ΑΝΙΧΝΕΥΣΗ ΙΧΝΗΛΑΤΩΝ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ

Andoena Balla

Η Διατριβή αυτή

Υποβλήθηκε προς Μερική Εκπλήρωση των

Απαιτήσεων για την Απόκτηση

Τίτλου Σπουδών Master

σε Προηγμένες Τεχνολογίες Πληροφορικής

στο

Πανεπιστήμιο Κύπρου

Συστήνεται προς Αποδοχή

από το Τμήμα Πληροφορικής

Φεβρουάριος, 2009

ΣΕΛΙΔΑ ΕΓΚΡΙΣΗΣ

Διατριβή Master

ΑΝΙΧΝΕΥΣΗ ΙΧΝΗΛΑΤΩΝ ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΧΡΟΝΟ

Παρουσιάστηκε από

Andoena Balla

Ερευνητικός Σύμβουλος

_____

Αναπλ. Καθ. Μάριος Δικαιάκος

Μέλος Επιτροπής

_____

Καθ. Ελπίδα Κεραυνού-Παπαηλίου

Μέλος Επιτροπής

_____

Καθ. Αθηνά Στασοπούλου

Πανεπιστήμιο Κύπρου

Φεβρουάριος , 2009

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Μάριο Δικαιάκο, για την σημαντική βοήθεια και καθοδήγηση που μου παρείχε κατά την εκπόνηση της μεταπτυχιακής μου διατριβής.

Ιδιαίτερα ευχαριστώ την οικογένεια μου και τον Λεόντιο, για την υποστήριξη και την συμπαράστασή τους.

ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας διατριβής είναι η ανάπτυξη μιας μεθόδου για την ανίχνευση των ιχνηλατών σε πραγματικό χρόνο. Ένας ιχνηλάτης είναι ένα πρόγραμμα το οποίο διασχίζει αυτόματα τους υπερσυνδέσμους στον παγκόσμιο ιστό, με σκοπό την ανάκτηση και την αποθήκευση πληροφοριών από το διαδίκτυο. Ενώ η χρήση των ιχνηλατών γίνεται για διαφορετικούς σκοπούς, οι οποίοι είναι σημαντικοί για την σωστή λειτουργία πολλών εφαρμογών του διαδικτύου, υπάρχουν περιπτώσεις που είναι αναγκαία η ανίχνευση και η διάκρισή τους από τους χρήστες του διαδικτύου.

Στην παρούσα εργασία αναπτύξαμε μια μέθοδο, η οποία χρησιμοποίει δέντρα αποφάσεων, για την ανίχνευση των ιχνηλατών καθώς η σύνοδος τους με τον εξυπηρετητή είναι ακόμα ανοιχτή. Η μέθοδος αυτή είναι απλή, έτσι ώστε να μην επιβαρύνεται ο εξυπηρετητής σε μνήμη και σε χρόνο αλλά ταυτόχρονα είναι πολύ αποτελεσματική στην ανίχνευση των ιχνηλατών. Η προτεινόμενη μέθοδος στηρίζεται στα πρότυπα πλοήγησης των ιχνηλατών και των χρηστών. Για την εύρεση αυτών των προτύπων αναλύσαμε τις HTTP αιτήσεις στα αρχεία απογραφής των εξυπηρετητών ακλουθώντας μια Πιθανοθεωριτική συλλογιστική προσέγγιση και στατιστικές μεθόδους.

Τρέξαμε πειραματικά την προτεινόμενη μέθοδο σε πραγματικό χρόνο με την χρήση ενός προσομοιωτή ο όποιος αναπτύχθηκε επίσης στα πλαίσια της παρούσας εργασίας. Τα αποτελέσματα της αξιολόγησης έδειξαν ότι το σύστημα ανιχνεύει τους ιχνηλάτες με μεγάλο ποσοστό επιτυχίας χρησιμοποιώντας μονό ένα μικρό αριθμό αιτημάτων.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ ΣΧΗΜΑΤΩΝ

1. Κεφάλαιο 1

Εισαγωγή

1.1 Γενικά

Ένας ιχνηλάτης (web robot) είναι ένα πρόγραμμα το όποιο διασχίζει αυτόματα τους υπερσυνδέσμους (hyperlinks) στον παγκόσμιο ιστό, με σκοπό την ανάκτηση και την αποθήκευση πληροφοριών από το διαδίκτυο. Ο ιχνηλάτης ξεκάνει με μια λίστα από διευθύνσεις (URLs) που ονομάζεται seeds και καθώς επισκέπτεται αυτές τις διευθύνσεις βρίσκει όλους τους υπερσυνδέσμους στην ιστοσελίδα αυτή και τους προσθέτει σε μια λίστα η οποία περιέχει τις διευθύνσεις που θα επισκεφτεί με μια συγκεκριμένη στρατηγική στο μέλλον[1]. Αυτά τα robot ιχνηλατούν το παγκόσμιο ιστό για διαφορετικούς σκοπούς, για παράδειγμα, για να μαζεύουν στατιστικές για τη δομή του παγκόσμιου ιστού, οι μηχανές αναζήτησης χρησιμοποιούν τους ιχνηλάτες για την ανάκτηση και την ευρετηρίαση πληροφοριών, οι διαχειριστές ιστοσελίδων τους χρησιμοποιών για έλεγχο σπασμένων υπερσυνδέσεων (broken links), επίσης χρησιμοποιούνται για την αγορά στο διαδίκτυο, για να μαζεύουν ηλεκτρονικές διευθύνσεις κλπ.

Υπάρχουν περιπτώσεις που είναι αναγκαία η ανίχνευση των ιχνηλατών και η διάκριση τους από τους χρήστες του διαδικτύου: (1)-Η κίνηση που δημιουργείται στο διαδίκτυο από τους ιχνηλάτες επηρεάζει άμεσα την απόδοση και την ποιότητα εξυπηρέτησης από τους εξυπηρετητές. (2)-Οι ιστοσελίδες του ηλεκτρονικού εμπορείου μπορεί να μην επιθυμούν εισερχόμενες αιτήσεις (HTTP requests) από μη-εξουσιοδοτημένους ιχνηλάτες, σε αυτές τις περιπτώσεις είναι χρήσιμο η ανίχνευση και η απαγόρευση των ιχνηλατών στις συγκεκριμένες σελίδες. (3)-Αν θέλουμε να

εφαρμόσουμε κάποιον τρόπο εξόρυξης δεδομένων (data mining) στα αρχεία απογραφής (log files) για την εξαγωγή προτύπων (patterns)για την συμπεριφορά των χρηστών του διαδίκτυο, οι αιτήσεις που έχουν δημιουργηθεί από ιχνηλάτες θα πρέπει να αφαιρούνται, διαφορετικά μπορεί να βγάλουμε λανθασμένα συμπεράσματα όσον αφορά την πλοήγηση του χρήστη στον παγκόσμιο ιστό. (4)- Επίσης στις περισσότερες διαφημίσεις που εμφανίζονται στις μηχανές αναζήτησης όπως στο Google, Yahoo κλπ, οι διαφημιστές τους πληρώνουν για κάθε κλικ που κάνουν οι χρήστες στη διαφήμιση τους [1, 2, 3]. Σε αυτήν την περίπτωση είναι πολύ σημαντική η έγκυρη ανίχνευση των κλικ που κάνουν οι ιχνηλάτες.

Ενώ οι ηθικοί ιχνηλάτες αυτοπροσδιορίζονται, οι κακόβουλοι κρύβουν την ταυτότητα τους και ακόμα μπορεί να χρησιμοποιούν ταυτότητα κάποιου φυλλομετρητή για να μη γίνουν αντιληπτοί από τις κοινές μεθόδους ανίχνευσης ιχνηλατών.

Σκοπός αυτής της διπλωματικής είναι η ανάπτυξη μιας μεθόδου για την έγκυρη ανίχνευση των ιχνηλατών καθώς βρίσκονται σε δραστηριότητα. Αυτές οι μέθοδοι πρέπει να είναι απλές για να μην επιβαρύνουν τον εξυπηρετητή σε μνήμη και σε χρόνο και επίσης πρέπει να είναι αποτελεσματικές για την ανίχνευση ιχνηλατών με μεγάλο ποσοστό βεβαιότητας. Η μέθοδος που θα αναπτύξουμε σε αυτήν την ερευνά στηρίζεται στα πρότυπα πλοήγησης των ιχνηλατών τα οποία είναι διαφορετικά από αυτά των χρηστών.[4, 3] Για το λόγο αυτό, ένα πρώτο βήμα σε αυτό το έργο είναι εύρεση των προτύπων αυτών των δυο ομάδων. Η ανάλυση των HTTP αιτήσεων στα αρχεία απογραφής (log files) των εξυπηρετητών θα μας βοηθήσει για την εύρεση των προτύπων και ιδιοτήτων που ξεχωρίζουν τους ιχνηλάτες από τους χρήστες. Σε αυτήν την ανάλυση ακολουθούμε μια πιθανοτική συλλογιστική προσέγγιση για την ταξινόμηση των συνόδων ως χρηστών ή ιχνηλατών [2] και στη συνεχεία

εφαρμόζουμε στατιστικές μεθόδους για την ανακάλυψη στατιστικών ιδιοτήτων και χαρακτηριστικών στους συνόδους που δημιουργήθηκαν από τους ιχνηλάτες. Αυτά τα χαρακτηριστικά συνδυάζονται μαζί με βάρη για την ανάπτυξη ενός μοντέλου που θα ανιχνεύει τους ιχνηλάτες σε πραγματικό χρόνο.

1.2 Δομή αναφοράς

Η δομή αυτής της διπλωματικής εργασίας έχει ως εξής: Στο Κεφάλαιο 2 εξετάζουμε υφιστάμενες τεχνικές για το διαχωρισμό των συνόδων ως παραγόμενα από χρήστες ή ιχνηλάτες. Στο Κεφάλαιο 3 περιγράφουμε την ανάλυση αρχείων απογραφής: χωρισμός των αρχείων απογραφής σε συνόδους, ταξινόμηση των συνόδων σε χρήστες ή ιχνηλάτες με τον ταξινομητή (classifier), στατιστική ανάλυση των συνόδων, συμπεράσματα. Το Κεφαλαίο 4 παρουσιάζει το μοντέλο που αναπτύξαμε για την ανίχνευση των ιχνηλατών σε πραγματικό χρόνο. Στο Κεφάλαιο 5 παρουσιάζουμε τα αποτελέσματα και την αξιολόγηση του μοντέλου. Τέλος παρουσιάζουμε συμπεράσματα και μελλοντική δουλειά στο κεφάλαιο 6.

2. Κεφάλαιο 2

Σχετική έρευνα

Οι υφιστάμενες τεχνικές για ανίχνευση ιχνηλατών μπορούν να χωριστούν σε κατηγορίες με βάση την ανάλυση ή το μοντέλο που ακολουθούν. Σε αυτό το κεφάλαιο περιγράφουμε αρχικά κάποιες απλές τεχνικές και τους περιορισμούς τους, και στη συνέχεια παρουσιάζουμε δυο άλλες κατηγορίες μαζί με τις τεχνικές που ανήκουν σε αυτές.

2.1 Απλές τεχνικές για ανίχνευση ιχνηλατών

2.1.1 IP address check

Ένας τρόπος ανίχνευσης ιχνηλατών είναι να συγκρίνουμε την IP διεύθυνση του πελάτη (web client) με τις IP διευθύνσεις των ιχνηλατών οι οποίοι είναι ήδη γνωστοί. Το πρόβλημα της μεθόδου αυτής είναι ότι παρόλο ότι υπάρχουν πολλές ιστοσελίδες οι οποίες παρέχουν λίστες με τις IP διευθύνσεις των γνωστών ιχνηλατών [5] πρακτικά είναι δύσκολη η συντήρηση μιας ενημερωμένης βάσης δεδομένων με όλους τους ιχνηλάτες. Η ανίχνευση των ιχνηλατών με αυτήν τη μέθοδο είναι αποτελεσματική μόνο στην περίπτωση που ο ιχνηλάτης έχει γίνει γνωστός προηγουμένως με την χρήση κάποιας άλλης μεθόδου ή ευριστικά από τους διαχειριστές των εξυπηρετητών ιστού.

2.1.2 Αίτηση για το αρχείο robots.txt

Το Robot Exclusion Standard (RES) [6] προτάθηκε για να επιτρέψει στους διαχειριστές διαδικτύου να ορίζουν ποιες ιστοσελίδες των ιστοτόπων, μπορούν να επισκέπτονται από τους ιχνηλάτες. Σύμφωνα με αυτό το πρότυπο κάθε φορά που ένας ιχνηλάτης επισκέπτεται μια ιστοσελίδα, (π.χ. www.foo.com) πρέπει πρώτα να εξετάσει το αρχείο robots.txt που βρίσκεται στη ρίζα της ιστοσελίδας (π.χ. http://www.foo.com/robots.txt). Αυτό το αρχείο περιέχει μια λίστα περιορισμών πρόσβασης του ιστοτόπου, όπως αυτή έχει καθοριστεί από το διαχειριστή του. Οι ιχνηλάτες μπορούν να ανιχνευτούν εύκολα όταν οι συνόδοι τους περιέχουν ένα αίτημα για το αρχείο robots.txt. Αυτός είναι ένας λογικός τρόπος ανίχνευσης ιχνηλατών για το λόγο ότι, οι περισσότεροι ιστοτόποι δεν περιέχουν εμφανείς σύνδεσμους για το αρχείο αυτό και οι περισσότεροι χρήστες δεν ξέρουν την ύπαρξη του. Παρ'όλα αυτά δεν μπορούμε να στηριζόμαστε αποκλειστικά σε αυτό το κριτήριο επειδή η συμμόρφωση με το RES είναι προαιρετική και πολλοί ιχνηλάτες μπορεί να μην το εφαρμόσουν.

2.1.3 Το ποσοστό των αιτήσεων με την μέθοδο HEAD

Υπάρχουν κάποιοι κανόνες για τον σωστό σχεδιασμό των ιχνηλατών [6] ένας εκ των οποίων εισηγείται ότι ένας ιχνηλάτης θα πρέπει να χρησιμοποιεί την μέθοδο HEAD όποτε αυτό είναι δυνατόν. Οι μέθοδοι GET, HEAD και POST ενός HTTP αιτήματος, προσδιορίζουν την πράξη που θα εκτελέσει ο εξυπηρετητής πάνω στο αρχείο που ζήτησε ο πελάτης. Ο εξυπηρετητής απαντά στη μέθοδο GET στέλνοντας ένα μήνυμα

το οποίο αποτελείται από μια επικεφαλίδα η οποία περιλαμβάνει κάποιες πληροφορίες, μαζί με το αρχείο που ζητήθηκε από τον πελάτη. Στην μέθοδο HEAD ο εξυπηρετητής επιστρέφει μονό την επικεφαλίδα η οποία έχει μικρή επιβάρυνση στην επικοινωνία. Αυτός είναι και ο λόγος που οι ιχνηλάτες πρέπει να χρησιμοποιούν όποτε αυτό είναι δυνατόν την μέθοδο HEAD. Παρόλα αυτά πολλοί ιχνηλάτες χρησιμοποιούν κατευθείαν την μέθοδο GET για τη ανάκτηση αρχείων με αποτέλεσμα αυτή η μέθοδος ανίχνευσης ιχνηλατών να μην είναι αποτελεσματική.

2.1.4 User agent check

Σύμφωνα με τους κανόνες υλοποίησης ιχνηλατών, ένας ιχνηλάτης πρέπει να δηλώσει την ταυτότητα του στον εξυπηρετητή μέσω του πεδίου user agent. Παρόλα αυτά πολλοί ιχνηλάτες δεν δηλώνουν την ταυτότητα τους σε αυτό το πεδίο ή κρύβουν την ταυτότητά τους χρησιμοποιώντας την ταυτότητα ενός γνωστού φυλλομετρητή.

2.1.5 Null referrer

Το πρωτόκολλο HTTP παρέχει το πεδίο referrer για να επιτρέπει στον πελάτη να ορίζει τη διεύθυνση της ιστοσελίδας στην οποία βρισκόταν ο υπερσύνδεσμος τον οποίο ακολούθησε για να φτάσει στην τρέχουσα ιστοσελίδα [7]. Οι περισσότερες αιτήσεις που προέρχονται από τους ιχνηλάτες δεν έχουν κάποια τιμή σε αυτό το πεδίο, αντίθετα με τις αιτήσεις των χρηστών. Παρόλα αυτά δεν μπορούμε να στηριζόμαστε σε αυτό το κριτήριο για το λόγο ότι υπάρχουν περιπτώσεις που και οι χρήστες μπορούν να παράγουν αιτήσεις οι οποίες δεν θα έχουν κάποια τιμή στο πεδίο referrer.

Αυτό μπορεί να συμβεί για παράδειγμα όταν ο χρήστης πατάει σε έναν σελιδοδείκτη (bookmark) ή γράφει κατευθείαν την URI διεύθυνση σε ένα νέο παράθυρο του φυλλομετρητή.

2.2 Ανάλυση κίνησης

Οι τεχνικές ανάλυσης της κίνησης στα αρχεία απογραφής, βασίζονται στην ιδέα ότι τα χαρακτηριστικά τα οποία επιδεικνύουν οι ιχνηλάτες σε αυτά τα αρχεία, διαφέρουν από αυτά των χρηστών. Αυτές οι τεχνικές βασίζονται είτε σε πρότυπα πλοήγησης και χαρακτηριστικά που είναι ήδη γνωστά για τη συμπεριφορά των ιχνηλατών, είτε σε υποθέσεις οι οποίες αποδείχνονται ότι ισχύουν όταν οι τεχνικές αυτές ανιχνεύουν τους ιχνηλάτες με μεγάλο ποσοστό επιτυχίας. Τα πρότυπα πλοήγησης εμπλέκουν αναζήτηση κατά βάθος ή κατά πλάτος, των ενσωματωμένων συνδέσμων και πόρων σε μια ιστοσελίδα. Τα χαρακτηριστικά συμπεριλαμβάνουν τα είδη των πόρων που ζητήθηκαν, το μέγεθος της αίτησης και τον τρόπο με τον όποιο έγινε η πλοήγηση σε μια ιστοσελίδα (κατά βάθος ή κατά πλάτος). Επιπλέον, οι περισσότεροι ιχνηλάτες έχουν προγραμματιστεί να κάνουν αιτήματα για συγκεκριμένο είδος πόρου (για παράδειγμα μερικοί ιχνηλάτες μπορεί να έχουν υλοποιηθεί να ζητάνε μόνο .pdf αρχεία). Για το λόγο ότι αυτές οι τεχνικές βασίζονται περισσότερο στην αναμενόμενη συμπεριφορά των ιχνηλατών παρά στην γνώση των χαρακτηριστικών που περιγράψαμε πιο πάνω, θεωρούνται πιο δυνατές τεχνικές για ανίχνευση ιχνηλατών. Παρόλα αυτά, αυτές οι τεχνικές ανίχνευσης δεν έχουν επιτυχία στη περίπτωση που ένας ιχνηλάτης συμπεριφέρεται διαφορετικά από τα αναμενόμενα πρότυπα.

2.2.1 Είδος αιτουμένου πόρου

Στο [8] οι συγγραφείς εκμεταλλεύονται την υπόθεση ότι οι ιχνηλάτες ζητούν μονό συγκεκριμένα είδη πόρων κατά την πλοήγηση τους σε μια ιστοσελίδα. Χρησιμοποιούν μια στατιστική προσέγγιση για την ανίχνευση των ιχνηλατών η οποία βασίζεται στα είδη των πόρων που ζητήθηκαν. Πιο συγκεκριμένα, υλοποιούν δυο νέους αλγορίθμους για ανίχνευση ιχνηλατών, ο καθένας από τους οποίους χρησιμοποιεί διαφορετικά χαρακτηριστικά των συνόδων των ιχνηλατών.

Στον πρώτο αλγόριθμο γίνεται στατιστική ανάλυση του είδους του πόρου που ζητήθηκε σε κάθε αίτημα. Αυτοί οι πόροι ομαδοποιούνται σε οκτώ διαφορετικές ομάδες: ιστοσελίδα (web page), έγγραφα (document), script, εικόνες (images), μουσική (music), βίντεο (video), download και όλα τα υπόλοιπα. Για την εφαρμογή των αλγορίθμων, κάθε εγγραφή στο αρχείο απογραφής ταξινομείται σε μια από τις ομάδες ανάλογα με τον πόρο που ζητήθηκε. Για παράδειγμα, ένα αίτημα για htm, asp, ή php θα ταξινομηθεί σαν ιστοσελίδα ενώ ένα αίτημα για jpg, gif, ή png θα ταξινομηθεί σαν εικόνα.

Ο αλγόριθμος αυτός λειτουργεί σε πέντε βήματα. Στο πρώτο βήμα ομαδοποιούνται οι εγγραφές των αρχείων απογραφής σε 8 ομάδες με βάση το είδος του πόρου που αφορούν. Στο δεύτερο βήμα, σαρώνονται σε όλες τις ομάδες τα πεδία για το όνομα κόμβου (hostname) και τον πράκτορα χρήστη (user agent). Αιτήματα σε μια ομάδα τα οποία έχουν το ίδιο όνομα κόμβου και πράκτορα χρήστη θεωρούνται ότι προέρχονται από τον ίδιο χρήστη. Αν ο χρόνος ανάμεσα σε δυο συνεχόμενα αιτήματα είναι μεγαλύτερος από ένα κατώφλι (15-30 λεπτά) τότε τα αιτήματα θεωρούνται ότι ανήκουν σε διαφορετικές συνόδους. Με αυτόν τον τρόπο χωρίζονται οι σύνοδοι σε κάθε ομάδα. Στο τρίτο βήμα, αν όλα τα αιτήματα σε μια σύνοδο ζητάνε το ίδιο είδος

πόρου τότε αυτή η συνοδός θεωρείται ότι δημιουργήθηκε από ιχνηλάτη. Αυτό βασίζεται στην υπόθεση ότι οι ιχνηλάτες κάνουν αιτήματα για συγκεκριμένο είδος πόρου. Στο τέταρτο βήμα ελέγχονται τα τρία πρώτα ψηφία της IP διεύθυνσης και του πεδίου πράκτορα χρήστη για κάθε σύνοδο που θεωρείται ότι προέρχεται από ιχνηλάτη, και αυτά που συμφωνούν θεωρούνται ότι ανήκουν στον ίδιο ιχνηλάτη. Κατά το τελευταίο βήμα μετριέται ο αριθμός των συνόδων της κάθε ομάδας που προέρχονται από τον ίδιο ιχνηλάτη και ο αντίστοιχος αριθμός επισκέψεων. Αυτά τα δεδομένα χρησιμοποιούνται για την αναγνώριση συνόδων που ζητάνε ένα μοναδικό είδος πόρου και περιέχουν σημαντικό αριθμό αιτήσεων. Με βάση αυτές τις μετρικές ορίζονται τα κατώφλια για να ξεχωρίσουν αιτήσεις που προέρχονται από χρήστες οι όποιοι κάνουν αίτηση για ένα μόνο είδος πόρου και στη συνεχεία δεν κάνουν άλλες αιτήσεις. Αυτές οι σύνοδοι δημιουργούνται όταν χρήστες κάνουν αναζήτηση στο διαδίκτυο για ένα συγκεκριμένο πόρο στέλνοντας μόνο λίγα αιτήματα για να πάρουν απευθείας τα αποτελέσματα χωρίς να πλοηγούνται ξεκινώντας από την αρχική σελίδα του ισοτόπου. Αν ο αριθμός των αιτημάτων σε μια σύνοδο ξεπερνά ένα ορισμένο κατώφλι τότε η σύνοδος ταξινομείται ως δημιουργημένη από ιχνηλάτη.

Ο δεύτερος αλγόριθμος λαμβάνει υπόψη την ταχύτητα με την οποία τα αιτήματα στέλνονται στον εξυπηρετητή και τον αριθμό όλων των ενσωματωμένων συνδέσμων σε μια ιστοσελίδα. Η ιδέα εδώ είναι να δημιουργηθεί για κάθε επισκέπτη μια λίστα με όλα τα αιτήματα και τους αντίστοιχους χρόνους που στάλθηκαν τα αιτήματα. Αν η διαφορά του χρόνου σε όλα τα αιτήματα είναι μικρότερη από ένα κατώφλι (συνήθως 0-30 δευτερόλεπτα) τότε η σύνοδος ταξινομείται ως ιχνηλάτης. Επιπλέον, αν ένας επισκέπτης μιας ιστοσελίδας δεν έχει κάνει αιτήσεις και για όλα τα ενσωματωμένα αντικείμενα (π.χ. εικόνες) της σελίδας τότε αυτός θεωρείται ότι μπορεί να είναι ιχνηλάτης. Ο αλγόριθμος υποθέτει ότι η σύνοδος ενός χρήστη θα

ξεκινήσει με ένα αίτημα για μια κύρια ιστοσελίδα και θα επακολουθήσουν αιτήματα για όλους τους ενσωματωμένους πόρους της ιστοσελίδας.

Οι δυνατότητες ανίχνευσης των δυο αλγορίθμων συγκρίνονται με το ίδιο σύνολο δεδομένων και όλοι οι ιχνηλάτες που ανιχνεύονται με τον πρώτο αλγόριθμο ανιχνεύονται και με το δεύτερο. Αυτό επιβεβαιώνει την υπόθεση των συγγραφέων ότι οι ιχνηλάτες χαρακτηρίζονται από το είδος των πόρων που αναζητούν, από τον χρόνο που μεσολαβεί ανάμεσα σε δυο διαδοχικά αιτήματα και από το ποσοστό των ενσωματωμένων πόρων που ζητήθηκαν σε μια ιστοσελίδα. Τα πειραματικά αποτελέσματα δείχνουν ότι οι αλγόριθμοί τους ανιχνεύουν όλους τους ηθικούς ιχνηλάτες δηλαδή όλους τους ιχνηλάτες οι όποιοι ζητάνε το αρχείο robot.txt. Σχετικά με την ορθότητα των αλγορίθμων, από τους 253 επισκέπτες υποψήφιους για ιχνηλάτες, οι 28 κατηγοριοποιήθηκαν ως ιχνηλάτες όταν η τιμή του κατωφλίου για τον αριθμό των συνόδων αποτελούμενες από αιτήματα για ένα συγκεκριμένο πόρο είναι μεγαλύτερη του 2 και ο αντίστοιχος αριθμός αιτημάτων για κάθε σύνοδο είναι μεγαλύτερος του 5. Οι συγγραφείς όρισαν αυτές τις τιμές με βάση την υπόθεση ότι ένας ιχνηλάτης χωρίζει τις δουλειές του σε υποκατηγορίες δημιουργώντας έτσι περισσότερες συνόδους από έναν χρήστη. Μια άλλη υπόθεση που κάνουν οι συγγραφείς είναι ότι οι ιχνηλάτες κατεβάζουν παραπάνω υλικό παρά τους χρήστες. Από τους 28 ιχνηλάτες οι 20 ήταν ηθικοί (well behaved) ενώ οι υπόλοιποι 8 δεν θα είχαν ανιχνευτεί για το λόγο ότι δεν είχαν ζητήσει το αρχείο robot.txt.

Η επιτυχία του δευτέρου αλγορίθμου επιβεβαιώνει ότι, όταν ένας ιχνηλάτης λαμβάνει μια http απόκριση από τον εξυπηρετητή για μια ιστοσελίδα δεν ακολουθεί μια σειρά από αποκρίσεων για όλα τα ενσωματωμένα αντικείμενα αυτής της ιστοσελίδας. Αυτή η συμπεριφορά είναι κοινή για ιχνηλάτες οι οποίοι ενδιαφέρονται

στην αναζήτηση μόνο ενός είδους πόρου ή έχουν συγκεκριμένους στόχους αναζήτησης.

2.2.2 Πρότυπα πλοήγησης

Μια άλλη προσπάθεια για εξεύρεση των ιχνηλατών στηρίζεται στην υπόθεση ότι τα πρότυπα πλοήγησης τους διαφέρουν σημαντικά από αυτά των χρηστών. Στο [3] προτείνεται ένας νέος αλγόριθμος για την εξαγωγή συνόδων από τα αρχεία απογραφής με βάση την πιο πάνω υπόθεση.

Για να αντιστοιχηθεί μια αίτηση στην σύνοδο στην οποία ανήκει, χωρίζεται η λίστα με τις ενεργές συνόδους σε τέσσερις ομάδες με βάση την IP διεύθυνση και το πεδίο του πράκτορα χρήστη. Στην πρώτη ομάδα συμπεριλαμβάνονται οι σύνοδοι οι οποίες έχουν την ίδια IP διεύθυνση και τον ίδιο πράκτορα χρήστη με αυτά της εγγραφής. Στην δεύτερη ομάδα συμπεριλαμβάνονται οι σύνοδοι οι οποίες έχουν τον ίδιο πράκτορα χρήστη με αυτά της εγγραφής και η IP διεύθυνση ανήκει στο ίδιο domain. Η τρίτη ομάδα περιέχει τις συνόδους με τον ίδιο πράκτορα χρήστη και το ίδιο πρόθεμα IP διεύθυνσης. Στην τελευταία ομάδα συμπεριλαμβάνονται οι σύνοδοι που δεν έχουν κανένα κοινό πεδίο. Στη συνέχεια υπολογίζονται διάφορα χαρακτηριστικά για την κάθε σύνοδο χωρίζοντας τις συνόδους σε επεισόδια, όπου κάθε επεισόδιο είναι ένα αίτημα για ένα HTTP αρχείο. Οι συγγραφείς υπολογίζουν 25 διαφορετικά χαρακτηριστικά πλοήγησης και χρησιμοποιούν τα τρία από αυτά για το labeling των συνόδων. Σε αυτά τα χαρακτηριστικά συμπεριλαμβάνονται ο έλεγχος αν το αρχείο robot.txt έχει ζητηθεί, το ποσοστό των αιτημάτων που έγιναν με την μέθοδο HEAD και το ποσοστό των αιτημάτων που δεν είχαν καμιά εγγραφή στο πεδίο της αναφοράς (unsigned referrer). Ο λόγος που χρησιμοποιούν αυτά τα χαρακτηριστικά για το

labeling είναι ότι αντιπροσωπεύουν πιο πολύ την συμπεριφορά ενός ιχνηλάτη παρά ενός χρήστη.

Όλα τα πεδία του πράκτορα χρήστη χωρίζονται σε ομάδες γνωστών ιχνηλατών, γνωστών φυλλομετρητών, πιθανών ιχνηλατών και πιθανών φυλλομετρητών με τον παρακάτω τρόπο. Αν μια σύνοδος περιλαμβάνει μια αίτηση για το αρχείο robots.txt, τότε η σύνοδος θεωρείται ως ιχνηλάτης. Ειδάλλως γίνεται ανάλυση των πεδίων πράκτορα χρήστη των αιτήσεων. Αν η σύνοδος έχει αιτήσεις μόνο από ένα πράκτορα χρήστη ο οποίος είναι γνωστός ιχνηλάτης ή πιθανός ιχνηλάτης τότε η σύνοδος κατηγοριοποιείται ως ιχνηλάτης. Διαφορετικά κατηγοριοποιείται ως άνθρωπος. Αν η σύνοδος έχει αιτήσεις από πολλούς πράκτορες χρήστη τότε κατηγοριοποιείται ως ιχνηλάτης μόνο (α) αν δεν υπάρχουν σύνοδοι που να είναι γνωστοί φυλλομετρητές ή πιθανοί φυλλομετρητές ή (β) αν όλες οι αιτήσεις της συνόδου χρησιμοποιούν τη μέθοδο HEAD HTTP ή το πεδίο της αναφοράς είναι άδειο. Τέλος, η τεχνική υιοθετεί τον αλγόριθμο δέντρου αποφάσεων C4.5 στις κατηγοριοποιημένες συνόδους χρησιμοποιώντας όλα τα παραγόμενα πρότυπα πλοήγησης. Ο σκοπός είναι η ανάπτυξη ενός καλού μοντέλου για πρόγνωση συνόδων ιχνηλατών βασισμένο μόνο στα χαρακτηριστικά πρόσβασης και η ανίχνευση της κίνησης ιχνηλατών όσο το δυνατό συντομότερα κατά την επίσκεψη ενός ιχνηλάτη σε ένα ιστοτόπο. Η ακρίβεια αυτού του μοντέλου είναι 90% μετά από τέσσερεις αιτήσεις. Το μοντέλο είναι επίσης ικανό να ανιχνεύει ιχνηλάτες οι οποίοι δεν αυτοπροσδιορίζονται χρησιμοποιώντας παρόμοια πρότυπα πλοήγησης με άλλους ιχνηλάτες.

Η επιτυχία αυτού του μοντέλου συνεπάγεται ότι η μέθοδος HTTP HEAD δεν χρησιμοποιείται σχεδόν ποτέ από τους ανθρώπους. Επιπλέον οι συγγραφείς κάνουν την σημαντική παρατήρηση ότι οι ιχνηλάτες πλοηγούνται κατά πλάτος σε ένα ιστοτόπο και λιγότερο κατά βάθος. Αντίθετα οι άνθρωποι πλοηγούνται οδηγούμενοι

από τους στόχους και σε μια σύνοδο, διασχίζουν τον ιστοτόπο περισσότερο κατά βάθος.

2.3 Αναλυτική μοντελοποίηση

Η τυχαιότητα στην συμπεριφορά ενός ιχνηλάτη και η πιθανοτική κατανομή της κίνησης του εξυπηρετητή, οδηγούν στη δημιουργία αναλυτικών μοντέλων για μια καλύτερη περιγραφή της συμπεριφοράς των ιχνηλατών. Οι αναλυτικές τεχνικές στηρίζονται στις τεχνικές ανάλυσης κίνησης κτίζοντας μοντέλα για την αναπαράσταση των χαρακτηριστικών που επιδεικνύουν οι ιχνηλάτες κατά την διάρκεια των επισκέψεών τους. Οι αναλυτικές τεχνικές μπορούν να κατηγοριοποιηθούν σύμφωνα με το μοντέλο που υλοποιούν και τα χαρακτηριστικά των ιχνηλατών που λαμβάνουν υπόψη. Τα παρατηρούμενα χαρακτηριστικά των αρχείων απογραφής ενός εξυπηρετητή, μπορούν να οριστούν χρησιμοποιώντας διάφορες στατιστικές οι οποίες ξεχωρίζουν την συμπεριφορά ενός χρήστη από αυτήν ενός ιχνηλάτη. Αυτές οι τεχνικές οδηγούν σε πιο συγκεκριμένους και πολυ-διάστατους τρόπους ανίχνευσης ιχνηλατών και θεωρούνται πιο δυνατές από τις τεχνικές ανάλυσης της κίνησης στους εξυπηρετητές. Ένα σωστά εκπαιδευμένο μοντέλο διασφαλίζει ότι η ανίχνευση οποιουδήποτε χρήσιμου ιχνηλάτη (δηλαδή ενός ιχνηλάτη που δεν έχει τυχαία συμπεριφορά) θα είναι επιτυχής. Ένα μειονέκτημα αυτών των τεχνικών όμως, είναι ότι χρειάζονται πολλά δεδομένα για την εκπαίδευση μοντέλων με μεγάλο ποσοστό επιτυχίας.

2.3.1 Η Bayesian Προσέγγιση

Στο [2] παρουσιάζεται μια Μπεισιανή προσέγγιση για την ανίχνευση ιχνηλατών στην οποία αρχικά εξάγουν τις συνόδους ομαδοποιώντας τις εγγραφές με βάση την IP διεύθυνση και τον χρόνο που μεσολαβεί ανάμεσα σε δυο διαδοχικά αιτήματα. Σε αυτή την προσέγγιση χρησιμοποιείται ένας δυναμικός εξωχρονισμός (timeout) ο οποίος προσαρμόζεται ανάλογα με τον αριθμό αιτήσεων σε μια σύνοδο. Στη συνέχεια υπολογίζονται τα ακόλουθα χαρακτηριστικά για την κάθε σύνοδο: μέγιστος ρυθμός κλικ, διάρκεια, το ποσοστό των εικόνων που ζητήθηκαν, το Ποσοστό των αιτήσεων για pdf/ps αρχεία, το ποσοστό των αποκρίσεων με κωδικό 4xx και μια τιμή 0 ή 1 ανάλογα αν ζητήθηκε το αρχείο robot.txt. Αυτά τα χαρακτηριστικά αποτελούν τους κόμβους του δικτύου Bayes και η ρίζα αποτελεί την κλάση (ιχνηλάτης/άνθρωπος) στην οποία ανήκει μια σύνοδος. Συνδυάζοντας όλα τα πιο πάνω χαρακτηριστικά υπολογίζεται μια πιθανότητα για την κλάση στην οποία ανήκει η σύνοδος. Επίσης η τεχνική αυτή λαμβάνει υπόψη ότι η κλάση μιας συνόδου επηρεάζει άμεσα τα παρατηρητέα χαρακτηριστικά της συνόδου.

Το δίκτυο έχει εκπαιδευτεί με ένα μεγάλο σύνολο δεδομένων αποτελούμενο από χιλιάδες συνόδους. Για την κατηγοριοποίηση των συνόδων σε ανθρώπους και ιχνηλάτες χρησιμοποιείται μια ημιαυτόματη μέθοδος η οποία βασίζεται σε ευρετικές συναρτήσεις. Πιο συγκεκριμένα μια σύνοδος κατηγοριοποιείται σαν ιχνηλάτης αν: (α) η διεύθυνση ανήκει στις IP διευθύνσεις γνωστών ιχνηλατών, (β) αν υπάρχει αίτηση για το robot.txt αρχείο (γ) αν η διάρκεια της συνόδου είναι μεγαλύτερη από 3 ώρες ή (δ) αν το ποσοστό των ζητούμενων εικόνων σε σχέση με τα HTML αρχεία είναι λιγότερο από 1 στα 10. Τα χαρακτηριστικά τα οποία παίρνουν μη-διακριτές τιμές,

όπως παράδειγμα η διάρκεια της συνόδου, έχουν μετατραπεί σε διακριτές τιμές χρησιμοποιώντας την εντροπία.

Οι συγγραφείς έκαναν πέντε διαφορετικά πειράματα χρησιμοποιώντας κάθε φορά διαφορετικό αριθμό συνόδων ιχνηλατών και ανθρώπων για την εκπαίδευση του δικτύου. Για την αξιολόγηση των αποτελεσμάτων ορίζουν τη μεταβλητή *Recall* για να μετρήσουν το ποσοστό των συνόδων ιχνηλατών οι οποίες έχουν κατηγοριοποιηθεί σωστά σε σχέση με τον πραγματικό αριθμό συνόδων ιχνηλατών. Τα αποτελέσματα δείχνουν ότι η κατηγοριοποίηση έχει *Recall* 95% ποσοστό επιτυχίας στην καλύτερη περίπτωση (όταν για την εκπαίδευση του δικτύου χρησιμοποιούν τον ίδιο αριθμό συνόδων από ιχνηλάτες και ανθρώπους) και 80% στην χειρότερη (όταν χρησιμοποιούν μεγαλύτερο αριθμό συνόδων ανθρώπων παρά ιχνηλατών).

Τα αποτελέσματα επιβεβαιώνουν ότι οι ιχνηλάτες αντιπροσωπεύονται από χαρακτηριστικά όπως: (1) ο μέγιστος ρυθμός των κλικ είναι μεγαλύτερος από αυτόν των ανθρώπων, (2) η σύνοδος διαρκεί πολύ περισσότερο, (3) ένα μεγάλο ποσοστό των αιτημάτων είναι για pdf/ps αρχεία, και τέλος (4) δημιουργούν ένα μεγάλο ποσοστό αποκρίσεων με κωδικό σφάλματος 4xx.

2.3.2 Κρυμμένο Μαρκοβιανό μοντέλο (ΚΜΜ)

Στο [9] παρουσιάζεται μια τεχνική ανίχνευσης βασισμένη σε ένα κρυμμένο μαρκοβιανό μοντέλο στην οποία διακρίνονται οι ιχνηλάτες από τους ανθρώπους με βάση τα πρότυπα αφίξεων των αιτήσεων. Υποστηρίζεται ότι ένας ανθρώπινος επισκέπτης σε μια ιστοσελίδα φαίνεται από ένα burst αιτήσεων HTTP από τον φυλλομετρητή για όλους τους ενσωματωμένους πόρους στην ιστοσελίδα, ακολουθούμενο από μια αδρανή περίοδο ενόσω ο χρήστης διαβάζει την σελίδα. Ένας

ιχνηλάτης από την άλλη μεριά, αποστέλλει αιτήσεις για πόρους με πιο χαμηλό ρυθμό και με σταθερό χρόνο ανάμεσα στις αιτήσεις. Για να περιγράψουν καλύτερα αυτό το φαινόμενο, οι συγγραφείς διαχωρίζουν το χρόνο σε διακριτά διαστήματα του ίδιου μήκους. Μια ή περισσότερες αιτήσεις από τον ίδιο χρήστη που καταφθάνουν στο ίδιο διάστημα χρόνου ονομάζεται batch arrival. Τέτοιες αφίξεις είναι χαρακτηριστικές για τους ανθρώπινους χρήστες παρά για τους ιχνηλάτες. Για να ανιχνεύσει τα batch arrivals, η τεχνική μετρά τον αριθμό των αιτήσεων σε μια συγκεκριμένη διάρκεια χρόνου. Ένα batch arrival ορίζεται ως μια ομάδα αιτήσεων σε ένα χρονικό διάστημα. Κάθε χρονικό διάστημα αντιπροσωπεύει μια παρατήρηση για ένα ΚΜΜ, αλλά μόνο αυτά με batch arrival λαμβάνονται υπόψη.

Για την εκπαίδευση του μοντέλου προτύπων πλοήγησης των ιχνηλατών, το οποίο αντιπροσωπεύεται από ένα ΚΜΜ, οι συγγραφείς χρησιμοποιούν προηγούμενες παρατηρούμενες ακολουθίες ιχνηλατών. Στη συνέχεια χρησιμοποιούν εισερχόμενες αιτήσεις για να προσδιορίσουν το ενδεχόμενο ότι οι αιτήσεις αυτές προέρχονται από ιχνηλάτες. Οι υπολογισμοί γίνονται με τον αλγόριθμο forward-backward.

Τα αρχεία απογραφής χωρίζονται σε συνόδους λαμβάνοντας υπόψη ένα αδρανές χρονικό διάστημα μεγαλύτερο των 30 λεπτών ως το τέλος της συνόδου. Από αυτές τις συνόδους διαλέγουν ένα τυχαίο δείγμα το οποίο αποτελείται από 500 συνόδους ιχνηλατών και 500 συνόδους ανθρώπων για τον έλεγχο του μοντέλου ΚΜΜ. Οι σύνοδοι αυτές συνενώνονται και στη συνέχεια εφαρμόζεται το μοντέλο στις αιτήσεις για την ανίχνευση των ιχνηλατών. Το μοντέλο τους ανιχνεύει τους ιχνηλάτες με ποσοστό 97.6% με μόνο 2% false positive.

Σαν συμπέρασμα παρατηρείται ότι οι ιχνηλάτες οι όποιοι ανιχνεύονται εύκολα κάνουν περίπου 1.5 αιτήματα σε ένα χρονικό διάστημα των 15 δευτερόλεπτων. Τα

αποτελέσματα επιβεβαιώνουν τον ισχυρισμό τους ότι οι ιχνηλάτες στέλνουν αιτήματα με ένα πιο σταθερό ρυθμό σε σύγκριση με τους ανθρώπους.

3. Κεφάλαιο 3

Ανάλυση αρχείων απογραφής

3.1 Εισαγωγή

Για να επιτευχθεί ο κύριος στόχος της εργασίας που είναι η ανίχνευση των ιχνηλατών σε πραγματικό χρόνο, εκτός από την ανάπτυξη και υλοποίηση μιας μεθόδου για την ανίχνευση πρέπει να γίνουν και κάποια άλλα βήματα. Αυτά τα βήματα περιλαμβάνουν την προεπεξεργασία των αρχείων απογραφής, την εξαγωγή των συνόδων, την κατηγοριοποίηση τους σε ιχνηλάτες και ανθρώπους, τον υπολογισμό καινούριων χαρακτηριστικών και τέλος την στατιστική ανάλυση αυτών των χαρακτηριστικών. Αυτά είναι απαραίτητα ώστε στο τέλος να δημιουργηθεί ένα χρήσιμο εργαλείο το οποίο να μπορεί να ανιχνεύει τους ιχνηλάτες με μεγάλο ποσοστό επιτυχίας.

Σε αυτό το κεφάλαιο περιγράφουμε τα πιο πάνω βήματα. Αρχικά δίνουμε μια σύντομη περιγραφή των αρχείων απογραφής και των πεδίων των HTTP αιτημάτων. Στη συνέχεια εισάγουμε την έννοια της συνόδου και περιγράφουμε τον αλγόριθμο τον οποίο εφαρμόζουμε για την εξαγωγή των συνόδων από τα αρχεία απογραφής. Ακλουθεί η περιγραφή της μεθόδου που χρησιμοποιούμε για την ταξινόμηση των συνόδων σε ιχνηλάτες ή ανθρώπους. Στη συνέχεια περιγράφουμε τα καινούρια χαρακτηριστικά τα οποία υπολογίζονται στις συνόδους όπως αυτά διαχωρίστηκαν από τον ταξινομητή. Τέλος, παρουσιάζουμε την στατιστική ανάλυση που κάναμε με το στατιστικό πακέτο SPSS και τα συμπεράσματα αυτής της ανάλυσης.

3.2 Αρχεία Απογραφής (Server logs)

Οι εξυπηρετητές, καθώς εξυπηρετούν τις εισερχόμενες αιτήσεις δημιουργούν παράλληλα τα αρχεία απογραφής. Κάθε εγγραφή του αρχείου αντιπροσωπεύει μια αίτηση-απάντηση προς και από τον εξυπηρετητή. Η εγγραφή αυτή περιλαμβάνει έναν αριθμό από πεδία που αφορούν πληροφορίες για τον πελάτη και τον εξυπηρετητή.

Τα αρχεία απογραφής κωδικοποιούνται σύμφωνα με το Common Log Format (CLF). Κάθε εγγραφή σε ένα CLF αποτελείται από τα πιο κάτω επτά πεδία[7]:

1. IP διεύθυνση ή όνομα του απομακρυσμένου κόμβου (Remote hostname)
2. Κωδικός πρόσβασης του απομακρυσμένου χρήστη (Remote login name)
3. Όνομα χρήστη όπως αυτός αυτοπροσδιορίζεται (Username)
4. Ημερομηνία (μμ/μμ/χχχχ:ΩΩ:λλ:δδ) (Date dd/mmm/yyyy:HH:mm:ss)
5. Το αίτημα του HTTP μηνύματος (HTTP request message)
6. Κωδικός απάντησης από τον εξυπηρετητή προς τον πελάτη (HTTP response code)
7. Μέγεθος απόκρισης σε bytes

Τα διάφορα πεδία στα αρχεία απογραφής χωρίζονται με κενό, ενώ αν κάποιο πεδίο δεν είναι διαθέσιμο τότε παριστάνεται από μια παύλα. Το Extended log file format περιλαμβάνει όλα τα πεδία που αναφέραμε πιο πάνω με δυο επιπλέον πεδία: τα Referer και User agent. Πιο κάτω φαίνονται παραδείγματα αρχείων απογραφής:

66.249.65.107 - - [08/Oct/2007:04:54:20 -0400] "GET /support.html HTTP/1.1" 200 11179

66.249.65.107 - - [08/Oct/2007:04:54:20 -0400] "GET /support.html HTTP/1.1" 200 11179 - "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"

Για το σκοπό της εργασίας χρησιμοποιήσαμε αρχεία απογραφής από τους παρακάτω εξυπηρετητές:

- Ένα σύνολο αρχείων απογραφής από τον εξυπηρετητή του Τμήματος της πληροφορικής (CS-UCY) του Πανεπιστημίου Κύπρου και ένα σύνολο από τον εξυπηρετητή όλου του πανεπιστημίου (CC-UCY). Τα αρχεία απογραφής του CS-UCY καταγράφουν την κίνηση του εξυπηρετητή για 176 μέρες και περιέχουν συνολικά 1,767,101 HTTP αιτήσεις. Οι εγγραφές του εξυπηρετητή CC-UCY καταγράφουν την κίνηση για 114 μέρες και περιέχουν 1,467,266 HTTP αιτήσεις.
- Το Ινστιτούτο της Επιστήμης της Πληροφορικής στην Ελλάδα, (Foundation of Research and Technology ICS-FORTH): τα αρχεία απογραφής αυτού του εξυπηρετητή καταγράφουν την κίνηση για 45 μέρες και περιέχουν συνολικά 2,724.074 HTTP αιτήσεις.
- Ο εξυπηρετητής του Εργαστηρίου Τεχνολογιών Λογισμικού του Μετσόβιου Πολυτεχνείου της Αθήνας (SL-NTUA): τα αρχεία απογραφής αυτού του εξυπηρετητή καταγράφουν την κίνηση για 58 μέρες και περιέχουν συνολικά 102,090 HTTP αιτήσεις.
- Το Πανεπιστήμιο της Πληροφορικής στο Toronto (CSE-TOR): ο εξυπηρετητής αυτού του τμήματος περιέχει συνολικά 2,56,5214 εγγραφές σε χρονικό διάστημα 42 ήμερες.
- Ο εξυπηρετητής του safeweb από το πανεπιστήμιο Κύπρου: περιέχει εγγραφές για 30 μέρες και συνολικά καταγράφει 1,56,6414 HTTP αιτήσεις.
- Τέλος, χρησιμοποιήσαμε αρχεία απογραφής από τους εξυπηρετητές του grid, mediinfo, webmail. Το χρονικό διάστημα αυτών των εγγραφών είναι από τις 20-12-2005 μέχρι 22-01-2006 και περιέχει συνολικά 1,42,12434 HTTP αιτήσεις.

3.3 Εύρεση συνόδων (Session identification)

Αυτό που θα περιγράψουμε σε αυτό το υποκεφάλαιο, είναι πως βρίσκουμε τις συνόδους από τα αρχεία απογραφής. Ένα αρχείο απογραφής αποτελείται από χιλιάδες εγγραφές με την κάθε εγγραφή να αποτελεί ένα HTTP αίτημα προς τον εξυπηρετητή, από κάποιον πράκτορα χρήστη. Τα αρχεία απογραφής είναι ταξινομημένα με βάση την ώρα που έφτασαν στον εξυπηρετητή. Κάθε αρχείο περιλαμβάνει έναν αριθμό από συνόδους και η κάθε σύνοδος αποτελείται από έναν αριθμό από αιτήματα τα οποία εξήχθηκαν από τον ίδιο χρήστη σε έναν συγκεκριμένο εξυπηρετητή. Μια σύνοδος τελειώνει όταν ο χρήστης ολοκληρώνει την πλοήγηση του σε μια ιστοσελίδα. Η εύρεση των συνόδων, είναι η διαδικασία που χωρίζει τις εγγραφές στο αρχείο απογραφής σε ξεχωριστές ομάδες αποτελούμενες από αιτήσεις οι οποίες πιστεύεται ότι έχουν έρθει από τον ίδιο χρήστη. Ο πιο απλός και συνηθισμένος τρόπος για την εύρεση των συνόδων είναι: πρώτα ομαδοποιούνται τα αιτήματα τα οποία προέρχονται από την ιδία IP διεύθυνση και χρησιμοποιώντας ένα εξωχρονισμό (timeout) χωρίζουμε τις εγγραφές σε ξεχωριστές συνόδους. Μια συνηθισμένη τιμή για το χρόνο που χρησιμοποιείται ως κατώφλι από διάφορες έρευνες οι οποίες ασχολούνται με εξόρυξη ιστού (web mining) είναι 30 λεπτά. Με την πιο πάνω μέθοδο (του εξωχρονισμού) μια σύνοδος ορίζεται ως μια σειρά από αιτήματα που εκδόθηκαν από έναν πράκτορα χρήστη, και ο χρόνος ανάμεσα σε δυο διαδοχικά αιτήματα είναι μικρότερος από ένα κατώφλι (threshold) [7]. Υπάρχει ένα μειονέκτημα σε αυτήν την μέθοδο το όποιο είναι η δυσκολία εύρεσης σωστής τιμής για το κατώφλι. Ο λόγος είναι ότι διαφορετικοί χρήστες παρουσιάζουν διαφορετική συμπεριφορά κατά την πλοήγησή τους στο διαδίκτυο.

Για το σκοπό της παρούσας εργασίας χρησιμοποιούμε μια διαφορετική τεχνική για την εξαγωγή των συνόδων από τα αρχεία απογραφής. Πιο συγκεκριμένα δεν χρησιμοποιούμε ένα σταθερό κατώφλι για την εισαγωγή της κάθε εγγραφής στην αντίστοιχη σύνοδο, αλλά ένα που προσαρμόζεται δυναμικά ανάλογα με τον αριθμό των αιτημάτων σε μια σύνοδο. Για παράδειγμα, για μια σύνοδο η οποία έχει r_{max} αιτήματα, ορίζουμε την τιμή του κατωφλίου t_1. Για συνόδους με περισσότερα αιτήματα από r_{max} αυξάνουμε την τιμή του κατωφλίου σε $t_2 > t_1$. Οι τιμές που χρησιμοποιούμε είναι 100 για το r_{max}, 30 λεπτά για το t_1 και 60 λεπτά για το t_2 [2].

Στην πιο πάνω μέθοδο και σε όλες τις μεθόδους οι οποίες βασίζονται στην IP διεύθυνση για την εξαγωγή των συνόδων, υπάρχει πιθανότητα να συμπεριλαμβάνουμε αιτήματα που προέρχονται από διαφορετικούς χρήστες, σε μια σύνοδο. Αυτό συμβαίνει όταν πολλοί χρήστες χρησιμοποιούν την ίδια IP διεύθυνση για την πλοήγηση στο διαδίκτυο, για παράδειγμα μέσω ενός πληρεξούσιου (proxy). Σε αυτήν την περίπτωση τα αιτήματα διαφόρων χρηστών γράφονται στον εξυπηρετητή με την ιδία IP διεύθυνση παρόλο που αυτά ανήκουν σε διαφορετικούς χρήστες. Υπάρχει επίσης μια αβεβαιότητα στη σωστή επιλογή του χρόνου για τον χωρισμό των αιτημάτων σε συνόδους για το λόγο ότι διαφορετικοί χρήστες συμπεριφέρονται διαφορετικά στην πλοήγηση τους [2].

Όπως αναφέραμε και στην εισαγωγή, ένας πρώτος στόχος αυτής της μελέτης, είναι η ανακάλυψη στατιστικών ιδιοτήτων των συνόδων που δημιουργούνται από τους ιχνηλάτες. Την ταξινόμηση της κάθε συνόδου στην αντίστοιχη κατηγορία (άνθρωπος ή ιχνηλάτης) την πραγματοποιούμε με τη βοήθεια του ταξινομητή. Με βάση αυτές τις ιδιότητες αναπτύσσουμε στην συνέχεια ένα εργαλείο για την έγκυρη ανίχνευση των ιχνηλατών, πριν τελειώσει η σύνοδος που έχει εγκατασταθεί μεταξύ ιχνηλάτη και εξυπηρετητή.

3.4 Διαχωρισμός των συνόδων σε ιχνηλάτες ή χρήστες με το δίκτυο Bayesian

3.4.1 Εισαγωγή

Το εργαλείο το οποίο χρησιμοποιούμε για την ταξινόμηση των συνόδων σε ιχνηλάτες ή χρήστες, ακολουθεί μια πιθανοτική συλλογιστική προσέγγιση. Πιο συγκεκριμένα το μοντέλο είναι ένα δίκτυο Bayesian το οποίο ταξινομεί αυτόματα συνόδους ως ιχνηλάτες ή ανθρώπους, συνδυάζοντας διάφορα χαρακτηριστικά, που έχουν αποδειχτεί προηγουμένως να αντιπροσωπεύουν την συμπεριφορά τους. Η προσέγγιση αυτή χρησιμοποιεί τεχνικές μηχανικής μάθησης για τον προσδιορισμό των παραμέτρων του μοντέλου. Τα αποτελέσματα της ταξινόμησης εκφράζονται με πιθανότητες οι οποίες προσδιορίζουν την τάξη που ανήκει μια σύνοδο[2].

3.4.2 Επισκόπηση του συστήματος

Το επόμενο βήμα μετά την εξαγωγή των συνόδων από τα αρχεία απογραφής είναι να προσδιορίζουμε ποιες συνόδους ανήκουν σε ιχνηλάτες και ποιες σε ανθρώπους. Για αυτόν τον σκοπό χρησιμοποιήσαμε ένα δίκτυο (Naïve Bayesian) το οποίο σε προγενέστερο στάδιο έχει εκπαιδευτεί για την εκμάθηση των παραμέτρων του. Το σύστημα αυτό συνδυάζει στοιχεία / χαρακτηριστικά τα οποία έχουν υπολογιστεί από τις συνόδους και η ταξινόμηση βασίζεται στη μέγιστη πιθανότητα δεδομένων των χαρακτηριστικών. Πιο κάτω περιγράφουμε τα χαρακτηριστικά των συνόδων τα οποία αποτελούν τις παραμέτρους του δικτύου Bayesian.

Μέγιστος συνεχιζόμενος ρυθμός κλικ (Maximum sustained click rate): Ένα κλικ αποτελεί μια αίτηση για ένα αρχείο HTML. Το χαρακτηριστικό αυτό αντιστοιχεί στο μέγιστο αριθμό αιτήσεων HTML που πετυχαίνονται σε ένα συγκεκριμένο χρονικό

πλαίσιο σε μια σύνοδο. Η λογική πίσω από αυτό το χαρακτηριστικό είναι ότι υπάρχει ένα όριο στο μέγιστο αριθμό κλικ που μπορεί να κάνει ένας χρήστης μέσα σε ένα συγκεκριμένο χρονικό πλαίσιο. Ο τρόπος με τον οποιο υπολογίζεται αυτό το χαρακτηριστικό είναι ο εξής: αρχικά ορίζεται ένα χρονικό πλαίσιο t και στη συνέχεια χρησιμοποιείται ένα κυλιόμενο παράθυρο για το συγκεκριμένο χρονικό πλαίσιο σε μια δεδομένη σύνοδο για να υπολογίσουμε το μέγιστο συνεχιζόμενο ρυθμό κλικ. Για παράδειγμα αν ορίζουμε τον χρόνο t=12 δευτερόλεπτα και βρούμε ότι ο μέγιστος αριθμός των κλικ μέσα στο κυλιόμενο παράθυρο σε μια σύνοδο είναι 36, συμπεραίνουμε ότι ο μέγιστος συνεχιζόμενος ρυθμός των κλικ είναι 3. Το κυλιόμενο παράθυρο ξεκινάει από την πρώτη HTML αίτηση της συνόδου και κρατάει το μέγιστο αριθμό αιτήσεων σε κάθε παράθυρο προχωρώντας κάθε φορά μια HTML αίτηση μέχρι το τέλος της συνόδου[2].

Διάρκεια συνόδου (duration of session): Αυτός είναι ο συνολικός χρόνος σε δευτερόλεπτα, από το πρώτο αίτημα HTTP μέχρι το τελευταίο σε μια σύνοδο. Γενικά οι σύνοδοι που δημιουργούνται από τους ιχνηλάτες διαρκούν περισσότερο από τις συνόδους που δημιουργεί ένας χρήστης του φυλλομετρητή. Επιπλέον η συμπεριφορά ενός χρήστη κατά την πλοήγηση του είναι πιο συγκεκριμένη και οδηγούμενη από στόχους σε αντίθεση με την πλοήγησή του ιχνηλάτη που παρουσιάζει μια άπληστη συμπεριφορά. Ακόμα, υπάρχει κάποιο όριο στο χρόνο που μπορεί ένας χρήστης να πλοηγείται στο διαδίκτυο.

Ποσοστό των ζητούμενων εικόνων σε μια σύνοδο (Percentage of image requests) : Αυτό το χαρακτηριστικό δηλώνει το συνολικό ποσοστό των εικόνων (αρχεία .jpg .gif κλπ) που ζητηθήκαν σε μια σύνοδο. Προηγούμενες έρευνες δείχνουν ότι οι ιχνηλάτες

δεν ζητάνε εικόνες. Σε αντίθεση, οι σύνοδοι που δημιουργούνται από χρήστες έχουν ένα μεγάλο ποσοστό από εικόνες. Ο λόγος είναι ότι, όταν ο χρήστης στέλνει ένα αίτημα HTTP για μια ιστοσελίδα, ο φυλλομετρητής στέλνει αυτόματα αιτήματα για όλα τα ενσωματωμένα αντικείμενα της ιστοσελίδας 2.

Ποσοστό των ζητούμενων pdf/ps σε μια σύνοδο (Percentage of pdf/ps requests) :
Το χαρακτηριστικό αυτό δηλώνει το συνολικό ποσοστό των αρχείων pdf/ps που ζητηθήκαν σε μια σύνοδο. Συνήθως οι ιχνηλάτες ζητούν αυτά τα έγγραφα σε μεγάλο ποσοστό σε σχέση με τους χρήστες.

Ποσοστό αποκρίσεων 4xx (percentage of 4xx error responses) : Τα αιτήματα των ιχνηλατών έχουν ως αποτέλεσμα ένα μεγάλο ποσοστό αποκρίσεων με κωδικό *4xx*. Αυτό δεν συμβαίνει με αιτήματα τα οποία έχουν σταλεί από χρήστες, για το λόγο ότι οι άνθρωποι είναι σε θέση να αναγνωρίζουν, να απομνημονεύουν και να αποφεύγουν σπασμένους υπερσυνδέσμους και μη διαθέσιμες πηγές.

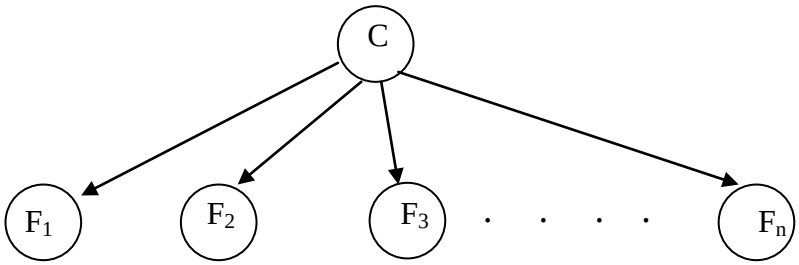
Αίτηση για το αρχείο robots.txt (robots.txt file reuests) Σκοπός της ύπαρξης του αρχείου *robots.txt* προτάθηκε για να επιτρέψει στους διαχειριστές διαδικτύου να ορίζουν ποιες ιστοσελίδες των ιστοτόπων, είναι προσβάσιμες από τους ιχνηλάτες. Σύμφωνα με αυτό το πρότυπο κάθε φορά που ένας ιχνηλάτης επισκέπτεται μια ιστοσελίδα, για παράδειγμα www.xyz.com πρέπει πρώτα να ελέγξει το αρχείο http://www.xyz.com/robots.txt. Αυτό το αρχείο περιέχει μια λίστα περιορισμών πρόσβασης του ιστοτόπου, όπως αυτή έχει καθοριστεί από το διαχειριστή του. Οι ιχνηλάτες μπορούν να ανιχνευτούν εύκολα όταν οι σύνοδοι περιέχουν ένα αίτημα για το αρχείο robots.txt. Αυτός είναι ένας λογικός τρόπος ανίχνευσης ιχνηλατών για το

λόγο ότι, οι περισσότεροι ιστοτόποι δεν περιέχουν εμφανές σύνδεσμο για αυτό το αρχείο και οι περισσότεροι χρήστες δεν γνωρίζουν την ύπαρξη του. Παρ'όλα αυτά δεν μπορούμε να στηριζόμαστε αποκλειστικά σε αυτό το κριτήριο επειδή η συμμόρφωση με το RES είναι προαιρετική και πολλοί ιχνηλάτες δεν το εφαρμόζουν.

3.4.3 Η δομή του δικτύου Bayesian

Το δίκτυο Bayesian είναι ένας κατευθυνόμενος ακυκλικός γράφος (Directed Acyclic Graph, DAG) όπου κάθε κόμβος αντιπροσωπεύει μια τυχαία μεταβλητή. Κάθε κόμβος περιλαμβάνει τις πιθανές καταστάσεις της μεταβλητής που αναπαριστά και ένα πίνακα με υποθετικές πιθανότητες ή γενικότερα μια συνάρτηση που υπολογίζει αυτές τις πιθανότητες. Ο πίνακας περιέχει τις πιθανότητες κάθε κατάστασης της μεταβλητής με δεδομένες τις καταστάσεις των κόμβων-γονέων (parent nodes). Οι ακμές αντιπροσωπεύουν σχέση αιτίας - αποτελέσματος (cause-effect) ανάμεσα στις συνδεδεμένες μεταβλητές και η κατεύθυνση των ακμών είναι από την αίτια στο αποτέλεσμα. Η αξιοπιστία του κάθε αποτελέσματος μοντελοποιείται σαν μια πιθανότητα [10].

Naïve Bayesian είναι μια ειδική περίπτωση των δικτύων Bayesian, όπου ένα μοναδικό αίτιο - η κλάση - επηρεάζει άμεσα έναν αριθμό από αποτελέσματα - τα χαρακτηριστικά - και η κλάση δεν έχει γονέα. Επιπλέον ένας ταξινομητής naive Bayes υποθέτει ότι, η ύπαρξη (ή η απουσία) ενός χαρακτηριστικού μιας δεδομένης κλάσης δεν επηρεάζει την ύπαρξη (ή την απουσία) οποιουδήποτε άλλου χαρακτηριστικού. Στο Σχήμα 3.1 φαίνεται ένα παράδειγμα αυτού του δικτύου. Τα F_1 F_2… F_n αναπαριστούν n χαρακτηριστικά και το f_i αντιπροσωπεύει την τιμή του Fι χαρακτηριστικού. Η μεταβλητή C αναπαριστά την τάξη και c είναι μια πιθανή τιμή (ετικέτα) αυτής της τάξης[10].



Σχήμα 3.1: Παράδειγμα δικτύου Naïve Bayesian

Το δίκτυο Bayesian που χρησιμοποιούμε έχει την δομή που φαίνεται στο Σχήμα 1. Κάθε κόμβος παιδί (child node) αντιπροσωπεύει ένα από τα χαρακτηριστικά που περιγράψαμε στο τμήμα 3.4.2. Ο κόμβος ρίζα αντιπροσωπεύει την τάξη, στην περίπτωση μας χρήστης (human) ή ιχνηλάτης (robot).

Όλοι οι κόμβοι του δικτύου μπορούν να περιγραφούν σε συντομία ως εξής:

- *Τάξη (class)*: Η ταξινόμηση των συνόδων. Αυτή η μεταβλητή παίρνει δυο τιμές: *χρήστης* ή *ιχνηλάτης* και αντιπροσωπεύει τον κόμβο ρίζα του δικτύου.
- *Κλικ (clicks):* Μέγιστος αριθμός αιτήσεων HTML που πετυχαίνονται σε ένα συγκεκριμένο χρονικό παράθυρο σε μια σύνοδο. Αυτή η μεταβλητή παίρνει τιμή από 0 μέχρι ένα μέγιστο αριθμό κλικ, ο οποίος ορίζεται με βάση την εκμάθηση των δεδομένων.
- *Διάρκεια (Duration):* Ο χρόνος σε δευτερόλεπτα ανάμεσα στο πρώτο και τελευταίο αίτημα (request). Η μεταβλητή παίρνει τιμή από 0 μέχρι ένα μέγιστο αριθμό διάρκειας, και αυτό ορίζεται με βάση την εκμάθηση των δεδομένων.
- *Εικόνες (images):* Το συνολικό ποσοστό των εικόνων (αρχεία .jpg .gif κλπ) που ζητήθηκαν σε μια σύνοδο. Η τιμή την οποία παίρνει αυτή η μεταβλητή βρίσκεται στο διάστημα 0 μέχρι 100.
- *PDF/PS:* Αυτός ο κόμβος δηλώνει το συνολικό ποσοστό των αρχείων pdf/ps που ζητήθηκαν σε μια σύνοδο και παίρνει τιμή από 0 μέχρι 100.

- *Ποσοστό αποκρίσεων 4xx:* Το συνολικό ποσοστό των αποκρίσεων με κωδικό 4xx σε μια συνοδό. Αυτή η μεταβλητή παίρνει τιμή από 0 μέχρι 100.
- *Αίτηση για το αρχείο robots.txt (robots.txt file requests):* Η μεταβλητή παίρνει δύο τιμές 0 και 1. Αν σε μια σύνοδο υπαρχή αίτηση για το αρχείο *robots.txt* η μεταβλητή παίρνει την τιμή 1 διαφορετικά παίρνει την τιμή 0.

3.5 Κατηγοριοποίηση

Μετά τον υπολογισμό των πιο πάνω χαρακτηριστικών, ταξινομούμε τις συνόδους σε ιχνηλάτες ή χρήστες με τον ταξινομητή (classifier). Για την ταξινόμηση της κάθε συνόδου σε ιχνηλάτη ή χρήστη, εξάγουμε το σύνολο των έξι ιδιοτήτων τα οποία χαρακτηρίζουν την συμπεριφορά του πελάτη (χρήστης ή ιχνηλάτης) και με αυτόν τον τρόπο προσδιορίζουμε τις μεταβλητές του δικτύου Bayesian. Ένα παράδειγμα διανύσματος για τα χαρακτηριστικά που περιγράψαμε πιο πάνω είναι: (15, 140, 65, 2, 0, 0). Σε αυτό το παράδειγμα ο μέγιστος συνεχιζόμενος ρυθμός κλικ (maximum sustained click rate) είναι 15, η διάρκεια της συγκεκριμένης συνόδου είναι 140 δευτερόλεπτα, το ποσοστό των αιτήσεων για εικόνες είναι 65%, 2% των αιτήσεων είχαν ζητήσει για pdf/ps αρχεία, το ποσοστό των αποκρίσεων με κωδικό σφάλματος μεγαλύτερο του 400 είναι 0%, και τέλος δεν υπάρχει αίτηση για το αρχείο robot.txt. Για το λόγο ότι το δίκτυο περιέχει μόνο διακριτές μεταβλητές, οι συνεχιζόμενες τιμές αντιστοιχίζονται σε διακριτές τιμές [2].

Ο Πίνακας 3.1 δείχνει συνοπτικά τους εξυπηρετητές από τους οποίους πήραμε τα αρχεία απογραφής, των αντίστοιχων αριθμών συνόδων, όπως επίσης και την ταξινόμηση των συνόδων σε ιχνηλάτες ή ανθρώπους. Τα αποτελέσματα στον πίνακα είναι όπως ταξινομήθηκαν από το δίκτυο Bayesian.

Access-Logs	Total Sessions	Human Sessions	Robot Sessions
Forthnet_access_log	30089	75%	25%
Data_set_access_log	26199	66%	34%
ntua-softlab_access_log	1047	54%	46%
safeweb_access_log	2445	80%	20%
Toronto_access_log	104116	87%	13%
UCY_access_log	37760	86%	14%
zeus-server_access_log	63179	50%	50%
Total	**264835**	**75%**	**25%**

Πίνακας 3.1: Αρχεία απογραφής από διάφορους εξυπηρετητές

3.6 Εύρεση νέων χαρακτηριστικών

Στην παράγραφο αυτή θα περιγράψουμε κάποια νέα χαρακτηριστικά [3, 9, 11, 12] τα οποία υπολογίσαμε στις συνόδους όπως αυτά έχουν ταξινομηθεί από το ταξινομητή. Ο λόγος για τον οποίο υπολογίζουμε νέα χαρακτηριστικά είναι για να ανακαλύψουμε νέες ιδιότητες που μπορεί να έχουν οι σύνοδοι που δημιουργούνται από τους ιχνηλάτες. Στόχος μας είναι να βρούμε ποια χαρακτηριστικά ξεχωρίζουν καλύτερα τους ιχνηλάτες από τους ανθρώπους. Πιο κάτω περιγράφουμε όλα τα χαρακτηριστικά που υπολογίζουμε για κάθε σύνοδο:

HEAD: Αυτό το χαρακτηριστικό δηλώνει το ποσοστό των αιτήσεων σε μια σύνοδο, τα οποία σταλθήκαν στον εξυπηρετητή ιστοτόπου με τη μέθοδο HEAD. Οι χρήστες μπορούν να χρησιμοποιούν τις μεθόδους HEAD και GET για την απόκτηση πληροφοριών. Χρησιμοποιώντας τη μέθοδο GET ο εξυπηρετητής επιστρέφει όλο το αντικείμενο ενώ με τη μέθοδο HEAD επιστρέφει μόνο πληροφορίες για το αντικείμενο. Οι οδηγίες σχεδιασμού ιχνηλατών, εισηγούνται τη χρήση της μεθόδου

HEAD, όπου αυτό είναι δυνατό. Ο λόγος είναι ότι, το αποτέλεσμα της απόκρισης του εξυπηρετητή για τη μέθοδο αυτή είναι πολύ πιο μικρό (σε μέγεθος byte) από αυτής της μεθόδου GET με αποτέλεσμα η λειτουργία ενός ιχνηλάτη να μην επιφέρει βαρύ φορτίο στο δίκτυο.

Ποσοστό αποκρίσεων 2xx (percentage of 2xx responses): Ποσοστό των αιτήσεων με κωδικό *2xx.* Έρευνες δείχνουν ότι οι σύνοδοι που ανήκουν σε χρήστες έχουν ένα μεγάλο ποσοστό αποκρίσεων με κωδικό *2xx.*

Ποσοστό αποκρίσεων 3xx (percentage of 3xx responses): Στις συνόδους των χρηστών έχει παρατηρηθεί ότι ένα ποσοστό των αιτήσεων έχουν κωδικό 304 το όποιο δηλώνει ότι το αντικείμενο δεν έχει τροποποιηθεί (not modified).

Ποσοστό αιτήσεων για σελίδες (Pages): Αυτό το χαρακτηριστικό δηλωνει το Ποσοστό των αιτήσεων για σελίδες: htm, html, php, jsp.

Night: Ποσοστό των αιτήσεων που έχουν γίνει κατά την διάρκεια της νύχτας. Σε αυτό το χαρακτηριστικό υπολογίζουμε αν οι σύνοδοι δημιουργήθηκαν μεταξύ ωρών 2am-8am τοπική ώρα.

AvgTime: Αυτό το χαρακτηριστικό δηλώνει το μέσο όρο του χρόνου ανάμεσα σε δυο διαδοχικά αιτήματα, σε μια σύνοδο. Πιο συγκεκριμένα υπολογίζουμε για την κάθε σύνοδο όλους τους ενδιάμεσους χρόνους αλλά μόνο για αιτήματα που αφορούν ανάκτηση σελίδων, δηλαδή htm, html, php, asp και στη συνέχεια βρίσκουμε το μέσο όρο.

StdDevTime: Όταν μία παράμετρος έχει μεγάλες αποκλίσεις, τότε ο μέσος όρος από μόνος του είναι μια παραπλανητική στατιστική, η οποία μπορεί να επηρεαστεί από ένα μικρό αριθμό από μεγάλες τιμές. Για το λόγο αυτό υπολογίζουμε για την κάθε σύνοδο και την απόκλιση ως μετρική για την ανάλυση των δεδομένων.

Ποσοστό αιτήσεων για αρχεία Zip, Multimedia, Ascii, Binary: Τέλος υπολογίζουμε για την κάθε σύνοδο το ποσοστό των αιτήσεων που ζήτησαν αρχεία *Zip, Multimedia, Ascii, Binary.*

3.7 Στατιστική ανάλυση με την βοήθεια του στατιστικού πακέτου SPSS

Όπως αναφέραμε και στην εισαγωγή στόχος αυτής της διπλωματικής είναι η υλοποίηση ενός συστήματος για την ανίχνευση ιχνηλατών σε πραγματικό χρόνο. Υπάρχει ένα αντιστάθμισμα στην υλοποίηση μεταξύ ορθότητας (accuracy) και πολυπλοκότητας του συστήματος. Χρειαζόμαστε ένα σύστημα το οποίο να ανιχνεύσει τους ιχνηλάτες με μεγάλο ποσοστό επιτυχίας και ταυτόχρονα να είναι όσο απλό γίνεται και να έχει ελάχιστες απαιτήσεις σε μνήμη και σε χρόνο και ως αποτέλεσμα να μην επιβαρύνεται η δουλειά του εξυπηρετητή. Για την επίτευξη αυτών των στόχων κάνουμε μια στατιστική ανάλυση των δεδομένων με σκοπό να βρούμε ποια από τα χαρακτηριστικά που υπολογίζουμε στην παράγραφο χαρακτηρίζουν περισσότερο τις συνόδους των ιχνηλατών/χρηστών.

Η ανάλυση στηρίχτηκε στην κατηγοριοποίηση του δικτύου Bayesian μόνο για τις συνόδους οι οποίες είχαν ποσοστό εμπιστοσύνης μεγαλύτερο του 80%. Συνολικός αριθμός των συνόδων είναι 270000 από τις οποίες 70000 έχουν κατηγοριοποιηθεί ως ιχνηλάτες και 180000 ως χρήστες. Παρακάτω θα περιγράψουμε τα αποτελέσματα της στατιστικής ανάλυσης καθώς και τα συμπεράσματα.

3.7.1 Στατιστική ανάλυση των δεδομένων

Η ανάλυση των χαρακτηριστικών έγινε με τη βοήθεια του στατιστικού πακέτου SPSS. Το SPSS (Statistical Package for the Social Sciences) είναι ένα από τα πιο πολύ

χρησιμοποιημένα πακέτα για στατιστική ανάλυση στις κοινωνικές επιστήμες. Ο λόγος που διαλέξαμε το SPSS για την ανάλυση των δεδομένων είναι ότι εκτός από ένα στατιστικό πακέτο, προσφέρει διαχείριση δεδομένων (case selection, file reshaping, creating derived data) και τεκμηρίωση δεδομένων (a metadata dictionary is stored with the data).

Στις παραγράφους που ακολουθούν θα αναφέρουμε τις διάφορες στατιστικές που τρέξαμε για την ανάλυση των χαρακτηριστικών μας.

QQPlot: Στη στατιστική, το **QQPlot** είναι μια γραφική μέθοδος για την αναγνώριση διαφορών ανάμεσα σε μια κατανομή πιθανοτήτων ενός στατιστικού πληθυσμού από τον οποίο έχει παρθεί ένα τυχαίο δείγμα, και μια συγκριτική κατανομή. Ένα παράδειγμα του είδους των διάφορων που μπορούν να ελεγχθούν είναι η κανονικότητα της κατανομής του πληθυσμού. Για το δικό μας σκοπό ελέγξαμε για όλα τα χαρακτηριστικά των συνόδων, αν ακολουθούν κάποια από τις γνωστές κατανομές. Το QQPlot έχει δείξει ότι κανένα από τα χαρακτηριστικά που ελέγξαμε δεν ακολουθεί κάποια από τις γνωστές κατανομές.

Means: Αυτό το στατιστικό μας δίνει μια πρώτη εικόνα για τα χαρακτηριστικά που διαφέρουν περισσότερο ανάμεσα στις δυο κατηγορίες.

Crosstabs: Στο SPSS το crosstabs είναι μια διαδικασία η οποία cross-tabulates δυο μεταβλητές, δείχνοντας την σχέση μεταξύ τους σε μορφή πίνακα. Σε αντίθεση με την κατανομή συχνοτήτων η οποία παρέχει την κατανομή μόνο μιας μεταβλητής, ο contingency πίνακας περιγράφει ταυτόχρονα την κατανομή δυο ή περισσότερων μεταβλητών.

Τα crosstabs τα τρέξαμε για όλα μας τα χαρακτηριστικά και τα αποτελέσματα δείχνουν ότι οι δυο ομάδες ιχνηλάτη/άνθρωπος δεν διαφέρουν στα περισσότερα από

αυτά. Στη συνέχεια θα περιγράψουμε τα αποτελέσματα των χαρακτηριστικών στα οποία οι δυο ομάδες έχουν διαφορετική κατανομή.

Στον πίνακα που ακολουθεί βλέπουμε το ποσοστό των αιτήσεων για σελίδες (htm, html, php, jsp) που υπήρχαν στις συνόδους των ιχνηλατών και των χρηστών. Η μεταβλητή "Percentage of pages" έχει τρεις κατηγορίες ποσοστών: 0-50%, 50-80%, 80-100%. Στην μεταβλητή "Category" έχουμε τις δυο κατηγορίες: ιχνηλάτης και άνθρωπος. Μπορούμε να συμπεραίνουμε εύκολα από τον πίνακα, ότι το χαρακτηριστικό το οποίο δηλώνει το ποσοστό των αιτήσεων για σελίδες, σε μια σύνοδο διαφέρει πολύ στις δυο ομάδες. Για παράδειγμα, 90% των συνόδων που δημιουργήθηκαν από ανθρώπους έχουν το πολύ 50% αιτήσεις για σελίδες. Το ποσοστό των σελίδων σε μια σύνοδο θα είναι σημαντικό χαρακτηριστικό για την υλοποίηση του μοντέλου για ανίχνευσης ιχνηλατών.

Percentage of HTML requests in session	Category	
	Robot	Human
80-100%	50%	1%
50-80%	30%	9%
0-50%	20%	90%

Πίνακας 3.2: Ποσοστό αιτήσεων για σελίδες στις κατηγορίες άνθρωπος/ιχνηλάτης

Τα αποτελέσματα για το χαρακτηριστικό "Image percentage" φαίνονται στο Πίνακα 3.2. Στον πίνακα φαίνονται οι δυο κατηγορίες robot/human και η μεταβλητή "Image percentage" η οποία έχει δυο κατηγορίες ποσοστών 0-10% και 10-100%. Βλέπουμε από τον πίνακα ότι οι ιχνηλάτες δημιουργούν συνόδους οι οποίες έχουν ποσοστό αιτήσεων για εικόνες λιγότερο από 10%. Το χαρακτηριστικό αυτό αποτελεί επίσης σημαντικό στοιχείο για την ανιχνεύσει των ιχνηλατών σε πραγματικό χρόνο.

Percentage of images requests in session	Category	
	Robot	Human
10-100%	2%	10%
0-10%	98%	90%

Πίνακας 3.3: Ποσοστό αιτήσεων για εικόνες στις κατηγορίες άνθρωπος/ιχνηλάτης

Ένα άλλο χαρακτηριστικό στο οποίο οι δυο ομάδες έχουν διαφορετική κατανομή, είναι η απόκριση του εξυπηρετητή με κωδικό σφάλματος 4XX. Τα αποτελέσματα στο Πίνακα 3.4, δείχνουν ότι στις συνόδους των χρηστών το ποσοστό των αποκρίσεων με κωδικό σφάλματος 4XX δεν ξεπερνά το 20%.

Percentage of 4XX requests in session	Category	
	Robot	Human
20-100%	35%	3%
0-20%	65%	97%

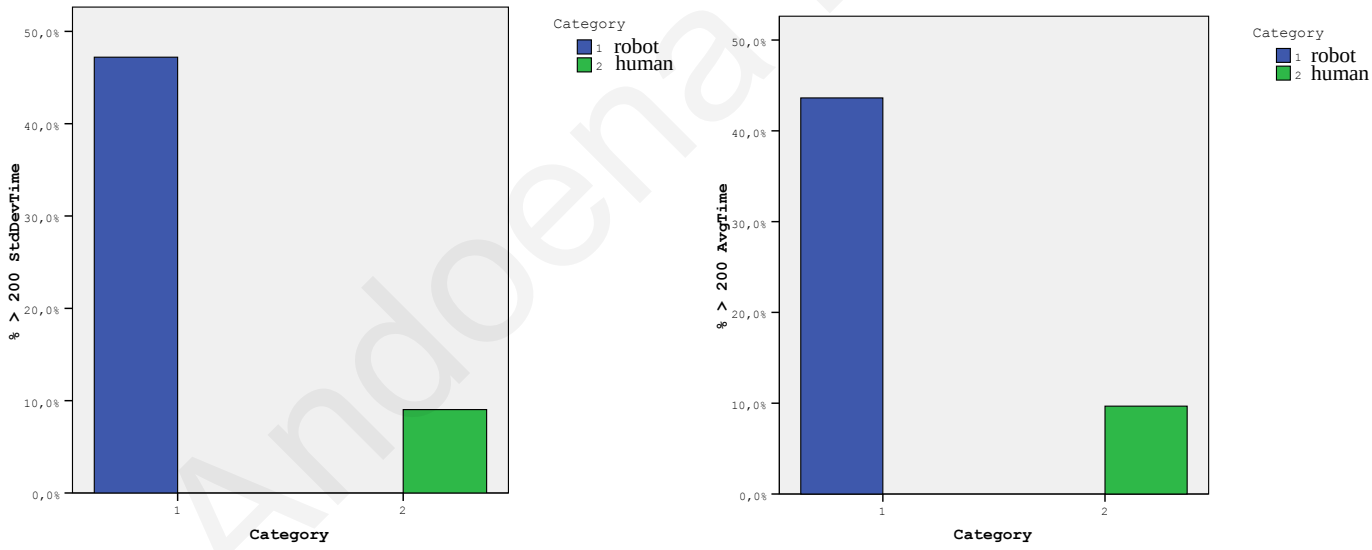
Πίνακας 3.4: Ποσοστό αποκρίσεων με κωδικό 4XX στις κατηγορίες άνθρωπος/ιχνηλάτης

Το "Maximum click rate" είναι ακόμα ένα σημαντικό χαρακτηριστικό το οποίο έχει διαφορετική κατανομή στις 2 κατηγορίες. Τα αποτελέσματα της ανάλυσης αυτού του χαρακτηριστικού δείχνουν ότι ο μέγιστος ρυθμός κλικ που μπορεί να έχουν οι σύνοδοι των χρηστών είναι 10.(Πίνακας 3.5)

Maximum click rate in session	Category	
	Robot	Human
<10	40%	99%
10>	60%	1%

Πίνακας 3.5: Μέγιστος ρυθμός των κλικ στις κατηγορίες άνθρωπος/ιχνηλάτης

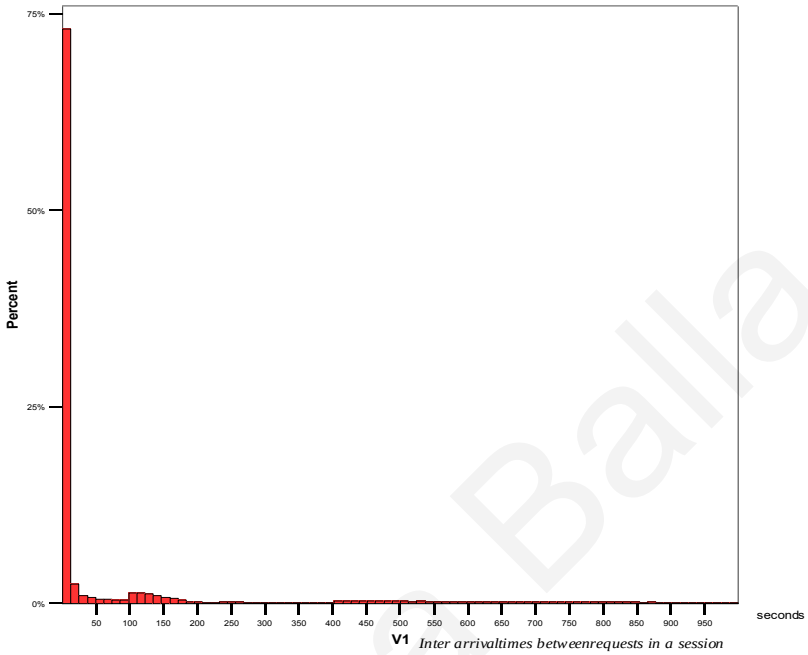
Τέλος, αναλύοντας δυο άλλα χαρακτηριστικά: το μέσο όρο του χρόνου ανάμεσα σε δυο διαδοχικά αιτήματα HTML και την τυπική απόκλιση αυτών των χρόνων φτάνουμε σε ένα σημαντικό συμπέρασμα όσον αφόρα το χρόνο (timeout) που διαλέγουμε για την εξαγωγή των συνόδων. Όπως φαίνεται και από τις πιο κάτω γραφηκές παραστάσεις περίπου 50% των συνόδων δημιουργημένες από ιχνηλάτες, έχουν μέσο όρο περισσότερο των 200 δευτερόλεπτων. Επίσης το stdDeviation στις περισσότερες συνόδους (45%) είναι μεγαλύτερο από 200. Ενώ περιμέναμε να δούμε ότι οι ιχνηλάτες στέλνουν τα αιτήματα τους με ένα συγκεκριμένο ρυθμό ο οποίος μπορεί να είναι αργός αλλά σταθερός, τα αποτελέσματα της ανάλυσης είναι πολύ διαφορετικά.



Σχήμα 3.2: Μέσος όρος του χρόνου ανάμεσα σε δυο διαδοχικά αιτήματα HTML

Για να εξηγήσουμε αυτά τα αποτελέσματα κάναμε μια άλλη ανάλυση για το χαρακτηριστικό *inter arrival times between requests in a session* (Σχήμα). Από την γραφική παράσταση φαίνεται ότι οι ιχνηλάτες κάνουν το επόμενο αίτημα σε λιγότερο από 50 δευτερόλεπτα και στη συνέχεια μπορεί να κάνουν αιτήσεις σε άλλους εξυπηρετητές και να επιστρέφουν. Άρα οι παραπάνω τιμές για το μέσο όρο και την

τυπική απόκλιση εξηγούνται από τον μεγάλο χρόνο που έχουμε ορίσει εμείς (1800 δευτερόλεπτα) για την εξαγωγή των συνόδων.



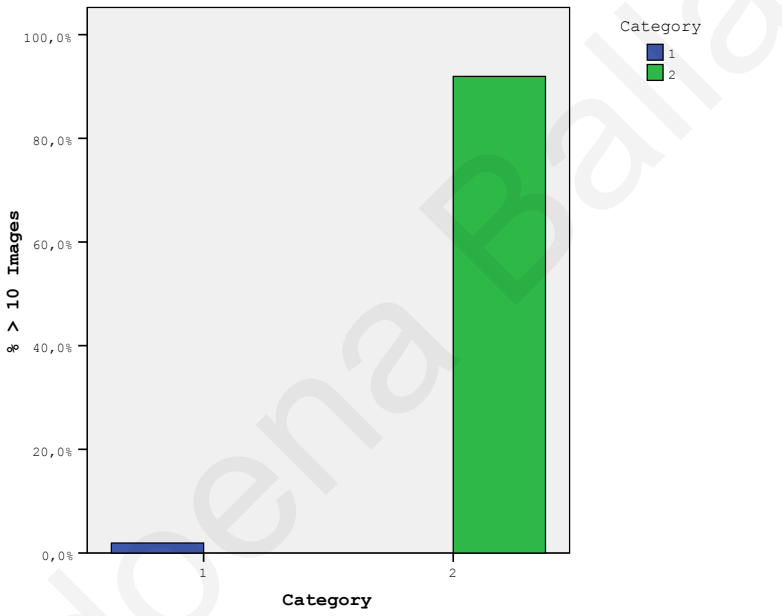
Σχήμα 3.3 Χρόνος ανάμεσα σε δυο διαδοχικά αιτήματα HTML

3.8 Συμπεράσματα

Σε αυτό το κεφάλαιο περιγράψαμε αναλυτικά όλη την ανάλυση των αρχείων απογραφής. Ένα πρώτο βήμα ήταν μια απλή προεπεξεργασία των αρχείων απογραφής για την αφαίρεση άκυρων εγγραφών. Ακλούθησε η περιγραφή του αλγορίθμου για την εξαγωγή των συνόδων. Στη συνέχεια χωρίσαμε τις συνόδους σε ιχνηλάτες και ανθρώπους με την βοήθεια του δικτύου Naïve Bayesian. Υπολογίσαμε επιπλέον χαρακτηριστικά στις συνόδους των δυο κατηγοριών. Τέλος κάναμε τη στατιστική ανάλυση όλων των χαρακτηριστικών με το SPSS. Η ανάλυση αυτή έδειξε ότι στα περισσότερα χαρακτηριστικά, η συμπεριφορά των ιχνηλατών και των ανθρώπων είναι σχεδόν η ίδια με μόνο κάποιες αμελητέες διαφορές. Τα πιο σημαντικά

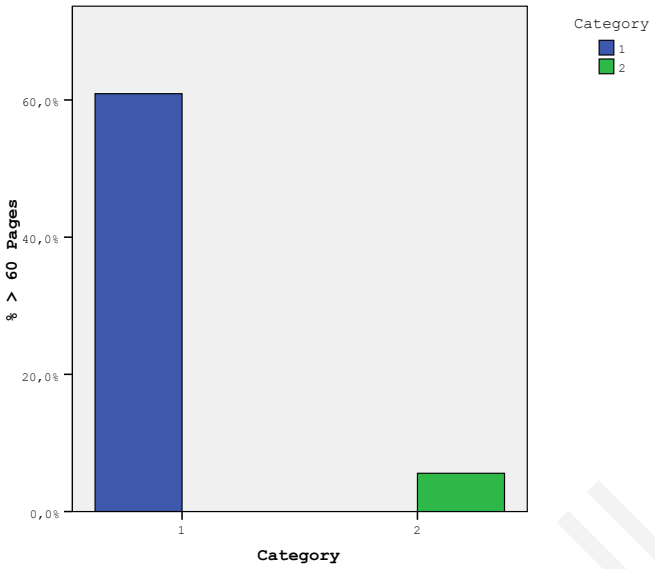
χαρακτηριστικά τα οποία έχουν διαφορετική κατανομή στις δυο κατηγορίες είναι τα παρακάτω:

Ποσοστό αιτήσεων για εικόνες(% of images) (Σχήμα. 3.4) Όπως φαίνεται και από την γραφική παράσταση, όλες οι σύνοδοι που έχουν δημιουργηθεί από τους χρήστες έχουν τουλάχιστον 10% αιτήσεις για εικόνες, ενώ ένα πολύ μικρό ποσοστό (1%) των συνόδων των ιχνηλατών έχουν ζητήσει εικόνες.



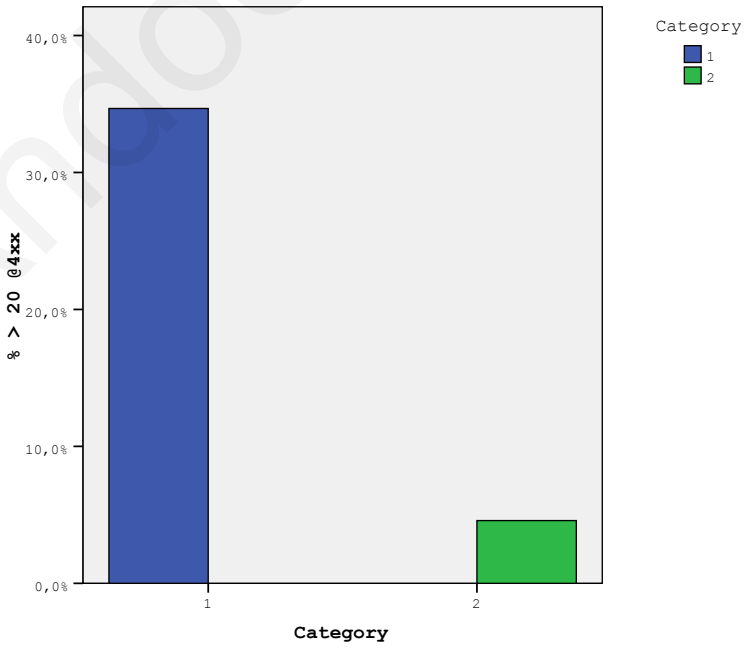
Σχήμα 3.4 Ποσοστό των ζητούμενων εικόνων στις συνόδους

Ποσοστό αιτήσεων για σελίδες (% of html, htm, php, jsp) (Σχήμα. 3.5) Το ποσοστό των αιτήσεων για σελίδες στις συνόδους των χρηστών είναι μικρότερο του 60% ενώ στις συνόδους των ιχνηλατών το Ποσοστό αυτό είναι μεγαλύτερο του 60%.



Σχήμα 3.5: Ποσοστό των ζητούμενων σελίδων στις συνόδους

Ποσοστό αποκρίσεων με κωδικο4xx (% of 4xx response code) (Σχήμα 3.6) Το ποσοστό των αποκρίσεων με κωδικό σφάλματος 4xx στις συνόδους των ανθρώπων δεν ξεπερνάει το 20%.



Σχήμα 3.6: Ποσοστό των αποκρίσεων με κωδικό σφάλματος 4XX

Χρόνος για το χωρισμό των ερχόμενων αιτημάτων σε συνόδους Ο χρόνος για την δημιουργία των συνόδων δεν χρειάζεται να είναι παραπάνω των 120 δευτερολέπτων. Η ανάλυση των χρόνων ανάμεσα σε δυο διαδοχικά αιτήματα για σελίδες (html, htm, php, asp) έδειξε ότι στις 75% των περιπτώσεων, οι ιχνηλάτες κάνουν το επόμενο αίτημα σε λιγότερο από 50 δευτερόλεπτα.

Τέλος, ο μέγιστος ρυθμός των κλικ στους ανθρώπους δεν ξεπερνά τα 10 κλικ το λεπτό ενώ στις συνόδους των ιχνηλατών το μέγιστο κλικ είναι περισσότερο από 10 κλικ το δευτερόλεπτο.

Τα πιο πάνω χαρακτηριστικά αποτελούν σημαντικά στοιχεία για την υλοποίηση του μοντέλου για την ανίχνευση των ιχνηλατών σε πραγματικό χρόνο. Στο επόμενο κεφάλαιο θα περιγράψουμε το μοντέλο το οποίο υλοποιήσαμε.

4. Κεφάλαιο 4

Ανίχνευση ιχνηλατών σε πραγματικό χρόνο

4.1 Εισαγωγή

Το σημαντικότερο μέρος αυτής της διατριβής είναι η ανάπτυξη και η υλοποίηση ενός συστήματος για την ανίχνευση των ιχνηλατών σε πραγματικό χρόνο. Τα όσα έχουν ήδη περιγραφεί στο προηγούμενο κεφάλαιο, όπως ο διαχωρισμός των αρχείων απογραφής σε συνόδους, η ταξινόμηση τους σε ιχνηλάτες ή ανθρώπους με τον ταξινομητή (δίκτυο Naïve Bayesian), ο υπολογισμός των χαρακτηριστικών των συνόδων και η ανάλυση των χαρακτηριστικών με το στατιστικό πακέτο SPSS, είναι απαραίτητες προϋποθέσεις για την ανάπτυξη του συστήματος.

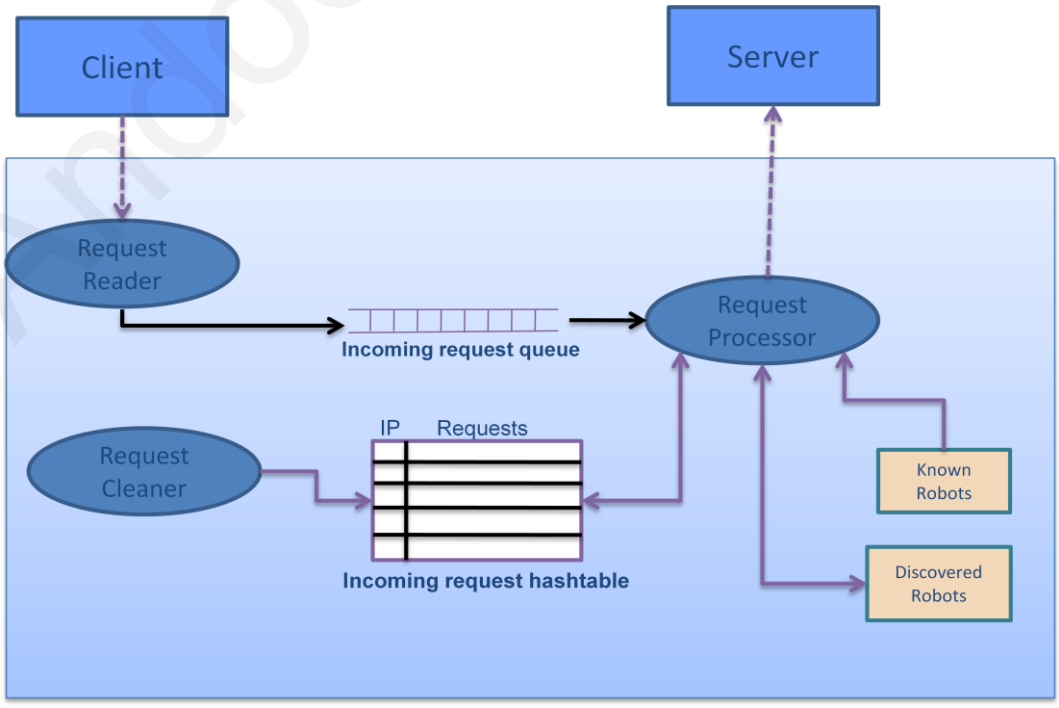
Έχουμε αναπτύξει και υλοποιήσει ένα σύστημα το όποιο δρα σαν ένα ενδιάμεσο φίλτρο ανάμεσα στον πελάτη και στον εξυπηρετητή. Αυτό, μπορεί να υλοποιηθεί ως ένα πληρεξούσιο ή ως ένα plug-in για τον εξυπηρετητή.

Η γλώσσα προγραμματισμού Java έχει χρησιμοποιηθεί για την υλοποίηση του συστήματος. Το σύστημα λαμβάνει εισερχόμενες αιτήσεις http και τις επεξεργάζεται βασισμένο στα χαρακτηριστικά τα οποία περιγράψαμε στο προηγούμενο κεφάλαιο. Στη συνέχεια το σύστημα αποφασίζει αν οι εισερχόμενες αιτήσεις ανήκουν σε

ιχνηλάτη και στη περίπτωση που συμβαίνει αυτό, καταγράφει την IP διεύθυνση σε μια λίστα με ανιχνευμένους ιχνηλάτες.

4.2 Αρχιτεκτονική του συστήματος

Το σύστημα αποτελείται από τρία συστατικά μέρη το κάθε ένα από τα οποία τρέχει σε ένα διαφορετικό νήμα (Σχήμα 4.1). Αυτά είναι το *Request Reader*, το *Request Processor* και το *Request Cleaner*. Το *Request Reader* είναι υπεύθυνο για να διαβάζει τις εισερχόμενες αιτήσεις HTTP τις οποίες στέλνει ο πελάτης. Στη συνέχεια τοποθετεί αυτές τις αιτήσεις σε μια ουρά FIFO (first in first out), την *Incoming request queue*. Το *Request Processor* είναι το πιο σημαντικό συστατικό μέρος του συστήματος, το όποιο περιέχει όλη την λογική που χρειάζεται για να διαχωρίζει τις εισερχόμενες αιτήσεις σε ιχνηλάτες ή ανθρώπους. Είναι υπεύθυνο για να διαβάζει τις αιτήσεις από την ουρά και να τις επεξεργάζεται με τη χρήση ενός δέντρου αποφάσεων, με σκοπό να αποφασίζει αν μια αίτηση προέρχεται από ιχνηλάτη.



Σχήμα 4.1: Αρχιτεκτονική του συστήματος

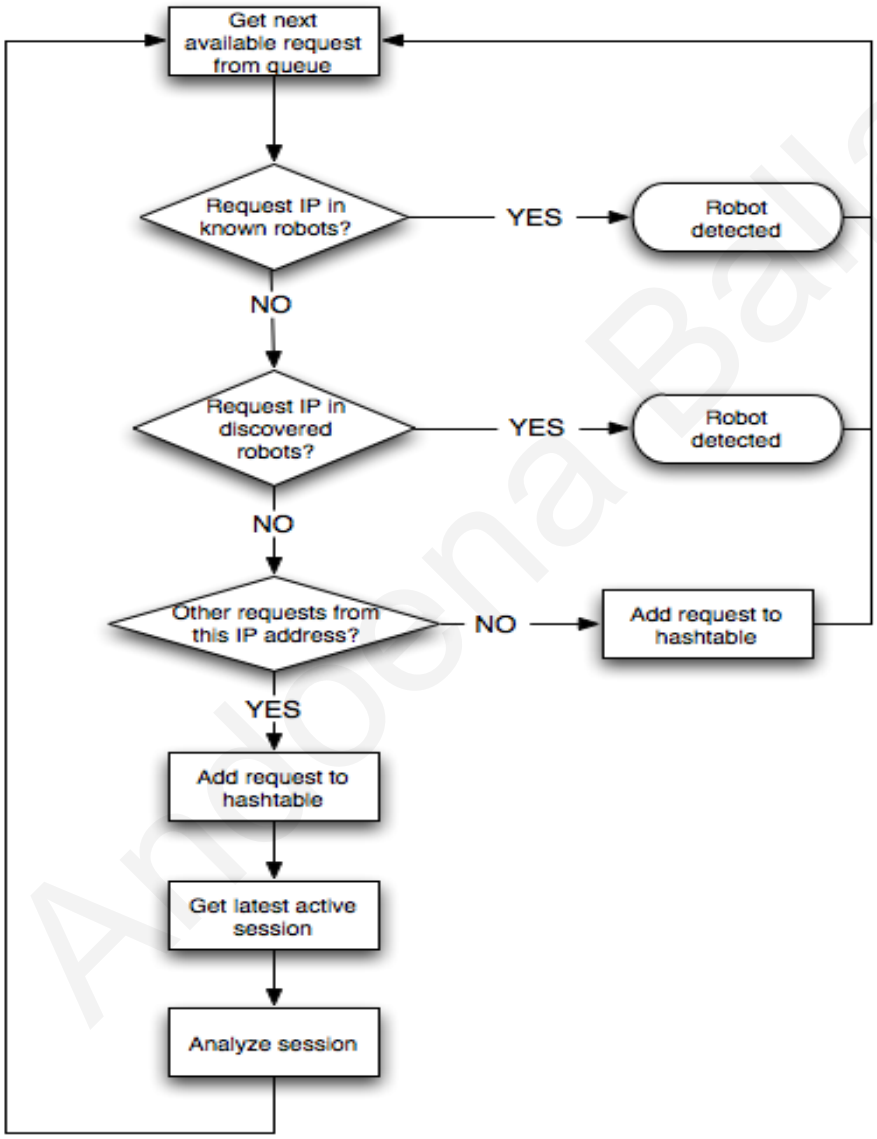
Για να μπορεί να εφαρμοστεί ο αλγόριθμος αποφάσεων, είναι απαραίτητο να υπάρχει ένας ελάχιστος αριθμός από αιτήσεις για την συγκεκριμένη διεύθυνση IP. Μέχρι να συμβεί αυτό, το νήμα τοποθετεί τις εισερχόμενες αιτήσεις σε ένα hashtable (Incoming request hash table).

Οι αιτήσεις τοποθετούνται στο hashtable ομαδοποιημένες με βάση την IP διεύθυνση. Το *Known Robots* είναι ένα αρχείο το οποίο περιέχει μια λίστα με IP διευθύνσεις, οι οποίες είναι γνωστό ότι ανήκουν σε ιχνηλάτες. Στο *Discovered Robots* βρίσκονται όλες οι IP διευθύνσεις οι οποίες έχουν ταξινομηθεί από το *Request Processor* σαν ιχνηλάτες. Τέλος το *Request Cleaner* είναι υπεύθυνο για να καθαρίσει από την μνήμη όλες τις αιτήσεις οι οποίες δεν είναι πλέον ενεργές.

4.3 Request Processor

Το σύστημα αποφασίζει αν οι εισερχόμενες αιτήσεις προέρχονται από ένα ιχνηλάτη με βάση τα ευρήματα της στατιστικής ανάλυσης την οποία περιγράψαμε στο προηγούμενο κεφάλαιο. Το υπεύθυνο συστατικό μέρος του συστήματος το οποίο εξεργάζεται τις εισερχόμενες αιτήσεις και αποφασίζει την τάξη (ιχνηλάτης/άνθρωπος) στην οποία ανήκουν είναι το *Request Processor*. Το διάγραμμα ροής αυτού του νήματος φαίνεται στο Σχήμα 4.2. Όπως βλέπουμε και από το σχήμα, το *Request Processor* διαβάζει την πρώτη αίτηση η οποία βρίσκεται στην ουρά (Incoming request queue) και στην συνέχεια κάνει δυο ελέγχους. Ο πρώτος έλεγχος αφορά το αν η IP διεύθυνση από την οποία εκδόθηκε η συγκεκριμένη αίτηση βρίσκεται στην λίστα με τους γνωστούς ιχνηλάτες (Known Robots). Αν αυτό ισχύει, τότε την προσθέτει στην λίστα με τους ανιχνευμένους ιχνηλάτες (Discovered Robots) και συνεχίζει με την

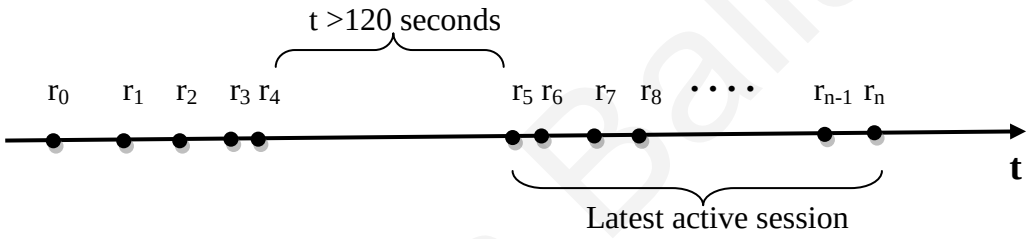
επόμενη αίτηση που βρίσκεται στην ουρά. Αν δεν βρει την IP διεύθυνση στη λίστα των γνωστών ιχνηλατών τότε ελέγχει αν βρίσκεται η IP διεύθυνση στη λίστα "Discovered Robots". Αν ναι, τότε συνεχίζει με την επόμενη αίτηση, διαφορετικά ελέγχει αν έχει έρθει άλλη αίτηση από τη συγκεκριμένη IP διεύθυνση. Αν βρει ότι αυτή είναι η πρώτη αίτηση τότε απλώς την προσθέτει στο hashtable.



Σχήμα 4.2: Διάγραμμα ροής του νήματος "Request Processor"

Αν υπάρχει στο hashtable και άλλη αίτηση από την ίδια IP διεύθυνση τότε ανακτάται η τελευταία ενεργή σύνοδος από την συγκεκριμένη διεύθυνση. Η τελευταία ενεργή

σύνοδο υπολογίζεται με τον εξής τρόπο (σχήμα 4.3): Θεωρούμε ότι έχουμε n+1 αιτήσεις από αυτήν την IP διεύθυνση, r_0 μέχρι r_n και οι αιτήσεις αυτές είναι ταξινομημένες με βάση τη χρονική στιγμή αφιξής τους ξεκινώντας από την πιο παλιά μέχρι την πιο πρόσφατη αίτηση. Ξεκινώντας από την τελευταία αίτηση r_n ελέγχουμε αν η διαφορά του χρόνου ανάμεσα σε αυτήν την αίτηση, με την προηγούμενη της είναι μικρότερη από 2 λεπτά. Αν είναι, τότε αυτές οι αιτήσεις ανήκουν στην ιδία ενεργή σύνοδο. Αυτό συνεχίζεται μέχρι να βρούμε διαφορά χρόνου της τρέχουσας αίτησης με την προηγούμενη της είναι μεγαλύτερη από 2 λεπτά.



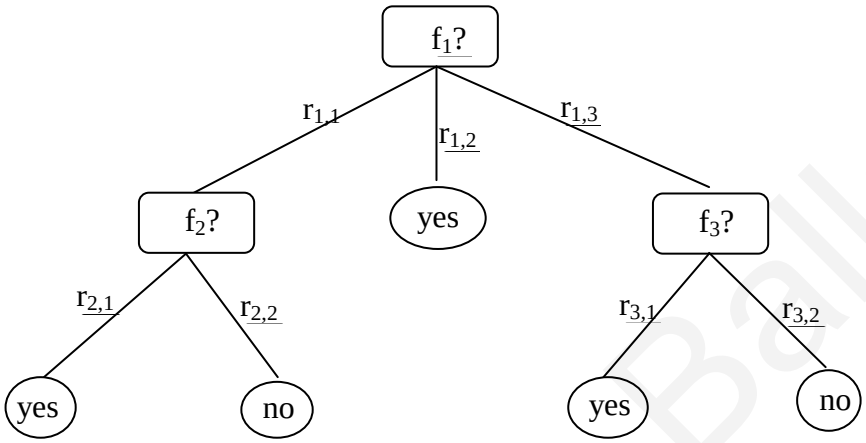
Σχήμα 4.3: Τελευταία ενεργή συνοδός

Στη συνέχεια το *Request Processor* αναλύει την τελευταία ενεργή σύνοδο. Η ανάλυση της ενεργής συνόδου πραγματοποιείται χρησιμοποιώντας ένα δέντρο αποφάσεων. Στο επόμενο υποκεφάλαιο περιγράφουμε συνοπτικά τι είναι τα δέντρα αποφάσεων και στη συνέχεια δίνουμε μια λεπτομερή περιγραφή της ανάλυσης της ενεργής συνόδου.

4.3.1 Ταξινόμηση με δέντρα αποφάσεων

Ένα δέντρο αποφάσεων είναι μια δεντρική δομή που μοιάζει με ένα διάγραμμα ροής, όπου ο κάθε εσωτερικός κόμβος αντιπροσωπεύει έναν έλεγχο ως προς ένα χαρακτηριστικό, το κάθε κλαδί αντιπροσωπεύει ένα αποτέλεσμα του ελέγχου, και κάθε φύλο δείχνει την ετικέτα της κλάσης (class label). Παράδειγμα ενός δέντρου

αποφάσεων φαίνεται στο Σχήμα 4.4. Οι εσωτερικοί κόμβοι του δέντρου αναπαριστούνται από ορθογώνια, και τα φύλλα αναπαριστούνται από οβάλ. Κάποιοι αλγόριθμοι δέντρων αποφάσεων παράγουν μόνο δυαδικά δέντρα, το οποίο σημαίνει ότι κάθε εσωτερικός κόμβος έχει μόνο δυο κλαδιά, ενώ άλλοι αλγόριθμοι παράγουν μη-δυαδικά δέντρα.



Σχήμα 4.4: Παράδειγμα δέντρου αποφάσεων

Τα δέντρα αποφάσεων χρησιμοποιούνται για ταξινόμηση με τον εξής τρόπο: Δεδομένης μιας πλειάδας **X**, της οποίας η κλάση στην οποία ανήκει είναι άγνωστη, ελέγχονται τα χαρακτηριστικά της σε κάθε κόμβο του δέντρου. Με αυτόν τον τρόπο δημιουργείται ένα μονοπάτι το οποίο ξεκινάει από την ρίζα του δέντρου και τελειώνει σε ένα φύλλο. Το φύλλο δηλώνει την κλάση στην οποία ανήκει το **X.**

Υπάρχουν διάφοροι λόγοι για τους οποίους τα δέντρα αποφάσεων χρησιμοποιούνται πολύ στην ταξινόμηση. Η δημιουργία ταξινομητών με δέντρα αποφάσεων δεν απαιτεί ιδιαίτερη γνώση για εξόρυξη δεδομένων και είναι κατάλληλη για εξερεύνηση και ανακάλυψη γνώσεων. Επίσης, τα δέντρα αποφάσεων μπορούν να διαχειριστούν πολυδιάστατα δεδομένα . Επιπλέον, η εκμάθηση και η δημιουργία των δέντρων αποφάσεων είναι απλή και πολύ γρήγορη. Τέλος, τα δέντρα αποφάσεων έχουν πολύ καλή ακρίβεια στην ταξινόμηση [10].

Στην επόμενη παράγραφο θα περιγράψουμε τον αλγόριθμο ID3 τον οποίο χρησιμοποιούμε για την εκμάθηση του δέντρου αποφάσεων, και στην συνέχεια για την ταξινόμηση των συνόδων σε ανθρώπους ή ιχνηλάτες. Κατά τη δημιουργία του δέντρου αποφάσεων, χρησιμοποιούμε μετρικές για επιλογή χαρακτηριστικών, τα οποία βρήκαμε από την στατιστική ανάλυση ότι ξεχωρίζουν "καλύτερα" τις δυο ομάδες ιχνηλάτες/ανθρώπους.

4.3.2 Decision Tree Induction

Ο αλγόριθμος ID3 υιοθετεί μια άπληστη προσέγγιση, στην οποία τα δέντρα αποφάσεων χτίζονται με μια αναδρομική τεχνική, διαίρει και βασίλευε, από πάνω προς τα κάτω. Ο αλγόριθμος ξεκίνα με ένα σύνολο από πλειάδες μαζί με την αντίστοιχη κλάση στην οποία ανήκουν. Το σύνολο αυτό, το οποίο χρησιμοποιείται για την εκπαίδευση του δέντρου χωρίζεται αναδρομικά σε μικρότερα υποσύνολα κατά το χτίσιμο του δέντρου (Σχήμα 4.5).

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition D.
Input:
- Data partition, D, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly either a *split point* or *splitting subset*.

Output:
A decision tree.
Method:
(1) create a node N;
(2) **if** tuples in D are all the same class, C **then**

```
(3)   return N as a leaf node labelled with the class C;
(4) if attribute_list is empty then
(5)   return N as a leaf node labelled with the majority class in D;
//majority voting
(6) apply  Attribute_selection_method (D attribute_list) to find the
"best" splitting_criterion;
(7  label node N with splitting_criterion;
(8) if splitting_attribute is discrete-valued and
      multiway splits allowed then
(9)   attribute_list←attribute_list – splitting_attribute;
(10) for each outcome j of splitting_criterion
//partition the tuples and grow subtrees for each partition
(11)  let Dj be the set of data tuples in D satisfying outcome j; //a
partition
(12) if Dj is empty then
(13)  attach a leaf labelled with the majority class in D to node N;
(14)  else attach the node returned by Generate_decision_tree(Dj,
attribute_list) to node N;
endfor
(15) return N;
```

Σχήμα 4.5: Αλγόριθμος για την δημιουργία ενός δέντρου αποφάσεων

- Ο αλγόριθμος καλείται με τρεις παραμέτρους: D, *attribute_list,* και *Attribute_selection_method.* Το D είναι ένα σύνολο από δεδομένα τα οποία θα χωρίσουμε σε δυο ομάδες ανθρώπους/ιχνηλάτες. Το σύνολο αυτό αποτελεί τις πλειάδες εκπαίδευσης μαζί με τις αντίστοιχες κλάσεις στις οποίες ανήκουν. Η παράμετρος *attribute_list* είναι μια λίστα από χαρακτηριστικά τα οποία χαρακτηρίζουν τις πλειάδες**.** Το *Attribute_selection_method* ορίζει μια ευριστική διαδικασία, για την επιλογή του χαρακτηριστικού το όποιο διακρίνει "καλύτερα" τις δυο ομάδες σύμφωνα με την κλάση στην οποία ανήκουν. Η διαδικασία αυτή χρησιμοποιεί μια μετρική - το "information gain" - για την επιλογή του χαρακτηριστικού.

- Το δέντρο ξεκινά με ένα μοναδικό κόμβο, *N*, αναπαριστώντας τις πλειάδες εκπαίδευσης στο *D*.
- Αν οι πλειάδες στο *D* ανήκουν όλες στην ίδια κλάση τότε το *N* γίνεται φύλλο του δέντρου με ετικέτα αυτής της κλάσης.
- Ειδάλλως, ο αλγόριθμος καλεί την διαδικασία *Attribute_selection_method* για να ορίσει το κριτήριο διαχωρισμού. Το κριτήριο διαχωρισμού δείχνει ποιο χαρακτηριστικό να ελέγξουμε στον κόμβο *N*, καθορίζοντας τον καλύτερο τρόπο για τον διαχωρισμό τον πλειάδων στο *D* σε ανεξάρτητες κλάσεις. Το κριτήριο διαχωρισμού λέει επίσης ποια κλαδιά του δέντρου να μεγαλώσουμε από τον κόμβο *N* σε σχέση με το αποτέλεσμα του επιλεγμένου ελέγχου. Ποιο συγκεκριμένα, το κριτήριο διαχωρισμού υποδηλώνει το χαρακτηριστικό διαχωρισμού και μπορεί να υποδηλώνει επίσης, ή ένα σημείο διαχωρισμού ή ένα υποσύνολο διαχωρισμού. Το κριτήριο διαχωρισμού ορίζεται έτσι ώστε, στη ιδανική περίπτωση, τα αποτελέσματα του διαχωρισμού σε κάθε κλαδί του δέντρου, είναι όσο ποιο "καθαρά" γίνεται. Ένας διαχωρισμός ονομάζεται "καθαρός" αν όλες οι πλειάδες σε αυτό ανήκουν στην ιδία κλάση.
- Ο κόμβος *N* ονομάζεται με βάση το χαρακτηριστικό διαχωρισμού, το οποίο χρησιμοποιείται σαν έλεγχο σε αυτόν τον κόμβο. Για κάθε αποτέλεσμα του ελέγχου στον κόμβο, δημιουργείται ένα νέο κλαδί. Ανάλογα με το χαρακτηριστικό διαχωρισμού μπορούν να δημιουργηθούν διάφορα ειδή δέντρων. Στην δική μας περίπτωση δημιουργείται ένα δυαδικό δέντρο (Σχήμα 6). Για κάθε έλεγχο σε κάθε κόμβο του δέντρου, υπάρχουν δυο πιθανά αποτελέσματα το "yes" και το "no". Το αριστερό κλαδί του κόμβου έχει την τιμή "yes" και συμπεριλαμβάνει τις πλειάδες οι οποίες ικανοποιούν την

συνθήκη και το δεξί κλαδί την τιμή "no" και σε αυτό ανήκουν οι πλειάδες οι οποίες δεν ικανοποιούν την συνθήκη.

- Ο αλγόριθμος χρησιμοποιεί την ίδια διαδικασία αναδρομικά για να δημιουργήσει το δέντρο αποφάσεων για τις πλειάδες σε κάθε διαχωρισμό, D_j από το σύνολο *D.*
- Ο αναδρομικός διαχωρισμός σταματά μόνο όταν ισχύει μια από τις παρακάτω συνθήκες τερματισμού:
    1. Όλες οι πλειάδες στο σύνολο D ανήκουν στην ιδία κλάση, ή
    2. Δεν υπάρχουν άλλα χαρακτηριστικά τα οποία μπορούν να διαχωρίσουν τις ομάδες. Σε αυτήν την περίπτωση χρησιμοποιείται η πλειοψηφία των ψηφοφόρων. Αυτό σημαίνει τη μετατροπή του κόμβου *N* σε φύλο του δέντρου και η ετικέτα του είναι αυτή η οποία έχει τις παραπάνω πλειάδες στο σύνολο *D.*
    3. Δεν υπάρχουν άλλες πλειάδες για ένα δεδομένο κόμβο, το οποίο σημαίνει ότι το σύνολο D_j είναι άδειο.
- Τέλος, επιστρέφεται το δέντρο αποφάσεων

Ο αλγόριθμος ID_3 χρησιμοποιεί την μετρική "information gain" για την επιλογή του χαρακτηριστικού σε κάθε κόμβο, για το χτίσιμο του δέντρου αποφάσεων [10]. Στην επόμενη παράγραφο θα περιγράψουμε την μετρική αυτή, για να δούμε πως κτίζεται και το δέντρο αποφάσεων για τον σκοπό αυτής της εργασίας.

4.3.2.1 Attribute Selection Measure και Information gain

Το "Attribute Selection Measure" είναι ένα ευρετικό για τη επιλογή του κριτηρίου διαχωρισμού, το οποίο θα ξεχωρίζει ένα σύνολο δεδομένων D το οποίο περιέχει δυο

διαφορετικές κλάσεις. Αν θέλουμε να διαχωρίσουμε το σύνολο D σε μικρότερα σύνολα ανάλογα με το αποτέλεσμα του κριτηρίου διαχωρισμού, στην ιδανική περίπτωση τα σύνολα αυτά θα συμπεριλάμβαναν μόνο δεδομένα της ίδιας κλάσης. Άρα το καλύτερο κριτήριο διαχωρισμού θα ήταν αυτό το οποίο θα έχει τέτοια αποτελέσματα. Το "Attribute Selection Measure" παρέχει μια κατάταξη για κάθε χαρακτηριστικό των προς εκπαίδευση πλειάδων. Το χαρακτηριστικό το οποίο έχει την καλύτερη βαθμολογία για την μετρική, επιλέγεται σαν κριτήριο διαχωρισμού. Ο αλγόριθμος ID3 χρησιμοποιεί την μετρική information gain για την επιλογή χαρακτηριστικού σε κάθε κόμβο του δέντρου. Το χαρακτηριστικό με το ψηλότερο information gain επιλέγεται σαν κριτήριο διαχωρισμού για τον κόμβο N. Αυτό το χαρακτηριστικό ελαχιστοποιεί την πληροφορία η οποία χρειάζεται για τον διαχωρισμό των πλειάδων στις αντίστοιχες κλάσεις. Μια τέτοια προσέγγιση ελαχιστοποιεί τον αριθμό των ελέγχων που χρειάζεται για των διαχωρισμό των πλειάδων και εγγυάται την εύρεση ενός απλού δέντρου.

Η αναμενόμενη πληροφορία η οποία χρειάζεται για την ταξινόμηση των πλειάδων δίνεται από τον τύπο:

$$\mathrm{Info(D)} = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Όπου p_i είναι η πιθανότητα ότι μια οποιαδήποτε πλειάδα στο σύνολο D ανήκει στη κλάση C_i και υπολογίζεται από $|C_{i,D}|/|D|$. Ο λογάριθμος με βάση το 2 χρησιμοποιείται για το λόγο ότι η πληροφορία κωδικοποιείται σε bits. Το *Info(D)* είναι ο μέσος όρος της πληροφορίας η οποία χρειάζεται για την εύρεση της κλάσης μιας πλειάδας στο σύνολο D.

Για να υπολογίσουμε πόση πληροφορία θα χρειαστεί μετά τον τρέχον διαχωρισμό μέχρι να φτάσουμε σε ακριβή ταξινόμηση, χρησιμοποιούμε τον τύπο:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

Ο Όρος $\frac{|D_j|}{|D|}$ είναι το βάρος για το j-διαχωρισμό. $To\ Info_A(D)$ είναι η αναμενόμενη πληροφορία η οποία χρειάζεται για την ταξινόμηση μιας συνόδου στην κλάση που ανήκει βασισμένο στο διαχωρισμό από το Α. Όσο ποιο μικρή είναι η αναμενόμενη πληροφορία τόσο πιο μεγάλο είναι το "purity" του διαχωρισμού. Το information gain ορίζεται ως την διάφορα ανάμεσα στην αρχική απαιτούμενη πληροφορία (βασισμένη μόνο στο ποσοστό των δυο κλάσεων) και της καινούριας απαίτησης (που παρατηρείται μετά από τον διαχωρισμό στο Α):

Gain(A) = Info(D) - Info(A)

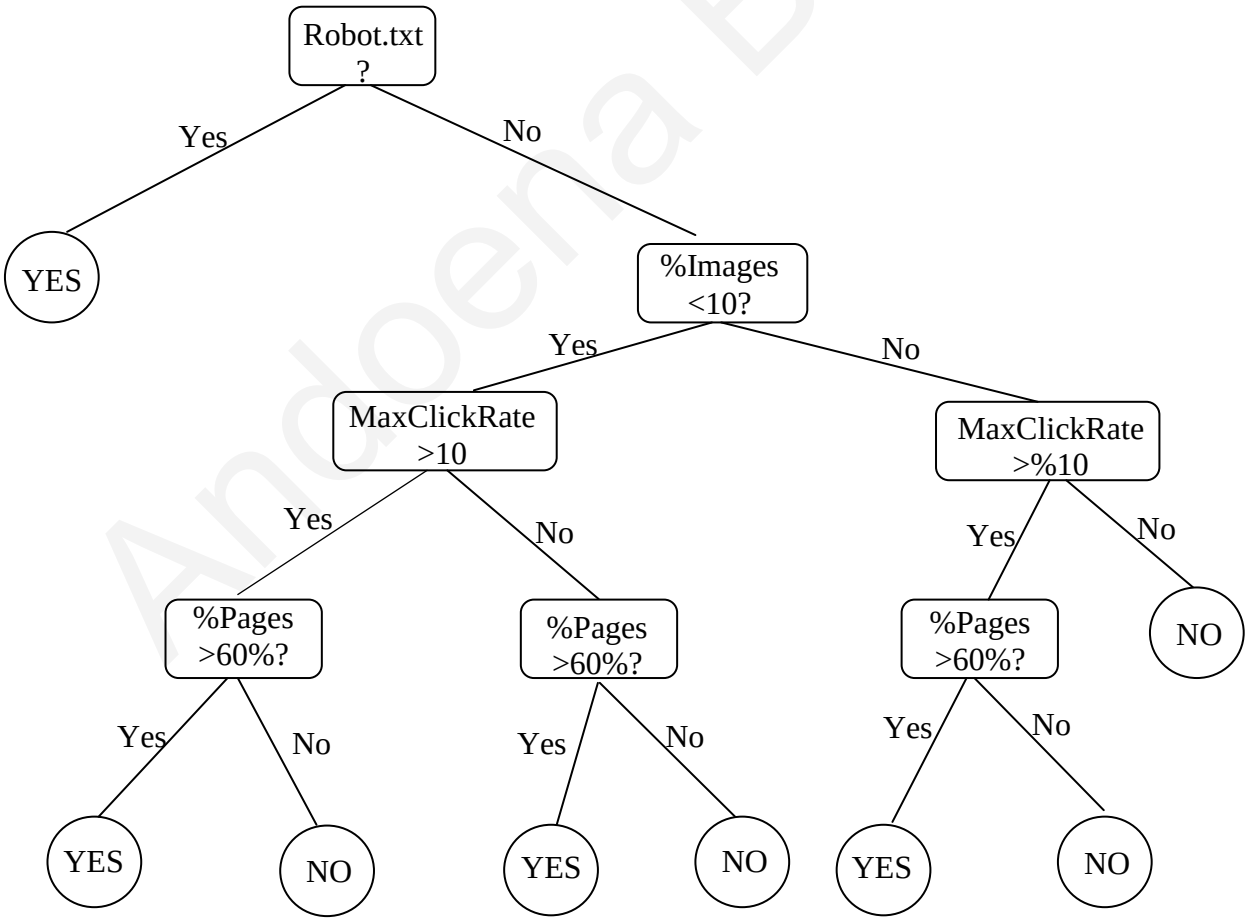
Ποιο συγκεκριμένα το *Gain(A)* μας δείχνει το κέρδος αν διαχωρίσουμε στο χαρακτηριστικό Α. Το χαρακτηριστικό Α με το μεγαλύτερο Information gain, *Gain(A)* επιλέγεται σαν κριτήριο διαχωρισμού στον κόμβο Ν [10].

4.3.3 Δημιουργία του δέντρου αποφάσεων χρησιμοποιώντας το Information gain

Σκοπός μας σε αυτήν την εργασία είναι να ταξινομήσουμε τις εισερχόμενες συνόδους σε ιχνηλάτες ή ανθρώπους σε πραγματικό χρόνο. Για την επίτευξη αυτού του στόχου δημιουργήσαμε ένα δέντρο αποφάσεων. Για τη δημιουργία του δέντρου αποφάσεων χρησιμοποιήσαμε ένα εκπαιδευτικό σύνολο από 500 συνόδους από τις οποίες οι 250 ανήκουν σε ιχνηλάτες και οι υπόλοιπες 250 ανήκουν σε ανθρώπους. Τα χαρακτηριστικά τα οποία χρησιμοποιήσαμε για κριτήριο διαχωρισμού σε κάθε κόμβο του δέντρου είναι: 1) Αν στη σύνοδο υπάρχει αίτηση για το αρχείο robot.txt , 2) Το

ποσοστο των αιτημάτων για εικόνες σε μια σύνοδο, 3) Το ποσοστο των αιτημάτων για σελιδες σε μια σύνοδο 4) Το μέγιστο ρυθμό των κλικ σε μια σύνοδο, 5) Η διάρκεια της συνόδου.

Με βάση το information gain το οποίο έχουμε υπολογίσει για κάθε χαρακτηριστικό δημιουργήσαμε το δέντρο αποφάσεων που φαίνεται στο Σχήμα 4.6. Για κάθε εισερχόμενη σύνοδο, αρχικά ελέγχουμε αν υπάρχει αίτηση για το αρχείο robot.txt. Αν ναι τότε αυτή η σύνοδος ταξινομείται ως ερχόμενη από ιχνηλάτη. Αν δεν υπάρχει αίτηση για το αρχείο, τότε ελέγχεται αν το ποσοστό των εικόνων σε αυτήν την σύνοδο είναι μικρότερο από 10%. Αν αυτό ισχύει τότε ελέγχουμε το μέγιστο ρυθμό κλικ είναι μεγαλύτερο του 8. Αν η τελευταία συνθήκη



Σχήμα 4.6: Δέντρο αποφάσεων για την ανίχνευση των ιχνηλατών

ικανοποιείται τότε ελέγχουμε αν το ποσοστο των αιτημάτων για σελίδες είναι μεγαλύτερο του 60% και αν ισχύει τότε έχουμε ανιχνεύσει έναν ιχνηλάτη. Βασικά για κάθε εισερχόμενη σύνοδο ακολουθούμε ένα μονοπάτι στο δέντρο (το οποίο δημιουργείται με βάση τους ελέγχους σε κάθε κόμβο του δέντρου), μέχρι να φτάσουμε σε ένα κόμβο φύλλο. Ο κόμβος φύλλο μας λέει την τάξη στην οποία ανήκει η τρέχουσα σύνοδος.

4.4 Request Cleaner

Το Request Cleaner είναι ένα νήμα του οποίου ο σκοπός είναι να απελευθερώνει την μνήμη την οποία κατακρατεί η λίστα με τα αιτήματα. Το νήμα αυτό εκτελεί αυτήν την εργασία κάθε λίγα δευτερόλεπτα. Διαγράφονται όλα τα προηγούμενα αιτήματα από κάθε IP διεύθυνση εκτός από αυτά της ενεργής συνόδου. Η λειτουργία αυτού του νήματος διαβεβαιώνει ότι η μνήμη η οποία είναι δεσμευμένη από τον ανιχνευτή ιχνηλατών διατηρείται σε χαμηλά επίπεδα.

5. Κεφάλαιο 5

Πειράματα και αξιολόγηση του συστήματος

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα πειράματα τα οποία τρέξαμε, με στόχο να εφαρμόσουμε την μεθοδολογία μας και να αξιολογήσουμε την απόδοση του συστήματος για ανίχνευση ιχνηλατών σε πραγματικό χρόνο.

5.1 Το σύνολο δεδομένων για εκπαίδευση

Για την αξιολόγηση της απόδοσης του συστήματος ανίχνευσης ιχνηλατών χρησιμοποιήσαμε αρχεία απογραφής από τους εξυπηρετητές του safeweb και του geclipse. Τα αρχεία απογραφής που καταγράφηκαν από τον εξυπηρετητή του safeweb τα χρησιμοποιήσαμε για την εκπαίδευση του δέντρου αποφάσεων. Αρχικά χωρίσαμε τα αρχεία απογραφής σε συνόδους με το πρόγραμμα ανάλυσης συνόδων και στη συνεχεία ταξινομήσαμε τις συνόδους ως δημιουργημένες από ιχνηλάτες ή ανθρώπους. Συνολικά από τον εξυπηρετητή του safeweb δημιουργήθηκαν 1000 συνόδοι από τις οποίες οι 300 ανήκαν σε ιχνηλάτες και οι 700 ταξινομήθηκαν ως δημιουργημένες από ανθρώπους. Για την εκπαίδευση του δέντρου αποφάσεων, πήραμε όλες τις συνόδους των ιχνηλατών και από το σύνολο των συνόδων των ανθρώπων διαλέξαμε τυχαία 300 συνόδους. Όλες οι σύνοδοι ελέγχτηκαν από έναν εμπειρογνώμονα για να είμαστε σίγουροι ότι δεν χρησιμοποιούμε λάθος ταξινομημένα στοιχεία κατά την διάρκεια της εκπαίδευσης του δέντρου.

Ένα σημαντικό θέμα που έπρεπε να διευθετήσουμε στην ανίχνευση των ιχνηλατών σε πραγματικό χρόνο, ήταν ο αριθμός των αιτημάτων που πρέπει να

μαζευτεί από μια IP διεύθυνση, πριν προχωρήσουμε στην ανάλυση αυτών των αιτημάτων. Πιο συγκεκριμένα, για να ταξινομηθεί μια IP διεύθυνση σε ιχνηλάτη ή άνθρωπο, πρέπει πρώτα να υπάρχει ένας αριθμός από αιτήματα στα οποία θα υπολογιστούν τα απαραίτητα χαρακτηριστικά και στην συνέχεια με το δέντρο αποφάσεων θα ταξινομηθεί στην αντίστοιχη ομάδα. Στο σημείο αυτό είναι πολύ σημαντικό να έχουμε έναν αριθμό από αιτήματα τα οποία θα περιέχουν όλη την πληροφορία που χρειαζόμαστε για να ταξινομήσουμε τις εισερχόμενες IP διευθύνσεις αλλά ταυτόχρονα θα πρέπει να πάρουμε την απόφαση όσο πιο γρήγορα γίνεται. Άρα στόχος είναι να βρούμε ποιος είναι ο ελάχιστος αριθμός από αιτήματα που πρέπει να μαζευτεί πριν προχωρήσουμε στην ανάλυση για να έχουμε τα καλύτερα αποτελέσματα ταξινόμησης σε ακρίβεια και σε χρόνο.

5.2 Έλεγχος του συστήματος

Για τον έλεγχο του συστήματος χρησιμοποιήσαμε ένα διαφορετικό αρχείο απογραφής το οποίο πήραμε από τον εξυπηρετητή του geclipse. Σε αυτό το αρχείο καταγράφεται η κίνηση για μια περίοδο τεσσάρων μηνών από τις 23 Αύγουστου του 2007 μέχρι τις 30 Δεκεμβρίου του 2007. Για να στέλνονται τα αιτήματα στο σύστημα ανίχνευσης ιχνηλατών το όποιο υλοποιήσαμε σε πραγματικό χρόνο, δημιουργήσαμε και υλοποιήσαμε έναν προσομοιωτή (emulator). Ο προσομοιωτής διαβάζει ένα-ένα τα αιτήματα από το αρχείο απογραφής και τα στέλνει στο σύστημα όπως θα έρχονταν σε πραγματικό χρόνο κατευθείαν από το πελάτη στον εξυπηρετητή.

Πραγματοποιήσαμε 5 διαφορετικά σενάρια, αλλάζοντας κάθε φορά τον αριθμό των αιτημάτων που πρέπει να μαζευτεί από μια IP διεύθυνση, πριν προχωρήσουμε στην ανάλυση τους: (Να τονίσουμε σε αυτό το σημείο ότι αναφερόμαστε μονό στα αιτήματα τα οποία ζητούν σελίδες http, php, htm html..)

1. Πρώτο σενάριο: Σε αυτόν τον έλεγχο περιμένουμε να μαζευτούν 5 αιτήματα από μια IP διεύθυνση και στην συνέχεια τα αναλύουμε για να αποφασίσουμε σε ποια ομάδα ανήκει.
2. Δεύτερο σενάριο: Αυξάνουμε τον αριθμό των αιτημάτων προς ανάλυση σε 10.
3. Τρίτο σενάριο: Θέλουμε να δούμε πως θα είναι τα αποτελέσματα της ταξινόμησης εάν περιμένουμε ένα μεγαλύτερο αριθμό από αιτήματα. Για το λόγο αυτό ορίζουμε τον αριθμό των αιτημάτων σε 20.
4. Τέταρτο σενάριο: Λογικά, όσο πιο μεγάλος είναι ο αριθμός των αιτημάτων προς ανάλυση τόσο πιο ακριβές θα είναι και τα αποτελέσματα της ταξινόμησης. Για το λόγο αυτό κάνουμε ακόμα έναν έλεγχο με πολύ μεγαλύτερο αριθμό αιτημάτων και συγκεκριμένα περιμένουμε 50 αιτήματα.
5. Πέμπτο σενάριο: Αυτός ο έλεγχος προέκυψε μετά από τα αποτελέσματα των προηγούμενων ελέγχων. Όπως θα δούμε και στην συνέχεια από τα αποτελέσματα των μετρικών για τους παραπάνω ελέγχους, φαίνεται ότι τα καλύτερα αποτελέσματα ταξινόμησης σε ακρίβεια και σε χρόνο θα ήταν αν περιμέναμε έναν αριθμό από αιτήματα ανάμεσα στα 10 και στα 20. Για το λόγο αυτό πραγματοποιούμε ακόμα έναν έλεγχο περιμένοντας 15 αιτήματα κάθε φορά.

Για την αξιολόγηση του συστήματος τρέξαμε τα πιο πάνω πειράματα και υπολογίσαμε τρεις μετρικές τις οποίες θα παρουσιάσουμε στην συνέχεια [14]:

1. Η πρώτη μετρική *True Positive* ή *Recall* δείχνει τον αριθμό των IP διευθύνσεων οι οποίες έχουν κατηγοριοποιηθεί σωστά από το σύστημα ως ιχνηλάτες δια το συνολικό αριθμό των IP διευθύνσεων που ανήκουν σε ιχνηλάτες:

$$True\ Positive\ = \frac{No.of\ Crawler\ IP\ addresses\ \ correctly\ classified}{Total\ No.\ of\ actual\ Crawler\ IP\ adresses}$$

2. Η δεύτερη μετρική False Positive, δείχνει τον αριθμό των IP διευθύνσεων οι οποίες έχουν κατηγοριοποιηθεί λάθος ως ιχνηλάτες (ενώ στην πραγματικότητα ανήκουν σε ανθρώπους), δια τον συνολικό αριθμό IP διευθύνσεων που βρήκε το σύστημα ότι ανήκουν σε ιχνηλάτες:

$$False\ Positive\ = \frac{No.of\ Crawler\ IP\ addresses\ \ wrongly\ classified}{Total\ No.\ of\ IP\ adresses\ classified\ as\ Crawler}$$

3. Μια άλλη μετρική την οποία υπολογίζουμε είναι η *Precision,* η μετρική αυτή δείχνει τον αριθμό των IP διευθύνσεων σωστά κατηγοριοποιημένα από το σύστημα ως ιχνηλάτες, δια τον συνολικό αριθμό IP διευθύνσεων τον οποίο το σύστημα βρήκε ως ιχνηλάτες:

$$Precision\ = \frac{No.of\ Crawler\ IP\ addresses\ \ correctly\ classified}{Total\ No.\ of\ IP\ adresses\ classified\ as\ Crawler}$$

Οι δυο μετρικές Recall(R) και Precision(P) μπορούν να συνδυαστούν σε μια μετρική την F_1 :

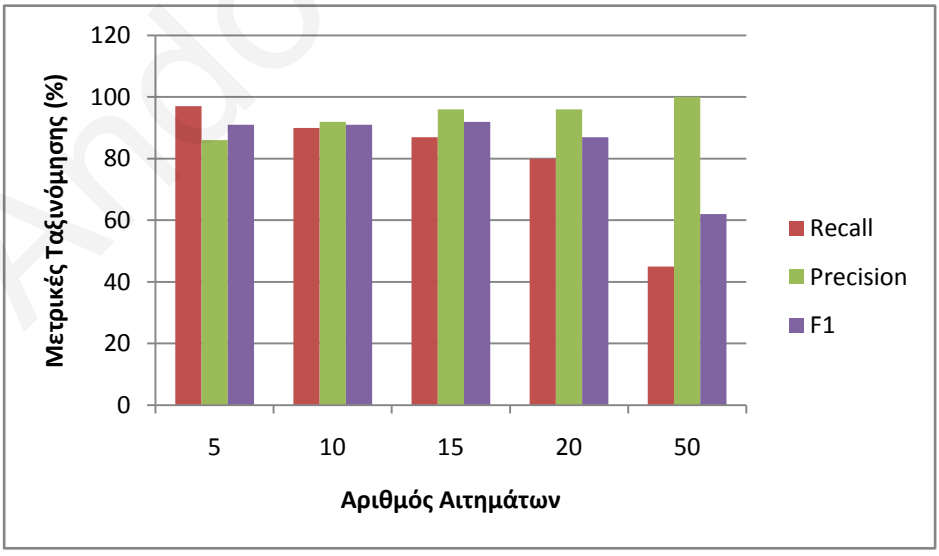
$$F1 = \frac{2RP}{R+P}$$

Τα αποτελέσματα αυτών των μετρικών για κάθε σενάριο ξεχωριστά φαίνονται στον παρακάτω πίνακα (πίνακα 5.1).

Ελάχιστος αριθμός αιτημάτων	True Positive	False Positive	Precision	F_1 measure
5	0.97	0.12	0.86	0.91
10	0.90	0.07	0.92	0.91
15	0.87	0.04	0.96	0.912
20	0.80	0.036	0.96	0.87
50	0.45	0.00	1.00	0.62

Πίνακας 5.1: Αποτελέσματα αξιολόγησης

Η μετρική F_1 είναι ο αρμονικός μέσος όρος των μετρικών precision and recall. Η μετρική αυτή συνδυάζει τις δυο μετρικές σε μια μοναδική τιμή, δίνοντάς τους έτσι το ίδιο βάρος σημαντικότητας. Βασικά ένας ταξινομητής πρέπει να έχει ψηλό precision και recall μαζί, διαφορετικά η τιμή της μετρικής F_1 θα είναι κοντά στην πιο μικρή τιμή των δυο μετρικών. Η γραφική παράσταση που αντιστοιχεί στις τιμές αυτών των μετρικών για το σύστημα ανίχνευσης ιχνηλατών σε πραγματικό χρόνο φαίνεται στο σχήμα 1.



Σχήμα 5.1: Γραφική παράσταση για τα αποτελέσματα αξιολόγησης του συστήματος

Όπως παρατηρούμε και από την γραφική παράσταση το μέγιστο recall (97%) επιτυχαίνεται όταν ορίζουμε τον αριθμό των αιτημάτων σε 5. Ταυτόχρονα όμως, στην περίπτωση αυτή έχουμε και το ελάχιστο precision (86%). Είναι πολύ σημαντικό ένας ταξινομητής να χαρακτηρίζεται και από ένα ψηλό recall και από ψηλό precision. Ο ταξινομητής ο όποιος έχουμε υλοποιήσει παρουσιάζει σε όλα τα σενάρια που τρέξαμε precision μεγαλύτερο του 86 και recall μεγαλύτερο του 80. Φυσικά παρουσιάζει ένα πολύ χαμηλό recall στην περίπτωση που ορίζουμε τον αριθμό των αιτημάτων σε 50. Στην τελευταία περίπτωση ο ταξινομητής ανίχνευσε μονό τις μισές IP διευθύνσεις οι οποίες ανήκαν σε ιχνηλάτες, αλλά δεν ταξινόμησε καμιά IP διεύθυνση λάθος ως ιχνηλάτης. Στην περίπτωση που αριθμός των αιτημάτων είναι 10 ο ταξινομητής δεν ανιχνεύει μόνο 13 IP διευθύνσεις από τις 131 που υπάρχουν, ενώ κάνει λάθος ταξινόμηση μόνο σε 10 IP διεύθυνσεις ανθρώπων από ένα σύνολο 453 IP διευθύνσεων. Τα καλύτερα αποτελέσματα εμφανίζονται στη περίπτωση που ο αριθμός των αιτημάτων είναι 15, για το λόγο ότι βρίσκει 87% των ιχνηλατών και ταξινομεί λάθος μονό 5 IP διευθύνσεις από τις 453. Επιπλέον σε αυτήν την περίπτωση έχουμε και την ψηλότερη τιμή για την μετρική F_1. Αν αυξήσουμε τον αριθμό των αιτημάτων σε 20 το recall πέφτει σε 80% ενώ το precision δεν αλλάζει σημαντικά με αποτέλεσμα τα 20 αιτήματα να ρίχνουν την τιμή της μετρικής F_1 και ως αποτέλεσμα την απόδοση του συστήματος.

Τέλος, να αναφέρουμε ότι το σύστημα έτρεχε για δυο συνεχόμενους μήνες χωρίς να παρουσιάζει οποιαδήποτε προβλήματα. Επιπλέον όλες οι IP διευθύνσεις οι οποίες ταξινομήθηκαν από το σύστημα ως ιχνηλάτες, έχουν ελεγχθεί από έναν εμπειρογνώμονα για να αποφασιστεί αν όντως στην πραγματικότητα προέρχονται από ιχνηλάτες.

6. Κεφάλαιο 6

Συμπεράσματα και Μελλοντική εργασία

6.1 Συμπεράσματα

Στα πλαίσια αυτής της μεταπτυχιακής διατριβής, αρχικά αναλύσαμε τα χαρακτηριστικά των συνόδων των ιχνηλατών και των ανθρώπων και στην συνέχεια αναπτύξαμε και υλοποιήσαμε ένα σύστημα για την ανίχνευση των ιχνηλατών σε πραγματικό χρόνο. Επιπλέον, υλοποιήσαμε και ένα προσομοιωτή για να δοκιμάσουμε και να αξιολογήσουμε το σύστημα.

Για τη στατιστική ανάλυση των χαρακτηριστικών των συνόδων των δυο ομάδων ιχνηλάτης/άνθρωπος, χωρίσαμε τα αρχεία απογραφής τα οποία έχουμε πάρει από διαφορετικούς εξυπηρετητές, σε συνόδους. Στη συνέχεια ταξινομήσαμε τις συνόδους σε ανθρώπους και ιχνηλάτες χρησιμοποιώντας τον ταξινομητή, ένα δίκτυο Naïve Bayesian το οποίο είχε εκπαιδευτεί προηγουμένως. Υπολογίσαμε 25 διαφορετικά χαρακτηριστικά πάνω στις συνόδους των δυο ομάδων και κάναμε μια στατιστική ανάλυση αυτών των χαρακτηριστικών με τη βοήθεια του στατιστικού πακέτου SPSS.

Η στατιστική ανάλυση έδειξε ότι οι δυο ομάδες δεν διαφέρουν στα περισσότερα χαρακτηριστικά εκτός από κάποιες αμελητέες διαφορές. Οι πιο σημαντικές διαφορές φαίνονται στα χαρακτηριστικά όπως το μέγιστο ρυθμό κλικ των δυο ομάδων (το μέγιστο ρυθμό κλικ που παρατηρείται στις συνόδους των ανθρώπων δεν ξεπερνά τα 8 κλικ το λεπτό, ενώ στους ιχνηλάτες το μέγιστο ρυθμό κλικ είναι μεγαλύτερο), στο ποσοστό των αιτησεων για εικόνες (jpeg, gif..) που υπάρχουν στις

συνόδους (90% των αιτήσεων στις συνόδους των ανθρώπων είναι για εικόνες), στο ποσοστό των αιτήσεων για σελίδες (htm, html, php, asp..) (οι σύνοδοι των ιχνηλατών έχουν ποσοστό για σελιδες πάνω από 60%), στο ποσοστό των αποκρίσεων του εξυπηρετητή με κωδικό σφάλματος 4XX (στις συνόδους των ιχνηλατών το ποσοστό αυτό είναι τουλάχιστον 20% ενώ στους ανθρώπους είναι αμελητέο). Επίσης μια σημαντική διαφορά υπάρχει στον χρόνο στον οποίο ένας ιχνηλάτης και ένας άνθρωπος κάνουν την επόμενη αίτηση. Η στατιστική ανάλυση έδειξε ότι ένας ιχνηλάτης κάνει την επόμενη αίτηση σε λιγότερο από 2 λεπτά και διατηρεί ένα σταθερό ρυθμό, ενώ αυτό δεν ισχύει για του ανθρώπους οι οποίοι κάνουν την επόμενη αίτηση μετά από ένα χρονικό διάστημα το οποίο παίρνει τιμές από 1 μέχρι 30 λεπτά.

Στη συνέχεια αναπτύξαμε και υλοποιήσαμε έναν σύστημα για ανίχνευση των ιχνηλατών σε πραγματικό χρόνο. Το σύστημα το όποιο βασίστηκε στα αποτελέσματα της στατιστικής ανάλυσης, ανιχνεύει τους ιχνηλάτες με υψηλή ακρίβεια 92%. Βασισμένοι στα χαρακτηριστικά τα οποία προτείναμε για την ανίχνευση, τα αποτελέσματα μας έδειξαν ότι οι ιχνηλάτες του διαδικτύου μπορούν να ανιχνευτούν με μια ικανοποιητική ακρίβεια μετά από 10 αιτήματα.

6.2 Μελλοντική εργασία

Όπως σε κάθε ερευνητικό θέμα έτσι και στην περίπτωση της ανίχνευσης και διάκρισης των ιχνηλατών από τους χρήστες του διαδικτύου, υπάρχουν πάντα περιθώρια για βελτιώσεις και επεκτάσεις των μεθόδων που αναπτύσσονται στα πλαίσια μιας ερευνητικής εργασίας. Στο υποκεφάλαιο αυτό θα αναφέρουμε πιθανές βελτιώσεις και επεκτάσεις που μπορούν να γίνουν, αναφερόμενοι κατά κύριο λόγο στο μοντέλο που αναπτύχθηκε για την ανίχνευση των ιχνηλατών σε πραγματικό χρόνο.

Όπως παρατηρήσαμε από τα αποτελέσματα της στατιστικής ανάλυσης, ο χρόνος που μεσολαβεί ανάμεσα σε δυο διαδοχικά HTTP αιτήματα τα οποία στέλλονται από έναν χρήστη είναι διαφορετικός από αυτόν που μεσολαβεί στα αιτήματα των ιχνηλατών. Άρα θα μπορούσε κάποιος να ορίσει έναν εξοχρονισμό (timeout) το οποίο να αυτοπροσαρμόζεται με βάση τα εισερχόμενα αιτήματα. Στη συνέχεια θα μπορούσε να ξαναγίνει η στατιστική ανάλυση των συνόδων με σκοπό την ανακάλυψη καινούριων ιδιοτήτων ή ακόμα να παρατηρήσει κανείς αν αυτή η αλλαγή θα αύξανε την ακρίβεια του ταξινομητή κατά την ανίχνευση των ιχνηλατών.

Επίσης θα μπορούσε το σύστημα να τροποποιηθεί έτσι ώστε ,όταν ανιχνεύσει έναν ιχνηλάτη να μην καταχωρίσει αμέσως την IP διεύθυνση στη λίστα με τα Discovered Robots, αλλά να την καταχωρίσει αρχικά σε μια προσωρινή λίστα. Σε κάθε IP διεύθυνση να αντιστοιχίζεται ένας αριθμός ο όποιος θα αντιπροσωπεύει το ποσοστό εμπιστοσύνης για το ενδεχόμενο ότι IP διεύθυνση αυτή προέρχεται από έναν ιχνηλάτη. Αν αργότερα ξαναβρεί ιχνηλάτη από την ιδία IP διεύθυνση τότε το ποσοστο της εμπιστοσύνης για την συγκεκριμένη διεύθυνση αυξάνεται και ως αποτέλεσμα η πιθανότητα να ταξινομηθεί λάθος μια IP διεύθυνση μειώνεται.

Μια άλλη βελτίωση που μπορεί να γίνει, αφορά τις αποκρίσεις τις οποίες στέλνει ο εξυπηρετητής στον πελάτη. Από την στατιστική ανάλυση βρήκαμε ότι οι σύνοδοι των ιχνηλατών έχουν ένα ποσοστό αποκρίσεων του εξυπηρετητή με κωδικό σφάλματος 4XX. Αυτό μπορεί να χρησιμοποιηθεί σαν χαρακτηριστικό για την ανίχνευση των ιχνηλατών. Άρα το σύστημα θα μπορούσε να τροποποιηθεί για να λαμβάνει υπόψη τις αποκρίσεις του εξυπηρετητή στην ταξινόμηση των συνόδων.

Επίσης μια σημαντική επέκταση του συστήματος αφορά την χρησιμότητα του στην πραγματικότητα. Το σύστημα μπορεί εύκολα να τροποποιηθεί για να λειτουργεί ως ένα πληρεξούσιο ή σαν ένα module του εξυπηρετητή και στη συνέχεια να

χρησιμοποιηθεί από τους διαχειριστές εξυπηρετητών για την ανίχνευση των ιχνηλατών. Επιπλέον στο σύστημα μπορεί να προστεθούν και κανόνες για να λαμβάνει κάποια μέτρα απέναντι στις IP διευθύνσεις οι οποίες προέρχονται από ιχνηλάτες. Πιο συγκεκριμένα να αποτρέπει αυτές τι διευθύνσεις για ένα χρονικό διάστημα το οποίο αυξάνεται συγκριτικά με το πόσο κακόβουλα φέρεται ένας ιχνηλάτης.

7. Βιβλιογραφία

[1] The Web Robots Pages. http://www.robotstxt.org/ (last accessed Nov 2008)

[2] Athena Stassopoulou, Marios D. Dikaiakos, " Web robot detection: A probabilistic reasoning approach", *Computer Networks, Volume 53 February 2009, pp. 265-278*

[3] P. Tan, V, Kumar, "Discovery of Web Robot Sessions based on their Navigational Patterns", *Data Mining and Knowledge Discovery*, 2002, 6(1) pp.9-35.

[4] M.D. Dikaiakos, L. Papageorgiou and A. Stassopoulou, "An investigation of web crawler behavior: Characterization and metrics", *Computer Communications* 28(8), 2005 pp.880-897.

[5] The Web Robots Database. http://www.iplists.com/ (last accessed June 2008)

[6] A Standard for Robot Exclusion. http://www.robotstxt.org/orig.html (last accessed May 2008)

[7] B. Krishnamurthy and J. Rexford. *Web Protocol and Practice.* Addison-Wesley, 2001.

[8] W. Guo Sh. Ju Y. Gu, "Web robot detection techniques based on statistics of their requested URL resources", in *Computer Supported Cooperative Work in Design, 2005. Proceedings of the Ninth International Conference*, 2005 pp.302-306.

[9] W. Lu Sh. Yu, "Web Robot Detection Based on Hidden Markov Model", in *Communications, Circuits and Systems Proceedings, 2006 International Conference*, 2006, pp.1806-1810.

[10] J. Han and M Kamber, *"Data Mining Concepts and Techniques"* Elsevier Inc, 2006.

[11] K. Park, V. S. Pai, K.-W. Lee, and S. Calo, "Securing Web service by automatic robot detection", in *USENIX Technical Conference*, June 2006.

[12] V. Almeida, D. Menasce, R. Reidi, F. Peligrinelli, R. Fonseca, and W. M. Jr, "Analyzing Web Robots and their Impact on Caching", *In Proceedings of the 6th Web Caching and Content Delivery Workshop*, June 2001.

[13] SPSS tutorials http://www.spsstools.net/spss.htm (last visited May 2008)

[14] P. Tan, V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005

[15] N. Geens, J. Huysmans and J. Vanthienen "Evaluation of Web Robot Discovery Techniques: A Benchmarking Study" in *Advances in Data Mining,* Vol. 4065, Springer Berlin / Heidelberg, 2006, pp.121-130.

[16] Huntington P., Nicholas D., Jamali H., "Web robot detection in the scholarity information environment," *Journal of Information Science 2008, vol 34, pp. 726, May 8 2008.*

[17] X . Lin, L. Quan, H. Wu, "An Automatic Scheme to Categorize User Sessions in Modern HTTP Traffic", in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 2008, pp. 1-6.

[18] Google Advertising. http://www.google.com/ads/ (last accessed Sep 2008)