει σχεδόν σίγουρα σε ένα τοπικό ελάχιστο[Bott98, Kiwi01].

Οι βασικές συναρτήσεις κόστους που χρησιμοποιούνται είναι οι εξής [Shal11]:

- Hinge: $l(w;(x,y)) = \max(0, 1 - y(w,x))$
- Log: $l(w;(x,y)) = \log(1 + \exp(-y(w,x)))$

3.3.2 Πλεονεκτήματα

Τα πλεονεκτήματα του αλγορίθμου ΣΚΚ είναι τα εξής:

- Αποδοτικότητα
- Ευκολία υλοποίησης και δυνατότητα ευελιξίας με την έννοια ότι υπάρχουν πολλοί τρόποι *ρύθμισης* (tuning) της μεθόδου

3.3.3 Μειονεκτήματα

Τα μειονεκτήματα του αλγορίθμου ΣΚΚ είναι τα ακόλουθα:

- Ο αλγόριθμος απαιτεί τον καθορισμό πολλών παραμέτρων, όπως η παράμετρος κανονικοποίησης και ο αριθμός των επαναλήψεων
- Ο αλγόριθμος είναι ευαίσθητος στην αύξηση του αριθμού των χαρακτηριστικών που χρησιμοποιούνται στα παραδείγματα εκπαίδευσης

Κεφάλαιο 4

Πειραματική διαδικασία και αποτελέσματα

4.1 Περιγραφή των δεδομένων

Για την λειτουργία του συστήματος που υλοποιήσαμε, χρειαζόμασταν κάποια PHP αρχεία ως είσοδο για τους αλγορίθμους μηχανικής μάθησης. Το σύνολο των αρχείων αυτών θα έπρεπε να περιέχει τόσο μολυσμένα αρχεία όσο και μη μολυσμένα. Όπως έχει αναφερθεί και στο πρώτο κεφάλαιο της εργασίας, η αναλογία μολυσμένων και μη μολυσμένων αρχείων έπρεπε να είναι 1 προς 10, καθώς αυτή περίπου είναι η αναλογία των αρχείων που ο αλγόριθμος θα εξετάσει αν χρησιμοποιηθεί σε πραγματικές συνθήκες [Mosk09].

4.1.1 Μολυσμένα αρχεία

Ως παραδείγματα μολυσμένων αρχείων χρησιμοποιήσαμε 171 αρχεία συνολικού μεγέθους 1,05 MB, που αντλήσαμε από διάφορες πηγές στο Διαδίκτυο. Στο σύνολο αυτό περιλαμβάνονται διαφορετικοί τύποι μολυσμένων αρχείων όπως artificial, classic, obfuscators, real και undetected[1].

4.1.2 Μη μολυσμένα αρχεία

Ως παραδείγματα μη-μολυσμένου κώδικα PHP χρησιμοποιήσαμε τα αρχεία που αποτελούν το λογισμικό ανοιχτού κώδικα *WordPress*[2]. Το WordPress χρησιμοποιείται για την δημιουργία ιστοτόπων, blog ή εφαρμογών και περιλαμβάνει πληθώρα αρχείων PHP. Η μεγάλη απήχησή του[3], ο μεγάλος αριθμός καθώς και το μεγάλο μέγεθος των αρχείων, το καθιστά ως σημαντικό παραδείγματα εκπαίδευσης για μια εφαρμογή μηχανικής μάθησης. Συγκεκριμένα, η έκδοση 4.8 αποτελείται από 704 αρχεία μεγέθους 10,8 MB συνολικά.

4.2 Υλοποίηση

4.2.1 Διαδικασία υλοποίησης

Η διαδικασία υλοποίησης για τον εκάστοτε αλγόριθμο μηχανικής μάθησης περιγράφεται στα παρακάτω βήματα:

1. Αρχικά διαβάζονται τα διανύσματα εισόδου που αποτελούν τα χαρακτηριστικά που προέκυψαν από την προεπεξεργασία των δεδομένων εισόδου, η οποία για εξοικονόμηση χρόνου έχει πραγματοποιηθεί νωρίτερα και τα αποτελέσματα της έχουν αποθηκευτεί.

2. Έπειτα πραγματοποιείται μια τυχαιοποιημένη αναζήτηση για την εύρεση των βέλτιστων παραμέτρων του αλγορίθμου που εξετάζουμε κάθε φορά. Για να την αναζήτηση αυτή, καθορίζεται το διάστημα αναζήτησης[4] και βρίσκονται οι παράμετροι που έχουν το καλύτερο αποτέλεσμα. Από αυτές, για εξοικονόμηση χρόνου, αποθηκεύονται οι 5 καλύτερες.

[1] https://github.com/nbs-system/php-malware-finder/tree/master/php-malware-finder/samples
[2] https://wordpress.org/
[3] https://w3techs.com/technologies/history_overview/content_management
[4] μέσω της μεθόδου RandomizedSearchCV της βιβλιοθήκης scikit-learn

3. Στη συνέχεια για κάθε ένα από αυτούς τους συνδυασμούς και με την μέθοδο της *διασταυρούμενης επικύρωσης k-διπλωμάτων* (k-fold cross validation) με $k = 5$, η οποία περιγράφεται στον Αλγόριθμο 2, αξιολογούμε το αποτέλεσμα κάθε συνόλου παραμέτρων.

Αξίζει να σημειωθεί πως τα σύνολα διαχωρισμού που χρησιμοποιούνται στην διασταυρούμενη επικύρωση k-διπλωμάτων έχουν καθοριστεί από την αρχή και συγκεκριμένα ακριβώς μετά την ανάγνωση των διανυσμάτων εισόδου, έτσι ώστε να χρησιμοποιηθούν οι ίδιοι διαχωρισμοί στην αξιολόγηση όλων των αλγορίθμων μηχανικής μάθησης. Αυτό γίνεται για να μην επηρεαστούν τα αποτελέσματα του αλγορίθμου από τυχόν ευνοϊκό η δυσχερή διαχωρισμό των δεδομένων εισόδου σε κάποια περίπτωση.

Τέλος στις τιμές των διανυσμάτων εισόδου για τις περιπτώσεις των αλγορίθμων ΜΔΥ και ΣΚΚ πραγματοποιείται κανονικοποίηση, η οποία βελτιώνει τόσο την ακρίβεια των αποτελεσμάτων του αλγορίθμου, όσο και την ταχύτητα του [Meye01]. Αντιθέτως στον αλγόριθμο ΔΑ κάτι τέτοιο δεν είναι απαραίτητο καθώς δεν έχει κάποιο αποτέλεσμα στην λειτουργία του αλγορίθμου.

4.2.2 Προγραμματιστικά εργαλεία

Για την υλοποίηση του συνολικού συστήματος χρησιμοποιήθηκε το περιβάλλον της γλώσσας προγραμματισμού Python 2.7[5]. Πιο συγκεκριμένα, για τους αλγορίθμους μηχανικής μάθησης χρησιμοποιήσαμε τη βιβλιοθήκη *scikit-learn*[6], η οποία παρέχει την δυνατότητα εύκολης υλοποίησης τους ενώ παράλληλα δίνει τη δυνατότητα πειραματισμού με τις διάφορες παραμέτρους κάθε αλγορίθμου. Επίσης χρησιμοποιήσαμε τις βιβλιοθήκες *pandas*[7] και *numpy*[8], οι οποίες βοηθούν στις επεξεργασία των δεδομένων και στις διάφορες αριθμητικές πράξεις. Τέλος, όπως αναφέρθηκε στην Ενότητα 2.2, για την λεξικογραφική ανάλυση των αρχείων PHP χρησιμοποιήσαμε τη βιβλιοθήκη *phply*.

4.2.3 Ανισορροπία κλάσεων

Σε πολλά προβλήματα μηχανικής μάθησης τυχαίνει τα δεδομένα εισόδου να μην είναι ισορροπημένα. Αυτό σημαίνει ότι η μια κλάση (κλάση μειοψηφίας) περιέχει σημαντικά λιγότερα δεδομένα από αυτά της άλλης κλάσης (κλάση πλειοψηφίας). Τότε λέμε ότι υπάρχει *ανισορροπία κλάσεων* (class imbalance) στα δεδομένα του προβλήματος. Αυτό έχει ως αποτέλεσμα ο ταξινομητής να ταξινομεί με αρκετά μεγαλύτερη ακρίβεια τα δεδομένα που ανήκουν στην κλάση πλειοψηφίας, ενώ αποτυγχάνει να ταξινομήσει με επαρκή ακρίβεια τα δεδομένα της άλλης κλάσης [Gang12]. Αιτία αυτού είναι ο αριθμός των παραδειγμάτων που έχει σε κάθε περίπτωση ο αλγόριθμος για να «μάθει». Επίσης η ανισορροπία των κλάσεων των δεδομένων μας κάνει να δίνουμε μεγαλύτερη βαρύτητα στις μετρικές αξιολόγησης που αφορούν την κλάση μειοψηφίας [Elka01], όπως θα γίνει αντιληπτό και από τις μετρικές που χρησιμοποιήσαμε για την αξιολόγηση των αποτελεσμάτων των αλγορίθμων που εξετάσαμε. Αυτές παρουσιάζονται διεξοδικότερα στην Ενότητα 4.3.

Αυτό το πρόβλημα παρατηρείται και στην δική μας περίπτωση, όπου η αναλογία μολυσμένων και μη μολυσμένων αρχείων είναι περίπου 1 προς 10 (Ενότητα 4.1). Λόγω της σημαντικότητας της ανίχνευσης των μολυσμένων αρχείων, καθίσταται επιτακτική η αντιμετώπιση της ανισορροπίας των κλάσεων. Οι σημαντικότερες μέθοδοι που έχουν προταθεί είναι οι εξής:

Δειγματοληψία δεδομένων (Data sampling): Ο αριθμός των παραδειγμάτων εκπαίδευσης των δύο κλάσεων μεταβάλλεται, έτσι ώστε στα παραδείγματα οι δύο κλάσεις να παρουσιάζουν μια σχετική ισορροπία. Αυτό γίνεται είτε με δειγματοληψία, όπου κάποια από τα παραδείγματα εκπαίδευσης που ανήκουν στην κλάση μειοψηφίας αφαιρούνται από το σύνολο, είτε με υπερδειγματοληψία, όπου δημιουργούνται νέα χαρακτηριστικά παραδείγματα της κλάσης μειοψηφίας, είτε τέλος με κάποια υβριδική μέθοδο, η οποία συνδυάζει τους δυο προαναφερόμενους τρόπους [Japk00].

[5] https://www.python.org/download/releases/2.7/
[6] http://scikit-learn.org/stable/index.html
[7] http://pandas.pydata.org/
[8] http://www.numpy.org/

Αλγοριθμικές προσαρμογές (Algorithmic modifications): Είτε επιλέγονται αλγόριθμοι μηχανικής μάθησης που είναι αποτελεσματικοί σε περιπτώσεις ανισορροπίας κλάσεων είτε οι μέθοδοι εκπαίδευσης προσαρμόζονται έτσι ώστε να αντιμετωπίζουν το πρόβλημα αυτό [Zadr01].

Εκπαίδευση με ευαισθησία κόστους (Cost-sensitive learning): Κατά την εκπαίδευση του μοντέλου, η ποινή που επιβάλλεται σε περίπτωση λάθος ταξινόμησης παραδείγματος της κλάσης μειοψηφίας είναι σημαντικά μεγαλύτερη από την αντίστοιχη που αποδίδεται σε παράδειγμα της κλάσης πλειοψηφίας. Με τον τρόπο αυτό αντισταθμίζεται το μικρό πλήθος των παραδειγμάτων εκπαίδευσης με την προσπάθεια αποφυγής λάθους που θα επιφέρει μεγάλη ποινή [Zadr03].

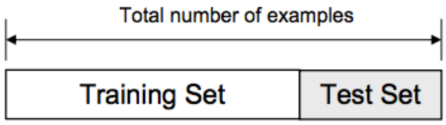
Η μέθοδος που επιλέξαμε εμείς είναι η εκπαίδευση με ευαισθησία κόστους, ορίζοντας κάποιες ποινές για λάθος ταξινόμηση της κλάσης μειοψηφίας, μέσω των μεθόδων της βιβλιοθήκης *scikit-learn*.

4.2.4 Μέθοδοι επικύρωσης

Στα προβλήματα μηχανικής μάθησης τα δεδομένα εισόδου χωρίζονται στις δύο παρακάτω κατηγορίες (Σχήμα 4.1):

Δεδομένα εκμάθησης (training set): για την εκπαίδευση του μοντέλου

Δεδομένα ελέγχου (test set): για την αξιολόγηση του εκπαιδευμένου μοντέλου.



Σχήμα 4.1: Παράδειγμα διαχωρισμού σε δεδομένα εκπαίδευσης και ελέγχου

Αυτός ο διαχωρισμός όμως δημιουργεί προβλήματα, όπως η μη αντιπροσωπευτική τιμή του αποτελέσματος της αξιολόγησης λόγω της ατυχούς επιλογής των δεδομένων. Επίσης πολλές φορές τα δεδομένα που απαιτούνται για την εκπαίδευση και την αξιολόγηση του μοντέλου μηχανικής μάθησης είναι δύσκολο να βρεθούν. Αυτό συμβαίνει και στην δική μας περίπτωση μας, όπου τα μολυσμένα αρχεία PHP είναι δυσεύρετα. Έτσι μοιάζει με πολυτέλεια η μη-χρήση όλων των δεδομένων για την εκπαίδευση του μοντέλου. Για να ξεπεραστούν τα προβλήματα αυτά υπάρχουν διάφορες μέθοδοι. Αυτή που επιλέξαμε να χρησιμοποιήσουμε εμείς είναι η μέθοδος της διασταυρούμενης επικύρωσης k-διπλωμάτων, η οποία περιγράφεται στον Αλγόριθμο 2

Αλγόριθμος 2 Διασταυρούμενη επικύρωση k-διπλωμάτων

1: Το σύνολο των δεδομένων χωρίζεται σε k κομμάτια
2: **Για** $i = 1, 2, \ldots k$
3: Όρισε ως δεδομένα ελέγχου το i-οστό κομμάτι
4: Όρισε ως δεδομένα εκπαίδευσης τα υπόλοιπα $k - 1$ κομμάτια
5: Εκπαίδευσε το μοντέλο στα δεδομένα εκπαίδευσης
6: Αξιολόγησε το μοντέλο στα δεδομένα ελέγχου
7: **Τέλος Για**
8: Υπολόγισε την τελική αξιολόγηση ως το μέσο όρο των k επιμέρους αξιολογήσεων

Με την μέθοδο αυτή αυξάνεται η αξιοπιστία των αποτελεσμάτων του αλγορίθμου καθώς τα αποτελέσματα του έχουν αξιολογηθεί σε περισσότερα από ένα κομμάτια. Το αρνητικό της είναι η αύξηση της υπολογιστικής πολυπλοκότητας, λόγω του αριθμού των επαναλήψεων.

Η παράμετρος που καθορίζει πόσο θα αυξηθεί η αξιοπιστία και η πολυπλοκότητα του αλγορίθμου, είναι η παράμετρος k. Η αύξηση της τιμής της προκαλεί αύξηση της αξιοπιστίας των αποτελεσμάτων ενώ παράλληλα αυξάνεται και το υπολογιστικό κόστος. Η μείωση του όπως είναι φυσικό επιφέρει τα αντίστροφα αποτελέσματα. Έτσι, για μεγάλο αριθμό δεδομένων είναι μάλλον αδύνατο να επιλέξουμε πολύ μεγάλες τιμές του k, κάτι το οποίο όμως δεν μας χρειάζεται αφού ταυτόχρονα έχουμε περισσότερα δεδομένα για την εκπαίδευση του μοντέλου κι έτσι δεν είναι πρόβλημα να κρατάμε ένα σχετικά μεγάλο ποσοστό των δεδομένων για τον έλεγχο του. Αντίθετα, για μικρό αριθμό δεδομένων έχουμε μεγαλύτερη ελευθερία επιλογής ενώ, ταυτόχρονα, μεγαλύτερη ανάγκη για οικονομία στο σύνολο των παραδειγμάτων εκπαίδευσης. Άρα η επιλογή της τιμής πρέπει να γίνεται με κριτήριο το μέγεθος του σώματος δεδομένων που έχουμε στη διάθεση μας. Στην περίπτωση μας θεωρήσαμε ότι η τιμή 5 είναι η πλέον καταλληλότερη.

4.2.5 Κανονικοποίηση Δεδομένων

Μετά την εξαγωγή των χαρακτηριστικών από κάθε αρχείο εισόδου, υπάρχουν διάφορα είδη επεξεργασίας τα οποία μπορούν να εφαρμοσθούν έτσι ώστε είτε να βελτιωθεί η ακρίβεια της πρόβλεψης του ταξινομητή ,είτε να μειωθεί ο χρόνος εκπαίδευσης του εκάστοτε μοντέλου διατηρώντας την ακρίβεια στο ίδιο επίπεδο. Μια βασική μέθοδος την οποία χρησιμοποιήσαμε στις περιπτώσεις που εξετάσαμε είναι η *κανονικοποίηση* (normalization). Η κανονικοποίηση των τιμών των χαρακτηριστικών που έχουν εξαχθεί από τα δεδομένα εισόδου είναι απαραίτητη για πολλές από τις μεθόδους μηχανικής μάθησης που χρησιμοποιούνται για την ταξινόμηση των δειγμάτων. Χωρίς αυτήν τα αποτελέσματα συνήθως είναι χειρότερα και ο χρόνος εκπαίδευσης αρκετά μεγαλύτερος [Meye01]. Η κανονικοποίηση συνήθως γίνεται αφαιρώντας από όλες τις τιμές ενός χαρακτηριστικού την μέση τιμή αυτού και διαιρώντας την με την τυπική του απόκλιση. Το μέγεθος που προκύπτει ονομάζεται z-score (Εξίσωση 4.1)

$$z = \frac{x - \mu}{\sigma} \tag{4.1}$$

όπου μ η μέση τιμή και σ η τυπική απόκλιση.

4.3 Μετρικές

Στο κομμάτι αυτό θα παρουσιάσουμε τις μετρικές που χρησιμοποιήσαμε για την αξιολόγηση του συστήματος ανίχνευσης κακόβουλων PHP αρχείων [Powe11]. Το γεγονός της ανισορροπίας των κλάσεων καθιστά απαραίτητη τη χρήση μετρικών που κατά την αξιολόγηση του μοντέλου μηχανικής μάθησης, λαμβάνουν υπόψη την ιδιαιτερότητα αυτή. Η χρήση δηλαδή μετρικών όπως η *απόλυτη ακρίβεια* ($\frac{\text{Σωστές Προβλέψεις}}{\text{Σύνολο Προβλέψεων}}$) δεν θα μας έδινε αντιπροσωπευτικά αποτελέσματα. Αυτό γίνεται κατανοητό αν σκεφτεί κανείς πως αν το σύστημα προβλέψει πως όλα τα αρχεία είναι μη μολυσμένα, τότε η τιμή της παραπάνω μετρικής θα ήταν περίπου 0,9, λόγω της αναλογίας μολυσμένων και μη μολυσμένων αρχείων που αναφέρθηκε στην Ενότητα 4.1. Επίσης η μη ανίχνευση ενός μολυσμένου PHP αρχείου μπορεί να προκαλέσει σοβαρή ζημιά σε έναν διακομιστή ιστού. Γίνεται λοιπόν αντιληπτό ότι πρέπει να δοθεί μεγαλύτερη βαρύτητα στην κλάση που αφορά τα μολυσμένα αρχεία.

Όπως είδαμε παραπάνω σε κάθε αρχείο προσδίδεται μια δυαδική τιμή ανάλογα με το αν είναι μολυσμένο ή όχι (+1 για μη μολυσμένο και -1 για μολυσμένο). Έτσι για κάθε πρόβλεψη, τέσσερις καταστάσεις είναι πιθανές:

Αληθώς Θετικό (True Positive): Πρόβλεψη ως μολυσμένο για ένα μολυσμένο αρχείο.

Αληθώς Αρνητικό (True Negative): Πρόβλεψη ως μη μολυσμένο για ένα μη μολυσμένο αρχείο.

Ψευδώς Θετικό (False Positive): Πρόβλεψη ως μολυσμένο για ένα μη μολυσμένο αρχείο.

Ψευδώς Αρνητικό (False Negative): Πρόβλεψη ως μη μολυσμένο για ένα μολυσμένο αρχείο.

Οι παραπάνω καταστάσεις συνοψίζονται στον Πίνακα 4.1, με θετική κλάση να θεωρείται αυτή των μολυσμένων αρχείων:

Πίνακας 4.1: Πίνακας Σύγχυσης (Confusion Matrix)

	Πρόβλεψη ως μολυσμένο	**Πρόβλεψη ως μη-μολυσμένο**
Μολυσμένο	Αληθώς Θετικό (TP)	Ψευδώς Αρνητικό (FP)
Μη-μολυσμένο	Ψευδώς Θετικό (FN)	Αληθώς Αρνητικό (TN)

Οι μετρικές που χρησιμοποιήσαμε συνοψίζονται παρακάτω [Powe11]:

Ακρίβεια (Precision): Ορίζεται ως ο λόγος του αριθμού των σωστά προβλεπόμενων αρχείων ως μολυσμένα προς τον αριθμό των συνολικών αρχείων που προβλέφθηκαν ως μολυσμένα. Στη ουσία μας δίνει την ακρίβεια των θετικών προβλέψεων.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Ανάκληση (Recall): Ορίζεται ως ο λόγος του αριθμού των σωστά προβλεπόμενων αρχείων ως μολυσμένα προς τον αριθμό των συνολικών μολυσμένων αρχείων. Στη ουσία μας δίνει την ακρίβεια με την οποία ένα μολυσμένο αρχείο εντοπίζεται.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

Μετρική F1 (F1-score): Ορίζεται ως ο αρμονικός μέσος της *Ακρίβειας* και της *Ανάκλησης*. Υψηλή τιμή της μετρικής F1 σημαίνει ότι το σύστημα επιτυγχάνει μια καλή ισορροπία όσον αφορά τις δύο προαναφερόμενες μετρικές.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.4}$$

Ευαισθησία (Sensitivity): Ορίζεται με τον ίδιο τρόπο που ορίζεται και η *Ανάκληση*

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.5}$$

Ιδιαιτερότητα (Specificity): Ορίζεται ως ο λόγος του αριθμού των σωστά προβλεπόμενων αρχείων ως μη-μολυσμένα προς το πλήθος των προβλέψεων των μη-μολυσμένων αρχείων. Συμβολίζει την *Ακρίβεια* στα αρνητικά παραδείγματα.

$$Specificity = \frac{TN}{TN + FP} \tag{4.6}$$

Γεωμετρικός Μέσος (Geometric Mean): Ο γεωμετρικός μέσος μετράει την ισορροπία μεταξύ της ακρίβειας της κατηγοριοποίησης της πλειοψηφικής κλάσης και της μειοψηφικής κλάσης. Η μετρική αυτή λαμβάνει υπόψη την *Ευαισθησία* και την *Ιδιαιτερότητα*.

$$G - Mean = \sqrt{Sensitivity * Specificity} \tag{4.7}$$

Όπως γίνεται εύκολα αντιληπτό οι παραπάνω μετρικές αφορούν μόνο την θετική κλάση (αυτή δηλαδή που περιέχει τα μολυσμένα αρχεία). Ο λόγος που μας ενδιαφέρουν σχεδόν περισσότερο οι μετρικές της κλάσης αυτής είναι η ανισορροπία των κλάσεων όπως περιγράφηκε παραπάνω. Εξαίρεση αποτελεί ο Γεωμετρικός Μέσος ο οποίος αφορά και τις δύο κλάσεις.

4.4 Αποτελέσματα

Στο κομμάτι αυτό παρουσιάζονται τα καλύτερα αποτελέσματα της αξιολόγησης κάθε αλγορίθμου μηχανικής μάθησης που υλοποιήσαμε. Ως βέλτιστα θεωρήσαμε τα αποτελέσματα που είχαν την μεγαλύτερη τιμή στη μετρική F1 στην κλάση μειοψηφίας. Επίσης τα αποτελέσματα αυτά αναπαρίστανται γραφικά σε ραβδογράμματα (Σχήματα 4.2, 4.4, 4.6) για την ευκολότερη κατανόηση και σύγκριση μεταξύ τους. Ακόμη παρουσιάζονται σε διαγράμματα (ένα για κάθε αλγόριθμο) οι τιμές των μετρικών μέτρο F1 στην κλάση μειοψηφίας και Γεωμετρικός υπό τη μορφή box plots (Σχήματα 4.3, 4.5, 4.7), κάτι που μας βοηθάει να μελετήσουμε την ευστάθεια των αποτελεσμάτων.

4.4.1 Δέντρα αποφάσεων

Στον αλγόριθμο αυτό το σύνολο των χαρακτηριστικών που μας έδωσαν τα καλύτερα αποτελέσματα είναι αυτά που αφορούν τη συχνότητα εμφάνισης των συναρτήσεων της γλώσσας με F1 ίσο με 0,78. Οι παράμετροι του αλγορίθμου που μας έδωσαν το αποτέλεσμα αυτό συνοψίζονται στον Πίνακα 4.2

Πίνακας 4.2: Βέλτιστες παράμετροι για τα Δέντρα Αποφάσεων

Παράμετρος	**Τιμή**
Splitter	gini
Min samples split	18
Max depth	3
Min samples leaf	4
Max leaf nodes	12
Class weight	Balanced

Από την τιμή της παραμέτρου *class weight* παρατηρούμε ότι δεν χρειάστηκε εκπαίδευση με ευαισθησία κόστους για την βελτιστοποίηση του αλγορίθμου αυτού.

Πίνακας 4.3: Αποτελέσματα Αξιολόγησης Δέντρων Αποφάσεων

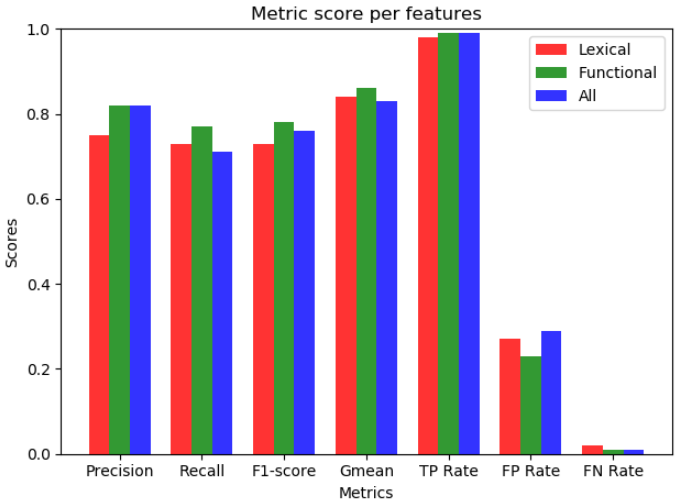
Χαρακτηριστικά	**Ακρίβεια**	**Ανάκληση**	**F1**	**Γεωμετρικός Μέσος**	**True Positive Rate**	**False Positive Rate**	**False Negative Rate**
Λεξικογραφικά	0,75	0,73	0,73	0,84	0,98	0,27	0,02
Εμφανίσεις Συναρτήσεων	0,82	0,77	0,78	0,86	0,99	0,23	0,01
Όλα	0,82	0,71	0,76	0,83	0,99	0,29	0,01

Στο Σχήμα 4.2 και στον Πίνακα 4.3 παρατηρούμε ότι σε όλες τις μετρικές αξιολόγησης τα χαρακτηριστικά που αφορούν τον αριθμό των εμφανίσεων των συναρτήσεων μας δίνουν τουλάχιστον καλύτερα αποτελέσματα από τα άλλα δυο σύνολα χαρακτηριστικών. Όσον αφορά τα άλλα δύο σύνολα χαρακτηριστικών, το καθένα υπερτερεί του άλλου σε κάποιες μετρικές με αυτό των συνδυαστικών να έχει μεγαλύτερο F1.

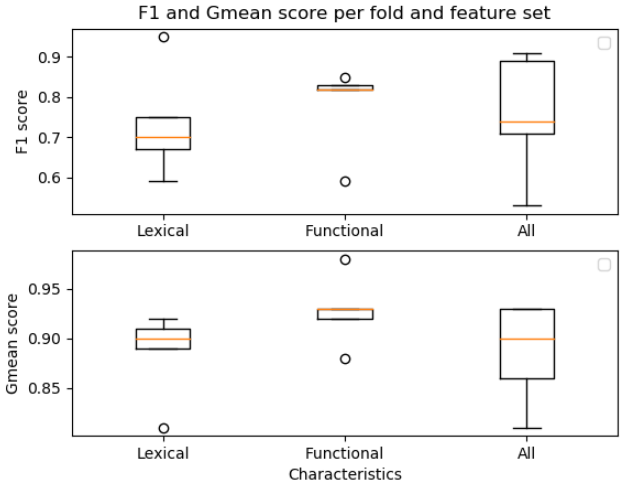
Ωστόσο, από το Σχήμα 4.3 φαίνεται πως και για τα τρία σύνολα χαρακτηριστικών οι τιμές της μετρικής F1 παρουσιάζουν μεγάλο εύρος ενώ του Γεωμετρικού Μέσου μικρότερο. Παρόλα αυτά, μπορούμε να δούμε ότι το δεύτερο σύνολο χαρακτηριστικών παρουσιάζει τις λιγότερες αυξομειώσεις.

4.4.2 Μηχανές Διανυσμάτων Υποστήριξης

Στον αλγόριθμο αυτό το σύνολο των χαρακτηριστικών που μας έδωσαν τα καλύτερα αποτελέσματα είναι αυτά που αφορούν τη συχνότητα εμφάνισης των συναρτήσεων της γλώσσας με F1 ίσο με



Σχήμα 4.2: Σύγκριση αποτελεσμάτων για τα 3 σύνολα χαρακτηριστικών στα Δέντρα Αποφάσεων



Σχήμα 4.3: Box plots για F1 και Γεωμετρικό Μέσο στα Δέντρα Αποφάσεων

0,84. Οι παράμετροι του αλγορίθμου που μας έδωσαν το αποτέλεσμα αυτό συνοψίζονται στον Πίνακα 4.4

Παρατηρούμε ότι τα βάρη που δόθηκαν σε κάθε κλάση για την εκπαίδευση με ευαισθησία κόστους είναι όση και αναλογία μεταξύ των κλάσεων.

Από το Σχήμα 4.4 και τον Πίνακα 4.5 γίνεται εμφανές ότι και σε αυτόν τον αλγόριθμο, όπως και στα ΔΑ, στις περισσότερες από μετρικές αξιολόγησης τα χαρακτηριστικά που αφορούν τον αριθμό των εμφανίσεων των συναρτήσεων μας δίνουν τουλάχιστον καλύτερα αποτελέσματα από τα άλλα δυο σύνολα χαρακτηριστικών.

Στο Σχήμα 4.5 φαίνεται πως και για τα τρία σύνολα χαρακτηριστικών οι τιμές του Γεωμετρικού μέσου και του μέτρου F1 παρουσιάζουν μεγαλύτερες αυξομειώσεις. Μοναδική εξαίρεση αποτελεί η τιμή του Γεωμετρικού μέσου για τα ΔΑ, όπου οι μεταβολές του είναι πολύ μικρές.

4.4.3 Στοχαστική κατάβαση κλίσης

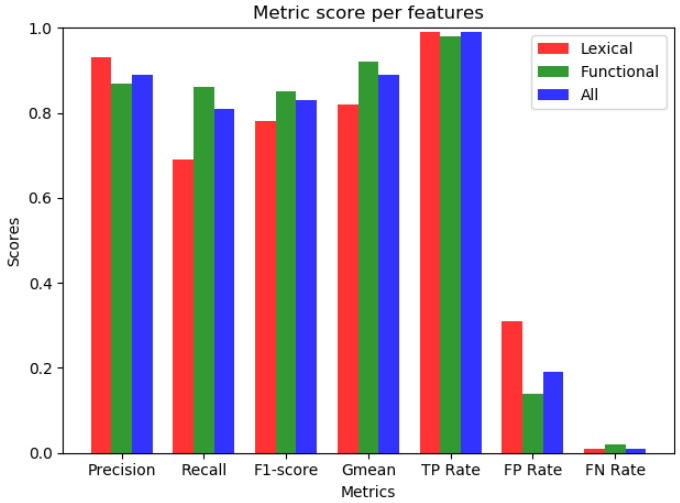
Στον αλγόριθμο αυτό το σύνολο των χαρακτηριστικών που μας έδωσαν τα καλύτερα αποτελέσματα είναι αυτά που αφορούν τη συχνότητα εμφάνισης των συναρτήσεων της γλώσσας με F1 ίσο με 0,84. Οι παράμετροι του αλγορίθμου που μας έδωσαν το αποτέλεσμα αυτό συνοψίζονται στον Πίνακα

Πίνακας 4.4: Βέλτιστες παράμετροι για τις Μηχανές Διανυσμάτων Υποστήριξης

Παράμετρος	**Τιμή**
Kernel	RBF
C	31,76
Gamma	$5,2 \times 10^{-5}$
Class weight	1 για τη θετική κλάση, 10 για την αρνητική

Πίνακας 4.5: Αποτελέσματα Αξιολόγησης Μηχανών Διανυσμάτων Υποστήριξης

Χαρακτηριστικά	**Ακρίβεια**	**Ανάκληση**	**F1**	**Γεωμετρικός Μέσος**	**True Positive Rate**	**False Positive Rate**	**False Negative Rate**
Λεξικογραφικά	0,83	0,71	0,75	0,83	0,99	0,29	0,01
Εμφανίσεις Συναρτήσεων	0,91	0,82	0,84	0,89	0,99	0,18	0,01
Όλα	0,81	0,75	0,76	0,85	0,98	0,25	0,02



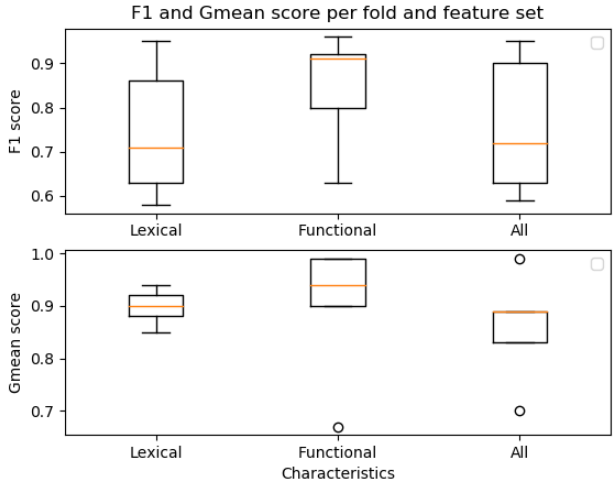
Σχήμα 4.4: Σύγκριση αποτελεσμάτων για τα 3 σύνολα χαρακτηριστικών στις Μηχανές Διανυσμάτων Υποστήριξης

4.6

Παρατηρούμε ότι όπως και στις ΜΔΥ, τα βάρη που δόθηκαν σε κάθε κλάση για την εκπαίδευση με ευαισθησία κόστους είναι όση και αναλογία μεταξύ των κλάσεων

Ακριβώς όπως και στα ΔΑ, παρατηρούμε στο Σχήμα 4.6 και στον Πίνακα 4.7, ότι σε όλες τις μετρικές αξιολόγησης τα χαρακτηριστικά που αφορούν τον αριθμό των εμφανίσεων των συναρτήσεων μας δίνουν τουλάχιστον καλύτερα αποτελέσματα από τα άλλα δυο σύνολα χαρακτηριστικών.

Όσον αφορά τις διακυμάνσεις των αποτελεσμάτων, στο Σχήμα 4.7 παρατηρούμε πως και για τα τρία σύνολα χαρακτηριστικών οι τιμές του Γεωμετρικού μέσου και του F1 παρουσιάζουν πολλές αυξομειώσεις, με εξαίρεση τη τιμή του Γεωμετρικού μέσου για τα ΔΑ, όπου οι μεταβολές του είναι πολύ μικρές.



Σχήμα 4.5: Box plots για F1 και Γεωμετρικό Μέσο στις Μηχανές Διανυσμάτων Υποστήριξης

Πίνακας 4.6: Βέλτιστες παράμετροι για τη Στοχαστική Κατάβαση Κλίσης

Παράμετρος	**Τιμή**
Loss	log
Penalty	elasticnet
Alpha	0,1
L1 ratio	0,15
Learning rate	optimal
Class weight	1 για τη θετική κλάση, 10 για την αρνητική

Πίνακας 4.7: Αποτελέσματα Αξιολόγησης Στοχαστικής Κατάβασης Κλίσης

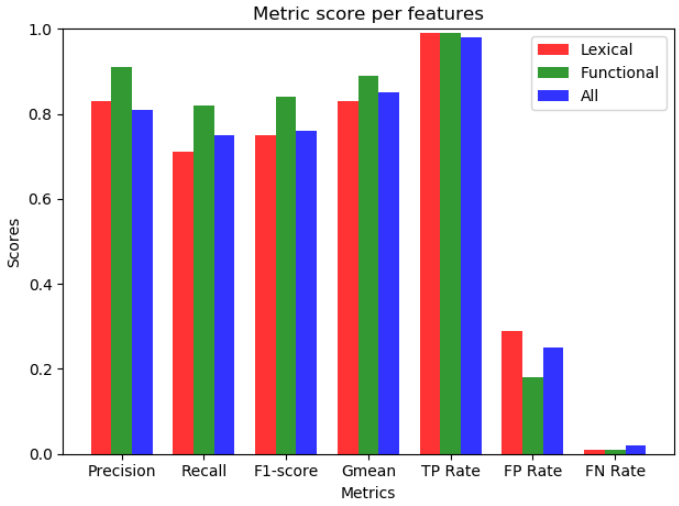
Χαρακτηριστικά	**Ακρίβεια**	**Ανάκληση**	**F1**	**Γεωμετρικός Μέσος**	**True Positive Rate**	**False Positive Rate**	**False Negative Rate**
Λεξικογραφικά	0,93	0,69	0,78	0,82	0,99	0,31	0,01
Εμφανίσεις Συναρτήσεων	0,87	0,86	0,85	0,92	0,98	0,14	0,02
Όλα	0,89	0,81	0,83	0,89	0,99	0,19	0,01

4.4.4 Σύγκριση των αλγορίθμων

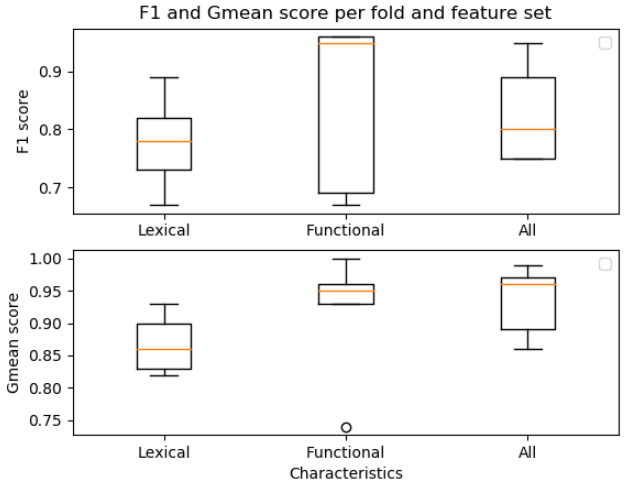
Στο Σχήμα 4.8 που ακολουθεί αναπαρίστανται οι καλύτερες τιμές του Γεωμετρικού μέσου και του F1 για τον κάθε αλγόριθμο. Παρατηρούμε ότι ο αλγόριθμος της ΣΚΚ με λογιστική συνάρτηση κόστους μας δίνει και στις δύο μετρικές καλύτερα αποτελέσματα (F1-score=0,85, Gmean-score=0,92) από τους άλλους δύο αλγορίθμους. Ακολουθεί ο αλγόριθμος των ΜΔΥ (F1-score=0,84, Gmean-score=0,89) και τελευταίος είναι ο αλγόριθμος των ΔΑ (F1-score=0,78, Gmean-score=0,86).

Τα γενικά συμπεράσματα που προκύπτουν είναι τα εξής:

- Τα χαρακτηριστικά που αφορούν τον αριθμό εμφανίσεων των συναρτήσεων έχουν τα καλύτερα αποτελέσματα και για τους τρεις αλγορίθμους.
- Ο αλγόριθμος της ΣΚΚ με λογιστική συνάρτηση κόστους είναι αυτός που μας δίνει τα καλύτερα αποτελέσματα.

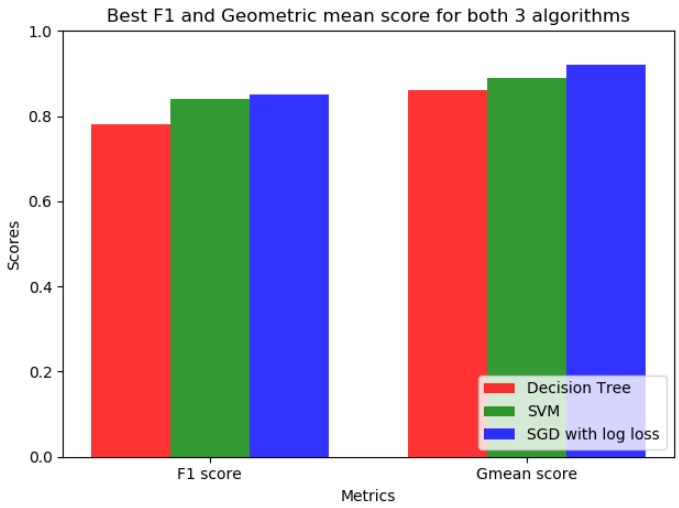


Σχήμα 4.6: Σύγκριση αποτελεσμάτων για τα 3 σύνολα χαρακτηριστικών στη Στοχαστική Κατάβαση Κλίσης



Σχήμα 4.7: Box plots για F1 και Γεωμετρικό Μέσο στη Στοχαστική Κατάβαση Κλίσης

- Υπάρχει αρκετή αστάθεια στα αποτελέσματα αφού οι τιμές τους έχουν μεγάλες διακυμάνσεις όπως φαίνεται και από τα αντίστοιχα box plots (ειδικά για την μετρική F1).
- Η ακρίβεια και των τριών αλγορίθμων είναι μεγάλη αφού το F1 κυμαίνεται μεταξύ 0,78 και 0,85 ενώ ο γεωμετρικός μέσος έχει τιμές μεταξύ 0,86 και 0,92.



Σχήμα 4.8: Σύγκριση μετρικής F1 και Γεωμετρικού Μέσου για τους 3 αλγορίθμους

Κεφάλαιο 5

Συμπεράσματα και Μελλοντικές Κατευθύνσεις

5.1 Συμπεράσματα

5.1.1 Χαρακτηριστικά

Όπως φάνηκε από τα αποτελέσματα που παρουσιάστηκαν στο Κεφάλαιο 4, ενώ και τα τρία σύνολα χαρακτηριστικών δίνουν καλά αποτελέσματα, τα χαρακτηριστικά που αφορούν τον αριθμό εμφανίσεων των συναρτήσεων είναι αυτά που ξεχωρίζουν. Έτσι μπορούμε συμπεράνουμε πως τα μολυσμένα αρχεία έχουν περισσότερα κοινά όσον αφορά τις συναρτήσεις που χρησιμοποιούν (όπως η *eval*), παρά τα λεξικογραφικά χαρακτηριστικά του κώδικα.

5.1.2 Εξισορρόπηση των δεδομένων εκπαίδευσης

Η μέθοδος που χρησιμοποιήσαμε για την εξισορρόπηση των δεδομένων εκπαίδευσης ήταν ιδιαίτερα επιτυχής, αφού χάρη σε αυτήν μπορέσαμε να εφαρμόσουμε κάποια μοντέλα εκμάθησης, όπως οι ΜΔΥ και η ΣΚΚ που δεν θα ανταποκρίνονταν χωρίς αυτή.

5.1.3 Ευστάθεια αποτελεσμάτων

Για την μελέτη της ευστάθειας των αποτελεσμάτων μελετήσαμε εκτός από το τελικό αποτέλεσμα κάθε μετρικής μετά την διασταυρούμενη επικύρωση k-διπλωμάτων, και τα ενδιάμεσα αποτελέσματα αυτής. Από τα αντίστοιχα box plots βγάλαμε το συμπέρασμα ότι σε αρκετές περιπτώσεις τα αποτελέσματα δεν ήταν αρκετά ευσταθή, κάτι το οποίο πρέπει να λάβουμε υπόψη μας. Βέβαια στην αστάθεια αυτή ίσως παίζει ρόλο και ο σχετικά μικρός αριθμός των δεδομένων εκπαίδευσης.

5.2 Μελλοντικές Κατευθύνσεις

Κάποιες κατευθύνσεις για την επέκταση της διπλωματικής εργασίας είναι οι εξής:

- Πειραματισμός με επιπλέον αλγορίθμους μηχανικής μάθησης, όπως τα *Τυχαία Δάση* (Random Forests).
- Μελέτη επιπλέον χαρακτηριστικών που μπορούν να προκύψουν από αρχεία PHP. Τέτοια θα μπορούσαν να είναι τα *n-grams* [Sant], τα οποία αν και χρησιμοποιούνται περισσότερο σε κείμενα που περιέχουν φυσική γλώσσα, ίσως να αποδειχθούν ενδιαφέροντα και στην περίπτωση που εξετάζουμε.
- Επίσης η κατασκευή ενός διαγράμματος ροής για το κάθε αρχείο και η περαιτέρω μελέτη του [Bazr13] μπορεί να μας δώσει κάποια χαρακτηριστικά χρήσιμα για την κατηγοριοποίηση των αρχείων ως μολυσμένα ή μη.
- Σύγκριση των αποτελεσμάτων του συστήματος που υλοποιήσαμε με κάποιο ήδη υπάρχων εργαλείο ανίχνευσης κακόβουλου PHP κώδικα χρησιμοποιώντας τα ίδια δεδομένα.

Βιβλιογραφία

[Bazr13] Z. Bazrafshan, H. Hashemi, S. M. H. Fard and A. Hamzeh, "A survey on heuristic malware detection techniques", in *The 5th Conference on Information and Knowledge Technology*, pp. 113–120, May 2013.

[Bose92] Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers", in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pp. 144–152, New York, NY, USA, 1992, ACM.

[Bott98] Leon Bottou, "On-line Learning in Neural Networks", chapter On-line Learning and Stochastic Approximations, pp. 9–42, Cambridge University Press, New York, NY, USA, 1998.

[Bram16] Max Bramer, *Principles of Data Mining*, Undergraduate Topics in Computer Science, Springer-Verlag London, 3 edition, 2016.

[Domi12] Pedro Domingos, "A Few Useful Things to Know About Machine Learning", *Commun. ACM*, vol. 55, no. 10, pp. 78–87, October 2012.

[Elka01] Charles Elkan, "The Foundations of Cost-sensitive Learning", in *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, pp. 973–978, San Francisco, CA, USA, 2001, Morgan Kaufmann Publishers Inc.

[Esla12] M. Eslahi, R. Salleh and N. B. Anuar, "Bots and botnets: An overview of characteristics, detection and challenges", in *2012 IEEE International Conference on Control System, Computing and Engineering*, pp. 349–354, Nov 2012.

[Gang12] Vaishali Ganganwar, "An overview of classification algorithms for imbalanced datasets", vol. 2, pp. 42–47, 01 2012.

[Gare13] James Gareth, Witten Daniela, Hastie Trevor and Tibshirani Robert, *An Introduction to Statistical Learning*, Springer Texts in Statistics, Springer-Verlag New York, 1 edition, 2013.

[Hear98] Marti A. Hearst, "Support Vector Machines", *IEEE Intelligent Systems*, vol. 13, no. 4, pp. 18–28, July 1998.

[Japk00] Nathalie Japkowicz, "The Class Imbalance Problem: Significance and Strategies", in *In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI*, pp. 111–117, 2000.

[JRus95] Stuart J Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, chapter Intelligent agents, pp. 31–52, Prentice Hall, 01 1995.

[Kiwi01] Krzysztof C. Kiwiel, "Convergence and efficiency of subgradient methods for quasiconvex minimization", *Mathematical Programming*, vol. 90, no. 1, pp. 1–25, Mar 2001.

[Lika09] P. Likarish, E. Jung and I. Jo, "Obfuscated malicious javascript detection using classification techniques", in *2009 4th International Conference on Malicious and Unwanted Software (MALWARE)*, pp. 47–54, Oct 2009.

[Meye01] David Meyer and FH Technikum Wien, "Support vector machines", *R News*, vol. 1, no. 3, pp. 23–26, 2001.

[Mitc97] Thomas M. Mitchell, *Machine Learning*, McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[Mitc06] Tom Michael Mitchell, "The discipline of machine learning", Technical report, School of Computer Science, Carnegie Mellon University, 2006.

[Mosk09] Robert Moskovitch, Dima Stopel, Clint Feher, Nir Nissim, Nathalie Japkowicz and Yuval Elovici, "Unknown malcode detection and the imbalance problem", *Journal in Computer Virology*, vol. 5, no. 4, p. 295, Jul 2009.

[Powe11] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation", *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[Rude16] Sebastian Ruder, "An overview of gradient descent optimization algorithms", *CoRR*, vol. abs/1609.04747, 2016.

[Safa91] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, May 1991.

[Sant] Igor Santos, Yoseba K. Penya, Jaime Devesa and Pablo G. Bringas, "N-Grams-based file signatures for malware detection", in *in: Proceedings of the 2009 International Conference on Enterprise Information Systems (ICEIS), Volume AIDSS*, pp. 317–320.

[Shal11] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro and Andrew Cotter, "Pegasos: primal estimated sub-gradient solver for SVM", *Mathematical Programming*, vol. 127, no. 1, pp. 3–30, Mar 2011.

[Simo13] Phil Simon, *Too Big to Ignore: The Business Case for Big Data*, Wiley Publishing, 1st edition, 2013.

[webr11] "Malicious PHP Scripts on the Rise", Online, February 2011.

[Zadr01] Bianca Zadrozny and Charles Elkan, "Learning and Making Decisions when Costs and Probabilities Are Both Unknown", in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pp. 204–213, New York, NY, USA, 2001, ACM.

[Zadr03] B. Zadrozny, J. Langford and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting", in *Third IEEE International Conference on Data Mining*, pp. 435–442, Nov 2003.

[Πα02] Νικόλαος Σ. Παπασπύρου και Εμμανουήλ Σ. Σκορδαλάκης, *Μεταγλωττιστές*, Συμμετρία, 202.

Παράρτημα Α

Ευρετήριο Ακρωνυμιών και Συντμήσεων

Α.0.1 Ελληνικών όρων

ΔΑ: Δέντρα Αποφάσεων (Decision Trees)
ΜΔΥ: Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines)
ΜΠΑ: Μη Ντετερμινιστικό Πεπερασμένο Αυτόματα (Non-deterministic Finite Automata)
ΝΠΑ: Ντετερμινιστικό Πεπερασμένο Αυτόματα (Deterministic Finite Automata)
ΣΚΚ: Στοχαστική Κατάβαση Κλίσης (Stochastic Gradient Descent)

Α.0.2 Αγγλικών όρων

DDoS: Distributed Denial-of-Service attach (Κατανεμημένη επίθεση Άρνησης Υπηρεσίας)
IRC: Internet Relay Chat
PHP: PHP: Hypertext Preprocessor