Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή στα Πληροφοριακά Συστήματα

Υλοποίηση Τεχνικών Απόκρυψης Κανόνων Συσχέτισης (Association Rule Hiding) στην Python

Πολυχρόνιος Ταμαμίδης

Επιβλέπων Καθηγητής
Βασίλειος Βερύκιος

Αύγουστος 2013

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Υλοποίηση Τεχνικών Απόκρυψης Κανόνων Συσχέτισης (Association Rule Hiding) στην Python

Πολυχρόνιος Ταμαμίδης

Επιβλέπων Καθηγητής
Βασίλειος Βερύκιος

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Αύγουστος 2013

Περίληψη

Η εξόρυξη δεδομένων είναι ο επιστημονικός κλάδος που επιτρέπει την εξαγωγή γνώσης μέσω της επεξεργασίας του τεράστιου όγκου ακατέργαστης πληροφορίας που βρίσκεται αποθηκευμένη μέσα σε αποθετήρια δεδομένων. Μια πολύ συνηθισμένη μορφή αναπαράστασης αυτής της γνώσης είναι μέσω της εύρεσης κανόνων συσχέτισης ή επαναλαμβανόμενων συνόλων δεδομένων. Ωστόσο, μέσα από αυτή τη διαδικασία ανακάλυψης γνώσης είναι δυνατό να παραβιαστεί η ιδιωτικότητα ή να αποκαλυφθούν ευαίσθητα δεδομένα ατόμων ή οργανισμών. Για την αποτροπή αυτής της ανεπιθύμητης κατάστασης, θα πρέπει κάποιοι κανόνες συσχέτισης, ή συχνά στοιχειοσύνολα, να μην γίνονται φανερά αλλά να αποκρύπτονται, κάτι το οποίο είναι κύριο μέλημα ενός νέου, σχετικά, ερευνητικού πεδίου, της Απόκρυψης Κανόνων Συσχέτισης (Α.Κ.Σ).

Ένα βασικό ερευνητικό ερώτημα της Α.Κ.Σ. είναι η ανάπτυξη ενός εργαλείου που θα παρέχει την δυνατότητα εύκολης διασύνδεσης και ανάπτυξης νέων τεχνικών και συγκριτικής μελέτης αυτών σε σχέση με άλλες ήδη υπάρχουσες τεχνικές, μέσω ενός ολοκληρωμένου περιβάλλοντος.

Σκοπός της παρούσης διατριβής, είναι η δημιουργία με χρήση της Python, ενός ολοκληρωμένου περιβάλλοντος πειραματισμού και αξιολόγησης ορισμένων αλγορίθμων που έχουν προταθεί για την επίλυση του προβλήματος της απόκρυψης των κανόνων συσχέτισης, και το οποίο θα δίνει την δυνατότητα στον χρήστη της συγκριτικής μελέτης των χαρακτηριστικών διαφόρων μεθόδων–αλγορίθμων Απόκρυψης Κανόνων Συσχέτισης. Μέσω του ολοκληρωμένου περιβάλλοντος θα είναι εφικτή η εύκολη δοκιμή των αλγορίθμων με χρήση διαφορετικών παραμέτρων, η συγκριτική μελέτη της απόδοσης διαφορετικών τεχνικών για δυναμικά ορισμένα σύνολα δεδομένων εισόδου και ελέγχου απόδοσης, και η δημιουργία οπτικοποίησης των αποτελεσμάτων σύγκρισης των ποιοτικών και ποσοτικών χαρακτηριστικών των αλγορίθμων αυτών.

Για την πραγματοποίηση του παραπάνω στόχου μελετήθηκαν αρκετοί αλγόριθμοι και τεχνικές Απόκρυψης και τελικά επιλέχθηκαν τρεις από αυτούς οι οποίοι υλοποιήθηκαν στη γλώσσα προγραμματισμού Python και ενσωματώθηκαν στο περιβάλλον που αναπτύχθηκε. Οι αλγόριθμοι δοκιμάσθηκαν, μέσω του εργαλείου που δημιουργήθηκε στο πλαίσιο της παρούσης διατριβής, σε πραγματικά σύνολα συναλλαγών, ώστε να παραχθούν όσο το δυνατόν πιο αξιόπιστα αποτελέσματα – συμπεράσματα.

Summary

Data mining is the discipline that enables the extraction of knowledge through the processing of huge amounts of raw information stored in data repositories. A very common form of representation of this knowledge is through finding association rules or repeated data sets. However, through this process of knowledge discovery, the privacy of individuals or organizations can be violated by revealing sensitive data. To prevent this undesirable situation, some specific association rules or frequent itemsets, shall become concealed. This activity is the main concern of a relatively new research field which is the Association Rule Hiding.

A key research question of Association Rule Hiding is to develop a software tool that provides a user-friendly interface and which shall give the capability that new techniques which are now developed or will be developed in the future shall be integrated to the tool and be easily compared with others already existing to the tool, through an integrated environment.

The purpose of the present study is to create, using the Python technology, an integrated environment for experimentation and evaluation of some algorithms that have been proposed to solve the problem of Association Rule Hiding, enabling the user to perform comparative studies over the characteristics of various methods-algorithms for Association Rule Hiding. The integrated environment will allow for easy testing of algorithms using different parameters, comparative performance analysis of different techniques for dynamic sets of input and control data and creation of visual representations of the results through the comparison of the qualitative and quantitative characteristics of these algorithms.

To achieve the above goal, several algorithms and techniques of Association Rule Hiding were studied and finally selected three of those which have been implemented in the Python programming language and integrated into the environment developed. The algorithms were tested through the tool created in the present study, in real data sets of transactions in order to produce as much as possible reliable results and conclusions.

Ευχαριστίες

Η παρούσα μεταπτυχιακή διατριβή ολοκληρώθηκε χάρις την πολύτιμη καθοδήγηση και ενθάρρυνση του επιβλέποντα καθηγητή κου Βασίλειου Βερύκιου, τον οποίο ειλικρινά ευχαριστώ, μεταξύ άλλων, και για την ηθική στήριξη και τα εποικοδομητικά του σχόλια, που μου παρείχε κατά την διάρκεια της εκπόνησής της.

Επιπλέον, ευχαριστώ την σύζυγό μου Ρωξάνη για την υπομονή που έδειξε όσο χρόνο διήρκεσε η προετοιμασία και παρουσίαση της μεταπτυχιακής διατριβής.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Στο παρόν κεφάλαιο γίνεται μία εισαγωγή στο ευρύτερο θέμα της διατριβής, δηλαδή στην Απόκρυψη των Κανόνων Συσχέτισης. Στην παράγραφο 1.1 κάνουμε μία σύντομη εισαγωγή στο πρόβλημα της διατήρησης της ιδιωτικότητας κατά την Εξόρυξη Δεδομένων ενώ στην επόμενη παρουσιάζονται οι διάφορες προσεγγίσεις επίλυσης του προβλήματος που έχουν παρουσιασθεί μέχρι σήμερα. Στην παράγραφο 1.3 παρουσιάζεται η συνεισφορά της παρούσης Διατριβής στην επίλυση του προβλήματος, ενώ τέλος στην παράγραφο 1.4 αναφέρουμε μέρος της σχετικής έρευνας που έχει γίνει μέχρι σήμερα μέσω της βιβλιογραφίας που μελετήθηκε για την εκπόνηση της παρούσης Διατριβής.

1.1 Το πρόβλημα της Διατήρησης της Ιδιωτικότητας κατά την Εξόρυξη Δεδομένων

Οι ραγδαίες τεχνολογικές εξελίξεις στη συλλογή δεδομένων και στις δυνατότητες αποθήκευσης και ανάκτησής τους έχουν επιτρέψει σε διάφορους οργανισμούς (ερευνητικούς κ.α.) και επιχειρηματικά κέντρα να αποθηκεύουν τεράστιες ποσότητες δεδομένων. Πράγματι, οι ερευνητές στην ιατρική αλλά και σε κάθε επιστημονικό κλάδο συσσωρεύουν με ταχύτατο ρυθμό

δεδομένα, τα οποία αποτελούν το κλειδί για νέες ανακαλύψεις. Επίσης, η συλλογή πληροφοριών στα καταστήματα πώλησης αγαθών μέσω χρήσης συσκευών γραμμωτού κώδικα έχει επιτρέψει στους εμπόρους να αντιληφθούν την καταναλωτική συμπεριφορά των πελατών τους και να την αξιοποιήσουν μέσω της εξόρυξης δεδομένων. Η εξόρυξη δεδομένων είναι ο επιστημονικός κλάδος που επιτρέπει την εξαγωγή γνώσης μέσω της επεξεργασίας του τεράστιου όγκου ακατέργαστης πληροφορίας που βρίσκεται αποθηκευμένη μέσα σε αποθετήρια δεδομένων. Μια πολύ συνηθισμένη μορφή αναπαράστασης αυτής της γνώσης είναι μέσω της εύρεσης κανόνων συσχέτισης των δεδομένων ή συχνά επαναλαμβανόμενων συνόλων δεδομένων [13].

Είναι επίσης αδήριτη ανάγκη στις μέρες μας οι διάφοροι οργανισμοί, επιχειρηματικά κέντρα, η επιστημονική κοινότητα κ.τ.λ. να συνεργάζονται μεταξύ τους ανταλλάσοντας πληροφορίες και δεδομένα. Αυτή η συνεργασία μεταξύ των οργανισμών μπορεί να αποτελεί απαραίτητο εργαλείο στην κατάκτηση νέας γνώσης, ωστόσο, μέσα από αυτή τη διαδικασία ανακάλυψης γνώσης είναι δυνατό να παραβιαστεί η ιδιωτικότητα ή να αποκαλυφθούν ευαίσθητα δεδομένα ατόμων ή οργανισμών. Ως τέτοιο παράδειγμα μπορούμε να θεωρήσουμε τα ιατρικά δεδομένα κάποιου οργανισμού παροχής υγείας που, για ερευνητικούς λόγους τα δίδει σε κάποιον ερευνητικό οργανισμό ο οποίος όμως στο τέλος αποδεικνύεται αναξιόπιστος με αποτέλεσμα την διαρροή ευαίσθητων ατομικών σε τρίτα μέρη [07].

Για την αποτροπή αυτής της ανεπιθύμητης κατάστασης, θα πρέπει κάποιοι κανόνες συσχέτισης, ή συχνά επαναλαμβανόμενα στοιχειοσύνολα δεδομένων τα οποία θα προκύψουν από την επεξεργασία των δεδομένων, να μην γίνονται εμφανείς αλλά να αποκρύπτονται. Αυτό μας οδηγεί στο συμπέρασμα ότι τα αρχικά δεδομένα που βρίσκονται σε κάποιο αποθετήριο δεδομένων, πριν πωληθούν ή δοθούν προς επεξεργασία σε κάποιον τρίτο οργανισμό, θα πρέπει να τύχουν κάποιας επεξεργασίας ώστε η μετέπειτα εφαρμογή των τεχνικών της εξόρυξης δεδομένων να μην οδηγήσει σε αποκάλυψη εκείνων των στοιχειοσυνόλων ή κανόνων συσχέτισης που έχουν εκτιμηθεί ως ευαίσθητοι. Επειδή ό μόνος τρόπος για να επιτευχθεί η παραπάνω απόκρυψη είναι η τροποποίηση των συναλλαγών ή των στοιχείων που περιέχονται στη βάση δεδομένων, μοιραία αλλοιώνεται, και η αρχικά περιεχόμενη πληροφορία η οποία ωστόσο, θα πρέπει να είναι η μικρότερη δυνατή ώστε να αποκρύπτονται μόνο οι ευαίσθητοι κανόνες ή στοιχειοσύνολα και να αποκαλύπτονται όλοι οι σημαντικοί κανόνες ή στοιχειοσύνολα.

1.2 Διάφορες προσεγγίσεις για την επίλυση του προβλήματος.

Σε γενικές γραμμές οι προτεινόμενες προσεγγίσεις, μπορούν να ταξινομηθούν σε δύο κατηγορίες: i) αυτές της απόκρυψης δεδομένων και ii) αυτές της απόκρυψης γνώσης, με τις οποίες ασχολείται και η παρούσα Διατριβή.

Η πρώτη κατηγορία (απόκρυψη δεδομένων), περιλαμβάνει όλες τις μεθοδολογίες που διερευνούν με ποιον τρόπο η ιδιωτικότητα των ακατέργαστων δεδομένων μπορεί να επιτευχθεί, πριν την εξόρυξη δεδομένων. Οι προσεγγίσεις αυτής της κατηγορίας, στοχεύουν στην απομάκρυνση ιδιωτικών ή εμπιστευτικών πληροφοριών από τα αρχικά δεδομένα πριν από την δημοσιοποίησή τους και ενεργούν εφαρμόζοντας τεχνικές όπως η δειγματοληψία, γενίκευση κλπ, με σκοπό να παράγουν ένα εξομαλυμένο μέρος της αρχικής βάσης δεδομένων. Ο βασικός τους στόχος είναι να παρέχουν στον κάτοχο της βάσης δεδομένων, την δυνατότητα να λάβει ακριβή αποτελέσματα εξόρυξης δεδομένων όταν δεν έχει τα πραγματικά δεδομένα [07].

Η δεύτερη κατηγορία (απόκρυψη γνώσης), και πεδίο έρευνας της Απόκρυψης Κανόνων Συσχέτισης, περιλαμβάνει τις μεθοδολογίες που σκοπεύουν να προστατεύσουν τα ευαίσθητα αποτελέσματα που παρήχθησαν με την εφαρμογή των εργαλείων εξόρυξης δεδομένων, παρά τα δεδομένα αυτά καθαυτά. Το κύριο μέλημα της Απόκρυψης Κανόνων Συσχέτισης (Κ.Σ.) και βασική ερευνητική πρόκληση είναι η ανάπτυξη διαφόρων τεχνικών μέσω των οποίων να τροποποιούνται (απολυμαίνονται) οι συναλλαγές της βάσης δεδομένων κατά τρόπον ώστε όλοι οι μη ευαίσθητοι Κ.Σ. που μπορούσαν να εξορυχθούν από αυτήν να εξακολουθούν να εξορύσσονται, ενώ την ίδια στιγμή, κανένας από τους ευαίσθητούς Κ.Σ. να μη μπορεί πλέον να εξορυχθεί. Επιπλέον, δεν θα πρέπει να υπεισέρχονται ψευδείς Κ.Σ. ή ψευδή συχνά στοιχειοσύνολα [07].

Αν και η Απόκρυψη Κανόνων Συσχέτισης είναι ένα σχετικά νέο ερευνητικό πεδίο με ύπαρξη που δεν υπερβαίνει τα 15 έτη, έχει ωστόσο, μέχρι και σήμερα, προτείνει αρκετές τεχνικές για την επίλυση του προβλήματος. Κάποιες από τις τεχνικές αυτές λύνουν το πρόβλημα της διατήρησης της ιδιωτικότητας ευριστικά, με χρήση άπληστων, ως προς την φύση τους αλγορίθμους, ενώ κάποιες άλλες με ακριβή, μαθηματικά, τρόπο.

Οι ευριστικές τεχνικές οι οποίες αναζητούν την βέλτιστη λειτουργία σε κάθε βήμα του αλγορίθμου και συνεπώς δεν εγγυώνται την εύρεση της βέλτιστης λύσης για ολόκληρο το

πρόβλημα, υποφέρουν από την ύπαρξη παράπλευρων απωλειών, δηλαδή αποκρύπτουν κανόνες ή συχνά σύνολα τα οποία δεν συμπεριλαμβάνονται στα ευαίσθητα δεδομένα. Αντίθετα, οι ακριβείς λύσεις παρουσιάζουν μεγάλη πολυπλοκότητα και αντίστοιχα υψηλό χρόνο επεξεργασίας και απαιτήσεις σε μνήμη, ωστόσο δεν παρουσιάζουν παράπλευρες απώλειες.

Επιπλέον, κάποιες τεχνικές προσεγγίζουν το θέμα με χρήση της υποστήριξης και μόνο, αποκρύπτοντας συγκεκριμένα συχνά στοιχειοσύνολα, ενώ άλλες χρησιμοποιούν και την εμπιστοσύνη και αποκρύπτουν και συγκεκριμένους (έναν ή και περισσότερους σε κάθε επανάληψη του αλγορίθμου) κανόνες συσχέτισης. Στην πρώτη περίπτωση ο αλγόριθμος ελαττώνει την υποστήριξη του στοιχειοσυνόλου από το οποίο παράγεται ο ευαίσθητος Κ.Σ., μέχρι το σημείο που η υποστήριξη του να πέσει κάτω από το κατώφλι υποστήριξης *msup.* Στην δεύτερη περίπτωση ο αλγόριθμος ελαττώνει την εμπιστοσύνη του ευαίσθητου Κ.Σ., είτε αυξάνοντας την μέτρηση υποστήριξης του αριστερού μέρους του κανόνα, είτε μειώνοντας την μέτρηση υποστήριξης του δεξιού μέρους.

Τέλος, ορισμένες λύσεις του προβλήματος επιτυγχάνονται μέσω της μεθόδου της παραμόρφωσης (*distortion*) δηλαδή της οριστικής διαγραφής κάποιων στοιχείων από την «αποστειρωμένη» Β.Δ. τη στιγμή που κάποιες άλλες χρησιμοποιούν την τεχνική του μπλοκαρίσματος (*blocking*) δηλαδή τις τροποποιούν, στα ίδια σημεία, εισάγοντας κάποιους ειδικούς – αναγνωρίσιμους χαρακτήρες μέσα στη Β.Δ. (π.χ. τον χαρακτήρα '?'), δίνοντας την δυνατότητα στον τελικό χρήστη να γνωρίζει τα σημεία που η Β.Δ. έχει τροποποιηθεί, χωρίς όμως να έχει την δυνατότητα να εκμαιεύσει την κρυμμένη πληροφορία.

1.3 Συνεισφορά της παρούσης Μεταπτυχιακής Διατριβής

Από τα παραπάνω είναι ασφαλές να συμπεράνει κανείς ότι η μέθοδος με την οποία θα πρέπει να προσεγγιστεί η επίλυση του προβλήματος της διατήρησης της ιδιωτικότητας κατά την διαδικασία της εξαγωγής γνώσης, κάθε άλλο παρά προφανής μπορεί να θεωρηθεί. Πράγματι, οι διάφορες προσεγγίσεις είναι πολλές και εξίσου πολλοί και οι αλγόριθμοι που έχουν αναπτυχθεί για κάθε προσέγγιση καθιστώντας επιτακτική τη δοκιμή διαφόρων μεθόδων, τη σύγκριση των αποτελεσμάτων της καθεμιάς και τελικά την επιλογή της βέλτιστης μεθόδου.

Η συνεισφορά της παρούσης εργασίας έγκειται στη δημιουργία ενός εργαλείου το οποίο θα δίδει την δυνατότητα στον χρήστη να επιλέγει και να εκτελεί διαφόρους αλγορίθμους απόκρυψης

ευαίσθητων στοιχειοσυνόλων και να εξάγει συμπεράσματα, συγκρίνοντας την αποδοτικότητά τους, πειραματιζόμενος με διάφορα σετ δεδομένων και κάνοντας χρήση διαφορετικών παραμέτρων. Ο καλύτερος τρόπος για την επίτευξη των παραπάνω είναι μέσω ενός ολοκληρωμένου περιβάλλοντος που περιλαμβάνει γραφική διεπαφή (Graphical User Interface) ώστε να είναι κατά το δυνατόν ευκολότερη και ταχύτερη η εισαγωγή των δεδομένων εισόδου. Επίσης τα αποτελέσματα της σύγκρισης των αλγορίθμων θα πρέπει να δίδονται με παραστατικό τρόπο μέσω γραφικών παραστάσεων των αποτελεσμάτων.

Βεβαίως, η ανάγκη ύπαρξης ενός τέτοιου εργαλείου θα γίνεται περισσότερο σαφής, ιδιαίτερα στον μη ειδικό αναγνώστη, όσο εξελίσσεται η παρούσα μεταπτυχιακή διατριβή. Για τον παραπάνω σκοπό έχει αναπτυχθεί το Εργαλείο της παρούσης Μεταπτυχιακής διατριβής, υλοποιημένο εξολοκλήρου στην γλώσσα προγραμματισμού Python και συγκεκριμένα στην έκδοση 3.3 αυτής. Η εφαρμογή παρουσιάζεται αναλυτικά στο Κεφάλαιο 4.

1.4 Σχετική Βιβλιογραφία

Για την πραγματοποίηση του παραπάνω στόχου μελετήθηκαν αρκετοί αλγόριθμοι και τεχνικές Απόκρυψης και τελικά επιλέχθηκαν τρεις από αυτούς οι οποίοι υλοποιήθηκαν στη γλώσσα προγραμματισμού Python και ενσωματώθηκαν στο περιβάλλον που αναπτύχθηκε. Επίσης, καθώς θα ήταν αδύνατον να γίνει απόκρυψη γνώσης πριν αυτή παραχθεί, επιλέχθηκαν και δύο αλγόριθμοι εύρεσης κανόνων συσχέτισης, ο πολύ γνωστός Apriori [01], αλλά και ο επίσης γνωστός στην επιστημονική κοινότητα E-Clat [18].

Εμπνευστής του δημοφιλούς αλγόριθμου Apriori είναι οι Agrawal et al οι οποίοι και εισήγαγαν το 1993 και την έννοια της εξόρυξης Κανόνων Συσχέτισης [01]. Ο αλγόριθμος υλοποιήθηκε με βάση την διαδικασία *apriori-gen(F_{k-1})*. Συμπληρωματικό, αλλά σημαντικό βοήθημα γενικά πάνω στο αντικείμενο της εξόρυξης δεδομένων αποτέλεσε το σύγγραμμα των P. N. Tan, M. Steinbach, and V. Kumar "Εισαγωγή στην Εξόρυξη Δεδομένων" το οποίο και χρησιμοποιήθηκε ως σύγγραμμα αναφοράς [13]. Αξίζει ίσως να σημειωθεί ότι διάφορες υλοποιήσεις του παραπάνω αλγορίθμου μπορεί κανείς να αναζητήσει επιτυχώς στο διαδίκτυο ακόμη και σε Python, ωστόσο οι εν' λόγω υλοποιήσεις δεν υιοθετήθηκαν καθώς δεν υλοποιούσαν με ακρίβεια τη διαδικασία *apriori-gen (F_{k-1})* αν και το τελικό αποτέλεσμα που προέκυπτε ήταν το σωστό. Οι Zaki et al. στην εργασία τους “New Algorithms for Fast Discovery of Association Rules” [18] παρουσιάζουν τον E-Clat, τον δεύτερο από τους αλγόριθμους Εξόρυξης Δεδομένων που υλοποιήθηκαν στην παρούσα Μεταπτυχιακή διατριβή.

Το σύγγραμμα των A. Gkoulalas-Divanis και V. S. Verykios, "Association Rule Hiding for Data Mining" [07] καθώς και η εργασία "A MaxMin approach for hiding frequent itemsets" [09] των G. V. Moustakides και V. S. Verykios αποτέλεσαν την πηγή μελέτης των αλγόριθμων Max-Min 1 και Max-Min 2. Και οι δύο αλγόριθμοι είναι ευριστικοί και βασίζονται στο κριτήριο Max-Min, ενώ χρησιμοποιούν το αναθεωρημένο θετικό όριο των συχνών στοιχειοσυνόλων, για να εξετάσουν την επίδραση που έχει τυχόν διαγραφή του κάθε υποψήφιου θύματος-στοιχείου, ενώ οι Xingzhi Sun και Philip S. Yu στην εργασία τους "Hiding Sensitive Frequent Itemsets by a Border-Based Approach" [13] παρουσιάζουν τον αλγόριθμο Border-Based Approach που είναι και αυτός ευριστικός αλλά αναθέτει συντελεστές βάρους σε κάθε υποψήφιο στοιχειοσύνολο – θύμα προκειμένου να διερευνήσει την επίδραση τυχόν διαγραφής του στο αναθεωρημένο θετικό όριο των συχνών στοιχειοσυνόλων, και τελικά να επιλέξει αυτό με την μικρότερη επίδραση.

Πέρα όμως από τους αλγορίθμους που υλοποιήθηκαν, μια μελέτη πάνω στο αντικείμενο της παρούσης Μεταπτυχιακής διατριβής δεν θα μπορούσε να μη συμπεριλάβει την εργασία των Agrawal, et al. [02] που εισήγαγαν την έννοια της Διατήρησης της Ιδιωτικότητας κατά την Εξόρυξη Δεομένων (Privacy Preserving Data Mining) το έτος 2000, καθώς και των Atallah et al. [03] που ήταν οι πρώτοι που εισήγαγαν τον όρο Απόκρυψη Κανόνων Συσχέτισης και πρότειναν κάποιο συγκεκριμένο αλγόριθμο για την απόκρυψη ευαίσθητων κανόνων συσχέτισης. Άλλες εργασίες που μελετήθηκαν και μπορούμε ν' αναφέρουμε είναι αυτές των Dasseni et al. [06] οι οποίοι μελέτησαν εντός κοινού πλαισίου τις δύο παραλλαγές του προβλήματος της απόκρυψης κανόνων συσχέτισης και ευαίσθητων συχνών στοιχειοσυνόλων, των Verykios et al. [15] που επέφεραν βελτιώσεις στους αλγορίθμους που εισήγαγαν οι προαναφερθέντες, την εργασία των Oliveira & Zaïane [10] που ήταν οι πρωτοπόροι στην εισαγωγή μεθόδων απόκρυψης πολλαπλών κανόνων σε κάθε επανάληψη του αλγορίθμου. Συνεχίζοντας, οι Pontikakis et al. [11] πρότειναν δύο ευριστικούς αλγορίθμους που βασίζονται στην *παραμόρφωση* της αρχικής βάσης δεδομένων ώστε να αποκρύψουν συγκεκριμένους ευαίσθητους κανόνες. Οι Saygin et al. [16] [17] πρότειναν για πρώτη φορά το *μπλοκάρισμα* (*blocking*) αντί της παραμόρφωσης της αρχικής βάσης δεδομένων, με τη χρησιμοποίηση αγνώστων τιμών (?) στις συναλλαγές αντί της εισαγωγής παραποιημένων δεδομένων, αν και οι Pontikakis et al [12] ισχυρίστηκαν ότι η τεχνική του μπλοκαρίσματος μειονεκτεί έναντι της παραμόρφωσης καθώς από τη τελική βάση δεδομένων που προκύπτει, η οποία δεν έχει ουσιαστικά τροποποιηθεί από την αρχική, είναι δυνατή η αποκάλυψη των κρυμμένων κανόνων. Τέλος, το θεωρητικό πλαίσιο αξιολόγησης των αλγορίθμων παρουσιάζεται διεξοδικά στην εργασία των Bertino et. al [04] και αποτέλεσε πηγή και για την παρούσα μεταπτυχιακή διατριβή.

Κεφάλαιο 2
Ορισμοί – Θεμελιώδεις έννοιες

Στο κεφάλαιο αυτό δίνουμε την ορολογία και το υπόβαθρο που είναι απαραίτητα για την κατανόηση εκ μέρους του αναγνώστη, της Απόκρυψης Κανόνων Συσχέτισης. Σημειώνουμε ότι καθώς η διαδικασία εξόρυξης Σ.Σ. και Κ.Σ αποτέλεσε γνώση υποβάθρου κατά την συγγραφή της παρούσης Μεταπτυχιακής διατριβής, περιγράφεται με συντομία και μόνο στον βαθμό που αφορά τη σημειολογία που θα ακολουθηθεί στα παρακάτω κεφάλαια, με εξαίρεση τη σημειολογία που έχει προφανή ή διαισθητικό χαρακτήρα η οποία παραλείπεται.

Το κεφάλαιο ξεκινάει με την ενότητα 2.1 όπου δίνονται οι θεμελιώδεις έννοιες και η αντίστοιχη σημειολογία που χρησιμοποιούνται στην Εξόρυξη των Κανόνων Συσχέτισης. Στην παράγραφο 2.2 επεξηγούνται οι δύο παραλλαγές του ίδιου προβλήματος δηλαδή η απόκρυψη κανόνων και συχνών στοιχειοσυνόλων ενώ στην επόμενη 2.3 δίνουμε επακριβώς τους στόχους της Απόκρυψης Κανόνων Συσχέτισης. Τέλος, καθώς θεωρούμε ότι είναι σημαντικό για την κατανόηση των αλγορίθμων που έχουν υλοποιηθεί στην παρούσα μεταπτυχιακή διατριβή, παρατίθενται τα απαραίτητα στοιχεία της Θεωρίας Συνόρων στην παράγραφο 2.4.

2.1 Θεμελιώδεις έννοιες

Η έννοια της Α.Κ.Σ. εισήχθη από τους Atallah, et al. [03], όπως έχει προαναφερθεί, το 1999. Κατά την Α.Κ.Σ. αρχικά γίνεται εξόρυξη όλων των συχνών Σ.Σ. και κατόπιν, χρησιμοποιώντας αυτά τα Σ.Σ., βρίσκονται οι Κ.Σ. σύμφωνα με τα κριτήρια (*confidence*) που έχουν τεθεί.

Το πρόβλημα της εξόρυξης συχνών Σ.Σ. μπορεί να ορισθεί ως ακολούθως: Έστω $I=\{i_1,i_2,\ldots,i_M\}$ ένα πεπερασμένο σύνολο Μ στοιχείων. Οποιοδήποτε υποσύνολο $\mathtt{I} \subseteq$ I ονομάζεται στοιχειοσύνολο (Σ.Σ.) του I. Ένα *k*-στοιχειοσύνολο είναι ένα Σ.Σ. μήκους *k* δηλαδή αποτελείται από *k* στοιχεία. Ως συναλλαγή T_n του I λογίζεται το ζεύγος $T_n=(tid, \mathtt{I})$, όπου tid είναι ένας μοναδικός αριθμός που δίδουμε σε κάθε μία συναλλαγή της Β.Δ. D η οποία, συνεπώς, αποτελείται από ένα σύνολο συναλλαγών, δηλαδή $D=\{T_1, T_2, \ldots, T_N\}$. Μία συναλλαγή $T=(tid, J)$ λέγεται ότι υποστηρίζει ένα Σ.Σ. $\mathtt{I}$ του I εάν $\mathtt{I} \subseteq$ J. Επίσης, με D_I συμβολίζουμε τις συναλλαγές της Β.Δ. D που υποστηρίζουν το Σ.Σ. $\mathtt{I}$ ενώ με S ένα οποιοδήποτε σύνολο στοιχείων. Ο συμβολισμός *P=P(I)* αναφέρεται στο σύνολο των δυνατών υποσυνόλων του I και είναι γνωστό και ως δυναμοσύνολο του *I*, ενώ με sup($\mathtt{I}$, D) ή πιο σύντομα sup($\mathtt{I}$) αναφερόμαστε στον αριθμό των συναλλαγών $T \in$ D που υποστηρίζουν το στοιχειοσύνολο $\mathtt{I}$. Ένα Σ.Σ. $\mathtt{I}$ λέγεται συχνό Σ.Σ. αν και μόνο αν $sup(\mathtt{I}, D) \geq msup$. Το σύνολο των συχνών στοιχειοσυνόλων το συμβολίζουμε με F_D και είναι $F_D=\{\mathtt{I} \subseteq I : sup(\mathtt{I}, D)\}$, ενώ όλα τα υπόλοιπα Σ.Σ είναι τα μη συχνά ή σπάνια Σ.Σ. Το πρόβλημα της εξόρυξης Κ.Σ. περιλαμβάνει το παραπάνω βήμα, δηλαδή την εξόρυξη των Σ.Σ. και την διαδικασία της αναγνώρισης των σημαντικών Κ.Σ. ανάμεσά τους. Ένας Κ.Σ. είναι σημαντικός όταν $sup(\mathtt{I} \cup J, D) / sup(\mathtt{I}, D) \geq mconf$.

2.2 Οι δύο προσεγγίσεις του προβλήματος

Όπως αναφέρθηκε και στην εισαγωγή το πρόβλημα της Α.Κ.Σ. παρουσιάζεται σε δύο παραλλαγές ή προσεγγίσεις.

Η πρώτη παραλλαγή στοχεύει στην απόκρυψη συγκεκριμένων συχνών Σ.Σ. από αυτά που εξορύσσονται από την αρχική Β.Δ. ενώ σύμφωνα με τη δεύτερη παραλλαγή αποκρύπτονται συγκεκριμένοι Κ.Σ. από αυτούς που εξορύσσονται από την αρχική βάση δεδομένων (Β.Δ.), Οι δύο παραλλαγές του προβλήματος είναι σχεδόν ισοδύναμες, ως προς την ουσία και την φύση τους, αφού όλοι οι Κ.Σ. παράγονται από στοιχειοσύνολα, οπότε αποκρύπτοντας τα ευαίσθητα

στοιχειοσύνολα που παράγουν τους συγκεκριμένους Κ.Σ. επιτυγχάνεται το ζητούμενο αποτέλεσμα.

2.2.1. Απόκρυψη ευαίσθητων στοιχειοσυνόλων.

Για την πρώτη παραλλαγή του προβλήματος, ας υποθέσουμε ότι έχουμε μία Β.Δ. D_O η οποία αποτελείται από Ν συναλλαγές και ένα κατώφλι υποστήριξης *msup* το οποίο έχει καθορισθεί από τον ιδιοκτήτη της Β.Δ. Μετά την εκτέλεση της λειτουργίας της εξόρυξης συχνών Σ.Σ. προκύπτει ένα σύνολο Σ.Σ. F_{D_O},τα οποία ικανοποιούν το κατώφλι υποστήριξης, και έστω ένα υποσύνολό της S το οποίο περιέχει κάποια Σ.Σ. που έχουν χαρακτηρισθεί ως ευαίσθητα. Ο στόχος της Α.Κ.Σ. είναι η δημιουργία μιας νέας Β.Δ. D η οποία μπορεί να χαρακτηρισθεί ως «αποστειρωμένη» και η οποία έχει την ιδιότητα, όταν εφαρμόζονται οι μέθοδοι της εξόρυξης, για τιμές υποστήριξης μεγαλύτερες ή ίσες από την *msup*, αφενός μεν να μην αποκαλύπτονται τα ευαίσθητα Σ.Σ., αφετέρου δε, όλα τα συχνά - μη ευαίσθητα Σ.Σ. (δηλαδή τα Σ.Σ. του συνόλου F_{D_O} - S) να εμφανίζονται. Για να το επιτύχει αυτό, ο αλγόριθμος Α.Κ.Σ. θα πρέπει να τροποποιήσει κατάλληλα κάποιες από τις συναλλαγές που περιέχει η αρχική Β.Δ. μειώνοντας τη υποστήριξη ειδικά επιλεγμένων Σ.Σ.

2.2.2 Απόκρυψη ευαίσθητων Κανόνων Συσχέτισης.

Για την δεύτερη παραλλαγή του προβλήματος, ας υποθέσουμε ότι έχουμε μία Β.Δ. D_O η οποία αποτελείται από Ν συναλλαγές και ένα κατώφλι υποστήριξης *msup* και εμπιστοσύνης *mconf*, τα οποία έχουν καθορισθεί από τον ιδιοκτήτη της Β.Δ. Μετά την εκτέλεση της λειτουργίας της εξόρυξης Κ.Σ. προκύπτει το σύνολο των Κ.Σ. R που ικανοποιούν το κατώφλι υποστήριξης και εμπιστοσύνης, και έστω ένα υποσύνολο κανόνων R_S το οποίο περιέχει κάποιους Κ.Σ. που έχουν χαρακτηρισθεί ως ευαίσθητοι. Το R_S εμπεριέχεται στο R. Κατά παρόμοιο τρόπο με την πρώτη παραλλαγή, στόχος της Α.Κ.Σ. είναι η δημιουργία μιας νέας Β.Δ. D η οποία μπορεί να χαρακτηριστεί ως «αποστειρωμένη» και η οποία έχει την ιδιότητα, όταν εφαρμόζονται οι μέθοδοι της εξόρυξης, για τιμές υποστήριξης ή εμπιστοσύνης μεγαλύτερες ή ίσες από τις αντίστοιχες τιμές κατωφλίου, αφενός μεν να μην αποκαλύπτει τους ευαίσθητους Κ.Σ., αφετέρου δε, όλοι οι συχνοί - μη ευαίσθητοι Κ.Σ. (δηλαδή οι Κ.Σ. του συνόλου $R - R_S$) να φανερώνονται.

2.3 Στόχοι της Α.Κ.Σ

Οι μεθοδολογίες και τεχνικές που εφαρμόζονται στην Απόκρυψη Κανόνων Συσχέτισης, στοχεύουν στο να τροποποιήσουν την αρχική βάση δεδομένων, με τέτοιον τρόπο ώστε να επιτυγχάνεται ένας ή και περισσότεροι από τους παρακάτω στόχους:

1. Κανένας από τους ευαίσθητους Κ.Σ. ή τα ευαίσθητα Σ.Σ. δεν πρέπει να εξάγονται από την τροποποιημένη βάση δεδομένων, όταν σ' αυτήν πραγματοποιείται εξόρυξη δεδομένων κάτω από τις ίδιες ή ψηλότερες τιμές υποστήριξης και εμπιστοσύνης *msup* και *minconf*. Σε αντίθετη περίπτωση μέρος της ευαίσθητης γνώσης θα αποκαλυφθεί με ανυπολόγιστες, ενίοτε, συνέπειες. Συνεπώς, ο πρώτος στόχος θα πρέπει πάντοτε και οπωσδήποτε να εκπληρώνεται.

2. Όλοι οι μη ευαίσθητοι Κ.Σ. ή τα μη ευαίσθητα Σ.Σ. που εξάγονταν από την αρχική βάση δεδομένων κάτω από συγκεκριμένες τιμές υποστήριξης και εμπιστοσύνης, να μπορούν να εξάγονται επιτυχώς και από την τροποποιημένη βάση και κάτω από τις ίδιες τιμές υποστήριξης και εμπιστοσύνης. Είναι προφανές ότι όσο περισσότεροι μη ευαίσθητοι κανόνες δεν παραχθούν από την τροποποιημένη Β.Δ. τόσο μεγαλύτερη απώλεια γνώσης θα έχει ο κάτοχός της.

3. Κανένας Κ.Σ. ή κανένα Σ.Σ. που δεν είχε εξαχθεί από την αρχική βάση κάτω από συγκεκριμένες τιμές κατωφλίων *msup* και *minconf*, να μην εξάγεται από την τροποποιημένη βάση (*ghost rule*) με τις ίδιες ή ψηλότερες τιμές υποστήριξης και εμπιστοσύνης, καθώς κάτι τέτοιο θα αντιπροσώπευε ψευδή γνώση. Τόσο η απώλεια μη ευαίσθητων Κ.Σ. ή Σ.Σ. στη νέα βάση δεδομένων, όσο και η εμφάνιση ορισμένων που δεν υπήρχαν στην αρχική, ονομάζονται *παρενέργειες (side effects).* Τέλος, αξίζει να σημειωθεί ότι οι παρενέργειες αποτελούν μία από τις πλέον συνηθισμένες μετρικές αξιολόγησης ενός αλγορίθμου Α.Κ.Σ. και χρησιμοποιείται και στην παρούσα μεταπτυχιακή διατριβή.

2.4 Στοιχεία Θεωρίας Συνόρων (Border Theory)

Η θεωρία των συνόρων [08] των συχνών Σ.Σ έχει παίξει σημαντικό ρόλο στην εκπόνηση της παρούσης Μεταπτυχιακής διατριβής καθώς οι αλγόριθμοι που έχουν υλοποιηθεί και ενσωματωθεί στο εργαλείο είναι βασισμένοι πάνω σ' αυτή τη θεωρία. Επίσης, πολλές ευριστικές και ακριβείς προσεγγίσεις βασίζονται πάνω σ' αυτή τη θεωρία. Για το λόγο αυτό μελετήθηκε ιδιαίτερα και παρατίθεται εδώ εν' συντομία.

2.4.1 Ορισμοί της Θεωρίας Συνόρων

Αρχικό Σύνορο *(Original Border)* ενός συνόλου από στοιχειοσύνολα είναι το νοητό επίπεδο που το χωρίζει σε δύο υποσύνολα. Στο σύνολο των συχνών στοιχειοσυνόλων F_{Do} από τα μη συχνά στοιχειοσύνολα $P - F_{Do}$.

Το θετικό σύνορο του συχνού Σ.Σ. F_D είναι αυτό που αποτελείται από τα Σ.Σ. του F_D των οποίων όλα τα κατάλληλα υπέρ-σύνολα (supersets) είναι μη συχνά. Ο συμβολισμός του θετικού συνόρου του F_D είναι $Bd^+(F_D)$ [08] και με μαθηματικούς όρους εκφράζεται ως [08] [07]

$$Bd^+(F_D) = \{I \in F_D \mid for\ all\ J \in P\ with\ I \subset J \Rightarrow J \notin F_D\}$$

Αντίστοιχα, το αρνητικό σύνορο του συχνού Σ.Σ. είναι αυτό που αποτελείται από όλα τα ελάχιστα σπάνια Σ.Σ. του P, ήτοι από τα Σ.Σ. του $P \backslash F_D$ των οποίων όλα τα κατάλληλα υποσύνολα είναι συχνά. Ο συμβολισμός του αρνητικού συνόρου του F_D είναι $Bd^-(F_D)$ [08] και με μαθηματικούς όρους εκφράζεται ως [08] [07]

$$Bd^-(F_D) = \{I \in P \backslash F_D \mid for\ all\ J \subset I \Rightarrow J \in F_D\}$$

Τέλος, το σύνορο του συχνού Σ.Σ. F_D, το οποίο ορίσαμε προηγουμένως, συμβολίζεται ως $Bd(F_D)$ [08] είναι η ένωση του θετικού και του αρνητικού συνόρου, δηλαδή [08][07]

$$Bd(F_D) = Bd^+(F_D) \cup Bd^-(F_D)$$

Επίσης, εάν θεωρήσουμε το σύνολο των ευαίσθητων Σ.Σ. S που πρέπει να αποκλεισθούν από το σύνολο των Σ.Σ. τότε ορίζουμε ως ελάχιστο σύνολο ευαίσθητων Σ.Σ. S_{min} και αντίστοιχα ως μέγιστο σύνολο ευαίσθητων Σ.Σ. S_{max} τα παρακάτω [08] [07]

$$S_{max} = \{I \in F_{D_o} \mid \exists J \in S_{min}, J \subseteq I\}, \text{όπου}$$

$$S_{min} = \{I \in S \mid for\ all\ J \subset I,\ J \notin S\}$$

2.4.2 Α.Κ.Σ. με χρήση της Θεωρίας Συνόρων

Η Θεωρία Συνόρων [08] μπορεί να χρησιμοποιηθεί για την Α.Κ.Σ. ή συχνών Σ. Σ μέσω της αναθεώρησης του συνόρου *(Border Revision)*. Με απλά λόγια η διαδικασία της Αναθεώρησης Συνόρων (Α. Σ.) έχει σαν σκοπό να εντοπίσει και διαχωρίσει τα συχνά Σ.Σ. που πρέπει να παραμείνουν να είναι συχνά, από τα συχνά Σ.Σ. που, μέσα από την διαδικασία της απόκρυψης, θα πρέπει να γίνουν σπάνια. Στην ιδανική περίπτωση, η οποία αντιστοιχεί στην Α.Κ.Σ χωρίς απώλεια μη ευαίσθητων συχνών Κ.Σ., το σύνορο θα πρέπει να αναθεωρηθεί ώστε να αποκλεισθούν από τα συχνά Σ.Σ. τα τυχόν ευαίσθητα Σ.Σ. καθώς και τα υπέρ-σύνολά τους. Διατυπώνοντας τα παραπάνω με μαθηματικούς όρους, αυτό που ζητούμε είναι τα Σ.Σ. [08][07]

$$F'_D = F_{Do} - S_{max}$$

όπου τα S_{max} και S_{min} έχουν ορισθεί στην προηγούμενη παράγραφο.

Για την εύρεση του βέλτιστου συνόρου, θα πρέπει να υπολογισθούν τα μεγέθη «αναθεωρημένο θετικό σύνορο» $Bd^+(F'_D)$ και «αναθεωρημένο αρνητικό σύνορο» $Bd^-(F'_D)$ και κατόπιν η ένωσή τους, [08][07] δηλαδή

$$Bd(F'_D) = Bd^+(F'_D) \cup Bd^-(F'_D).$$

Το μόνο πλέον ζητούμενο είναι η εύρεση ενός τρόπου για να τροποποιήσουμε την αρχική Β.Δ. ώστε μετά την εφαρμογή των μεθόδων εξόρυξης, για το δοθέν *msup*, το σύνολο των συχνών Σ.Σ. που θα προκύπτει να είναι το F'_D , το οποίο υλοποιείται με διαφορετικό τρόπο από τον κάθε αλγόριθμο Α.Κ.Σ.

Κεφάλαιο 3

Αλγόριθμοι Απόκρυψης Κανόνων Συσχέτισης

Στο κεφάλαιο αυτό παρουσιάζονται οι τρεις αλγόριθμοι Α.Κ.Σ που έχουν υλοποιηθεί στο πλαίσιο της παρούσης Μεταπτυχιακής διατριβής. Οι τρεις αλγόριθμοι, οι οποίοι έχουν ενσωματωθεί στο εργαλείο που αναπτύχθηκε, είναι ο Border Based Approach των Sun and Yun [13], ο οποίος είναι ο πρώτος αλγόριθμος που βασίζεται στην έννοια του συνόρου, και οι δύο Max Min αλγόριθμοι των Moustakides and Verykios [09] οι οποίοι εισάγουν την έννοια του κριτηρίου Max-Min. Για κάθε έναν από τους τρεις αλγορίθμους παρουσιάζεται ένα παράδειγμα εκτέλεσής τους, δίδοντας τα αποτελέσματα όπως αυτά προκύπτουν από την υλοποίηση των αλγορίθμων μέσω της εφαρμογής που έχει αναπτυχθεί.

Στο τέλος του Κεφαλαίου θα επιχειρηθεί μια συγκριτική παρουσίαση των τριών αλγορίθμων παρουσιάζοντας τα δυνατά και αδύνατα σημεία τους.

3.1 Αλγόριθμοι Max-Min

Η πορεία των δύο αλγορίθμων είναι κοινή μέχρις ενός σημείου, αφού και οι δύο βασίζονται στην έννοια του κριτηρίου Max-Min. Στην συνέχεια η πορεία αυτή διαχωρίζεται καθώς ο Max-Min 2 διαφοροποιείται αισθητά εισάγοντας μια σειρά από βελτιώσεις. Οι αλγόριθμοι Max-Min καταφέρνουν να αποκρύπτουν ευαίσθητα συχνά Σ.Σ., ελαχιστοποιώντας ταυτόχρονα την αρνητική επίδραση αυτής της διαδικασίας στα μη ευαίσθητα δεδομένα. Η επίτευξη αυτού του στόχου επιτυγχάνεται μέσω της εξέτασης της επίδρασης, που έχει η διαδικασία απόκρυψης, στο Αναθεωρημένο Θετικό Σύνορο (Revised Positive Border - στο εξής Α.Θ.Σ) όπως παρουσιάζεται παρακάτω.

Ας υποθέσουμε ότι το ευαίσθητο Σ.Σ. {a,b,d} πρέπει να αποκρυφτεί και ότι το Α.Θ.Σ. αποτελείται από τα Σ.Σ. {a,b}, {b,d}, {a,c,d}, {c,d,e}. Για κάθε στοιχείο του συνόλου των ευαίσθητων στοιχείων δηλαδή των a, b και d δημιουργούμε μια δομή δεδομένων που αποκαλούμε λίστα συγγένειας (affinity list) με Σ.Σ. του Α.Θ.Σ. που σχετίζονται με το τρέχον ευαίσθητο στοιχείο. Έτσι θα έχουμε την παρακάτω δομή:

a	a, b /4	a, c, d/4	
b	a, b /4	b, d/ 5	
d	b, d/ 5	a, c, d/ 4	c, d, e/ 3

Πίνακας 3.1: Λίστα Συγγένειας του Σ.Σ {a,b,d}

Τα στοιχεία a ή b ή d κ. ο. κ που ανήκουν σε κάποιο ευαίσθητο Σ.Σ. ονομάζονται *υποψήφια στοιχεία -θύματα (tentative victim item*), ενώ το ίδιο το ευαίσθητο Σ.Σ. ονομάζεται «υποψήφιο Σ.Σ. – θύμα» (*tentative victim itemset*). Τέλος, το σύνολο των υποψήφιων Σ.Σ. -θυμάτων που αντιστοιχούν σε ένα υποψήφιο στοιχείο -θύμα, το ονομάζουμε «λίστα υποψήφιων Σ.Σ. –θυμάτων» (*vi-list*). Για παράδειγμα η vi-list για το υποψήφιο στοιχείο -θύμα a είναι {a,b}, {a,c,d}. Τώρα δημιουργείται μια νέα δομή δεδομένων η οποία περιλαμβάνει τα Σ.Σ. με την μικρότερη υποστήριξη από κάθε μία *vi-list*. Η δομή αυτή αποκαλείται «Σ.Σ. Ελάχιστης Υποστήριξης» (*minimum support itemsets*). Ο λόγος για τον οποίο επιλέγονται αυτά τα Σ.Σ. είναι διότι αποτελούν τα περισσότερο ευαίσθητα δεδομένα μιας *vi-list* καθώς βρίσκονται εγγύτερα στην οριογραμμή ανάμεσα στο Αναθεωρημένα Θετικό και Αρνητικό Σύνορα. Επιπλέον, για τον ίδιο λόγο η απόκρυψή τους είναι ευκολότερο να επιτευχθεί. Με βάση τα παραπάνω τα *minimum support itemsets* για το υποψήφιο Σ.Σ. -θύμα a θα είναι τα {a,b}, {a,c,d} καθώς και τα δύο έχουν την ίδια τιμή υποστήριξης, για το b θα είναι το {a,b}, ενώ για το

d το {c,d,e}. Τέλος, από όλα τα *minimum support itemsets* επιλέγουμε εκείνο με την μεγαλύτερη υποστήριξη. Το Σ.Σ. αυτό το αποκαλούμε *maxmin itemset* (*maxmin* Σ.Σ.). Στο απλοϊκό παράδειγμα μας το *maxmin itemset* θα προκύψει από την ένωση των συνόλων {a,b}, {a,c,d} και {a,b}. Οπότε το *maxmin itemset* θα αποτελείται από τα σύνολα {a,b}, {a,c,d} με τιμή υποστήριξης 4. Το *maxmin itemset* θα καθορίσει πιο από τα υποψήφια στοιχεία–θύματα θα αποτελέσει το τελικό στοιχείο-θύμα (*victim item*) και για τον λόγο αυτό καλείται κριτήριο *maxmin*. Στο παράδειγμά μας, η παραπάνω τεχνική υποδεικνύει ότι ένα από τα στοιχεία a ή b θα αποτελέσουν το «θύμα» καθώς τα *maxmin* Σ.Σ. ανήκουν στις vi-list των στοιχείων a και b. Είναι εμφανές ότι η επίδραση της διαγραφής ενός στοιχείο-θύματος από κάποια συναλλαγή της D_o , που υποστηρίζει το ευαίσθητο Σ.Σ, στην υποστήριξη των *maxmin* Σ.Σ. θα είναι η μικρότερη καθώς η απόσταση αυτού από το σύνορο είναι η μεγαλύτερη.

Από τη διαδικασία που περιγράφηκε παραπάνω είναι δυνατόν να προκύψει ένα maxmin Itemset ή και περισσότερα από ένα. Στο σημείο αυτό και ανάλογα της τιμής maxmin που θα προκύψει, οι δύο αλγόριθμοι Max-Min 1 και Max-Min 2 συνεχίζουν με διαφορετική πορεία. Στην περίπτωση κατά την οποία το Σ.Σ. Max-Min, που θα προκύψει από την παραπάνω διαδικασία, συμμετέχει σε ένα και μοναδικό «υποψήφιο Σ.Σ. –θύμα» (tentative victim item-set) με συνέπεια να υπάρχει ακριβώς ένα «στοιχείο – θύμα» (*victim item*) τότε χρησιμοποιείται αποκλειστικά ο αλγόριθμος Max-Min 1. Στην περίπτωση, όμως, κατά την οποία το Σ.Σ. Max-Min, που θα προκύψει από την παραπάνω διαδικασία, συμμετέχει σε περισσότερα από ένα «υποψήφιο Σ.Σ. – θύμα» (tentative victim item-set) με συνέπεια να υπάρχουν περισσότερα από ένα «στοιχείο – θύμα» (*victim item*) που θα μπορούσε να χρησιμοποιηθεί, τότε ο Max-Min 1 επιλέγει τυχαία το «στοιχείο –θύμα», ενώ ο Max-Min 2 εφαρμόζει μια βελτιστοποιημένη, πλην όμως περισσότερο πολύπλοκη διαδικασία καθορισμού «στοιχείο – θύματος». Έτσι, στο παραπάνω παράδειγμα, ο Max-Min 1 θα επιλέξει τυχαία «στοιχείο – θύμα» ανάμεσα στα a και b.

3.1.1. Ο αλγόριθμος Max-Min 1

Στο παρακάτω σχήμα παρουσιάζεται ο Αλγόριθμος Max-Min 1. Παρατηρούμε ότι ο αλγόριθμος ξεκινάει με ταξινόμηση των ευαίσθητων Σ.Σ. με βάση την τιμή της υποστήριξης. Γενικά, δοθέντος ενός συνόλου από ευαίσθητα Σ.Σ. προς απόκρυψη, είναι πολύ σημαντική, για την ποιότητα της «καθαρισμένης» Β.Δ., η σειρά με την οποία θα γίνει η απόκρυψη. Παρατηρούμε ότι το βήμα της ταξινόμησης επαναλαμβάνεται για κάθε Σ.Σ. του S. Ο λόγος είναι ότι διαφορετικά ευαίσθητα Σ.Σ.

ενδέχεται να υποστούν διαφορετική μείωση της υποστήριξής τους κατά την διαδικασία της απόκρυψης. Ωστόσο, μια σημαντική παρατήρηση που οι συγγραφείς αποδεικνύουν στο [09] είναι ότι κατά την διάρκεια απόκρυψης του Σ.Σ. με την μικρότερη υποστήριξη, αυτή παραμένει συνεχώς χαμηλότερη από την υποστήριξη των υπολοίπων ευαίσθητων Σ.Σ. Επίσης, σε περίπτωση ύπαρξης δύο Σ.Σ. της ίδιας υποστήριξης, πρώτα αποκρύπτεται αυτό με το μεγαλύτερο μήκος καθώς με αυτήν την επιλογή επιτυγχάνεται μεγαλύτερος βαθμός ελευθερίας όσον αφορά την επιλογή στοιχείο –θύματος. Ο αλγόριθμος συνεχίζεται με την εύρεση της λίστας συγγένειας για κάθε ευαίσθητο Σ.Σ. και στη συνέχεια εφαρμόζεται το κριτήριο Max-Min για την εύρεση του στοιχείο –θύματος και ακολουθεί διαγραφή του από την πρώτη συναλλαγή της Β.Δ. που υποστηρίζει το προς απόκρυψη Σ.Σ. Εν' συνεχεία αναθεωρείται η λίστα συγγένειας καθώς τα υποψήφια στοιχείο-θύματα ενδέχεται πλέον να έχουν διαφορετική τιμή υποστήριξης, η Β.Δ. αντικαθίσταται από την νέα που προκύπτει μετά την διαγραφή του στοιχείο – θύματος και η διαδικασία επαναλαμβάνεται μέχρις ότου η υποστήριξη του ευαίσθητου Σ.Σ. πέσει κάτω από το κατώφλι που έχει τεθεί οπότε και θα έχει επιτευχθεί η απόκρυψή του.

The Max-Min 1 Algorithm of Moustakides & Verykios [09]

1: function MAX-MIN 1 (Original database $\mathcal{D}_\sigma$ revised positive border $\mathcal{B}\mathrm{d}^+$, sensitive itemsets S, minimum support threshold *msup*)
2: $\mathcal{D}' \leftarrow \mathcal{D}_\sigma$
3: **while** $S \neq \emptyset$ **do**
4: Select $I \in S$ with minimum support, breaking ties in favor of maximum length
5: For each tentative victim item $j \in I$ compute its tentative victim itemsets $\mathcal{B}\mathrm{d}^+ |_j$
6: while sup (I, $\mathcal{D}'$)$\geq$ *msup* **do**
7: Compute the Max-Min itemset, splitting ties arbitrarily
8: Remove victim item $i \in I$, determined by the Max-Min itemset from the first transaction that supports the sensitive itemset I
9: Revise the tentative victim intersets
10: **end while**
11: Remove I from S
12: **end while**
13: **Return:** sanitized database $\mathcal{D} \leftarrow \mathcal{D}'$
14: **end function**

Σχήμα 3.1: Αλγόριθμος Max-Min 1

3.1.2. Ο αλγόριθμος Max-Min 2

Ο αλγόριθμος Max-Min 2 εισάγει μια σειρά από βελτιώσεις σε σχέση με τον Max-Min 1 όσον αφορά την επιλογή του στοιχείο – θύματος, ελαχιστοποιώντας τις παράπλευρες απώλειες στα μη ευαίσθητα Σ.Σ. Ο αλγόριθμος βρίσκει εφαρμογή στην περίπτωση κατά την οποία τα υποψήφια στοιχείο-θύματα είναι περισσότερα από ένα, οπότε θα πρέπει να γίνει επιλογή. Έχουμε δει παραπάνω ότι ο αλγόριθμος Max-Min 1 κάνει τυχαία επιλογή. Αντιθέτως ο Max-Min 2 αντιδρά σύμφωνα με 3 διαφορετικά σενάρια, που αναλύονται παρακάτω, και επιλέγει βάσει κριτηρίων το βέλτιστο στοιχείο-θύμα. Βεβαίως, στην περίπτωση του ενός στοιχείο – θύματος γίνεται η διαγραφή του από την Β.Δ. με βάση αυτά που περιγράφηκαν στην παραπάνω παράγραφο. Ο ψευδοκώδικας του αλγορίθμου Max-Min 2 παρουσιάζεται στο σχήμα 3.2.

A. Κατά το πρώτο σενάριο περισσότερα από ένα Max-Min Σ.Σ. αντιστοιχούν σε ένα υποψήφιο στοιχείο – θύμα (δηλαδή το υποψήφιο στοιχείο – θύμα συνάγεται από Σ.Σ. που βρίσκονται όλα στην ίδια vi-list της λίστας συγγένειας). Σ' αυτή τη περίπτωση ο Max-Min 2 προσπαθεί να ελαττώσει την υποστήριξη του ευαίσθητου Σ.Σ. μέσω του εν' λόγω στοιχείου χωρίς να επηρεάζεται, ει δυνατόν, η υποστήριξη κανενός Max-Min Σ.Σ. Στην περίπτωση κατά την οποία αυτό είναι εφικτό, τότε όπως αποδεικνύεται στο [09] κανένα άλλο Σ.Σ. από αυτά που απαρτίζουν τα Σ.Σ. ελάχιστης υποστήριξης θα επηρεαστεί. Η ελάττωση της υποστήριξης ενός ευαίσθητου Σ.Σ. χωρίς να επηρεαστεί η υποστήριξη των Max-Min Σ.Σ. μπορεί να επιτευχθεί μόνο εάν πληρούνται οι παρακάτω δύο συνθήκες: α) Τα Max-Min Σ.Σ. δεν είναι υποσύνολα του ευαίσθητου Σ.Σ. και β) Υπάρχουν συναλλαγές στην Β.Δ. που υποστηρίζουν το ευαίσθητο Σ.Σ. χωρίς, ωστόσο, να υποστηρίζουν τα Max-Min Σ.Σ.

   Προκειμένου να βρεθεί αυτή η πληροφορία ο Max-Min 2 υπολογίζει, για κάθε ευαίσθητο Σ.Σ. και για κάθε Max-Min Σ.Σ. τις συναλλαγές στην αρχική Β.Δ. που τα υποστηρίζει. Ονομάζοντας L_I και L_U τις λίστες με τις αντίστοιχες συναλλαγές, τότε η διαφορά τους L_I - L_U υποδηλώνει το σύνολο των συναλλαγών της D_o που υποστηρίζουν το ευαίσθητο Σ.Σ. I χωρίς να υποστηρίζουν το Max-Min Σ.Σ. U. Εάν το μέγεθος του συνόλου που προκύπτει είναι μεγαλύτερο ή τουλάχιστον ίσο με την διαφορά *sup(I)-msup-1* τότε υπάρχουν συναλλαγές που επιτρέπουν την απόκρυψη του I χωρίς την ελάττωση της υποστήριξης του Max-Min Σ.Σ.

B. Το δεύτερο σενάριο εφαρμόζεται όταν περισσότερα από ένα Max-Min Σ.Σ. αντιστοιχούν σε διαφορετικά υποψήφια στοιχείο-θύματα. Σ' αυτή τη περίπτωση ο αλγόριθμος εξετάζει όλες τις vi-lists που περιέχουν Max-Min Σ.Σ. και ελέγχει εάν μπορεί να μειωθεί η υποστήριξη του ευαίσθητου Σ.Σ. χωρίς να επηρεάζεται η υποστήριξη κανενός από τα Max-Min Σ.Σ. της vi-list.

   Στην περίπτωση που το παραπάνω είναι εφικτό, τότε όπως αποδεικνύεται στο [09] κανένα άλλο Σ.Σ. σε οποιαδήποτε άλλη *vi-list* δεν θα επηρεαστεί. Προκειμένου να εξετασθεί εάν μπορεί να μειωθεί η υποστήριξη του ευαίσθητου Σ.Σ. χωρίς να επηρεαστεί η υποστήριξη κανενός από τα Max-Min Σ.Σ. της *vi-list*, ο αλγόριθμος διατρέχει την Β.Δ. ψάχνοντας για συναλλαγές που υποστηρίζουν το ευαίσθητο Σ.Σ. αλλά δεν υποστηρίζει κανένα από τα Max-Min Σ.Σ. στη *vi-list*. Οπότε υπολογίζεται και πάλι η διαφορά L_I - L_U και αν προκύψει σύνολο διαφορετικό του κενού, γίνεται διαγραφή του στοιχείου (που αντιστοιχεί στη *vi-list*). Αν σε Σ.Σ. της *vi-list* που εξετάζεται προκύπτει το κενό σύνολο, τότε το σενάριο αυτό δεν μπορεί να εφαρμοστεί και εφαρμόζεται το παρακάτω Γ. σενάριο.

C. Στην περίπτωση αυτή το σενάριο Β δεν μπορεί να εφαρμοστεί διότι L_I - L_U=Ø. Τότε ο Max-Min 2 εκτελεί επαναλήψεις εξετάζοντας όλους τους δυνατούς συνδυασμούς ζευγών *vi-list* για την ανεύρεση συναλλαγών που υποστηρίζουν τα Σ.Σ. ελάχιστης υποστήριξης της πρώτης λίστας και ταυτόχρονα δεν υποστηρίζουν κανένα από τα Max-Min Σ.Σ. της δεύτερης λίστας. Στην περίπτωση που βρεθούν τέτοιες συναλλαγές τότε το αντίστοιχο στοιχείο-θύμα διαγράφεται από (κάποια από) αυτές. Σε αντίθετη περίπτωση το στοιχείο-θύμα διαγράφεται από κάποια συναλλαγή που υποστηρίζει τα Σ.Σ. ελάχιστης υποστήριξης της πρώτης λίστας (σειρά 32 του ψευδοκώδικα)

The Max-Min 2 Algorithm of Moustakides & Verykios [09]

1: function MAX-MIN 2 (Original database $\mathcal{D}_\sigma$ revised positive border $\mathcal{B}\mathrm{d}^+$, sensitive itemsets S, minimum support threshold *msup*)

2: $\mathcal{D}' \leftarrow \mathcal{D}_\sigma$
3: **while** $S \neq \emptyset$ **do**
4: Select $I \in S$ with minimum support, breaking ties in favor of maximum length
5: For each tentative victim item $j \in I$ compute its tentative victim itemsets $\mathcal{B}\mathrm{d}^+ |_j$

6: **while** sup (I, $\mathcal{D}'$)≥ *msup* **do**

7: Compute the Max-Min itemset, splitting ties arbitrarily

8: ***if*** $\exists\, j \in I: max - \min \subseteq Bd^+|_j$ and $\forall\, i \neq j: Bd^+|_i \cap max - min = \emptyset$ *then*

9: **if** $L \leftarrow L_I - L_U \neq \emptyset$ **then**

10: delete j from a transaction of list L

11: **else**

12: delete j from a transaction of list L_I

13: **end if**

14: **else**

15: $K \leftarrow \{$ *tentative victim items* $j \in I : \mathcal{B}\mathrm{d}^+|_j \cap \max - \min \neq \emptyset\}$

16: **for** each k ∈ K **do**

17: $U \leftarrow max - min \cap \mathcal{B}\mathrm{d}^+|_k$

18: **if** $L \leftarrow L_I - L_U \neq \emptyset$ **then**

19: delete k from a transaction of list L

20: **break**

21: **end if**

22: **end for**

23: **if** $L \neq \emptyset$ **then**

24: **for** each $k_1 \in K$ **do**

25: **for** each $k_2 (\neq k_1) \in K$ **do**

26: $U_1 \leftarrow max - min \cap \mathcal{B}\mathrm{d}^+|_{k_1}$

27: $U_2 \leftarrow max - min \cap \mathcal{B}\mathrm{d}^+|_{k_2}$

28: $L \leftarrow (L_{U_1} \cap L_I) - (L_{U_2} \cap L_I)$

29: **if** $L \neq \emptyset$ **then**

30: delete k_1 from a transaction of list L

31: **else**

32: delete k_1 from a transaction of list $L_{U_1} \cap L_I$

33: **end if**

34: **end for**

35: **end for**

36: **end if**

37: **end if**

38: **end while**

39: Remove I from S

40: **end while**

41: **Return:** sanitized database $\mathcal{D} \leftarrow \mathcal{D}'$

42: **end function**

Σχήμα 3.2: Αλγόριθμος Max-Min 2

3.2 Ο Αλγόριθμος Border Based Approach (BBA)

Οι Sun & Yu [13] ήταν αυτοί που πρώτοι εισήγαγαν αλγόριθμο Α. Σ.Σ. βασιζόμενο στη Θεωρία Συνόρων. Πρόκειται για έναν ευριστικό αλγόριθμο, που σε αντίθεση με τους Max-Min που εξετάσθηκαν παραπάνω και οι οποίοι χρησιμοποιούν το Max-Min κριτήριο, αναθέτει κάποιον συντελεστή βάρους σε κάθε ένα Σ.Σ. του Α.Θ.Σ. προσπαθώντας με αυτό τον τρόπο να ποσοτικοποιήσει το πόσο ευάλωτο είναι σε μια πιθανή διαγραφή κάποιου στοιχείου. Ο συντελεστής βάρους δεν είναι στατικός αλλά μεταβάλλεται δυναμικά, κατά την διάρκεια της διαδικασίας απόκρυψης, σαν συνάρτηση της τρέχουσας υποστήριξης των.

Μια αξιοσημείωτη ιδιότητα του αλγορίθμου BBA είναι ότι προσπαθεί να διατηρήσει την σχέση υποστήριξης ανάμεσα στα μη ευαίσθητα συχνά Σ.Σ., πριν και μετά το τέλος της διαδικασίας απόκρυψης. Για παράδειγμα, αν θεωρήσουμε δύο μη ευαίσθητα, συχνά Σ.Σ. I και J, τα οποία στην αρχική Β.Δ. D_o είχαν την ιδιότητα $sup(I, D_o) > sup(J, D_o)$. Αν στην Β.Δ. που θα προκύψει μετά την απόκρυψη δεν ισχύει η παραπάνω σχέση, αλλά ή αντίθετη $sup(I, D') < sup(J, D')$, τότε κατά την εξόρυξη της D' με $msup= sup(J, D')$, προφανώς, το Σ.Σ. I θα καταγραφεί ως σπάνιο, ενώ θα έπρεπε να είχε καταγραφεί ως συχνό. Γενικά, είναι επιθυμητό για τα δύο συχνά Σ.Σ. I και J να ισχύει:

$$sup(I, D_o)- sup(J, D_o)= sup(I, D')- sup(J, D')$$

στον μεγαλύτερο δυνατό βαθμό.

Για να δείξουμε την διαδικασία της Απόκρυψης, ας θεωρήσουμε ένα ευαίσθητο Σ.Σ. I το οποίο ανήκει στο σύνολο των ελάχιστων Σ.Σ. S_{min}, όπως αυτό έχει ορισθεί στη παράγραφο 2.4.1. Αξίζει να σημειωθεί ότι, σε αντίθεση με τους Max-Min αλγορίθμους, τα στοιχεία του συνόλου S_{min}, αποκρύπτονται κατά φθίνουσα τάξη μήκους Σ.Σ. και αυξανόμενης τιμής υποστήριξης. Έστω, επίσης, C_I το σύνολο που περιέχει όλα τα ζεύγη (T, i) των συναλλαγών T και των στοιχείων i ∈ I στην αρχική Β.Δ. D_o, τα οποία στοιχεία I μπορούν να χρησιμοποιηθούν για την μείωση της υποστήριξης του Σ.Σ. I. Οι συγγραφείς του [05] ονομάζουν το C_I σύνολο *υποψήφιων απόκρυψης (hiding candidates)* του Σ.Σ. I, και ο μαθηματικός ορισμός του είναι:

$$C_I=\{ (T, i)| T \in D_I \wedge i \in I \}$$

Όπου D_I είναι το σύνολο των συναλλαγών που υποστηρίζουν το Σ.Σ. I. Χρησιμοποιώντας την παραπάνω σημειολογία, το ζευγάρι (T_o, i_o) εκφράζει το υποψήφιο προς διαγραφή στοιχείο i_o από

την συναλλαγή T_o της αρχική Β.Δ. D_o με στόχο τη μείωση της υποστήριξης του Ι. Όπως, όμως, έχουμε αναφέρει και παραπάνω, ο αλγόριθμος χρησιμοποιεί ένα σχήμα απόδοσης συντελεστή βαρύτητας με βάση το οποίο κάθε Σ.Σ. του Α.Σ.Θ. «μεταφέρει» μαζί του και το «βάρος» του με στόχο να επιτευχθεί η μικρότερη δυνατή επίδραση σε αυτό. Όσο μεγαλύτερος είναι ο συντελεστής βαρύτητας τόσο πιο ευάλωτο είναι ένα Σ.Σ. στο να επηρεαστεί από μια διαγραφή ενός στοιχείου. Ο συντελεστής βάρους ορίζεται ως εξής:

$$w(I \in Bd+) = \begin{cases} \dfrac{sup(I, Do) - sup(I, D') + 1}{sup(I, Do) - msup}, & sup(I, D') \geq msup + 1 \\ \lambda + msup - sup(I, D'), & 0 \leq sup(I, D') \leq msup \end{cases}$$

Όπου λ είναι ένας ακέραιος αριθμός με τιμή μεγαλύτερη από τον αριθμό των Σ.Σ. που απαρτίζουν το Α.Θ.Σ. Ο συντελεστής βαρύτητας, μεταβάλλεται δυναμικά κατά την λειτουργία του αλγορίθμου (σειρά 11 του ψευδοκώδικα), και το γεγονός αυτό επιτρέπει την διατήρηση της σχετικής υποστήριξης των Σ.Σ. στην καθαρισμένη Β.Δ. Η παρακάτω εικόνα παρουσιάζει τον αλγόριθμο BBA.

The Border Based Approach of Sun & Yu [13]

1: **function** BBA (Original database $\mathcal{D}_o$ frequent itemsets $\mathcal{F}_{\mathcal{D}o}$, sensitive itemsets S, minimum support threshold *msup*)
2: $\mathcal{D}' \leftarrow \mathcal{D}_o$
3: Compute S_{min} and $\mathcal{B}\mathrm{d}^+$
4: Sort items in S_{min} in decreasing order of length and increasing order of support
5: **for** each $I \in S_{min}$ **do**
6: **Compute** $\mathrm{Bd}^+|_I$ and w ($I \in \mathrm{Bd}^+|_I$)
7: Initialize the set $\mathrm{C}|_I$ of hiding candidates for itemset I
8: **for** $i = 0;\ i < \sup(I, \mathcal{D}') - msup + 1;\ i++$ **do**
9: Find hiding candidate $c = (T_o, i_o)$ with minimal impact in C
10: $C \leftarrow C - \{(T, i) | T = T_o\}$
11: Update $w(I \in \mathcal{B}\mathrm{d}^+|_I)$
12: **end for**
13: Update database $\mathcal{D}'$
14: **end for**
15: **Return:** sanitized database $\mathcal{D} \leftarrow \mathcal{D}'$
16: **end function**

Πίνακας 3.4: Αλγόριθμος BBA

3.3. Παραδείγματα εκτέλεσης των τριών αλγορίθμων

Η χρήση ενός παραδείγματος για κάθε έναν από τους τρεις αλγορίθμους απόκρυψης θα βοηθήσει στην κατανόηση της λειτουργίας των. Για τον σκοπό αυτό, ας θεωρήσουμε την παρακάτω Β.Δ., δέκα συναλλαγών (transactions) και οκτώ στοιχείων (item) που δημιουργήθηκε για τον σκοπό αυτό, στον πίνακα 3.3a. Επίσης, ας θεωρήσουμε το σύνολο των ευαίσθητων στοιχειοσυνόλων προς απόκρυψη *S=[{a,b,d}, {a,c,d}, {c,d,e}]* και τέλος, το κατώφλι της υποστήριξης είναι *msup=0.3*. Στον πίνακα 3.3b δίδεται το σύνολο των Σ.Σ. το οποίο ακολουθείται από την τιμή της υποστήριξης, ενώ στον 3.3c δίδεται το Αναθεωρημένο Θετικό Σύνορο.

Tid	*Itemset*
1	a,b,c,d,e
2	a,c,d
3	a,b,d,f,g
4	b,c,d,e
5	a,b,d
6	b,c,d,f,h
7	a,b,c,g
8	a,c,d,e
9	a,c,d,h
10	a,b,f

Πίνακας 3.5: ΒΔ

Frequent Itemset: Support
{a,b,d}: 0.3, {a,c,d}: 0.4, {b,c,d}: 0.3, {c,d,e}: 0.3 {a,b}:0.5, {a,c}:0.5, {a,d}:0.6, {b,c}:0.4, {b,d}:0.5, {b,f}:0.3 {c,d}:0.6, {c,e}:0.3, {d,e}:0.3 {a}:0.8, {b}:0.7, {c}:0.7, {d}:0.8, {e}:3, {f}:0.3

Πίνακας 3.6: Το σύνολο των συχνών Σ.Σ.

{a,b}, {a,c}, {a,d}, {b,f}, {c,e}, {d,e}, {b,c,d}

Πίνακας 3.7: Το Α.Θ.Σ. της Β.Δ.

Ακολουθούν τα τρία παραδείγματα των ισάριθμων αλγόριθμων που παρουσιάστηκαν παραπάνω. Ο λόγος που παρατίθενται τα παραδείγματα είναι ότι μέσω αυτών θα γίνουν φανερές κάποιες λεπτές πτυχές των αλγορίθμων, κάτι το οποίο θα βοηθήσει στην συγκριτική παρουσίαση που θα ακολουθήσει. Επίσης πιστεύουμε ότι κάτι τέτοιο απουσιάζει από την υπάρχουσα βιβλιογραφία και επιπλέον θα αποτελέσει βοήθημα σε μελλοντικούς μελετητές των αλγορίθμων αυτών.

Σημειώνουμε ότι το print out των προγραμμάτων είναι επικεντρωμένα στην παρουσίαση των βημάτων των αλγορίθμων μάλλον, παρά σε αυτά τα ίδια τα αποτελέσματα που δίδουν.

3.3.1 Παράδειγμα εκτέλεσης του Αλγορίθμου Max-Min 1.

Εκτελώντας το module Max_Min_1.py το οποίο υλοποιεί τον ομώνυμο αλγόριθμο και χρησιμοποιώντας την Β.Δ. και το σύνολο των ευαίσθητων στοιχειοσυνόλων της προηγούμενης παραγράφου, παίρνουμε τα παρακάτω βήματα του αλγορίθμου.

Οι γραμμές 1 έως 10 παρουσιάζουν την vi-list του εκάστοτε sensitive itemset. Ο αλγόριθμος ξεκινάει με το στοιχειοσύνολο που έχει τη μικρότερη υποστήριξη. Επειδή τα δύο στοιχειοσύνολα {a,b,d} και {b,c,d} έχουν την ίδια υποστήριξη 0.3, επιλέγεται αυθαίρετα ένα από τα δύο ως το πρώτο για απόκρυψη. Ακολουθούν ο αύξων αριθμός του ευαίσθητου Σ.Σ., το υποψήφιο στοιχείο -θύμα, το αντίστοιχο στοιχειοσύνολο και η μέτρηση της υποστήριξής του. Για παράδειγμα, στην γραμμή 2, το "0" αντιστοιχεί στον α/α του ευαίσθητου Σ.Σ., κ.τ.λ. Στις γραμμές 11 έως 17 παρουσιάζεται το min itemset και στις 18 έως 20 τα δύο Max-Min itemset σύμφωνα, με τα προαναφερθέντα. Στη συνέχεια ο αλγόριθμος επιλέγει αυθαίρετα το victim item. Στη γραμμή 27 βλέπουμε ότι το συγκεκριμένο loop εφαρμόστηκε μόνο μια φορά καθώς η αρχική υποστήριξη του {a,b,d} ήταν 0.3. Επίσης, από την γραμμή 57 βλέπουμε ότι και για το Σ.Σ. {a, c, d} ο βρόχος επανάληψης είναι ένας, ενώ με βάση την αρχική υποστήριξη θα έπρεπε να είναι δύο. Αυτό οφείλεται στο γεγονός ότι κατά την διαγραφή του item a από τη συναλλαγή {e, d, c, b, a} μειώθηκε και η υποστήριξη του {a, c, d}. Τέλος, στη γραμμή 83 βλέπουμε τις επιδόσεις του αλγορίθμου, τον αριθμό των αλλαγών που έκανε στην αρχική Β.Δ., τις παρενέργειες, δηλαδή μη ευαίσθητα συχνά Σ.Σ. τα οποία απωλέσθηκαν κατά τη διαδικασία της απόκρυψης.

```
1   The vi-list for the 1 loop of [a, b, d] is:
2   0 a [a, d] 0.6
3   0 a [a, b] 0.5
4   0 a [a, c] 0.5
5   0 b [b, c, d] 0.3
6   0 b [a, b] 0.5
7   0 b [b, f] 0.3
8   0 d [a, d] 0.6
9   0 d [b, c, d] 0.3
10  0 d [d, e] 0.3
11  The min item set is:
12  a [a, b] 0.5
13  a [a, c] 0.5
14  b [b, c, d] 0.3
15  b [b, f] 0.3
16  d [b, c, d] 0.3
17  d [d, e] 0.3
18
19  The Max-Min item set is:
```

```
a [a, b] 0.5
a [a, c] 0.5

**********
vi(tent_v=a, it_set=[a, b], sup=0.5)
**********
a to be discarded from {e, d, c, b, a}

number of internal loops: 1

The vi-list for the 1 loop of [c, d, e] is:
1 c [c, e] 0.3
1 c [b, c, d] 0.3
1 c [a, c] 0.4
1 d [a, d] 0.5
1 d [b, c, d] 0.3
1 d [d, e] 0.3
1 e [c, e] 0.3
1 e [d, e] 0.3
The min item set is:
c [c, e] 0.3
c [b, c, d] 0.3
d [b, c, d] 0.3
d [d, e] 0.3
e [c, e] 0.3
e [d, e] 0.3
The Max-Min item set is:
c [c, e] 0.3
c [b, c, d] 0.3
d [b, c, d] 0.3
d [d, e] 0.3
e [c, e] 0.3
e [d, e] 0.3
**********
vi(tent_v=c, it_set=[c, e], sup=0.3)
**********
c to be discarded from {e, d, c, b}

number of internal loops: 1

The vi-list for the 1 loop of [a, c, d] is:
2 a [a, d] 0.5
2 a [a, b] 0.4
2 a [a, c] 0.3
2 c [c, e] 0.2
2 c [b, c, d] 0.2
2 c [a, c] 0.3
2 d [a, d] 0.5
2 d [b, c, d] 0.2
2 d [d, e] 0.3
```

```
69  The min item set is:
70  a [a, c] 0.3
71  c [c, e] 0.2
72  c [b, c, d] 0.2
73  d [b, c, d] 0.2
74  The Max-Min item set is:
75  a [a, c] 0.3
76  c [c, e] 0.2
77  d [b, c, d] 0.2
78  ***********
79  vi(tent_v=a, it_set=[a, c], sup=0.3)
80  ***********
81  a to be discarded from {d, c, a}
82
83  change_raw_data= 3, side_effects=2, CPU_time=0.011 sec
```

Πίνακας 3.8: Παράδειγμα εκτέλεσης του Αλγορίθμου Max-Min 1

3.3.2 Παράδειγμα εκτέλεσης του Αλγορίθμου Max-Min 2.

Η εκτέλεση του module Max_Min_2.py για τις ίδιες τιμές εισόδου με αυτές της προηγούμενης παραγράφου, παίρνουμε τα παρακάτω βήματα του αλγορίθμου. Οι γραμμές 0 έως 22 είναι πανομοιότυπες με αυτές του αλγορίθμου Max-Min 1. Στο σημείο αυτό, ο Max-Min 2, αντί να επιλέξει στην τύχη victim Itemset ακολουθεί την διαδικασία που περιγράφεται στο σενάριο Α της παραγράφου 3.1.1. καθώς δύο Max-Min itemsets εξάγονται από το ίδιο υποψήφιο στοιχειό- θύμα a. Τελικά επιλέγεται ή διαγραφή του a από το σύνολο *Li* καθώς η διαφορά *L= Li- Lu* είναι το κενό σύνολο (γραμμή 29). Στη συνέχεια σειρά παίρνει το ευαίσθητο Σ.Σ. *{c,d,e}* για το οποίο προκύπτει το δεύτερο σενάριο του αλγορίθμου, καθώς βρέθηκαν έξι Max-Min στοιχειοσύνολα για τρια διαφορετικά μεταξύ τους υποψήφια στοιχειό- θύματα. Το σύνολο *Li* θα πάρει την τιμή του ευαίσθητου Σ.Σ. *{c,d,e}* ενώ το *Lu* θα πάρει διαδοχικά τις τιμές *{d, e}, {b, c, d}* και τέλος *c, e}* αναζητώντας *L= Li- Lu* ≠ Ø (γραμμές 65-75). Σε περίπτωση που δεν προκύψει κάτι τέτοιο και είναι πάντα *L= 0*, τότε θα ισχύει το σενάριο Γ. Στο παράδειγμα που εξετάζουμε προκύπτει μη μηδενική τιμή για το Σ.Σ. {b, c, d} (γραμμή 74) οπότε γίνεται διαγραφή του c από τη συναλλαγή με α/α επτά.

Από τις τελευταίες γραμμές παρατηρούμε ότι ο αλγόριθμος εκτελέστηκε σε δύο βήματα, αντί των τριών του Max-Min 1 και είχε μία «παρενέργεια» αντί των δύο του Max-Min 1. Αυτό οφείλεται στο ότι κατά την μείωση της υποστήριξης του Σ.Σ. {a, b, d} μειώθηκε και η υποστήριξη του {a, c, d} και του {c, d, e}.

```
The vi-list for the 1 loop of [a, b, d] is:
0 a [a, b] 0.5
0 a [a, d] 0.6
0 a [a, c] 0.5
0 b [b, c, d] 0.3
0 b [a, b] 0.5
0 b [b, f] 0.3
0 d [b, c, d] 0.3
0 d [a, d] 0.6
0 d [d, e] 0.3
**********
The min item set is:
a [a, b] 0.5
a [a, c] 0.5
b [b, c, d] 0.3
b [b, f] 0.3
d [b, c, d] 0.3
d [d, e] 0.3

The Max-Min item set is:
a [a, b] 0.5
a [a, c] 0.5

first case scenario

Li= {0, 2, 4} /* τα 0, 2, 4 αποτελούν α/α συναλλαγών
στην αρχική Β.Δ

Lu= {0, 1, 2, 4, 6, 7, 8, 9}
------------------
L= Li- Lu =set()
####################
{a} {0, 2, 4}
####################
Item a shall be removed from {d, e, a, b, c}

number of internal loops: 1

The vi-list for the 1 loop of [c, d, e] is:
1 c [b, c, d] 0.3
1 c [c, e] 0.3
1 c [a, c] 0.4
1 d [b, c, d] 0.3
1 d [a, d] 0.5
1 d [d, e] 0.3
1 e [c, e] 0.3
1 e [d, e] 0.3
**********
The min item set is:
c [b, c, d] 0.3
c [c, e] 0.3
```

```
d [b, c, d] 0.3
d [d, e] 0.3
e [c, e] 0.3
e [d, e] 0.3

The Max-Min item set is:
c [b, c, d] 0.3
c [c, e] 0.3
d [b, c, d] 0.3
d [d, e] 0.3
e [c, e] 0.3
e [d, e] 0.3

second case scenario

{d, e, c}
{(d, e), (b, c, d), (c, e)}
-------(d, e)-------------
Li= {0, 3, 7}
Lu= {0, 3, 7}
L= set()
---------(b, c, d)----------
Li= {0, 3, 7}
Lu= {0, 3, 5}
L= {7}

c (b, c, d) {7}
####################
{c} {7}
####################
Item c shall be removed from {d, e, a, c}

number of internal loops: 1

change_raw_data= 2,

side_effects=1,

CPU_time=0.021 sec
```

Πίνακας 3.9: Παράδειγμα εκτέλεσης του Αλγορίθμου Max-Min 2

3.3.3 Παράδειγμα εκτέλεσης Αλγορίθμου BBA.

Ο αλγόριθμος BBA ξεκινάει με τον υπολογισμό του ελάχιστου συνόλου ευαίσθητων Σ.Σ. (γραμμή 1). Εν συνεχεία δημιουργείται μια κατάλληλη δομή δεδομένων για την ταξινόμηση του S_{min} σύμφωνα με τις επιταγές του αλγορίθμου (γραμμές 3–5) και ακολουθεί ο υπολογισμός του Α.Θ.Σ. στη γραμμή 8. Αυτό που ακολουθεί στις γραμμές 11 έως 81 είναι οι συντελεστές βαρύτητας που αναθέτει ο αλγόριθμος για κάθε στοιχείο κάθε ευαίσθητου Σ.Σ., μετρώντας την επίδραση έχει

τυχόν διαγραφή κάποιου στοιχείου του ευαίσθητου Σ. Σ στο Α.Θ.Σ. Αυτό που ακολουθεί στις γραμμές 84 έως 106 είναι η δημιουργία μια δομής δεδομένων για την ταξινόμηση των βαρών και την διαγραφή του στοιχείου με το μικρότερο βάρος από το αντίστοιχο Σ.Σ.. και η όλη διαδικασία επαναλαμβάνεται σε κάθε βήμα του αλγορίθμου.

Είναι εμφανής η υπολογιστική πολυπλοκότητα του BBA, κάτι το οποίο αποτυπώνεται και στον υψηλότερο χρόνο εκτέλεσης του προγράμματος. Οι παρενέργειες είναι επίσης αυξημένες καθώς κατά την διαγραφή των στοιχείων δεν έτυχε να μειωθεί και η υποστήριξη κάποιου επιπλέον ευαίσθητου Σ.Σ.. Έτσι, ο αλγόριθμος εκτελέστηκε τέσσερις φορές.

```
1   S_min=[ {d, c, a},{d, b, a},{e, d, c} ]
2
3   Length_sort= [s_set(i_set=[a, b, d], len_i_set=3, sup=0.3),
4   s_set(i_set=[a, c, d], len_i_set=3, sup=0.4), s_set(i_set=[c, d,
5   e], len_i_set=3, sup=0.3)]
6
7
8   Rev_pos_bord= [{b, a}, {d, a}, {e, c},
9
10   {d, c, b}, {e, d}, {f, b}, {c, a}]
11
12  C-I for [a, c, d] is:
13  0 TID= 0 , a , w= 3
14  ---------
15  0 TID= 0 , c , w= 3
16  ---------
17  0 TID= 0 , d , w= 3
18  ---------
19  ============
20  0 TID= 1 , a , w= 2
21  ---------
22  0 TID= 1 , c , w= 1
23  ---------
24  0 TID= 1 , d , w= 1
25  ---------
26  ============
27  0 TID= 7 , a , w= 2
28  ---------
29  0 TID= 7 , c , w= 2
30  ---------
31  0 TID= 7 , d , w= 2
32  ---------
33  ============
34  0 TID= 8 , a , w= 2
35  ---------
36  0 TID= 8 , c , w= 1
    ---------
```

```
0 TID= 8 , d , w= 1
---------
============
C-I for [a, b, d] is:
1 TID= 0 , a , w= 3
---------
1 TID= 0 , b , w= 2
---------
1 TID= 0 , d , w= 3
---------
============
1 TID= 2 , a , w= 2
---------
1 TID= 2 , b , w= 2
---------
1 TID= 2 , d , w= 1
---------
============
1 TID= 4 , a , w= 2
---------
1 TID= 4 , b , w= 1
---------
1 TID= 4 , d , w= 1
---------
============
C-I for [c, d, e] is:
2 TID= 0 , c , w= 3
---------
2 TID= 0 , d , w= 3
---------
2 TID= 0 , e , w= 2
---------
============
2 TID= 3 , c , w= 2
---------
2 TID= 3 , d , w= 2
---------
2 TID= 3 , e , w= 2
---------
============
2 TID= 7 , c , w= 2
---------
2 TID= 7 , d , w= 2
---------
2 TID= 7 , e , w= 2
---------
============
[w_set(index=0, trans=1, item=c, weight=1), w_set(index=0,
trans=1, item=d, weight=1), w_set(index=0, trans=8, item=c,
weight=1), w_set(index=0, trans=8, item=d, weight=1),
```

```
87  w_set(index=0, trans=1, item=a, weight=2), w_set(index=0, trans=7,
88  item=a, weight=2), w_set(index=0, trans=7, item=c, weight=2),
89  w_set(index=0, trans=7, item=d, weight=2), w_set(index=0, trans=8,
90  item=a, weight=2), w_set(index=0, trans=0, item=a, weight=3),
91  w_set(index=0, trans=0, item=c, weight=3), w_set(index=0, trans=0,
92  item=d, weight=3), w_set(index=1, trans=2, item=d, weight=1),
93  w_set(index=1, trans=4, item=b, weight=1), w_set(index=1, trans=4,
94  item=d, weight=1), w_set(index=1, trans=0, item=b, weight=2),
95  w_set(index=1, trans=2, item=a, weight=2), w_set(index=1, trans=2,
96  item=b, weight=2), w_set(index=1, trans=4, item=a, weight=2),
97  w_set(index=1, trans=0, item=a, weight=3), w_set(index=1, trans=0,
98  item=d, weight=3), w_set(index=2, trans=0, item=e, weight=2),
99  w_set(index=2, trans=3, item=c, weight=2), w_set(index=2, trans=3,
100 item=d, weight=2), w_set(index=2, trans=3, item=e, weight=2),
101 w_set(index=2, trans=7, item=c, weight=2), w_set(index=2, trans=7,
102 item=d, weight=2), w_set(index=2, trans=7, item=e, weight=2),
103 w_set(index=2, trans=0, item=c, weight=3), w_set(index=2, trans=0,
104 item=d, weight=3)]
105
106 INDEX= 0 , TID= 1 , ITEM= c
107 item c deleted from transaction 1{a,c,d}
108 Transaction 1 shall be deleted from C - I list.
109
110 [w_set(index=0, trans=8, item=c, weight=1), w_set(index=0,
111 trans=8, item=d, weight=1), w_set(index=0, trans=7, item=a,
112 weight=2), w_set(index=0, trans=7, item=c, weight=2),
113 w_set(index=0, trans=7, item=d, weight=2), w_set(index=0, trans=8,
114 item=a, weight=2), w_set(index=0, trans=0, item=a, weight=3),
115 w_set(index=0, trans=0, item=c, weight=3), w_set(index=0, trans=0,
116 item=d, weight=3), w_set(index=1, trans=2, item=d, weight=1),
117 w_set(index=1, trans=4, item=b, weight=1), w_set(index=1, trans=4,
118 item=d, weight=1), w_set(index=1, trans=0, item=b, weight=2),
119 w_set(index=1, trans=2, item=a, weight=2), w_set(index=1, trans=2,
120 item=b, weight=2), w_set(index=1, trans=4, item=a, weight=2),
121 w_set(index=1, trans=0, item=a, weight=3), w_set(index=1, trans=0,
122 item=d, weight=3), w_set(index=2, trans=0, item=e, weight=2),
123 w_set(index=2, trans=3, item=c, weight=2), w_set(index=2, trans=3,
124 item=d, weight=2), w_set(index=2, trans=3, item=e, weight=2),
125 w_set(index=2, trans=7, item=c, weight=2), w_set(index=2, trans=7,
126 item=d, weight=2), w_set(index=2, trans=7, item=e, weight=2),
127 w_set(index=2, trans=0, item=c, weight=3), w_set(index=2, trans=0,
128 item=d, weight=3)]
129
130 INDEX= 0 , TID= 8 , ITEM= c
131 item c deleted from transaction 8 {a,c,d,h}
132 Transaction 8 shall be deleted from C - I list.
133
134 [w_set(index=0, trans=7, item=a, weight=2), w_set(index=0,
135 trans=7, item=c, weight=2), w_set(index=0, trans=7, item=d,
136 weight=2), w_set(index=0, trans=0, item=a, weight=3),
    w_set(index=0, trans=0, item=c, weight=3), w_set(index=0, trans=0,
```

```
137  item=d, weight=3), w_set(index=1, trans=2, item=d, weight=1),
138  w_set(index=1, trans=4, item=b, weight=1), w_set(index=1, trans=4,
139  item=d, weight=1), w_set(index=1, trans=0, item=b, weight=2),
140  w_set(index=1, trans=2, item=a, weight=2), w_set(index=1, trans=2,
141  item=b, weight=2), w_set(index=1, trans=4, item=a, weight=2),
142  w_set(index=1, trans=0, item=a, weight=3), w_set(index=1, trans=0,
143  item=d, weight=3), w_set(index=2, trans=0, item=e, weight=2),
144  w_set(index=2, trans=3, item=c, weight=2), w_set(index=2, trans=3,
145  item=d, weight=2), w_set(index=2, trans=3, item=e, weight=2),
146  w_set(index=2, trans=7, item=c, weight=2), w_set(index=2, trans=7,
147  item=d, weight=2), w_set(index=2, trans=7, item=e, weight=2),
148  w_set(index=2, trans=0, item=c, weight=3), w_set(index=2, trans=0,
149  item=d, weight=3)]
150
151  INDEX= 1 , TID= 2 , ITEM= d
152  item d deleted from transaction 2 {a,b,d,f,g}
153  Transaction 2 shall be deleted from C - I list.
154
155  [w_set(index=0, trans=7, item=a, weight=2), w_set(index=0,
156  trans=7, item=c, weight=2), w_set(index=0, trans=7, item=d,
157  weight=2), w_set(index=0, trans=0, item=a, weight=3),
158  w_set(index=0, trans=0, item=c, weight=3), w_set(index=0, trans=0,
159  item=d, weight=3), w_set(index=1, trans=4, item=b, weight=1),
160  w_set(index=1, trans=4, item=d, weight=1), w_set(index=1, trans=0,
161  item=b, weight=2), w_set(index=1, trans=4, item=a, weight=2),
162  w_set(index=1, trans=0, item=a, weight=3), w_set(index=1, trans=0,
163  item=d, weight=3), w_set(index=2, trans=0, item=e, weight=2),
164  w_set(index=2, trans=3, item=c, weight=2), w_set(index=2, trans=3,
165  item=d, weight=2), w_set(index=2, trans=3, item=e, weight=2),
166  w_set(index=2, trans=7, item=c, weight=2), w_set(index=2, trans=7,
167  item=d, weight=2), w_set(index=2, trans=7, item=e, weight=2),
168  w_set(index=2, trans=0, item=c, weight=3), w_set(index=2, trans=0,
169  item=d, weight=3)]
170
171  INDEX= 2 , TID= 0 , ITEM= e
172  item e deleted from transaction 0 {a,b,c,d,e}
173  Transaction 0 shall be deleted from C - I list.
174
175  [w_set(index=0, trans=7, item=a, weight=2), w_set(index=0,
176  trans=7, item=c, weight=2), w_set(index=0, trans=7, item=d,
177  weight=2), w_set(index=1, trans=4, item=b, weight=1),
178  w_set(index=1, trans=4, item=d, weight=1), w_set(index=1, trans=4,
179  item=a, weight=2), w_set(index=2, trans=3, item=c, weight=2),
180  w_set(index=2, trans=3, item=d, weight=2), w_set(index=2, trans=3,
181  item=e, weight=2), w_set(index=2, trans=7, item=c, weight=2),
182  w_set(index=2, trans=7, item=d, weight=2), w_set(index=2, trans=7,
183  item=e, weight=2)]
184
185  change_raw_data= 4,side_effects=3,
186  CPU_time=0.04 sec
```

Πίνακας 3.8: Παράδειγμα εκτέλεσης του Αλγορίθμου BBA

3.4 Μια πρώτη συγκριτική προσέγγιση των τριών αλγορίθμων

Από την θεωρητική ανάλυση και τα παραδείγματα που την ακολούθησαν, μπορεί να γίνει μια πρώτη προσέγγιση στην σύγκριση των τριών αλγορίθμων. Πιο ολοκληρωμένη σύγκριση θα προκύψει στο κεφάλαιο πέντε, λόγω του ότι θα πλαισιώνεται με πειραματικές μετρήσεις των μετρικών απόδοσης συναρτήσει της τιμής υποστήριξης και του μήκους του ευαίσθητου Σ.Σ.

1. Εύκολα παρατηρούμε την υψηλή πολυπλοκότητα του αλγορίθμου BBA σε σχέση μέ τους άλλους δύο και ιδιαιτέρως σε σχέση με τον Max-Min 1, καθώς έχει υπολογισμούς βαρών για κάθε στοιχείο του ευαίσθητου Σ.Σ. και για όλες τις συναλλαγές που το υποστηρίζουν. Ο Max-Min 2 έχει επίσης αρκετά υψηλή πολυπλοκότητα, στην πράξη όμως η πολυπλοκότητα του Max-Min 2 αγγίζει αυτή του BBA μόνο στην χειρότερη περίπτωση. Για να συμβεί αυτό θα πρέπει να ισχύσει το σενάριο Γ της σελίδας 19 και μάλιστα ο αλγόριθμος να φθάσει μέχρι την γραμμή 32, όπως αυτός παρουσιάζεται στην σελίδα 17. Αυτό όμως είναι κάτι σχετικά σπάνιο να συμβεί καθώς ο αλγόριθμος τις περισσότερες φορές τερματίζει τον κάθε βρόχο πολύ ενωρίτερα.

2. Η μεθοδολογία που ακολουθεί ο αλγόριθμος BBA αποδίδει καλά αποτελέσματα στις περιπτώσεις που προκύπτει συντελεστής βαρύτητας ίσος με μηδέν, όταν δηλαδή η διαγραφή ενός item δεν έχει καθόλου επίδραση στο Α.Θ.Σ. Στην αντίθετη περίπτωση κατά την οποία δεν προκύπτει μηδενικό βάρος η απόδοσή του μειώνεται, και είναι συγκρίσιμη με αυτή του maxmin κριτηρίου του Max-Min 1. Η βελτιστοποιήσεις που εισάγει ο Max-Min 2 στην επιλογή των Σ.Σ. – θυμάτων τον καθιστά πιο αποδοτικό, ή στην χειρότερη περίπτωση ίδιας απόδοσης, με τον Max-Min 1.

3. Οι αλγόριθμοι Max-Min 1 λαμβάνουν ως εισόδους τα ευαίσθητα Σ.Σ. σε αντίθεση με το BBA που υπολογίζει ελάχιστο σύνολο ευαίσθητων Σ.Σ. Στην περίπτωση κατά την οποία το σύνολο των ευαίσθητων Σ.Σ. δεν είναι το ελάχιστο, τότε η απόδοση του BBA είναι δυνατόν να βελτιωθεί πολύ, των δε Max-Min να επιδεινωθεί αισθητά.

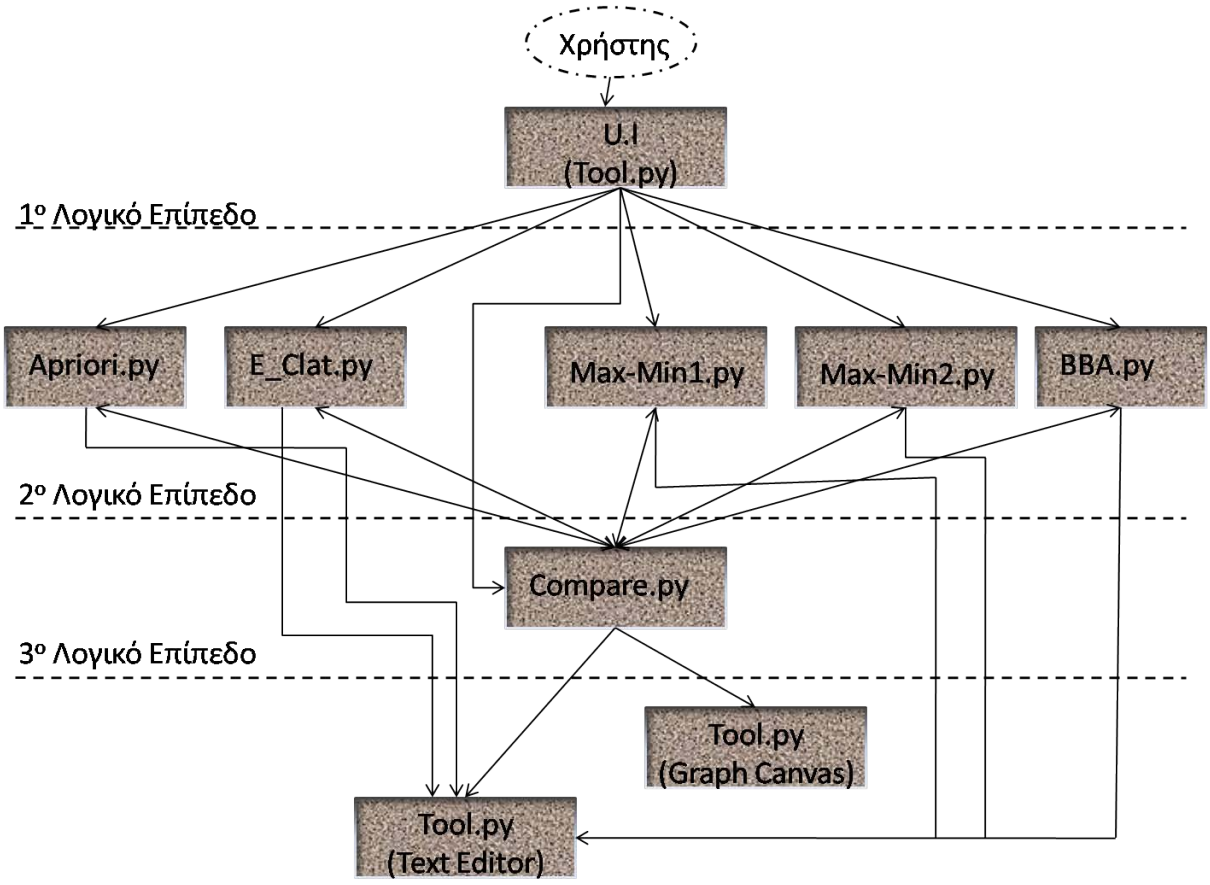
Κεφάλαιο 4
Περιγραφή του Εργαλείου

Στο παρόν κεφάλαιο παρουσιάζεται διεξοδικά το ολοκληρωμένο περιβάλλον πειραματισμού και αξιολόγησης των αλγορίθμων, που έχει αναπτυχθεί στο πλαίσιο της παρούσης μεταπτυχιακής διατριβής. Στην παράγραφο 4.1 παρουσιάζεται η αρχιτεκτονική του εργαλείου μέσω των διαφόρων modules από τα οποία αποτελείται και τον τρόπο της μεταξύ τους επικοινωνίας. Λόγω της σπουδαιότητας του για την εξαγωγή ορθών αποτελεσμάτων, η δομή των του αρχείου συναλλαγών και του αρχείου των ευαίσθητων στοιχειοσυνόλων αναλύεται διεξοδικά στις παραγράφους 4.2, και 4.3 αντίστοιχα, ενώ στην 4.4 επεξηγείται ο τρόπος αυτόνομης εκτέλεσης των modules. Στην ενότητα 4.5 παρουσιάζεται αναλυτικά το εργαλείο και η λειτουργικότητά του, ενώ τέλος στην ενότητα 4.6 δίδονται παραδείγματα λειτουργίας της εφαρμογής μέσω της εκτέλεσης διαφόρων χαρακτηριστικών σεναρίων.

4.1 Αρχιτεκτονική του Εργαλείου

Η εφαρμογή αποτελείται από επτά διακριτά προγράμματα ή modules. Τα δύο πρώτα modules υλοποιούν τους αντίστοιχους αλγορίθμους εξόρυξης, τρία modules αναλαμβάνουν να υλοποιήσουν τους ισάριθμους αλγορίθμους απόκρυψης, ενώ ένα έκτο module εκτελεί την σύγκριση, δύο κάθε φορά, αλγορίθμων. Τέλος, το κυρίως module με την ονομασία 'tool' υλοποιεί την γραφική διεπαφή (GUI) και παρουσιάζει τα αποτελέσματα της εφαρμογής σε μορφή text editor και εκτελεί γραφικές παραστάσεις των αποτελεσμάτων, καλώντας τα προηγούμενα έξι modules κατά περίσταση.

Την αρχιτεκτονική του Εργαλείου μπορούμε να χωρίσουμε σε τρία λογικά επίπεδα αναφορικά με την εγγύτητα του κάθε module προς τον χρήστη, όπως φαίνεται στο σχήμα 1.



Σχήμα 1: Διάγραμμα αρχιτεκτονικής του Εργαλείου

Από το παραπάνω διάγραμμα φαίνεται ότι ο χρήστης επικοινωνεί μέσω του User Interface με όλα τα υπόλοιπα modules της εφαρμογής αναλόγως των επιλογών που έχει κάνει. Αν ο χρήστης επιλέξει απλώς την εκτέλεση ενός αλγορίθμου (όχι σύγκριση) το module Tool.py στέλνει τις επιλογές του χρήστη στο αντίστοιχο module και λαμβάνει τα

αποτελέσματα του προγράμματος στον Text Editor. Αν όμως ο χρήστης επιλέξει σύγκριση αλγορίθμων, τότε μεσολαβεί το module Compare.py το οποίο λαμβάνει τα δεδομένα του χρήστη και επικοινωνεί αμφίδρομα με τα κατάλληλα modules των αλγορίθμων εξόρυξης και απόκρυψης. Τέλος, το module Tool.py λαμβάνει τα αποτελέσματα και τα παρουσιάζει σε μορφή κειμένου και γράφων.

Στις επόμενες παραγράφους ακολουθεί αναλυτική περιγραφή όλων των modules.

4.1.1 Modules Αλγόριθμων Εξόρυξης.

Όπως έχει αναφερθεί και σε άλλο σημείο του παρόντος οι δύο αλγόριθμοι Data Mining είναι οι Apriori και E-Clat και τα αντίστοιχα modules που τους υλοποιούν τα Apriori.py και E-Clat.py. Τα modules αυτά λαμβάνουν σαν δεδομένα εισόδου τα παρακάτω:

1. Το αρχείο που περιέχει τις συναλλαγές. Το αρχείο αυτό πρέπει να έχει κατάλληλη μορφή ώστε να μπορέσει να διαβαστεί ορθά από το πρόγραμμα και να παράξει τα σωστά αποτελέσματα. Η δομή που πρέπει να έχει το αρχείο συναλλαγών εξετάζεται διεξοδικά στην επόμενη παράγραφο.
2. Την τιμή της υποστήριξης σε μορφή δεκαδικού αριθμού. Σημειώνουμε ότι γίνεται χρήση του decimal point και όχι του συμβόλου ' , ' (π.χ. 0.3).

Και τα δύο modules επιστρέφουν, υπό μορφή κειμένου, τα αποτελέσματά τους, τα οποία είναι τα συχνά Σ.Σ. για την δοσμένη τιμή υποστήριξης καθώς και ο χρόνος εκτέλεσης του αλγορίθμου εξόρυξης.

4.1.2 Modules Αλγόριθμων Απόκρυψης.

Ως γνωστόν, οι τρεις αλγόριθμοι Απόκρυψης Σ.Σ. που έχουν υλοποιηθεί είναι οι Max-Min 1, Max-Min 2 και BBA και έχουν ενσωματωθεί στο εργαλείο μέσω των ομώνυμων modules. Και τα τρία modules λαμβάνουν σαν δεδομένα εισόδου τις ίδιες τιμές οι οποίες περιγράφονται παρακάτω:

1. Το αρχείο που περιέχει τις συναλλαγές της αρχικής Β.Δ. Η δομή που πρέπει να έχει το αρχείο συναλλαγών εξετάζεται διεξοδικά στην επόμενη παράγραφο.
2. Το αρχείο των συχνών στοιχειοσυνόλων, όπως αυτό παράγεται από τα modules της εξόρυξης δεδομένων.
3. Το αρχείο που περιέχει τα ευαίσθητα στοιχειοσύνολα. Η δομή που πρέπει να έχει το αρχείο των ευαίσθητων δεδομένων είναι σημαντική και μπορεί να έχει

διαφορετική μορφή ανάλογα με το εάν το module εκτελείται μεμονωμένα ή μέσω της εφαρμογής. Έτσι, για λόγους καλύτερης επεξήγησης, το θέμα αυτό θα αναλυθεί στην παράγραφο 4.3.

4. Την τιμή της υποστήριξης, όπως και στην προηγούμενη παράγραφο.

Και τα τρία modules επιστρέφουν σε μορφή text αρχείου τα αποτελέσματά τους. Τα αποτελέσματα αποτελούνται από την sanitized Β.Δ., τον χρόνο εκτέλεσης του αλγορίθμου, τον αριθμό των μεταβολών (Changes of Raw Data) που έχουν γίνει προκειμένου να καθαρισθεί η αρχική λίστα συναλλαγών και τέλος, το Α.Θ.Σ., το οποίο θα χρησιμοποιηθεί από το module compare για την εύρεση των παρενεργειών (Side Effects).

4.1.3 Το module Compare.py

Ο ρόλος του συγκεκριμένου module, όταν εκτελείται μέσω της εφαρμογής, είναι να λάβει τις επιλογές του χρήστη μέσω του GUI, ήτοι το αρχείο των συναλλαγών, το αρχείο των ευαίσθητων Σ.Σ. και το είδος της σύγκρισης μεταξύ δύο κάθε φορά αλγορίθμων, να τα μεταβιβάσει στα κατάλληλα module, να εκτελέσει την σύγκριση και τέλος, να επιστρέψει τα αποτελέσματα της σύγκρισης στο tool.py σε κατάλληλη μορφή, ώστε να αναπαρασταθούν γραφικά και σε μορφή text.

Το module έχει την δυνατότητα να εκτελεί συγκρίσεις των αλγορίθμων απόκρυψης για πολλές τιμές υποστήριξης για κάποιο συγκεκριμένο ευαίσθητο Σ.Σ. ή για μεγάλο αριθμό ευαίσθητων Σ.Σ. διαφορετικών μεταξύ τους μήκους, αλλά μιας και συγκεκριμένης τιμής υποστήριξης, και να επιστρέφει τον μέσο όρο των μετρικών μεγεθών αξιολόγησης που έχουν επιλεγεί. Για παράδειγμα, θεωρώντας το παράδειγμα της παραγράφου 3.3 και της Β.Δ. που απεικονίζεται στον πίνακα 3.5, και αν υποθέσουμε ως σύνολο ευαίσθητων Σ.Σ. τα σύνολα του πίνακα 4.1, τότε θα ήταν δυνατόν μέσω του module compare.py να γίνουν οι παρακάτω μετρήσεις απόδοσης των αλγορίθμων:

1. Υπολογισμός των μετρικών αξιολόγησης των αλγορίθμων για κάθε ένα από τα σύνολα ευαίσθητων Σ.Σ. του πίνακα 4.1 για κάποια τιμή υποστήριξης και υπολογισμό του μέσου όρου (των μετρικών αξιολόγησης) για τα διάφορα μήκη ευαίσθητου Σ.Σ. Δηλαδή για το συγκεκριμένο παράδειγμα θα υπολογίσει τις μέσες τιμές των μετρικών για μήκος ευαίσθητου Σ.Σ. ίσο με 2, 3, 5, 7, 8 και 9 για την επιθυμητή τιμή υποστήριξης.

2. Υπολογισμός των μετρικών αξιολόγησης των αλγορίθμων για κάθε ένα από τα σύνολα ευαίσθητων Σ.Σ. του πίνακα 4.1 για όλες τις τιμές υποστήριξης από 0.1 έως 0.9 με βήμα 0.1 και υπολογισμό του μέσου όρου. Δηλαδή για το συγκεκριμένο παράδειγμα θα υπολογίσει τις τιμές των μετρικών για κάθε σύνολο ευαίσθητων Σ.Σ. και για κάθε τιμή υποστήριξης. Στη συνέχεια θα υπολογίσει τη μέση τιμή των μετρικών για κάθε μία από τις τιμές υποστήριξης (χωρίς, ωστόσο, να κάνει διάκριση ανάμεσα στο μήκος των συνόλων προς απόκρυψη).

Ο τρόπος με τον οποίο γίνεται η σύγκριση των αλγορίθμων, ή με άλλα λόγια η μέθοδος που ακολουθεί το module compare.py για να συγκρίνει τους αλγορίθμους παρουσιάζεται στη παράγραφο 5.2 του κεφαλαίου 5.

Για την επίτευξη των παραπάνω το αρχείο των ευαίσθητων δεδομένων πρέπει να έχει κατάλληλη δομή η οποία εξετάζεται στην παράγραφο 4.3.

Α/Α	Σύνολο ευαίσθητων Σ.Σ.	Μήκος ευαίσθητου Σ. Σ
1	{a,d}	2
2	{c,d}	2
3	{a,b,d}	3
4	{c,d,e}	3
5	{ {a,b}, {a,c,d} }	5
6	{ {a,c}, {a,b,d} }	5
7	{ {a,d}, {b,c,d} }	5
8	{ { {a,b}, {d,e}, {a,c,d} } }	7
9	{ { {a,c}, {a,d}, {b,c,d} } }	7
10	{ { {a,c}, {a,b,d}, {c,d,e} } }	7
11	{ { { {a,b}, {b,c}, {c,d}, {d,e} } } }	8
12	{ { {a,b,d}, {a,c,d}, {c,d,e} } }	9

Πίνακας 4.1: Παράδειγμα δυνατότητας του compare.py να συγκρίνει την απόδοση των αλγορίθμων για πολλά και για διαφόρων μηκών σύνολα ευαίσθητων Σ.Σ.

4.1.4 Το module tool.py

Το τελευταίο module αναλαμβάνει να υλοποιήσει τα παρακάτω:

1. Την γραφική διεπαφή μηχανής - χρήστη (GUI) η οποία αναλαμβάνει να συγκεντρώσει τις επιλογές του χρήστη με γρήγορο και εύκολο τρόπο.
2. Έναν απλό text editor ο οποίος αναλαμβάνει να παρουσιάσει τα αποτελέσματα των αλγορίθμων εξόρυξης ή απόκρυψης και σύγκρισης αυτών σε μορφή κειμένου.
3. Ένα καμβά στον οποίο παρουσιάζονται σε μορφή γραφικής παράστασης τα αποτελέσματα της σύγκρισης των αλγορίθμων απόκρυψης που έχουν ενσωματωθεί στο εργαλείο.

Το module αφού παραλάβει τα δεδομένα του χρήστη μέσω της διεπαφής, τα μεταβιβάζει στο module compare.py αν ο χρήστης επιθυμεί σύγκριση ή στο αντίστοιχο module του αλγόριθμου εξόρυξης ή απόκρυψης σε περίπτωση που ο χρήστης επιθυμεί να δοκιμάσει κάποιον αλγόριθμο μόνο του, χωρίς σύγκριση. Τέλος, λαμβάνει τα αποτελέσματα που επιστρέφουν τα αντίστοιχα modules, τα επεξεργάζεται ώστε να τα παρουσιάσει στον text editor ή/ και στον καμβά γραφικών παραστάσεων.

4.2 Δομή του αρχείου συναλλαγών

Ο τύπος του αρχείου θα πρέπει να είναι text με κωδικοποίηση ANSI και η εσωτερική του δομή όπως αυτή που παρουσιάζεται ως παράδειγμα στον πίνακα 4.2. Παρατηρούμε ότι τα στοιχεία που απαρτίζουν τις συναλλαγές διαχωρίζονται μεταξύ του με ένα κενό χαρακτήρα, ενώ δεν υπάρχουν άλλοι ειδικοί χαρακτήρες ή σύμβολα ή κενά ανάμεσα στα στοιχεία των συναλλαγών ή στην αρχή ή στο τέλος του αρχείου. Τέλος, τα στοιχεία των συναλλαγών μπορεί να είναι strings ή characters ή αριθμοί. *Εάν είναι αριθμοί τότε αυτοί θα πρέπει να είναι αποκλειστικά και μόνο ακέραιοι, ενώ δεν θα πρέπει να περιέχονται ειδικά σύμβολα (πίνακας 4.3) ή μεικτά αλφανουμερικά strings*.

bread milk	111 567 87	bread milk
bread diapers beer eggs	687 5 444 111	bread milk 111
milk diapers beer cola	111 2222 8796324	444 111 bread
bread milk diapers beer	111	bread milk diapers 5
bread milk diapers cola	444 222 111	111

Πίνακας 4.2: παραδείγματα συμβατής δομής αρχείου συναλλαγών

Τα module εξόρυξης δεδομένων παράγουν ως έξοδο ένα αρχείο text στο οποίο περιλαμβάνονται τα συχνά Σ.Σ. και η αντίστοιχη υποστήριξη του καθενός από αυτά. Η μορφή του αρχείου είναι τέτοια ώστε να μπορεί να αποτελεί είσοδο στα modules που υλοποιούν τους αλγορίθμους απόκρυψης.

br22ad milk	11,1 567 87	bread,milk
bread di55ers beer eggs	687 5 0.444 111	bread,milk,diapers
milk dia55ers b*er cola	111 2222 8796324	bread,milk,diapers,beer
bread milk dia$ers beer	111	bread,milk,diapers,beer
bread milk dia@ers cola	444 22.2 111	bread,milk,diapers,cola

Πίνακας 4.3: παραδείγματα ΜΗ συμβατής δομής αρχείου συναλλαγών

4.3 Δομή του αρχείου ευαίσθητων Σ.Σ.

Λόγω της ιδιαιτερότητας αλλά και της σπουδαιότητας που έχει στην δομή του αρχείου ευαίσθητων Σ.Σ. για την ορθή λειτουργία της εφαρμογής, εξετάζεται σε ξεχωριστή παράγραφο.

Η ιδιαιτερότητα έγκειται στο ότι το εν' λόγω αρχείο έχει διαφορετική δομή αν τα module max_min_1.py, max_min_2.py και BBA.py εκτελούνται αυτόνομα, απ' ότι αν εκτελούνται μέσω του εργαλείου ή του module compare.py. Ο λόγος για τον οποίο συμβαίνει αυτό είναι για να έχει το εργαλείο ή το compare.py τη δυνατότητα να δοκιμάζει την συμπεριφορά των αλγορίθμων για πολλά σύνολα ευαίσθητων Σ. Σ και να υπολογίζει την μέση τιμή των μετρικών αξιολόγησης που έχουν επιλεγεί. Αντίθετα, τα max_min_1.py, max_min_2.py και BBA.py εκτελούνται για ένα και μόνο σύνολο ευαίσθητων Σ.Σ. Αυτό σημαίνει ότι το compare.py καλεί τους αλγορίθμους απόκρυψης για κάθε ένα από τα σύνολα ευαίσθητων Σ.Σ. που περιλαμβάνονται στο σωστά δομημένο αρχείο.

Δίνουμε από ένα παραδείγματα για την κάθε περίπτωση.

1. Δομή αρχείου για μεμονωμένη εκτέλεση των module Max_Min_1.py, Max_Min_2.py και BBA.py

butter bread	bread milk diapers beer

Πινακας 4.4: Παράδειγμα ευαίσθητου Σ.Σ.

Πίνακας 4.5: Παράδειγμα ευαίσθητου Σ.Σ.

Και στις δύο περιπτώσεις, αυτές των παραπάνω πινάκων, το ευαίσθητο Σ.Σ. είναι ένα. Στο παράδειγμα του πίνακα 4.4 το μήκος του ευαίσθητο Σ.Σ. είναι δύο, ενώ σε αυτό του πίνακα 4.5 είναι τέσσερα. Τα ευαίσθητα στοιχειοσύνολα διαχωρίζονται μεταξύ τους απλά και μόνο εάν είναι γραμμένα σε διαφορετική γραμμή στο αρχείο. Έτσι, στο παράδειγμα του πίνακα 4.4 το ευαίσθητο Σ.Σ είναι το {butter bread} ενώ για τον πίνακα 4.5 το ευαίσθητο Σ.Σ. είναι το {{bread milk}, {diapers beer}}. Σημειώνουμε και πάλι ότι δεν χρησιμοποιούμε κόμμα ή άλλους διαχωριστικούς χαρακτήρες ή σημεία στίξης, παρά μόνο τον κενό χαρακτήρα.

2. Δομή αρχείου για λειτουργία μέσω του εργαλείου ή του module compare.py.

Για την καλύτερη κατανόηση της δομής του αρχείου ας θεωρήσουμε ότι θέλουμε να δοκιμάσουμε την απόδοση των αλγορίθμων και για τα δύο σύνολα, αυτά του πίνακα 4.4 και 4.5. Στην περίπτωση αυτή θα πρέπει να δημιουργήσουμε το αρχείο του πίνακα 4.6.

```
butter bread

bread milk;diapers beer
```

Πίνακας 4.6: Παράδειγμα ευαίσθητου Σ.Σ.

Στην περίπτωση αυτή τα δεδομένα που βρίσκονται στις δύο διαφορετικές γραμμές σηματοδοτούν διαφορετικά σύνολα ευαίσθητων Σ.Σ. σε αντίθεση με αυτά που διαχωρίζονται από τον χαρακτήρα ' ; ' που σημαίνουν διαφορετικά σετ μέσα στο ίδιο σύνολο. Το module compare.py (μεμονωμένα ή μέσω της εφαρμογής) θα καλέσει τους αλγορίθμους απόκρυψης δύο φορές. Μία φορά για το {butter bread} και μία φορά για το {{bread milk}, {diapers beer}}.

4.4 Αυτόνομη λειτουργία των modules

Τα modules που περιγράφηκαν παραπάνω μπορούν να λειτουργήσουν και αυτόνομα, το καθένα χωριστά, μέσω της γραμμής εντολών. Στην παράγραφο αυτή εξετάζεται η

λειτουργικότητά τους ως μεμονωμένα προγράμματα, ενώ η λειτουργικότητα της εφαρμογής (του εργαλείου) εξετάζεται στην επόμενη παράγραφο.

1. Τα modules Apriori και E-Clat εκτελούνται από την γραμμή εντολών ως εξής. Εντός του τρέχοντος directory, από το οποίο τρέχει η python πληκτρολογούμε το όνομα του module: Apriori.py και πατούμε Enter. Το πρόγραμμα αρχίζει να εκτελείται και ζητάει από τον χρήστη το όνομα του αρχείου των συναλλαγών και εν' συνεχεία την τιμή της υποστήριξης. Τα αποτελέσματα λαμβάνονται στο αρχείο Apriori_results.txt στο τρέχον directory.

2. Για την εκτέλεση των modules των αλγορίθμων απόκρυψης ως μεμονωμένα προγράμματα θα πρέπει προηγουμένως να έχει τρέξει ένα από τα modules Apriori ή E-Clat ώστε να προκύψει το αρχείο συχνών Σ.Σ. Κατά τα λοιπά, τα modules αυτά εκτελούνται κατά παρόμοιο τρόπο, με τα προηγούμενα modules, από την γραμμή εντολών. Πληκτρολογούμε το όνομα του module και εν' συνεχεία, κατόπιν προτροπής του προγράμματος, το όνομα του αρχείου συναλλαγών, το όνομα του αρχείου συχνών Σ.Σ., το όνομα του αρχείου των ευαίσθητων Σ.Σ. το οποίο θα πρέπει να έχει την δομή που περιγράφηκε στην παράγραφο 4.2, και την τιμή της υποστήριξης. Τα αποτελέσματα λαμβάνονται στα αρχεία Max_Min_1_results.txt ή Max_Min_2_results.txt ή BBA_results.txt στο τρέχον directory.

3. Το module compare.py μπορεί και αυτό να εκτελεστεί μεμονωμένα με παρόμοιο με τα παραπάνω τρόπο. Πληκτρολογούμε το όνομα του module και εν' συνεχεία, κατόπιν προτροπής του προγράμματος, το όνομα του αρχείου συναλλαγών, το όνομα του αρχείου των ευαίσθητων Σ.Σ. το οποίο θα πρέπει να έχει την δομή που περιγράφηκε στην παράγραφο 4.2, την επιλογή των αλγορίθμων προς σύγκριση, η οποία θα πρέπει να είναι μία από τις τρεις επιλογές που παρουσιάζονται στην προτροπή ('Max Min 1 & Max Min 2' ή 'Max Min 1 & BBA' ή 'Max Min 2 & BBA'). Τέλος επιλέγουμε το είδος της σύγκρισης σύμφωνα με τα παρακάτω:

   3.1. Αν επιθυμούμε λειτουργία σύγκρισης των μετρικών αξιολόγησης για διάφορα μήκη ευαίσθητων Σ.Σ. και συγκεκριμένη τιμή υποστήριξης, δίνουμε τη τιμή της υποστήριξης.

   3.2. Αν επιθυμούμε λειτουργία σύγκρισης των μετρικών αξιολόγησης για διάφορες τιμές υποστήριξης, δίνουμε τον χαρακτήρα ' x '.

   Σημειώνουμε ότι, στην περίπτωση του compare.py δεν απαιτείται να έχει προηγηθεί η εκτέλεση των Apriori ή E-Clat.

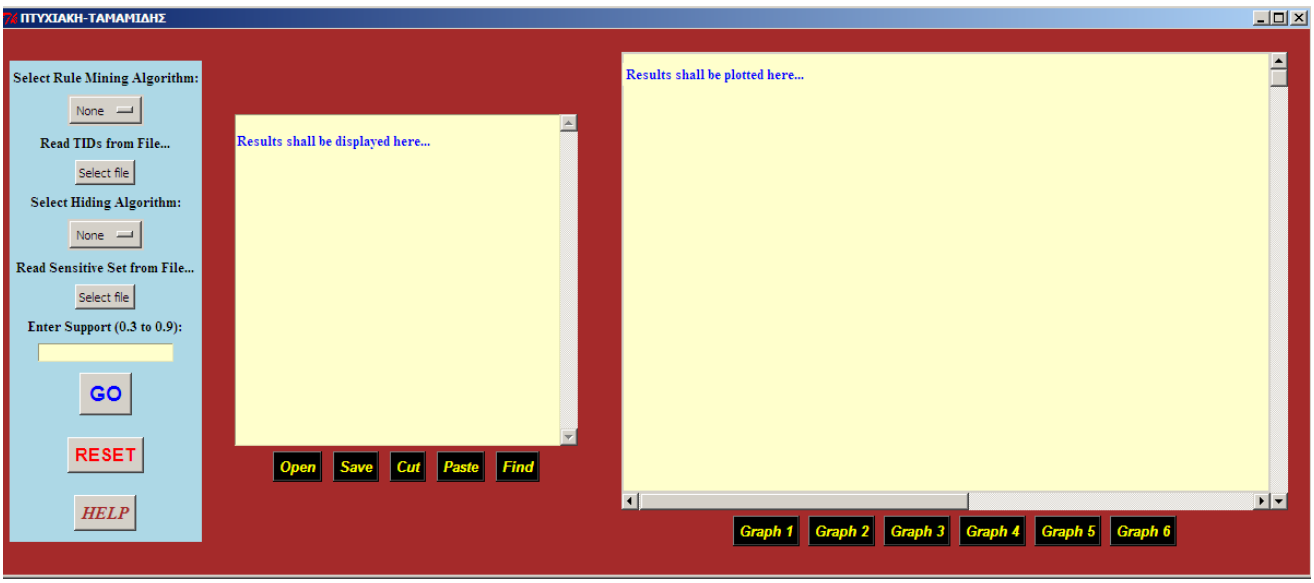
4.5 Περιγραφή του εργαλείου

Με την εκκίνηση της εφαρμογής αντικρίζουμε το περιβάλλον εργασίας όπως παρουσιάζεται στην εικόνα 1 (ενδέχεται να διαφέρει ελαφρώς αναλόγως του λειτουργικού συστήματος που χρησιμοποιούμε).

4.5.1 Γενική περιγραφή

Το περιβάλλον εργασίας αποτελείται από τρία μέρη.

Το πρώτο μέρος, στην αριστερή πλευρά, υλοποιεί τη γραφική διεπαφή χρήστη (GUI) και απαρτίζεται από τα button και τα μενού των επιλογών του χρήστη. Ο χρήστης μπορεί να επιλέξει αλγόριθμο εξόρυξης, αλγόριθμο απόκρυψης, το αρχείο που περιέχει την αρχική βάση δεδομένων, το αρχείο που περιέχει τα ευαίσθητα στοιχειοσύνολα και την επιθυμητή τιμή υποστήριξης. Στην συνέχεια πατώντας το κουμπί GO αναμένει τα αποτελέσματα στον text editor ή και στον καμβά γραφικών παραστάσεων, αναλόγως των επιλογών του. Στο κάτω μέρος του GUI υπάρχουν τα κουμπιά RESET και HELP.



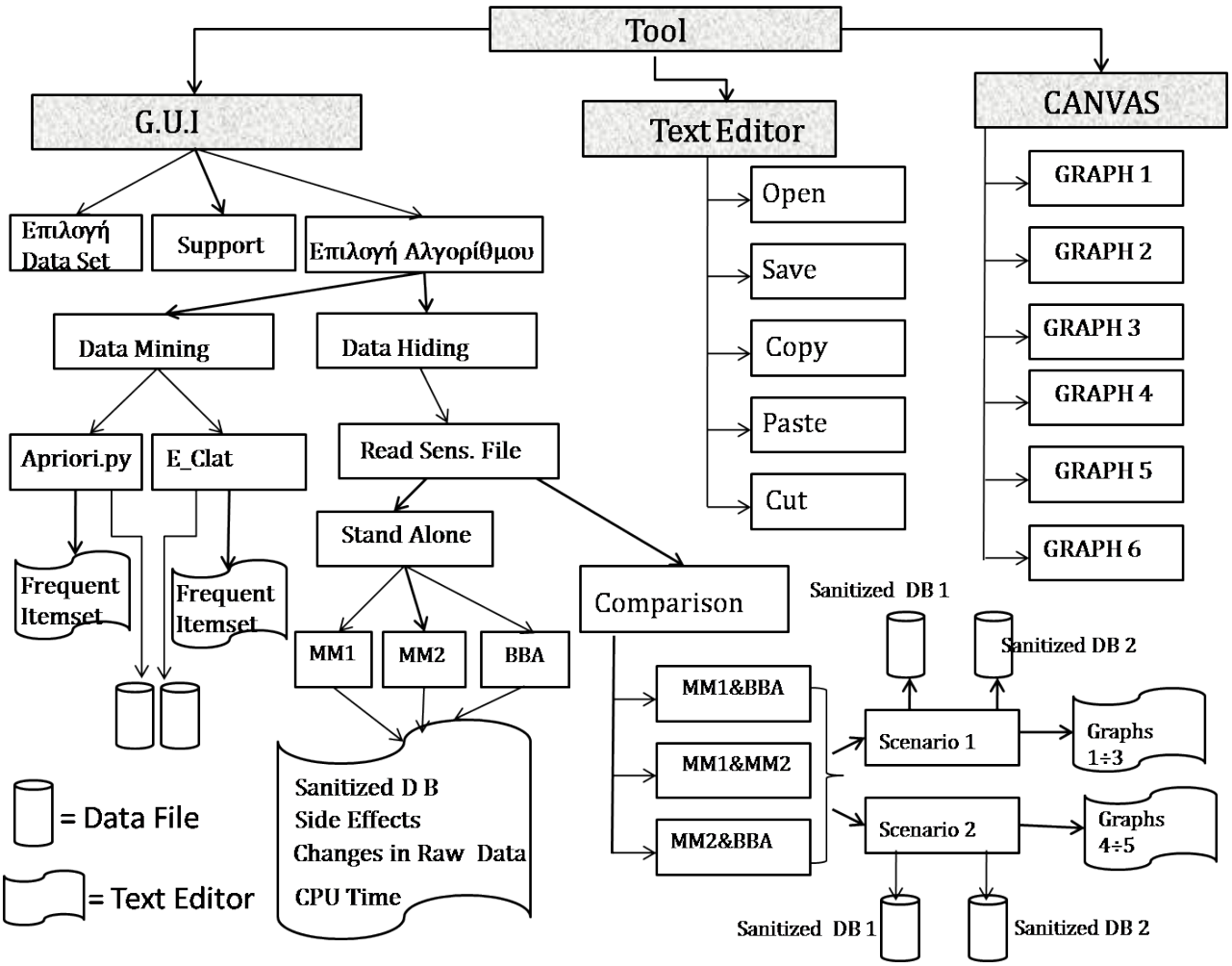
Εικόνα 4.1: Η αρχική εικόνα του εργαλείου

Στο μεσαίο τμήμα έχει υλοποιηθεί ένας απλός text editor στον οποίο παρουσιάζονται τα αποτελέσματα του προγράμματος σε μορφή κειμένου. Ο text editor, εκτός από την προβολή των αποτελεσμάτων δίδει την δυνατότητα στον χρήστη να εκτελεί απλές εργασίες επεξεργασίας κειμένου όπως αντιγραφή, επικόλληση, άνοιγμα, αποθήκευση κειμένου, καθώς και εύρεση χαρακτήρων μέσα στο κείμενο. Οι παραπάνω λειτουργίες αντιστοιχούν στα ισάριθμα μπουτόν “copy”, “paste”, “open”, “save” και “find”.

Στο δεξιό μέρος του εργαλείου έχει υλοποιηθεί ένας καμβάς όπου παρουσιάζονται σε μορφή γραφικών παραστάσεων τα αποτελέσματα του προγράμματος, υπό την προϋπόθεση βεβαίως ότι οι επιλογές εισόδου είναι τέτοιες που να οδηγούν στην δημιουργία γραφικών παραστάσεων. Αυτό συμβαίνει μόνο όταν ο χρήστης επιλέξει τη σύγκριση δύο αλγορίθμων. Η σύγκριση των αλγορίθμων έχει επιλεγεί να γίνεται μέσω της μέτρησης των μεγεθών που θα αναλυθούν παρακάτω. Στο κάτω μέρος του καμβά γραφικών υπάρχουν τα έξι μπουτόν Graph1 έως Graph6, καθένα από τα οποία αντιστοιχεί σε μία συγκριτική γραφική παράσταση.

4.5.2 Λειτουργική περιγραφή του εργαλείου

Το Εργαλείο έχει σχεδιαστεί ώστε να παρέχει μια σειρά από δυνατές επιλογές στον χρήστη. Στο παρακάτω διάγραμμα παρουσιάζεται το σύνολο των λειτουργικών δυνατοτήτων του Εργαλείου.



Σχήμα 4.2: Λειτουργικό διάγραμμα του Εργαλείου.

Όπως φαίνεται και από το παραπάνω διάγραμμα, οι δυνατές λειτουργικές επιλογές είναι οι παρακάτω:

1. Εύρεση συχνών στοιχειοσυνόλων. Από το μενού "Select Rule Mining Algorithm" επιλέγουμε έναν από τους δύο ενσωματωμένους αλγορίθμους (Apriori ή E-Clat), επιλέγουμε αρχείο εισόδου δεδομένων μέσω του Button "Read TIDs from File.." και τέλος εισάγουμε την επιθυμητή τιμή υποστήριξης και πατούμε το κουμπί GO. Στον Text Editor εμφανίζονται τα αποτελέσματα.

2. Απόκρυψη συχνών στοιχειοσυνόλων. Επιλέγουμε έναν από τους τρεις αλγόριθμους απόκρυψης από το αντίστοιχο μενού "Select Rule Hiding Algorithm" και τέλος επιλέγουμε το αρχείο στο βρίσκονται τα ευαίσθητα δεδομένα, εισάγουμε την επιθυμητή τιμή υποστήριξης και πατούμε το κουμπί GO. Στον Text Editor εμφανίζονται τα αποτελέσματα που σε αυτή τη περίπτωση είναι η sanitized βάση δεδομένων και οι μετρικές απόδοσης του αλγορίθμου.

3. Η αξιολόγηση της απόδοσης των αλγορίθμων επιτυγχάνεται μέσω των της συγκριτικής μελέτης μετρικών μεγεθών που έχουν επιλεγεί. Όπως θα παρουσιαστεί στο επόμενο κεφάλαιο οι μετρικές αυτές είναι οι "Changes in Raw Data", "Side Effects" και "CPU Time". Εδώ υπάρχουν οι παρακάτω δύο συγκριτικές δυνατότητες (Σενάρια Σύγκρισης):

    3.1. **Πρώτο Σενάριο**: Σύγκριση δύο αλγορίθμων για διάφορες τιμές υποστήριξης για ένα στοιχειοσύνολο προς απόκρυψη. Για την περίπτωση αυτή κάνουμε, μέσω της διεπαφής, τις απαραίτητες επιλογές για την επιλογή του αρχείου συναλλαγών, του αρχείου ευαίσθητων Σ.Σ., τους αλγορίθμους που θέλουμε να συγκρίνουμε (ανά δύο κάθε φορά). Το πεδίο εισαγωγής υποστήριξης το αφήνουμε κενό. Όπως έχει εξηγηθεί και παραπάνω, ο λόγος που το πεδίο της υποστήριξης το αφήνουμε κενό είναι ότι το πρόγραμμα εκτελεί τη σύγκριση των δύο αλγορίθμων για τιμές υποστήριξης που ξεκινούν από την τιμή 0,1 έως την τιμή 0,9 με βήμα 0,1.

    Στον Text Editor εμφανίζονται τα αποτελέσματα που σε αυτή τη περίπτωση είναι τα μετρικά μεγέθη για τις διάφορες τιμές υποστήριξης και για καθένα από τους δύο αλγόριθμους.

Στον καμβά παρουσιάζονται οι αντίστοιχες γραφικές παραστάσεις των παραπάνω μετρικών μεγεθών σαν συνάρτηση της υποστήριξης. Το button Graph1 παριστά γραφικά το μέγεθος "Changes in Raw Data" συναρτήσει της υποστήριξης, το button Graph2 παριστά το μέγεθος "Side Effects " συναρτήσει της υποστήριξης, ενώ το button Graph3 το "CPU Time" συναρτήσει της υποστήριξης.

Σημειώνουμε ότι είναι ευθύνη του χρήστη να δημιουργήσει κατάλληλο σετ δεδομένων ώστε οι αλγόριθμοι εξόρυξης και απόκρυψης (και συνεπώς το αποτέλεσμα της σύγκρισης των αλγορίθμων) για τις παραπάνω τιμές να έχει νόημα.

3.2. **Δεύτερο Σενάριο**: Σύγκριση δύο αλγορίθμων για απόκρυψη στοιχειοσυνόλων διαφόρων μηκών (αποδεκτά μήκη ευαίσθητων Σ.Σ. είναι από ένα έως δέκα), για κάποια (μία και συγκεκριμένη) τιμή υποστήριξης. Εκτελούμε τα βήματα της προηγούμενης παραγράφου, αλλά βεβαίως εισάγουμε την επιθυμητή τιμή υποστήριξης στο αντίστοιχο πεδίο.

Στον Text Editor εμφανίζονται τα αποτελέσματα που σε αυτή τη περίπτωση είναι τα ως άνω μετρικά μεγέθη για τα διάφορα μήκη των "ευαίσθητων" στοιχειοσυνόλων και για καθένα από τους δύο αλγόριθμους.

Αντίστοιχα, στον καμβά παρουσιάζονται οι γραφικές παραστάσεις των παραπάνω μετρικών μεγεθών σαν συνάρτηση του μήκους των στοιχειοσυνόλων προς απόκρυψη. Τα button Graph4, Graph5, Graph6 δημιουργούν αντίστοιχα γραφικές παραστάσεις των μεγεθών "Changes in Raw Data", " Side Effects " και "CPU Time" συναρτήσει του "Length of Sensitive Itemset".

4.6 Παραδείγματα χρήσης του εργαλείου

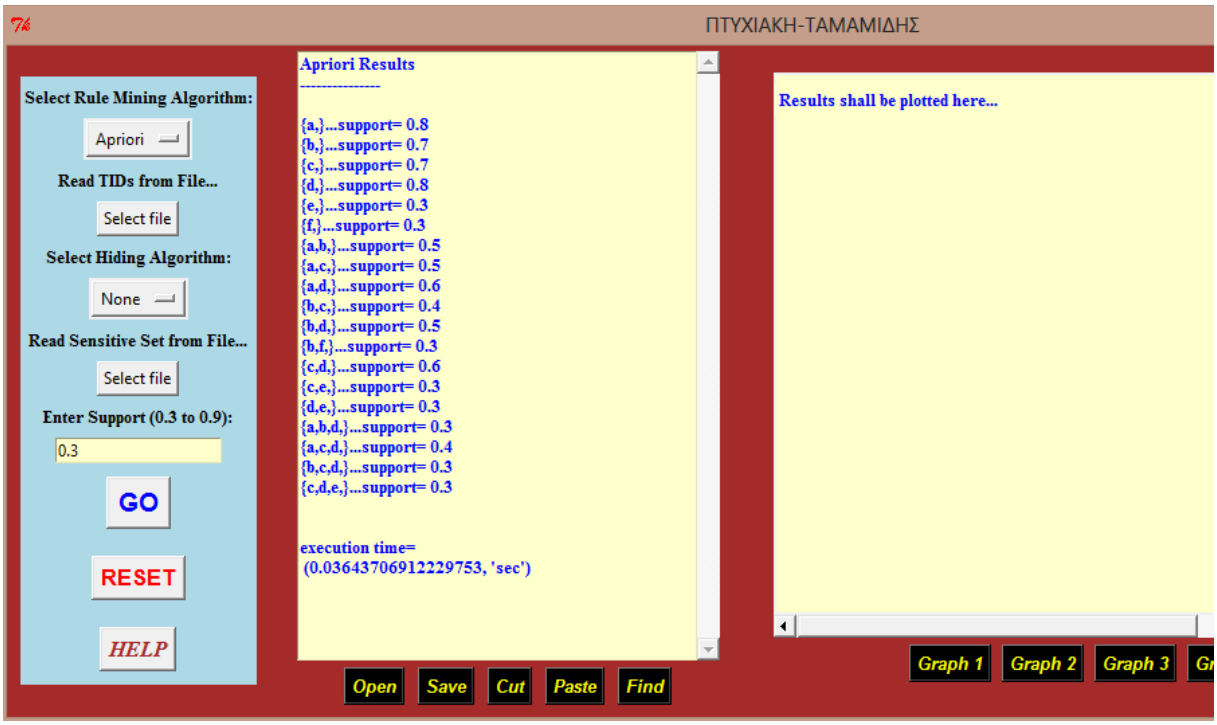
Στην παράγραφο αυτή εξετάζονται διάφορα "σενάρια" χρήσης του εργαλείου. Για τον σκοπό αυτό θα χρησιμοποιήσουμε τις συναλλαγές της Β.Δ. του πίνακα 3.5.

4.6.1 Εύρεση συχνών Σ.Σ.

Για τη περίπτωση αυτή επιλέγουμε μόνο αλγόριθμο data mining από το αντίστοιχο πεδίο, αρχείο εισόδου και τιμή υποστήριξης, αφήνοντας όλα τα υπόλοιπα πεδία κενά. Το πρόγραμμα θα παράξει τα αποτελέσματα, δηλαδή τα συχνά Σ.Σ. της εικόνας 4.2 για τιμή υποστήριξης 0.3.

4.6.2 Εύρεση Sanitized Data Base και απόδοσης αλγορίθμου απόκρυψης

Για τη λειτουργία αυτή, επιλέγουμε έναν από τους δύο αλγόριθμους εξόρυξης και έναν από τους τρείς αλγορίθμους απόκρυψης, τα αρχεία συναλλαγών και ευαίσθητων Σ.Σ. Τέλος, συμπληρώνουμε την επιθυμητή τιμή υποστήριξης. Το πρόγραμμα θα παράξει τα αποτελέσματα της παρακάτω εικόνας στον text editor, τα οποία περιλαμβάνουν την Sanitized Β.Δ., και τις μετρικές απόδοσης των αλγορίθμων απόκρυψης. Σημειώνουμε ότι το αρχείο των ευαίσθητων Σ.Σ. πρέπει να έχει την δομή του πίνακα 4.2 ή 4.3 όπως έχει αναλυθεί.



Εικόνα 4.2: Εύρεση συχνών Σ.Σ. με χρήση του εργαλείου.

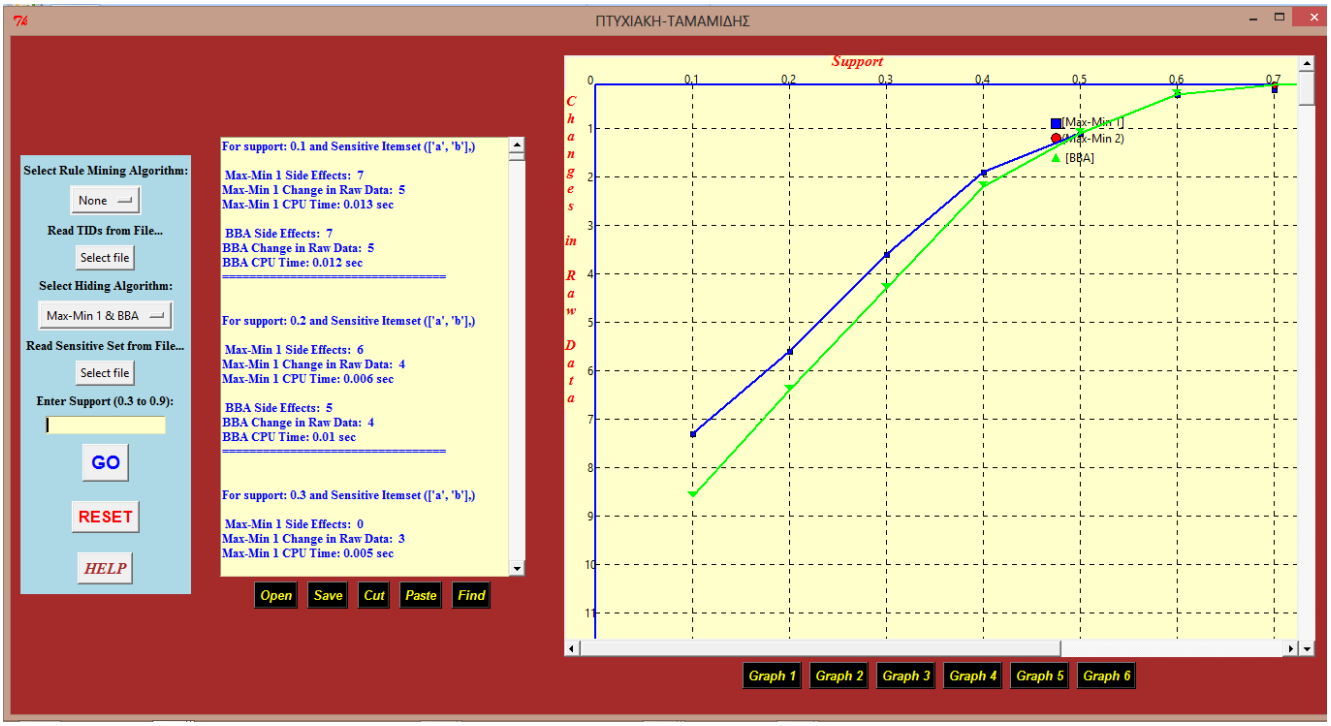
Εικόνα 4.3: Εύρεση της Sanitized Β.Δ. και των μετρικών απόδοσης αλγορίθμων απόκρυψης με χρήση του εργαλείου.

4.6.3 Σύγκριση απόδοσης δύο αλγορίθμων

Όπως έχει προαναφερθεί, τα σενάρια σύγκρισης των αλγορίθμων είναι δύο. Για κάθε ένα από αυτά μπορούν να παραχθούν τρείς γραφικές παραστάσεις, μία για κάθε μια μετρική απόδοσης.

Το πρώτο σενάριο αφορά στην συγκριτική απόδοση δύο αλγορίθμων σαν συνάρτηση της υποστήριξης. Αυτό επιτυγχάνεται πολύ απλά με την παρακάτω διαδικασία. Αφού γίνουν οι επιθυμητές επιλογές με χρήση των μενού επιλογής, αφήνουμε το πεδίο της εισαγωγής τιμής υποστήριξης κενό. Το πρόγραμμα θα υπολογίσει τις μετρικές απόδοσης για διάφορες τιμές υποστήριξης, θα υπολογίσει τις μέσες τιμές και θα παρουσιάσει τα αποτελέσματα αναλυτικά στον text editor και σε μορφή γραφικής παράστασης μέσω των κουμπιών Graph1 έως Graph3. Το κουμπί Graph1 εμφανίζει την πρώτη μετρική (Changes in Raw Data) συναρτήσει της υποστήριξης, το κουμπί Graph2 εμφανίζει την δεύτερη μετρική (Side Effects) συναρτήσει της υποστήριξης και τέλος το κουμπί Graph3 εμφανίζει την τρίτη και τελευταία μετρική (CPU Time) συναρτήσει της υποστήριξης.

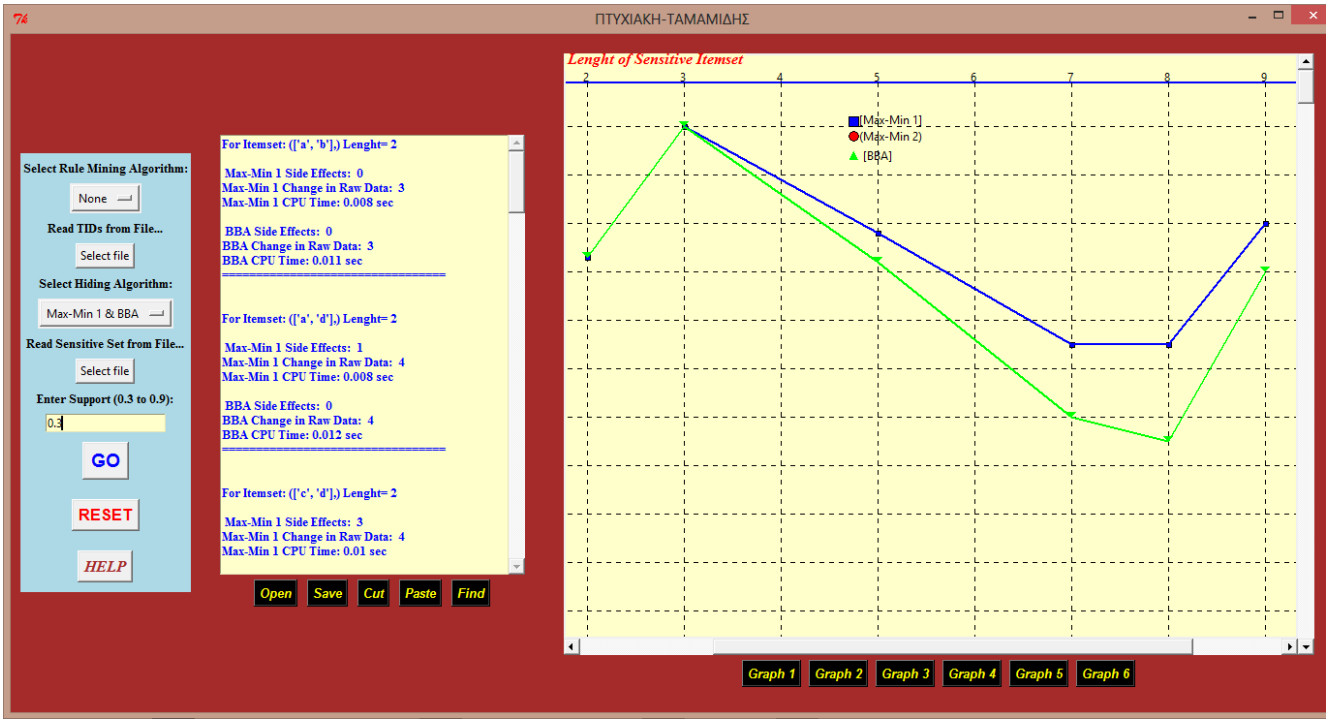
Η εικόνα 4.4 παρουσιάζει τη μετρική Changes in Raw Data συναρτήσει της υποστήριξης για τους δύο αλγορίθμους Max-Min 1 και BBA.



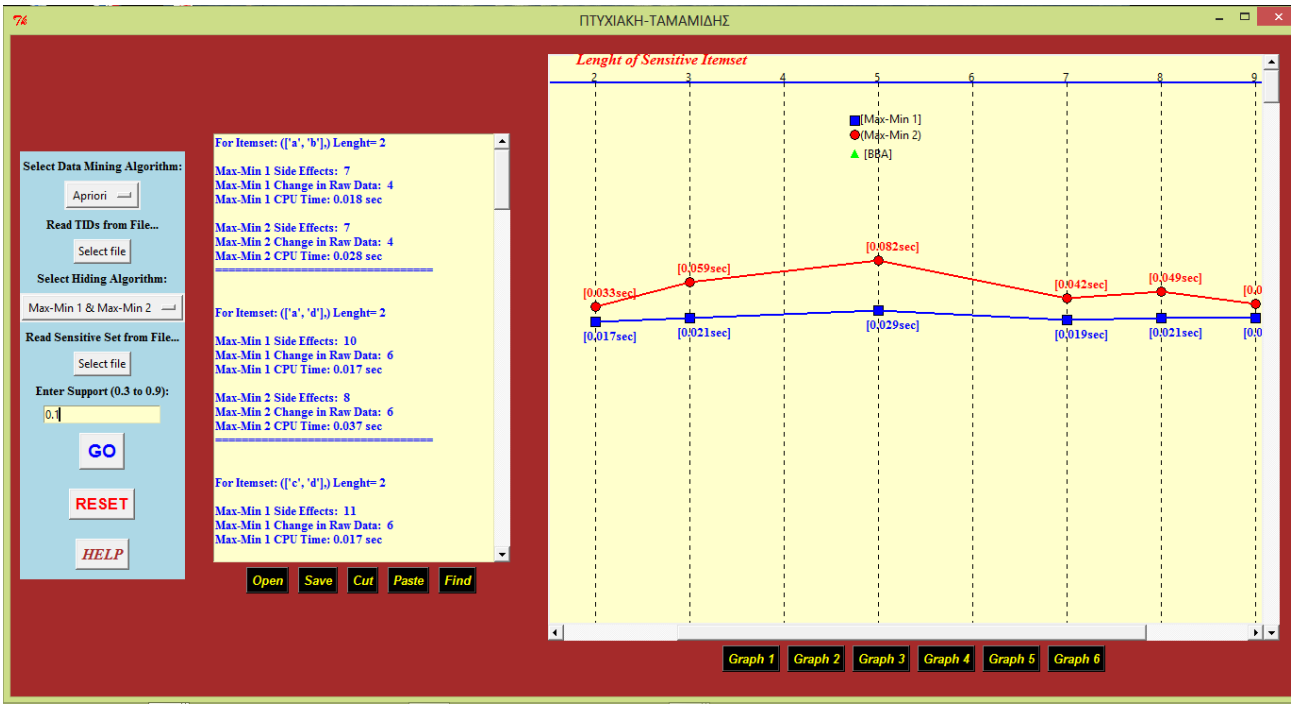
Εικόνα 4.4: Συγκριτική αξιολόγηση της απόδοσης δύο αλγορίθμων μέσω γραφικής παράστασης της μετρικής Changes in Raw Data συναρτήσει της υποστήριξης (Πρώτο Σενάριο σύγκρισης).

Το δεύτερο σενάριο αφορά στην συγκριτική απόδοση δύο αλγορίθμων σαν συνάρτηση της του μήκους του προς απόκρυψη Σ.Σ., για κάποια συγκεκριμένη τιμή υποστήριξης. Για να επιτευχθεί αυτό ακολουθούμε την ίδια με παραπάνω διαδικασία με μόνη διαφορά ότι στο πεδίο της εισαγωγής τιμής υποστήριξης ορίζουμε την επιθυμητή τιμή. Το πρόγραμμα θα υπολογίσει τις μετρικές απόδοσης για τα διάφορα μήκη ευαίσθητων Σ.Σ. που υπάρχουν στο αρχείο ευαίσθητων Σ.Σ., θα υπολογίσει τις μέσες τιμές και θα παρουσιάσει τα αποτελέσματα αναλυτικά στον text editor και σε μορφή γραφικής παράστασης μέσω των κουμπιών Graph4 έως Graph6. Το κουμπί Graph4 εμφανίζει την πρώτη μετρική (Changes in Raw Data), το κουμπί Graph5 εμφανίζει την δεύτερη μετρική (Side Effects) και τέλος το κουμπί Graph6 εμφανίζει την τρίτη και τελευταία μετρική (CPU Time) συναρτήσει του μήκους του προς απόκρυψη Σ.Σ.

Οι εικόνες 4.5 και 4.6 παρουσιάζουν τις μετρικές Changes in Raw Data και CPU Time αντίστοιχα, συναρτήσει του μήκους του ευαίσθητου Σ.Σ. για τους δύο αλγορίθμους Max-Min 1 και BBA.



Εικόνα 4.5: Συγκριτική αξιολόγηση της απόδοσης δύο αλγορίθμων μέσω γραφικής παράστασης της μετρικής Changes in Raw Data συναρτήσει του μήκους του Σ.Σ. προς απόκρυψη (Δεύτερο Σενάριο σύγκρισης).



Εικόνα 4.6: Συγκριτική αξιολόγηση της απόδοσης δύο αλγορίθμων μέσω γραφικής παράστασης της μετρικής CPU Time συναρτήσει του μήκους του Σ.Σ. προς απόκρυψη.

```
a b
a d
c d
a b d
c d e
a b;a c d
a c;a b d
a d;b c d
b c;c d e
c e;a b d
a b;d e;a c d
a c;a d;b c d
a c;a b d;c d e
a b;b c;c d;d e
a b d;a c d;c d e
a b d;a c d;b c d
```

Πίνακας 4.7: Ευαίσθητα Σ.Σ. για παραγωγή των εικόνων 4.5 και 4.6

Τέλος, για την όσο το δυνατόν καλύτερη κατανόηση του τρόπου παραγωγής των αποτελεσμάτων της εφαρμογής, παρουσιάζουμε στον πίνακα 4.8 αναλυτικά τα αποτελέσματα που παράγονται στον text editor για τα ευαίσθητα Σ.Σ. του πίνακα 4.7. Τα αποτελέσματα των γραφικών παραστάσεων των εικόνων 4.5 και 4.6 προκύπτουν από τις μέσες τιμές του πίνακα 4.8 για τα διάφορα μήκη Σ.Σ.

```
For Itemset: ([a, b],) Length= 2

Max-Min 1 Side Effects:  0
Max-Min 1 Change in Raw Data:  3
Max-Min 1 CPU Time: 0.008 sec

BBA Side Effects:  0
BBA Change in Raw Data:  3
BBA CPU Time: 0.011 sec
================================
For Itemset: ([a, d],) Length= 2

Max-Min 1 Side Effects:  1
Max-Min 1 Change in Raw Data:  4
Max-Min 1 CPU Time: 0.008 sec

BBA Side Effects:  0
BBA Change in Raw Data:  4
BBA CPU Time: 0.012 sec
================================
For Itemset: ([c, d],) Length= 2
```

```
Max-Min 1 Side Effects:  3
Max-Min 1 Change in Raw Data:  4
Max-Min 1 CPU Time: 0.01 sec

BBA Side Effects:  1
BBA Change in Raw Data:  4
BBA CPU Time: 0.01 sec
================================
For Itemset: ([a, b, d],) Length= 3

Max-Min 1 Side Effects:  0
Max-Min 1 Change in Raw Data:  1
Max-Min 1 CPU Time: 0.005 sec

BBA Side Effects:  0
BBA Change in Raw Data:  1
BBA CPU Time: 0.009 sec
================================
For Itemset: ([c, d, e],) Length= 3

Max-Min 1 Side Effects:  2
Max-Min 1 Change in Raw Data:  1
Max-Min 1 CPU Time: 0.006 sec

BBA Side Effects:  3
BBA Change in Raw Data:  1
BBA CPU Time: 0.009 sec
================================
For Itemset: ([a, b], [a, c, d]) Length= 5

Max-Min 1 Side Effects:  0
Max-Min 1 Change in Raw Data:  4
Max-Min 1 CPU Time: 0.007 sec

BBA Side Effects:  0
BBA Change in Raw Data:  5
BBA CPU Time: 0.017 sec
================================
For Itemset: ([a, c], [a, b, d]) Length= 5

Max-Min 1 Side Effects:  0
Max-Min 1 Change in Raw Data:  3
Max-Min 1 CPU Time: 0.008 sec

BBA Side Effects:  0
BBA Change in Raw Data:  4
BBA CPU Time: 0.014 sec
================================
For Itemset: ([a, d], [b, c, d]) Length= 5
```

```
Max-Min 1 Side Effects:  0
Max-Min 1 Change in Raw Data:  5
Max-Min 1 CPU Time: 0.008 sec

BBA Side Effects:  0
BBA Change in Raw Data:  5
BBA CPU Time: 0.016 sec
================================
For Itemset: ([b, c], [c, d, e]) Length= 5

Max-Min 1 Side Effects:  1
Max-Min 1 Change in Raw Data:  2
Max-Min 1 CPU Time: 0.006 sec

BBA Side Effects:  1
BBA Change in Raw Data:  3
BBA CPU Time: 0.013 sec
================================
For Itemset: ([c, e], [a, b, d]) Length= 5

Max-Min 1 Side Effects:  1
Max-Min 1 Change in Raw Data:  2
Max-Min 1 CPU Time: 0.006 sec

BBA Side Effects:  2
BBA Change in Raw Data:  2
BBA CPU Time: 0.012 sec
================================
For Itemset: ([a, b], [d, e], [a, c, d]) Length= 7

Max-Min 1 Side Effects:  2
Max-Min 1 Change in Raw Data:  5
Max-Min 1 CPU Time: 0.008 sec

BBA Side Effects:  2
BBA Change in Raw Data:  6
BBA CPU Time: 0.019 sec
================================
For Itemset: ([a, c], [a, d], [b, c, d]) Length= 7

Max-Min 1 Side Effects:  1
Max-Min 1 Change in Raw Data:  6
Max-Min 1 CPU Time: 0.008 sec

BBA Side Effects:  0
BBA Change in Raw Data:  8
BBA CPU Time: 0.02 sec
================================
For Itemset: ([a, c], [a, b, d], [c, d, e]) Length= 8
```

```
Max-Min 1 Side Effects:  2
Max-Min 1 Change in Raw Data:  4
Max-Min 1 CPU Time: 0.008 sec

BBA Side Effects:  1
BBA Change in Raw Data:  5
BBA CPU Time: 0.019 sec
================================
For Itemset: ([a, b], [b, c], [c, d], [d, e]) Length= 8

Max-Min 1 Side Effects:  1
Max-Min 1 Change in Raw Data:  7
Max-Min 1 CPU Time: 0.012 sec

BBA Side Effects:  3
BBA Change in Raw Data:  10
BBA CPU Time: 0.024 sec
================================
For Itemset: ([a, b, d], [a, c, d], [c, d, e]) Length= 9

Max-Min 1 Side Effects:  2
Max-Min 1 Change in Raw Data:  3
Max-Min 1 CPU Time: 0.01 sec

BBA Side Effects:  3
BBA Change in Raw Data:  4
BBA CPU Time: 0.021 sec
================================
For Itemset: ([a, b, d], [a, c, d], [b, c, d]) Length= 9

Max-Min 1 Side Effects:  0
Max-Min 1 Change in Raw Data:  3
Max-Min 1 CPU Time: 0.009 sec

BBA Side Effects:  0
BBA Change in Raw Data:  4
BBA CPU Time: 0.021 sec
```

Πίνακας 4.6: Αναλυτικά τα αποτελέσματα του text editor των εικόνων 4.5 και 4.6.

Κεφάλαιο 5

Δοκιμή του Εργαλείου Μετρήσεις – Πειράματα

Αντικείμενο του κεφαλαίου αυτού είναι οι μέθοδοι αξιολόγησης αλγορίθμων, η δοκιμή του εργαλείου σε πραγματικές βάσεις δεδομένων και η εξαγωγή συμπερασμάτων. Για το σκοπό αυτό στην πρώτη παράγραφο 5.1 παρουσιάζεται το γενικό θεωρητικό πλαίσιο αξιολόγησης αλγορίθμων καθώς και οι μετρικές που επιλέχθηκαν στο πλαίσιο της παρούσης Μεταπτυχιακής διατριβής, για την ενσωμάτωση στο εργαλείο. Στην παράγραφο 5.2 παρουσιάζουμε την μέθοδο που ακολουθεί το εργαλείο για τη σύγκριση των αλγορίθμων, ενώ στην επόμενη, 5.3 περιγράφονται τα πειράματα που έχουν γίνει με πραγματικές βάσεις δεδομένων, τα γενικά χαρακτηριστικά αυτών των Β.Δ., τα αντίστοιχα αποτελέσματα σε μορφή κειμένου και γράφων, ενώ τέλος στην παράγραφο 5.4 επιχειρούμε μια αξιολόγηση της απόδοσης τόσο του εργαλείου όσο και των αλγορίθμων που εξετάσθηκαν, κάνουμε αναφορά σε σχετικές εργασίες και τέλος, παρουσιάζουμε μια σειρά από δυνατότητες μελλοντικής επέκτασης του Εργαλείου που αναπτύξαμε.

5.1 Γενικό πλαίσιο αξιολόγησης αλγορίθμων

Ο μεγάλος αριθμός των αλγορίθμων που έχουν προταθεί στο πλαίσιο της απόκρυψης κανόνων συσχέτισης κατέστησε αναγκαία την ανάπτυξη μεθόδων αξιολόγησης για την ποσοτική σύγκριση καθώς και κατάταξη αυτών των αλγορίθμων βασιζόμενοι σε διαφορετικές μετρικές. Μέχρι στιγμής, υπάρχει μια εργασία των Bertino et al. [15], ενώ φαίνεται ότι οι νέες μελέτες αξιολόγησης είναι αναγκαίες και θα συμβάλλουν σημαντικά στην πρόοδο της περιοχής. Το έργο των Bertino et al. προτείνει ένα πλαίσιο αξιολόγησης που βασίζεται σε έναν συγκεκριμένο αριθμό μετρικών που επιλέγονται κατάλληλα για τη σύγκριση των αλγορίθμων. Ειδικότερα, οι συγγραφείς έχουν προσδιορίσει πέντε σημαντικές πτυχές της αξιολόγησης:

- Αποδοτικότητα (*Efficiency*): Η εκτίμηση των πόρων του υπολογιστικού συστήματος που χρησιμοποιούνται από έναν Α.Κ.Σ. προσδιορίζει την απόδοσή του. Οι πόροι αυτοί, στις πιο συνηθισμένες περιπτώσεις είναι η χρονική διάρκεια απασχόλησης της Κεντρικής Μονάδας Επεξεργασίας και η ποσότητα Μνήμης που δεσμεύεται για την εκτέλεση του προγράμματος. Μια σημαντική επισήμανση των Bertino et al. είναι ότι είναι χρήσιμο οι μέθοδοι αξιολόγησης να συγκρίνουν τις επιδόσεις του αλγορίθμου εξόρυξης με αυτή του αλγορίθμου απόκρυψης. Το προσδοκώμενο αποτέλεσμα θα ήταν μια αναλογική σχέση μεταξύ τους.

- Επεκτασιμότητα (*Scalability*): Η μετρική αυτή περιγράφει την τάση που ακολουθεί η μετρική της αποδοτικότητας καθώς μεταβάλλεται το μέγεθος των αποθηκευτικών δεδομένων. Συνεπώς, η μετρική της Επεκτασιμότητας αφορά όλες τις παραμέτρους από τις οποίες επηρεάζεται η Αποδοτικότητα. Όπως έχει αναφερθεί και στην εισαγωγή, η μεγάλη πρόοδος που σημειώθηκε στην τεχνολογία του υλικού έχει οδηγήσει στην δημιουργία μεγάλων αποθετηρίων δεδομένων, τα οποία και ολοένα γίνονται μεγαλύτερα. Είναι λοιπόν εύλογο ένας αλγόριθμος απόκρυψης να σχεδιαστεί ούτως ώστε να είναι επεκτάσιμος σε ολοένα και μεγαλύτερα data sets. Η επεκτασιμότητα ενός αλγορίθμου απόκρυψης είναι τόσο καλύτερη όσο λιγότερο απότομη είναι η μείωση της αποδοτικότητας για αυξανόμενων διαστάσεων data set.

- Ποιότητα Δεδομένων (*Data quality*): Είναι μια σύνθετη παράμετρος η οποία έχει διάφορους τρόπους ερμηνείας. Το κυρίως θέμα της Ποιότητας Δεδομένων είναι τα δεδομένα που προκύπτουν μετά την εφαρμογή ενός αλγορίθμου απόκρυψης να περικλείουν την πληροφορία που περιελάμβαναν πριν την εφαρμογή του αλγορίθμου απόκρυψης με αποκλεισμό των ευαίσθητων δεδομένων και μόνο αυτών. Αν και έχουν

προταθεί διάφορες μετρικές για την Ποιότητα Δεδομένων, οι οποίες είναι είτε γενικής φύσεως και μπορούν να εφαρμοστούν σε κάθε περίπτωση δεδομένων είτε όχι, ωστόσο, μέχρι και αυτή τη στιγμή δεν υπάρχει κάποια μετρική που να είναι καθολικά αποδεκτή από την επιστημονική κοινότητα. Ως παράδειγμα της παραμέτρου της Ποιότητας Δεδομένων μπορούμε να θεωρήσουμε την διατήρηση της σχέσης της τιμής υποστήριξης ανάμεσα στα μη ευαίσθητα συχνά Σ.Σ., πριν και μετά το τέλος της διαδικασίας απόκρυψης, κάτι το οποίο όπως είδαμε στην παράγραφο 3.2 αποτελεί ιδιότητα του αλγορίθμου BBA.

- Αποτυχία Απόκρυψης (*Hiding failure*): Το ποσοστό των ευαίσθητων πληροφοριών που είναι εξακολουθεί να ανακαλύπτεται, μετά την απολύμανση της αρχικής Β.Δ., δίνει μια εκτίμηση της παραμέτρου της Αποτυχίας Απόκρυψης. Οι περισσότεροι από τους γνωστούς αλγόριθμους απόκρυψης έχουν σχεδιαστεί με στόχο την επίτευξη μηδενικής αποτυχίας απόκρυψης. Έτσι, κρύβουν όλα τα πρότυπα που θεωρούνται ευαίσθητα. Ωστόσο, είναι γνωστό ότι όσο περισσότερες είναι οι ευαίσθητες πληροφορίες που κρύβουμε, τόσο περισσότερες είναι και οι μη-ευαίσθητες πληροφορίες που χάνονται. Έτσι, ορισμένοι από τους αλγορίθμους που έχουν αναπτυχθεί προσφάτως επιτρέπουν στο χρήστη να επιλέξει το ποσό των ευαίσθητων δεδομένων που πρέπει να κρυφτεί, προκειμένου να βρεθεί μια ισορροπία μεταξύ της ιδιωτικής ζωής και την ανακάλυψης γνώσης.

5.2 Εργαλείο: Μετρικές και Μέθοδος σύγκρισης αλγορίθμων

Στη παράγραφο αυτή εξετάζουμε τη μεθοδολογία σύγκρισης - αξιολόγησης των αλγορίθμων, όπως έχει υλοποιηθεί στο εργαλείο, και τις μετρικές αξιολόγησης που έχουν επιλεγεί στο πλαίσιο της παρούσης Μεταπτυχιακής διατριβής. Ωστόσο, κάποιες από τις παραμέτρους που συζητήθηκαν παραπάνω δεν υλοποιήθηκαν στο πλαίσιο της παρούσης Μεταπτυχιακής διατριβής αλλά τοποθετήθηκαν ως μελλοντικές δυνατότητες επέκτασης του εργαλείου. Ειδική μνεία για αυτές γίνεται στο κεφάλαιο 6.

5.2.1 Μετρικές που επιλέχθηκαν

Έχοντας θέσει το γενικό - θεωρητικό πλαίσιο αξιολόγησης των αλγορίθμων απόκρυψης της προηγούμενης παραγράφου μπορούμε να προχωρήσουμε στην περιγραφή των μετρικών που επιλέχθηκαν για να ενσωματωθούν στο εργαλείο της παρούσης Μεταπτυχιακής διατριβής. Έτσι, για την μέτρηση της αποδοτικότητας (*Efficiency*) έχουμε επιλέξει την μέτρηση του χρόνου

εκτέλεσης του αλγορίθμου (CPU time). Ο χρόνος αυτός έχει μετρηθεί για το βασικό βρόχο εκτέλεσης των αλγορίθμων, αφήνοντας εκτός μέτρησης την ανάγνωση του αρχείου των δεδομένων, τον χειρισμό των δεδομένων, την εγγραφή των αποτελεσμάτων στα αρχεία εξόδου κ.τ.λ. Σημειώνουμε ότι, καθώς ο χρόνος αυτός ενδέχεται να επηρεάζεται σημαντικά από διάφορες παράλληλες εργασίες που εκτελεί ο επεξεργαστής, καταβλήθηκε προσπάθεια εκτέλεσης των αλγορίθμων αφού πρώτα είχαν αποκλειστεί τυχόν άλλες εργασίες που εκτελούσε ο επεξεργαστής. Σε κάθε περίπτωση, και για την καλύτερη αποτύπωση του χρόνου εκτέλεσης, οι αλγόριθμοι έτρεξαν πολλές φορές για τα ίδια σετ δεδομένων και αποκλείστηκαν από την καταγραφή διάφορες τιμές που εμφανώς ήταν επηρεασμένες από την αιτία που αναφέρθηκε παραπάνω.

Η ποιότητα των δεδομένων μετρήθηκε μέσω των "παρενεργειών" (side effects). Όπως έχει αναφερθεί και σε προηγούμενα κεφάλαια η απόκρυψη ενός ευαίσθητου Σ.Σ. ή Κ.Σ. πολλές φορές έχει σαν συνέπεια και την απόκρυψη μη ευαίσθητων Σ.Σ. ή Κ.Σ. με συνεπακόλουθο την υποβάθμιση της πληροφορίας που μπορεί να εξαχθεί από την καθαρισμένη (Sanitized) Β.Δ. Ο αριθμός αυτός, των μη ευαίσθητων Σ.Σ. που από συχνά γίνονται σπάνια, αποτελεί τις παρενέργειες. Ο αριθμός των παρενεργειών περιορίζεται από τον αριθμό των συχνών Σ.Σ. του Αναθεωρημένου Θετικού Συνόρου.

Η τρίτη μετρική που ενσωματώθηκε στο εργαλείο συνίσταται στην μέτρηση του αριθμού των επεμβάσεων - αλλαγών στα αρχικά δεδομένα (changes in Raw Data) που πραγματοποιεί κάθε αλγόριθμος προκειμένου να πετύχει την απόκρυψη κάποιου Σ.Σ. Ο αριθμός αυτός, ο οποίος είναι σημαντικός σε ορισμένες εφαρμογές, έχει ως άνω φράγμα τον αριθμό των επαναλήψεων που απαιτούνται προκειμένου να μειωθεί η υποστήριξη του ευαίσθητου Σ.Σ. κάτω από το κατώφλι υποστήριξης. Ωστόσο, ο αριθμός αυτός μπορεί να είναι διαφορετικός για κάθε αλγόριθμο για τον λόγο ότι σε κάθε βήμα του αλγορίθμου μπορεί να μειώνεται και η υποστήριξη κάποιων Σ.Σ. του Α.Θ.Σ.

Τέλος, στο σημείο αυτό αναφέρουμε ότι δεν υλοποιήθηκε κάποια μετρική για την Αποτυχία Απόκρυψης (*Hiding failure*) καθώς και οι τρεις αλγόριθμοι που ενσωματώθηκαν στο εργαλείο εξασφαλίζουν την βέβαιη απόκρυψη όλων των ευαίσθητων Σ.Σ.

5.2.2 Μέθοδος σύγκρισης αλγορίθμων

Η μέθοδος που ακολουθεί το Εργαλείο προκειμένου να συγκρίνει τους, δύο κάθε φορά, αλγορίθμους συνίσταται στην μελέτη των συχνών Σ.Σ. πριν και μετά την διαδικασία απόκρυψης. Έστω για παράδειγμα ότι ο χρήστης του Εργαλείου επιλέγει την σύγκριση των Max-Min 1 και

BBA για κάποια τιμή υποστήριξης. Η διαδικασία σύγκρισης - αξιολόγησης μπορεί να επικεντρωθεί στα παρακάτω βήματα:

1. Αρχικά, μέσω του module εξόρυξης Apriori θα βρεθούν όλα τα συχνά Σ.Σ. και οι αντίστοιχες τιμές υποστήριξης που αντιστοιχούν στο αρχείο εισόδου .dat που περιέχει την αρχική Β.Δ.

2. Εκτελούνται διαδοχικά τα modules των αλγορίθμων απόκρυψης που έχουν επιλεγεί. Το βήμα αυτό προϋποθέτει την ύπαρξη κατάλληλα σχεδιασμένου αρχείου ευαίσθητων δεδομένων, ώστε η συγκριτική αξιολόγηση των αλγορίθμων, για την δοσμένη τιμή msup να έχει νόημα. Επίσης, στο βήμα αυτό υπολογίζεται το Α.Θ.Σ. των συχνών Σ.Σ. που αντιστοιχεί στο τρέχον σετ ευαίσθητων δεδομένων και καταγράφονται σε μεταβλητές ο χρόνος εκτέλεσης του αλγορίθμου, ο αριθμός των επεμβάσεων - αλλαγών στην αρχική Β.Δ.. Τέλος, στο βήμα αυτό δημιουργούνται τα αρχεία sanitized_1.txt και sanitized_2.txt που αντιστοιχούν στις καθαρές πλέον Β.Δ. των αλγορίθμων 1 (Max-Min 1) και 2 (BBA) αντίστοιχα.

3. Εκτελείται εκ' νέου το module εξόρυξης Apriori και παράγονται τα συχνά Σ.Σ. που αντιστοιχούν στις καθαρές Β.Δ. Αυτό σημαίνει ότι ο Apriori εκτελείται δύο φορές με εισόδους τα αρχεία sanitized_1.txt και sanitized_2.txt που έχουν παράξει, για το παράδειγμά μας, οι Max-Min 1 και BBA αντίστοιχα.

4. Στο βήμα αυτό θα συγκριθούν τα αποτελέσματα του βήματος 1 με αυτά του βήματος 3 προκειμένου να προκύψει ο αριθμός των side effects.

Αν και έχει αναλυθεί στο κεφάλαιο 4, ωστόσο σημειώνουμε και πάλι ότι, όλα τα παραπάνω βήματα εκτελούνται μέσω του module Compare.py. Ο ρόλος του module Tool.py είναι να υλοποιήσει την διεπαφή, να καλέσει το module Compare.py και να παρουσιάσει τα αποτελέσματα σε μορφή κειμένου και γραφικών παραστάσεων.

5.3 Εργαλείο: Εκτέλεση Πειραμάτων – Αποτελέσματα

Στην παράγραφο αυτή περιγράφουμε τα πειράματα που έχουν πραγματοποιηθεί με το εργαλείο κάνοντας χρήση έτοιμων βάσεων δεδομένων, τα βασικά χαρακτηριστικά των βάσεων δεδομένων με τις οποίες διενεργήθηκαν τα πειράματα και τα αντίστοιχα αποτελέσματα.

Οι βάσεις δεδομένων που χρησιμοποιήθηκαν για δοκιμές με το Εργαλείο είναι οι Chess, η Mushroom, και η Retail. Και οι τρεις αντλήθηκαν από το διαδίκτυο, και συγκεκριμένα από την

τοποθεσία http://fimi.ua.ac.be/data/. Ο αναγνώστης που ενδιαφέρεται για τον τρόπο παραγωγής των συγκεκριμένων data set μπορεί να ανατρέξει στον σχετικό σύνδεσμο.

Όπως αναφέρθηκε παραπάνω, στις παραγράφους που ακολουθούν, γίνεται μεταξύ των άλλων και μία σύντομη περιγραφή των χαρακτηριστικών των Β.Δ. Ο λόγος για τον οποίο γίνεται η παρουσίαση των γενικών χαρακτηριστικών των Β.Δ. είναι διότι με τον τρόπο αυτό θα γίνει καλύτερα αντιληπτή η μεθοδολογία σχεδίασης και εκτέλεσης των πειραμάτων αλλά, σε κάποιο βαθμό, και των αποτελεσμάτων που προέκυψαν. Όπως θα φανεί παρακάτω η κάθε βάση δεδομένων έχει τα δικά της ιδιαίτερα χαρακτηριστικά, κάτι που επιβάλει τον διαφορετικό σχεδιασμό των πειραμάτων για κάθε μία από αυτές.

Αξίζει ίσως να σημειωθεί ότι βασικό παράγοντα στο σχεδιασμό των πειραμάτων αποτέλεσε η ελαχιστοποίηση του χρόνου εκτέλεσης των πειραμάτων. Και αυτό διότι, όπως θα γίνει αντιληπτό στις παραγράφους που ακολουθούν, η εκτέλεση ενός πειράματος σε μια μεγάλη βάση δεδομένων μπορεί να είναι πολύ χρονοβόρα υπόθεση εάν δεν σχεδιαστεί κατάλληλα με βάση τους στόχους που έχουν τεθεί.

Ο κύριος στόχος των πειραμάτων ήταν η δοκιμή και η λειτουργική αξιολόγηση του εργαλείου σε μεγάλων διαστάσεων data set. Αν και η ορθότητα της υλοποίησης των αλγορίθμων ελέγχθηκε σε μικρότερες Β.Δ. όπου είναι ευκολότερος ο έλεγχος των αποτελεσμάτων, ωστόσο η δοκιμή σε μεγάλες Β.Δ. είναι δυνατόν να οδηγήσει σε αποκάλυψη τυχόν λειτουργικών δυσλειτουργιών ή αδυναμιών στο λογισμικό που αναπτύχθηκε. Επίσης σημαντικός στόχος ήταν και η μελέτη της συμπεριφοράς των αλγορίθμων μέσω της μεταβολής των χαρακτηριστικών τους, κάτι που αποτελεί κεντρικό θέμα της παρούσης Μεταπτυχιακής διατριβής. Δευτερεύοντα ρόλο στο σχεδιασμό και εκτέλεση των πειραμάτων αποτέλεσε η καθεαυτή σύγκριση των αλγορίθμων κι αυτό λόγω της ύπαρξης αρκετών επιστημονικών εργασιών με αναφορές σε αυτούς.

Τέλος, η παρουσίαση των αποτελεσμάτων έγινε με την βοήθεια γραφημάτων που κατασκευάστηκαν από το μέσο όρο των τιμών που απέδωσε το εργαλείο για τις διάφορες δοκιμές που έγιναν για κάθε ευαίσθητο Σ.Σ. (λεπτομέρειες δίδονται παρακάτω για κάθε Β.Δ.). Σε ορισμένες περιπτώσεις έχουν δοθεί και τα γραφήματα που δημιούργησε το ίδιο το εργαλείο. Οι λόγοι για τους οποίους δεν δόθηκαν τα γραφήματα του εργαλείου σε όλες τις περιπτώσεις είναι κυρίως δύο. Ο πρώτος λόγος είναι διότι ο καμβάς γραφικών απεικονίσεων του εργαλείου είναι scrollable και οι γραφικές παραστάσεις καταλαμβάνουν μεγάλο μέρος του. Ο χρήστης του εργαλείου κάνει scroll down για να δει τα αποτελέσματα, κάτι το οποίο φυσικά δεν μπορεί να γίνει παρουσιάζοντας τη γραφική παράσταση σε αρχείο κειμένου. Αποτέλεσμα αυτού του γεγονότος είναι το να απαιτείται κάποιου είδους επεξεργασία εικόνας ώστε να προκύψει καλό

αποτέλεσμα παρουσιάσιμο στο MS Word, κάτι το οποίο δεν είναι πάντα εφικτό. Ο δεύτερος λόγος είναι ότι οι εικόνες που προκύπτουν είναι σχετικά μεγάλες (περίπου 300 KB η καθεμιά), κάτι που ανέβαζε πολύ (πέραν των ορίων που έχουν τεθεί) το συνολικό μέγεθος του αρχείου της παρούσης Μεταπτυχιακής διατριβής.

5.3.1 Πειράματα με την Chess

Η πρώτη Β.Δ. με την οποία διενεργήσαμε πειράματα ήταν η Chess. Αυτή αποτελείται από 3196 συναλλαγές των σαράντα περίπου στοιχεία η κάθε μία, ενώ το σύνολο των στοιχείων που απαρτίζουν το – 1 Itemset είναι εβδομήντα πέντε. Η Β.Δ. Chess αποδίδει 622 συχνά Σ.Σ. για τιμή υποστήριξης *msup*=0.9, για τιμή υποστήριξης *msup*=0.8 αποδίδει 8.228 συχνά Σ.Σ. ενώ για μικρότερες τιμές υποστήριξης το σύνολο των Σ.Σ. ανέρχεται σε πολλές χιλιάδες, πράγμα που καθιστά την εκτέλεση πειραμάτων σε χαμηλότερες τιμές υποστήριξης χρονοβόρα. Έτσι, η Β.Δ. Chess χρησιμοποιήθηκε για δύο κατηγορίες δοκιμών:

a) Στην πρώτη κατηγορία δοκιμών χρησιμοποιήθηκε σταθερή τιμή στο κατώφλι υποστήριξης *msup*=0.9 και μεταβαλλόμενο μήκος ευαίσθητων Σ.Σ. με τιμές από Length=2 έως Length=6. Σε αυτή τη κατηγορία δοκιμών το αρχείο των ευαίσθητων δεδομένων σχεδιάστηκε ώστε να περιλαμβάνει τουλάχιστον πέντε και έως δέκα Σ.Σ. από κάθε κατηγορία μήκους, ενώ το πείραμα εκτελέστηκε τουλάχιστον δύο και έως πέντε φορές, ώστε τυχόν αποκλίσεις ανάμεσα στα αποτελέσματα (κυρίως όσον αφορά τη χρονική περίοδο εκτέλεσης) να εξομαλυνθούν με την εύρεση των μέσων τιμών των αποτελεσμάτων. Στα σχήματα 5.1 έως 5.3 δίδονται τα αποτελέσματα των πειραμάτων αυτής της κατηγορίας.

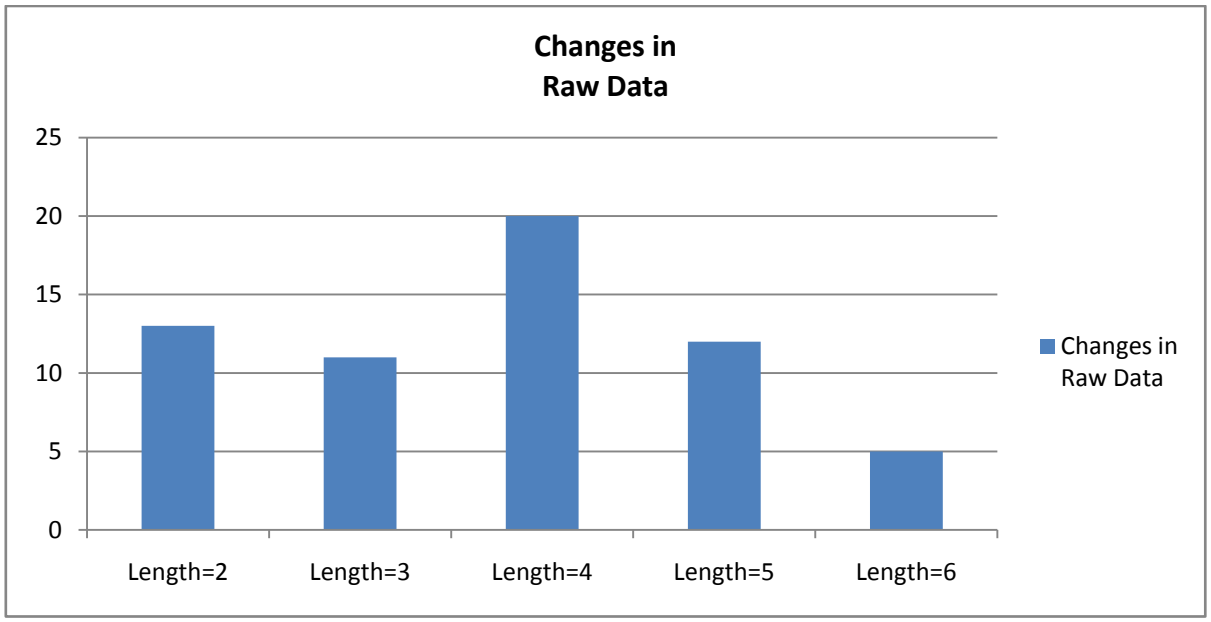
b) Στην δεύτερη κατηγορία πειραμάτων κρατήσαμε σταθερό μήκος ευαίσθητου Σ.Σ. και μεταβαλλόμενη τιμή υποστήριξης από 0,9 έως 0.95 με βήμα αύξησης 0.01. Το μήκος του ευαίσθητου Σ.Σ. επιλέχθηκε να έχει την τιμή 4. Επίσης, το αρχείο των ευαίσθητων δεδομένων σχεδιάστηκε ώστε να περιλαμβάνει τουλάχιστον πέντε Σ.Σ. του μήκους που επιλέχθηκε. Και σε αυτή τη περίπτωση το πείραμα εκτελέστηκε τουλάχιστον δύο και έως πέντε φορές, ώστε τυχόν αποκλίσεις ανάμεσα στα αποτελέσματα (κυρίως όσον αφορά τη χρονική περίοδο εκτέλεσης) να εξομαλυνθούν με την εύρεση των μέσων τιμών των αποτελεσμάτων. Στα σχήματα 5.4 έως 5.6 δίδονται τα αντίστοιχα αποτελέσματα.

Σημειώνουμε ότι και στις δύο κατηγορίες πειραμάτων, η επιλογή των ευαίσθητων Σ.Σ. δεν αφέθηκε στην τύχη, αλλά επιλέχθηκε ώστε να βρίσκεται σχετικά κοντά στο κατώφλι υποστήριξης, καθώς κάποιο ευαίσθητο Σ.Σ. με τιμή υποστήριξης λ. χ. 0.99 θα απαιτούσε πολλές

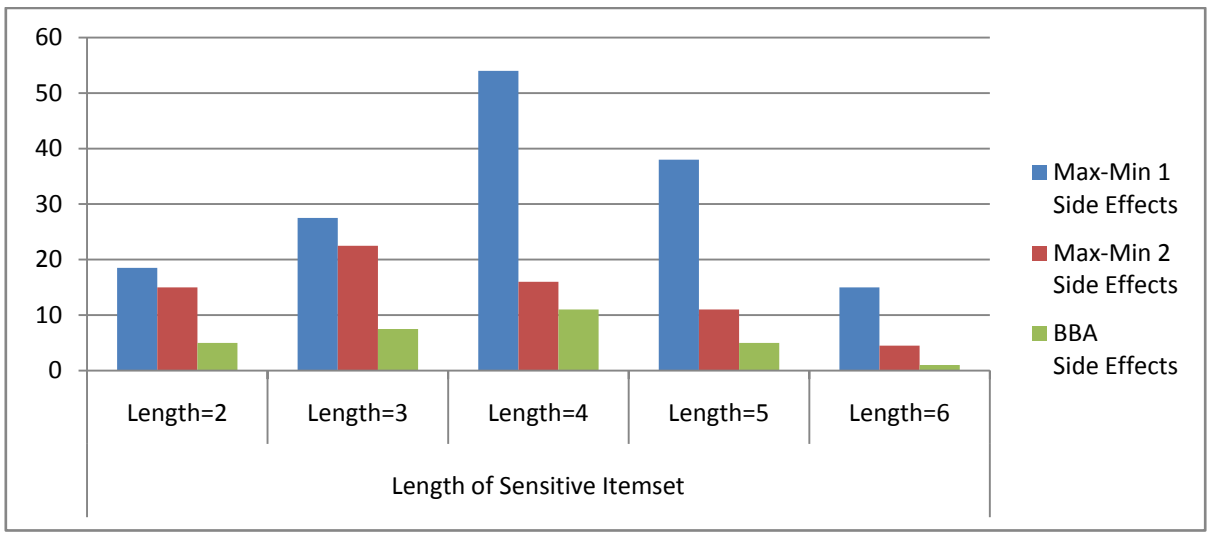
χιλιάδες επαναλήψεις για να αποκρυφτεί. Αντίθετα επιλέχθηκαν ευαίσθητα Σ.Σ. με τιμή υποστήριξης κοντά στην τιμή 0,95. Το γεγονός αυτό αντικατοπτρίζεται στο πρώτο διάγραμμα το οποίο παρουσιάζουμε, (σχήμα 5.1) το οποίο παριστά τον αριθμό των αλλαγών στη Β.Δ. συναρτήσει του μήκους του ευαίσθητου Σ.Σ.

Αποτελέσματα - Διαγράμματα με την Chess

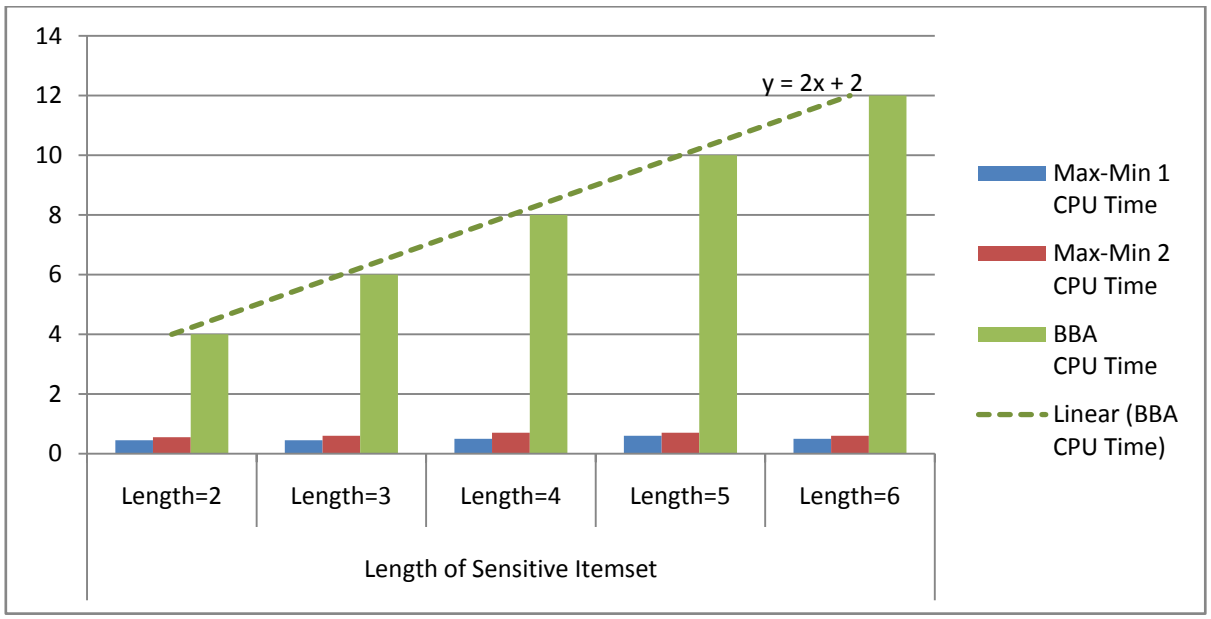
(a) κατηγορία πειραμάτων



Σχήμα 5.1: Η μετρική Changes in Raw Data σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (a). Η μετρική δεν αποδίδεται για κάθε αλγόριθμο χωριστά, καθώς οι διαφορές ανάμεσά τους ήταν της τάξης +/- 1.

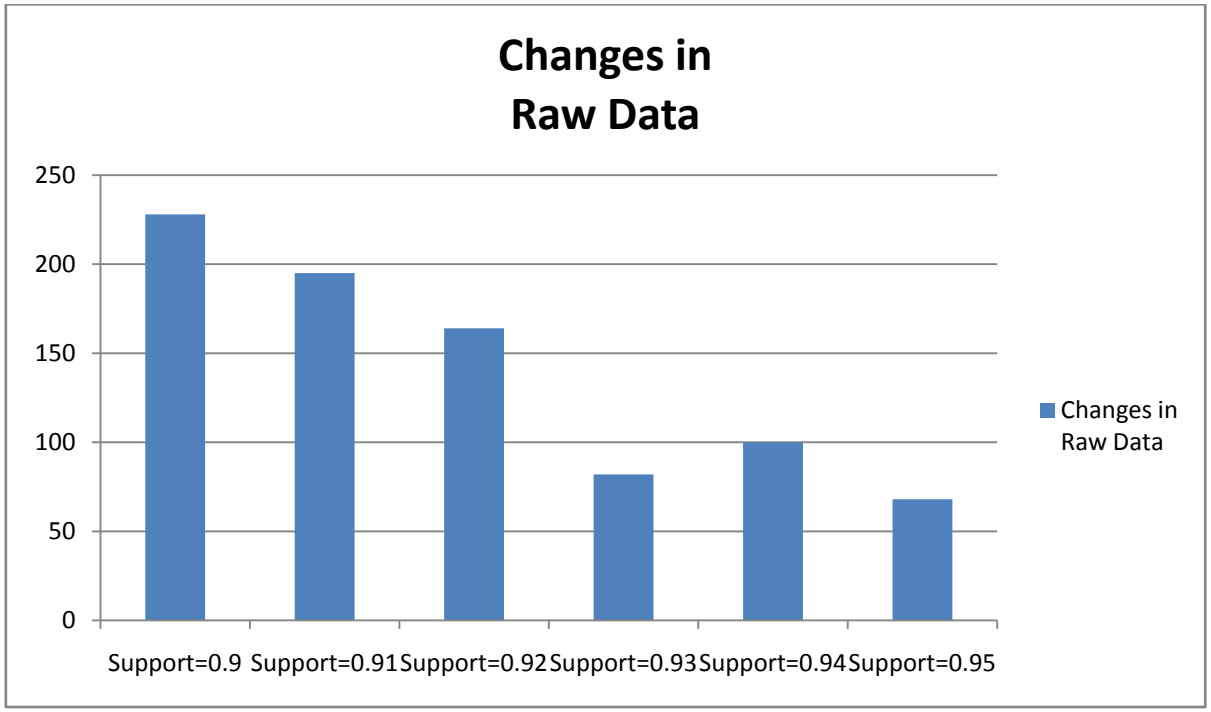


Σχήμα 5.2: Η μετρική Side Effects ανά αλγόριθμο σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (a) με την Β.Δ. Chess.

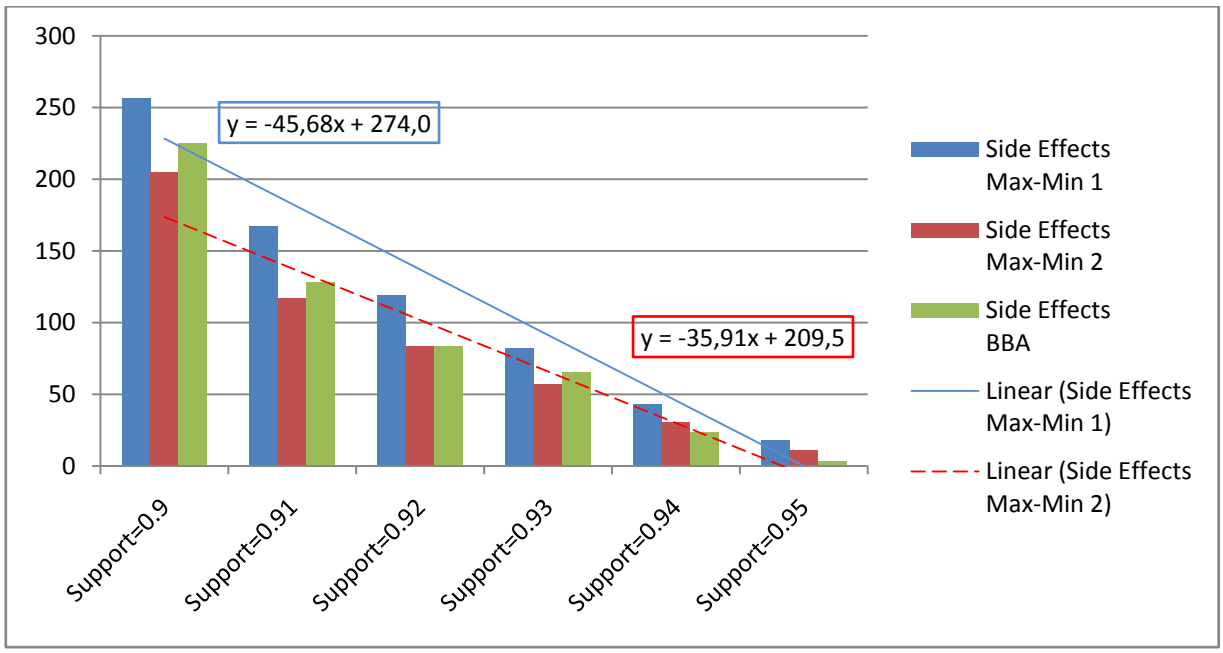


Σχήμα 5.3: Η μετρική CPU Time ανά αλγόριθμο σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (a) με την Β.Δ. Chess. Ο χρόνος εκτέλεσης του BBA αυξάνεται γραμμικά με το μήκος του ευαίσθητου Σ.Σ., ενώ για τους άλλους δύο αλγορίθμους είναι σχεδόν σταθερός.

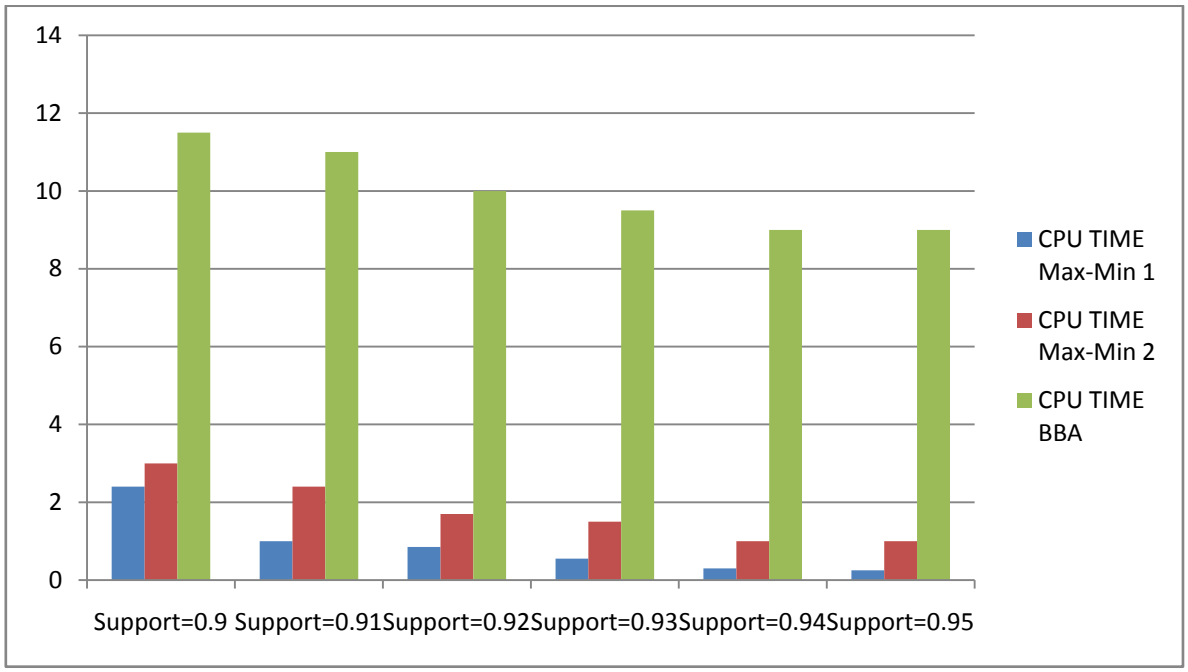
(β) κατηγορία πειραμάτων



Σχήμα 5.4: Η μετρική Changes in Raw Data σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (b) με την Β.Δ. Chess. Η μετρική δεν αποδίδεται για κάθε αλγόριθμο χωριστά, καθώς οι διαφορές ανάμεσά τους ήταν της τάξης +/- 1



Σχήμα 5.5: Η μετρική Side Effects ανά αλγόριθμο σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (b) με την Β.Δ. Chess. Η αποκλιμάκωση της Side Effects συναρήσει της υποστήριξης είναι ισχυρότερη για τον Max-Min 1.



Σχήμα 5.6: Η μετρική CPU Time ανά αλγόριθμο σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (b) με την Β.Δ. Chess.

5.3.2 Πειράματα με την Mushroom

Η δεύτερη Β.Δ. Mushroom αποτελείται από 8124 συναλλαγές, είκοσι τριών περίπου στοιχείων η κάθε μία, ενώ το σύνολο των στοιχείων που απαρτίζουν το – 1 Itemset είναι 119. Η Mushroom αποδίδει 565 συχνά Σ.Σ. για τιμή υποστήριξης *msup*=0.4, για τιμή υποστήριξης *msup*=0.3 το σύνολο των Σ.Σ. ανέρχεται σε 2.736, ενώ για μικρότερες τιμές υποστήριξης το σύνολο των Σ.Σ. ανέρχεται σε πολλές χιλιάδες, πράγμα που καθιστά την εκτέλεση πειραμάτων σε χαμηλότερες τιμές υποστήριξης χρονοβόρα. Για τον λόγο αυτό η Β.Δ. Mushroom χρησιμοποιήθηκε για κατώφλι υποστήριξης μεγαλύτερο ή το πολύ ίσο με 0.4. Έτσι, και η Β.Δ. Mushroom, σε αντιστοιχία με τα πειράματα που διενεργήθηκαν με την Chess, χρησιμοποιήθηκε για δύο κατηγορίες δοκιμών:

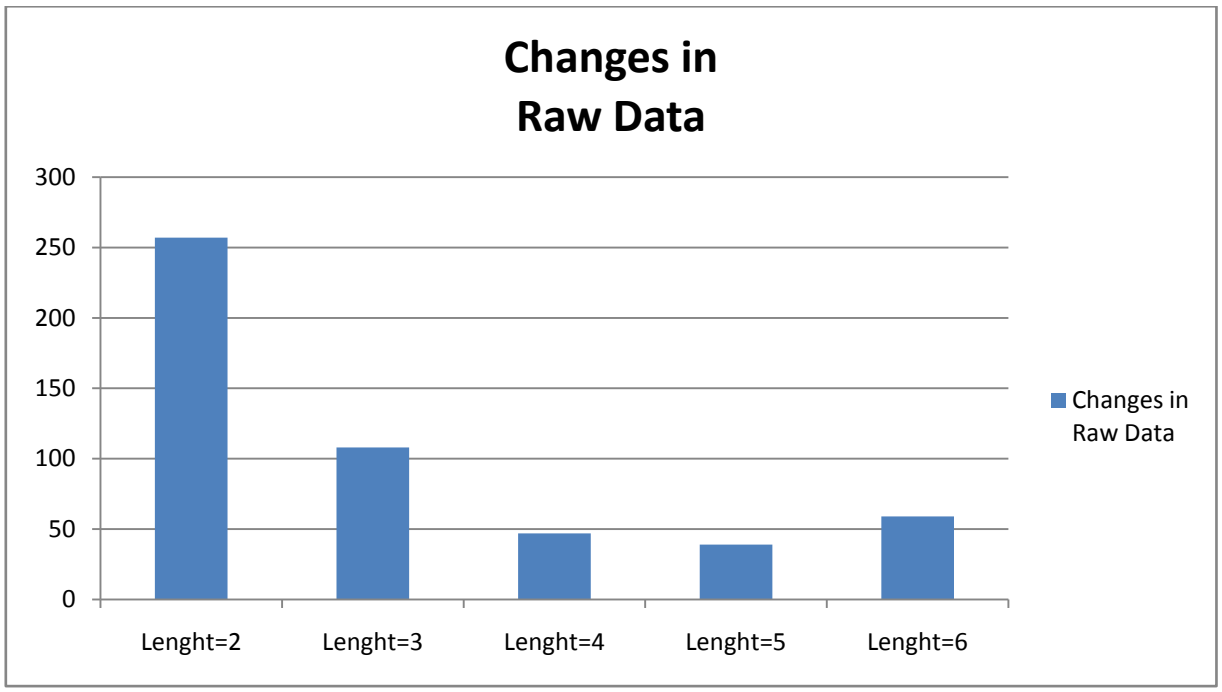
a) Στην πρώτη κατηγορία δοκιμών χρησιμοποιήθηκε σταθερή τιμή στο κατώφλι υποστήριξης *msup*=0.4 και μεταβαλλόμενο μήκος ευαίσθητων Σ.Σ. με τιμές από Length=2 έως Length=6. Στην περίπτωση αυτή ισχύουν όσα έχουν προαναφερθεί για την αντίστοιχη κατηγορία πειραμάτων που διενεργήθηκαν με την Β.Δ. Chess. Στα σχήματα 5.7 έως 5.9 δίδονται τα αποτελέσματα των πειραμάτων αυτής της κατηγορίας..

b) Στην δεύτερη κατηγορία πειραμάτων κρατήσαμε σταθερό μήκος ευαίσθητου Σ.Σ. και μεταβαλλόμενη τιμή υποστήριξης από 0,4 έως 0.45 με βήμα αύξησης 0.01. Το μήκος του ευαίσθητου Σ.Σ. επιλέχθηκε να έχει την τιμή 5. Και στην περίπτωση αυτή ισχύουν όσα έχουν προαναφερθεί για την αντίστοιχη κατηγορία πειραμάτων που διενεργήθηκαν με την Β.Δ. Chess. Στα σχήματα 5.10 έως 5.12 δίδονται τα αντίστοιχα αποτελέσματα.

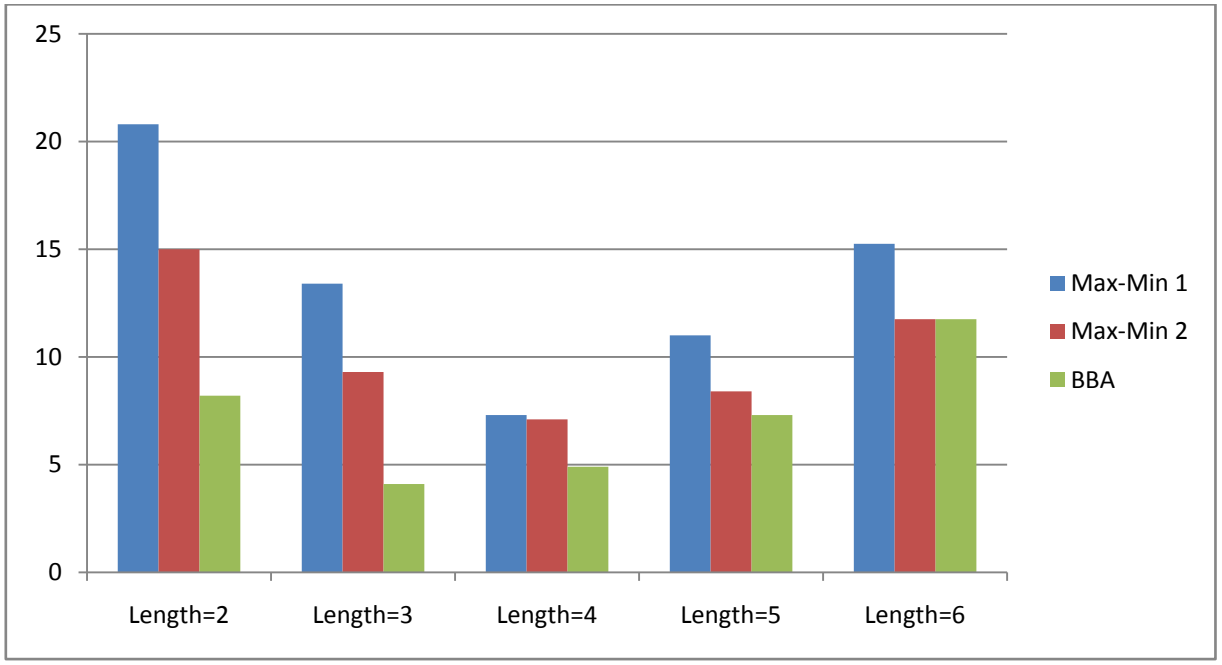
Τέλος, σημειώνουμε ότι, όπως και με την Chess, και στις δύο κατηγορίες πειραμάτων, η επιλογή των ευαίσθητων Σ.Σ. έγινε με βάση την τιμή της υποστήριξης τους, δηλαδή επιλέχθηκαν Σ.Σ. τα οποία να βρίσκονται σχετικά κοντά στο κατώφλι υποστήριξης, ώστε να αποφευχθεί τεράστιος αριθμός επαναλήψεων των αλγορίθμων απόκρυψης. Τα αποτελέσματα παρουσιάζονται παρακάτω, με την ίδια σειρά που παρουσιάστηκαν με την Β.Δ. Chess.

Αποτελέσματα - Διαγράμματα πειραμάτων με την Mushroom.

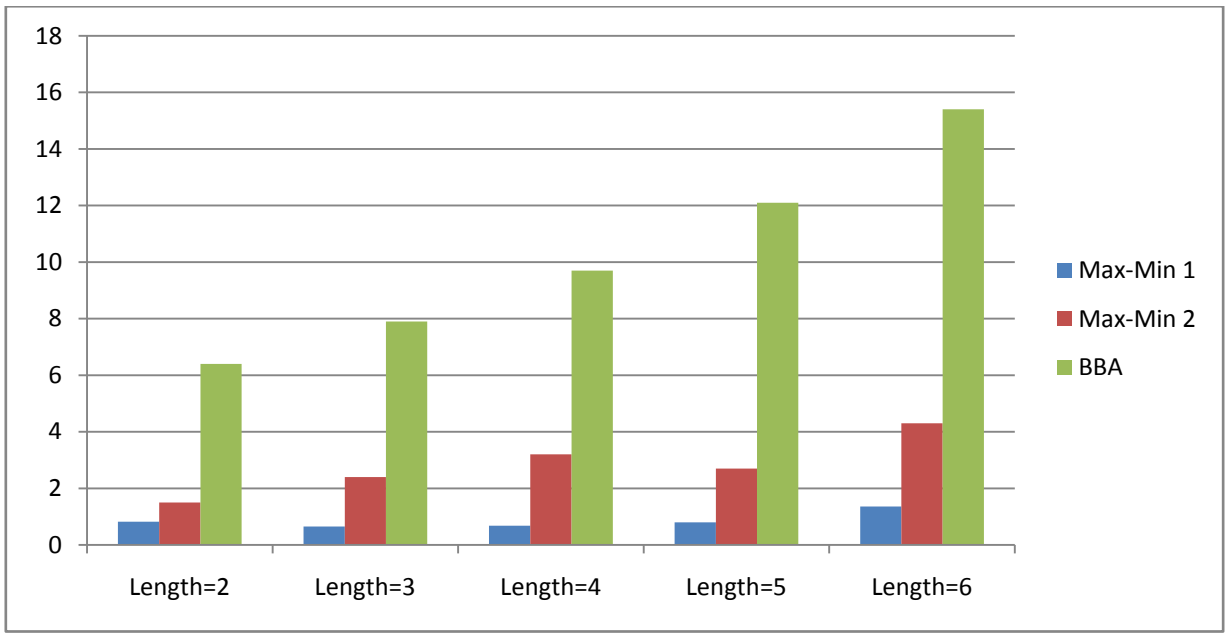
(α) κατηγορία πειραμάτων



Σχήμα 5.7: Η μετρική Changes in Raw Data σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (a) με την Β.Δ. Mushroom.

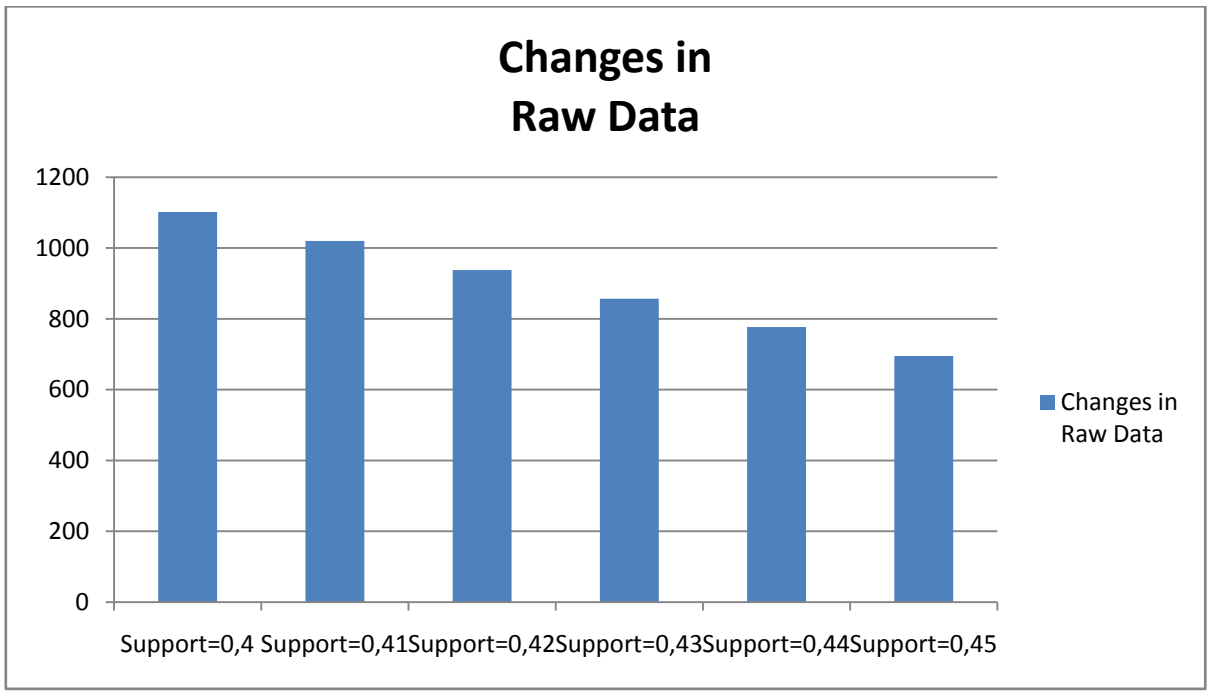


Σχήμα 5.8: Η μετρική Side Effects ανά αλγόριθμο σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (a) με την Β.Δ. Mushroom.

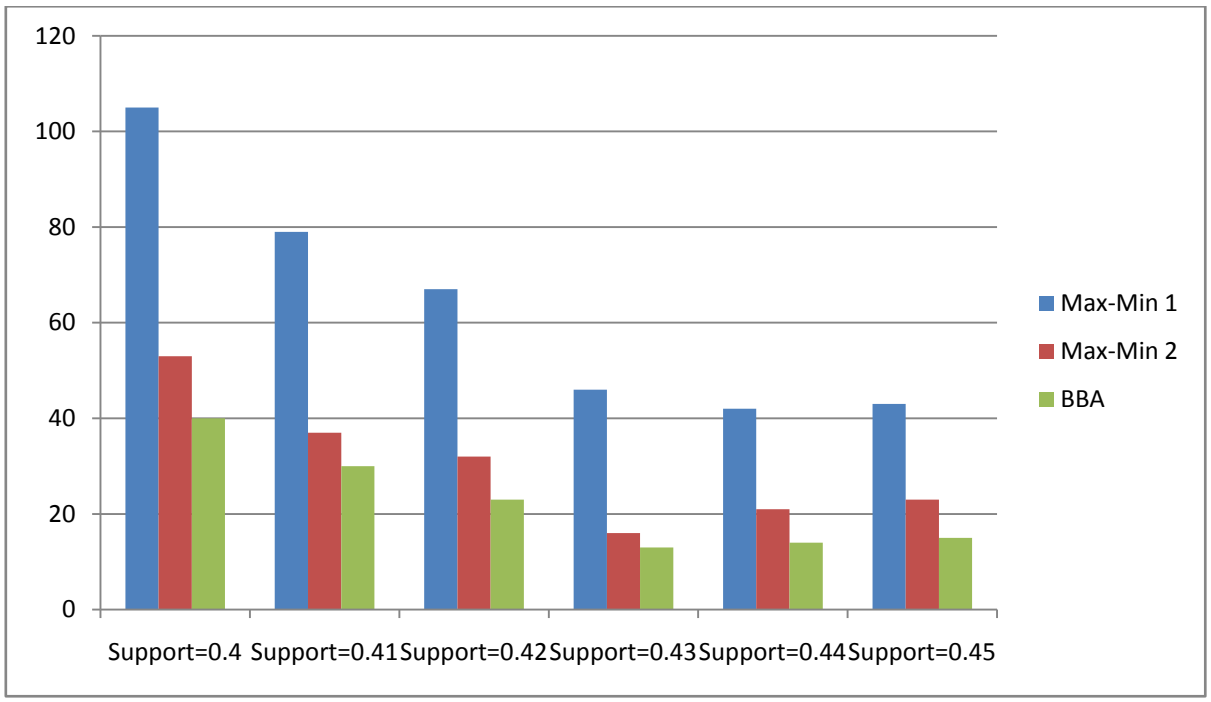


Σχήμα 5.9: Η μετρική CPU Time ανά αλγόριθμο σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (a) με την Β.Δ. Mushroom.

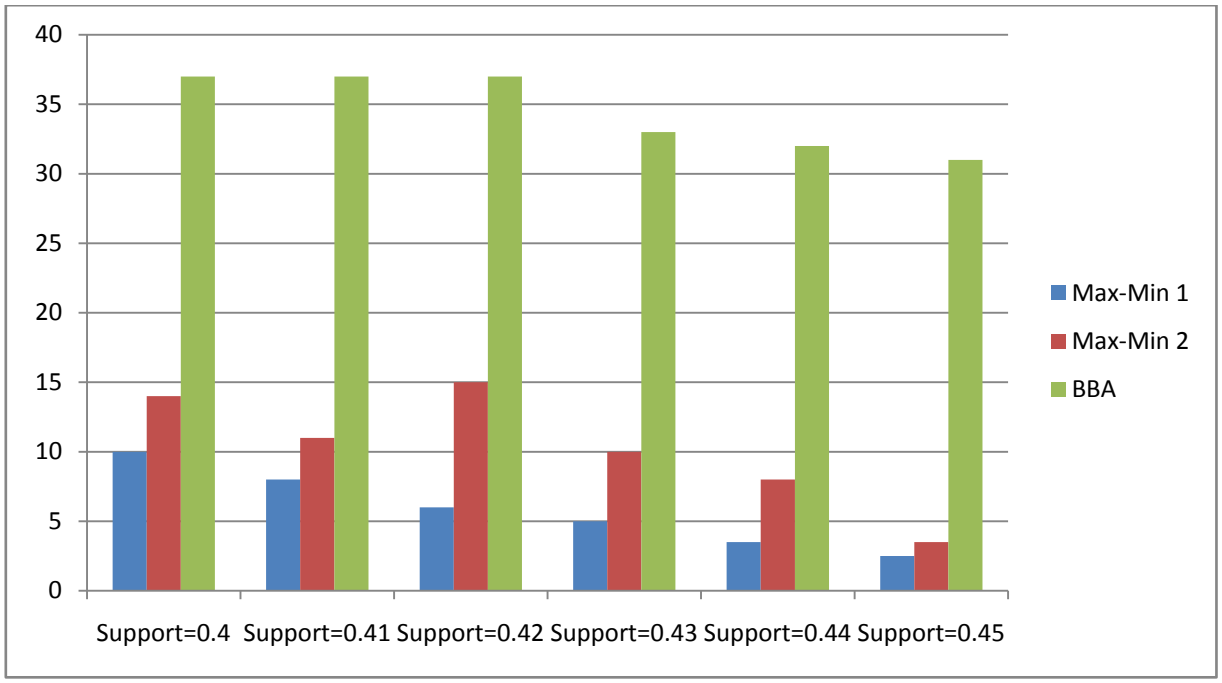
(β) κατηγορία πειραμάτων



Σχήμα 5.10: Η μετρική Changes in Raw Data σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (b) με την Β.Δ. Mushroom.



Σχήμα 5.11: Η μετρική Side Effects ανά αλγόριθμο σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (b) με την Β.Δ. Mushroom.



Σχήμα 5.12: Η μετρική CPU Time ανά αλγόριθμο σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (b) με την Β.Δ. Mushroom.

5.3.3 Πειράματα με την Retail

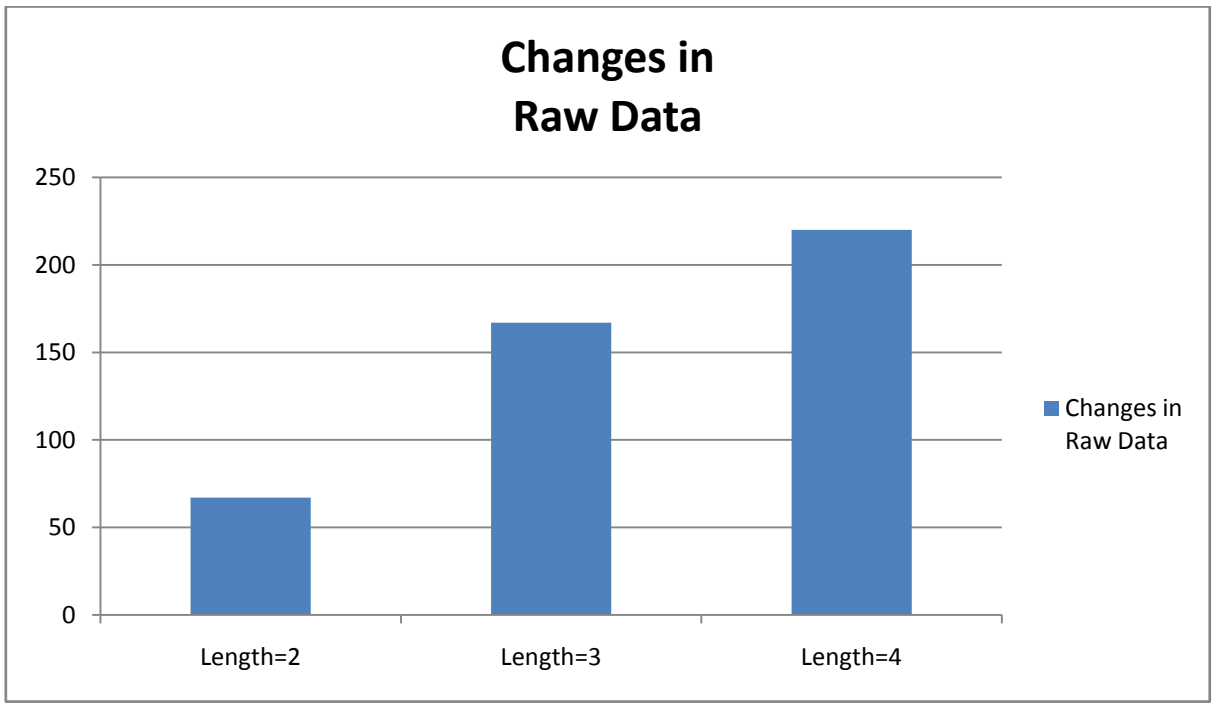
Η τρίτη και τελευταία Β.Δ. με την οποία διενεργήσαμε πειράματα είναι η Retail. Πρόκειται για μια βάση δεδομένων ενός ανώνυμου super market λιανικής σε κάποια πόλη του Βελγίου που αποτελείται από πραγματικά δεδομένα συναλλαγών που συλλέχθηκαν σε χρονικό διάστημα πέντε μηνών. Δημιουργός της Β.Δ. είναι ο Tom Brijs, Ph.D., ''Research Group Data Analysis and Modeling, Limburgs Universitair Centrum''. Αυτή η Β.Δ. αποτελείται από 88.162 συναλλαγές. Η κάθε συναλλαγή περιλαμβάνει από τουλάχιστον ένα item, ενώ μπορεί να περιέχει και πολύ περισσότερα, όπως ακριβώς συμβαίνει με το γνωστό μοντέλο του καλαθιού super market. Το σύνολο των items που απαρτίζουν την Β.Δ. είναι 16.470 στοιχεία. Κάνοντας εξόρυξη συχνών Σ.Σ. στην συγκεκριμένη Β.Δ. λάβαμε μόλις τρία item set για msup=0.3, εννέα συχνά Σ. Σ εξορύχθηκαν για msup=0.1, ενώ τέλος για msup=0,01 παράχθηκαν 158 συχνά Σ.Σ. Τέλος σημειώνουμε ότι ο μέσος χρόνος εκτέλεσης μόνο του αλγορίθμου εξόρυξης ξεπερνούσε τα δέκα λεπτά της ώρας. Οι δύο κατηγορίες πειραμάτων που έγιναν με την Retail περιγράφονται παρακάτω:

a) Στην πρώτη κατηγορία δοκιμών χρησιμοποιήθηκε σταθερή τιμή στο κατώφλι υποστήριξης *msup*=0.01 και μεταβαλλόμενο μήκος ευαίσθητων Σ.Σ. με τιμές από Length=2 έως Length=4. Το σύνολο των ευαίσθητων Σ.Σ. το αποτελούσαν πέντε Σ..Σ. μήκους δύο, τέσσερα Σ.Σ. μήκους τρία και τρία Σ.Σ. μήκους τέσσερα. Όπως γίνεται αντιληπτό και από το διάγραμμα της 'Changes in Raw Data' συναρτήσει του μήκους του ευαίσθητου Σ.Σ., τα sensitive Itemsets επιλέχθηκαν να έχουν τιμή υποστήριξης κοντά στην τιμή του κατωφλίου. Το αποτέλεσμα της απόκρυψης έδωσε μηδενικά 'side effects' κάτι το οποίο ερμηνεύεται εύκολα λόγω των ελάχιστων (σχετικά με το μέγεθος της Β.Δ.) συχνών Σ.Σ. Στα σχήματα 5.13 έως 5.15 δίδονται τα αποτελέσματα των πειραμάτων αυτής της κατηγορίας.

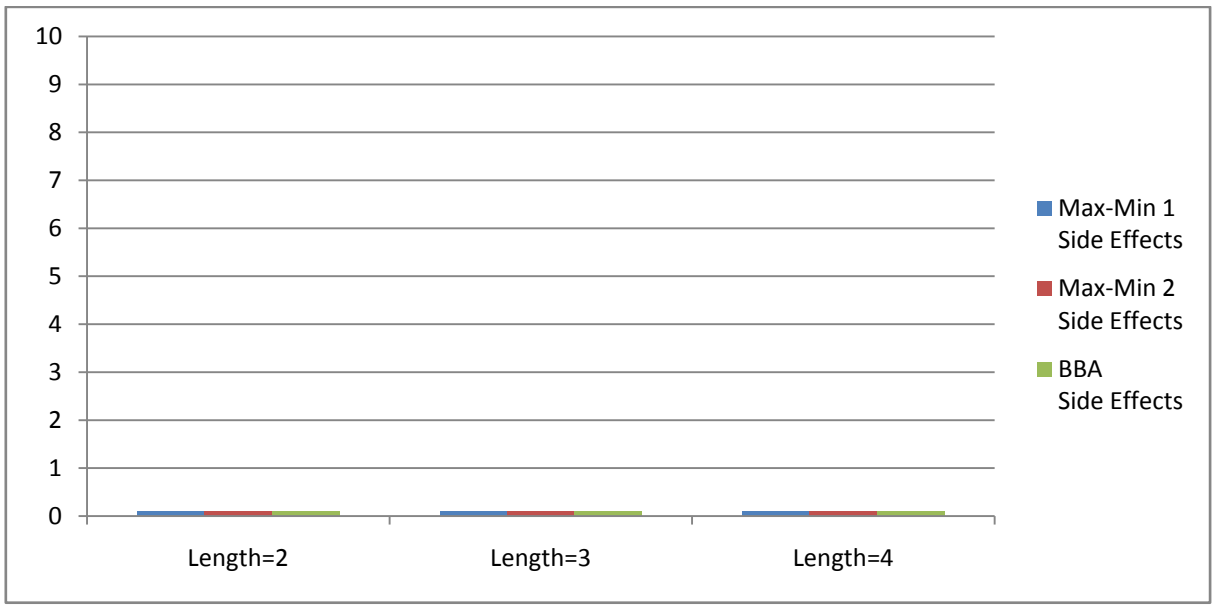
b) Στην δεύτερη κατηγορία πειραμάτων κρατήσαμε σταθερό το μήκος του ευαίσθητου Σ.Σ. και μεταβαλλόμενη τιμή υποστήριξης από 0,1 έως 0.3 με βήμα αύξησης 0.1. Το μήκος του ευαίσθητου Σ.Σ. επιλέχθηκε να έχει την τιμή 2. Στην περίπτωση αυτή, ως ευαίσθητο Σ.Σ. επιλέχθηκε ένα από τα πιο συχνά Σ..Σ. (συγκεκριμένα το '39', '48' με τιμή υποστήριξης 0,33) πράγμα το οποίο οδήγησε σε πολύ μεγάλο αριθμό επαναλήψεων τους αλγορίθμους απόκρυψης, κάτι το οποίο αντικατοπτρίζεται και στο διάγραμμα της 'Changes in Raw Data' συναρτήσει της υποστήριξης. Στην περίπτωση αυτή το αποτέλεσμα του πειράματος ήταν ένας πολύ μικρός αριθμός 'side effects' κάτι το οποίο ερμηνεύεται, όπως και παραπάνω λόγω των ελάχιστων (σχετικά με το μέγεθος της Β.Δ.) συχνών Σ.Σ. Στα σχήματα 5.16 έως 5.18 δίδονται τα αντίστοιχα αποτελέσματα.

Αποτελέσματα - Διαγράμματα με την Retail

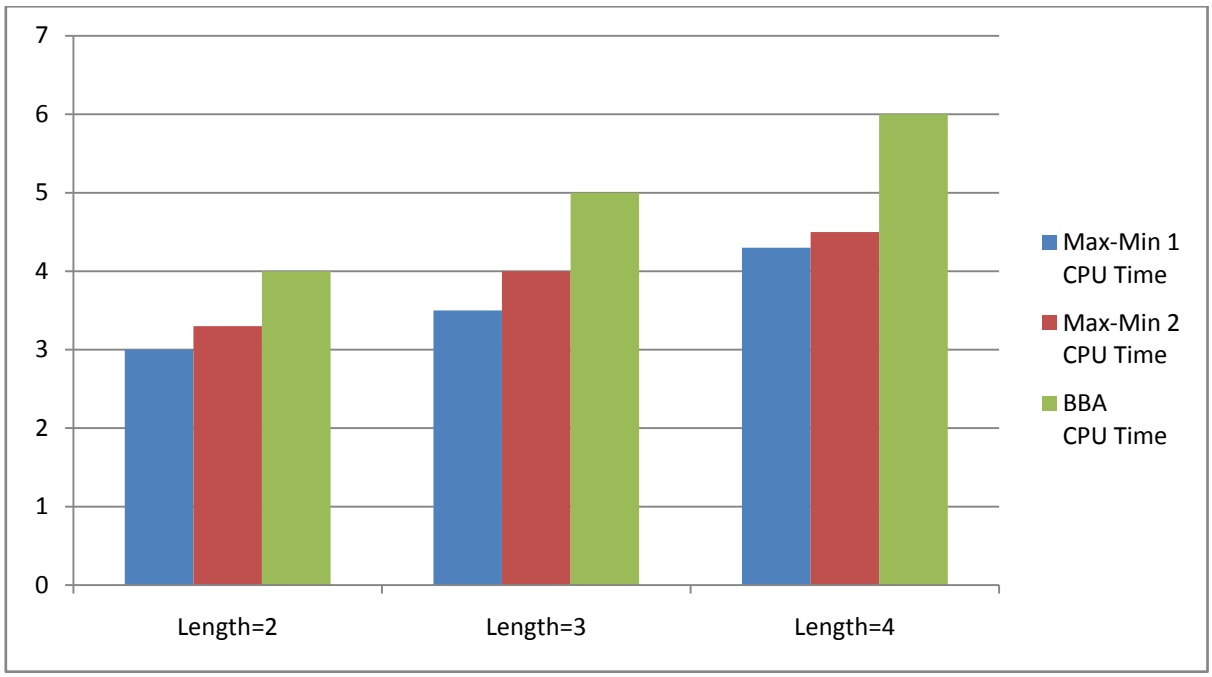
(a) κατηγορία πειραμάτων



Σχήμα 5.13: Η μετρική Changes in Raw Data σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (a) με την Β.Δ. Retail.

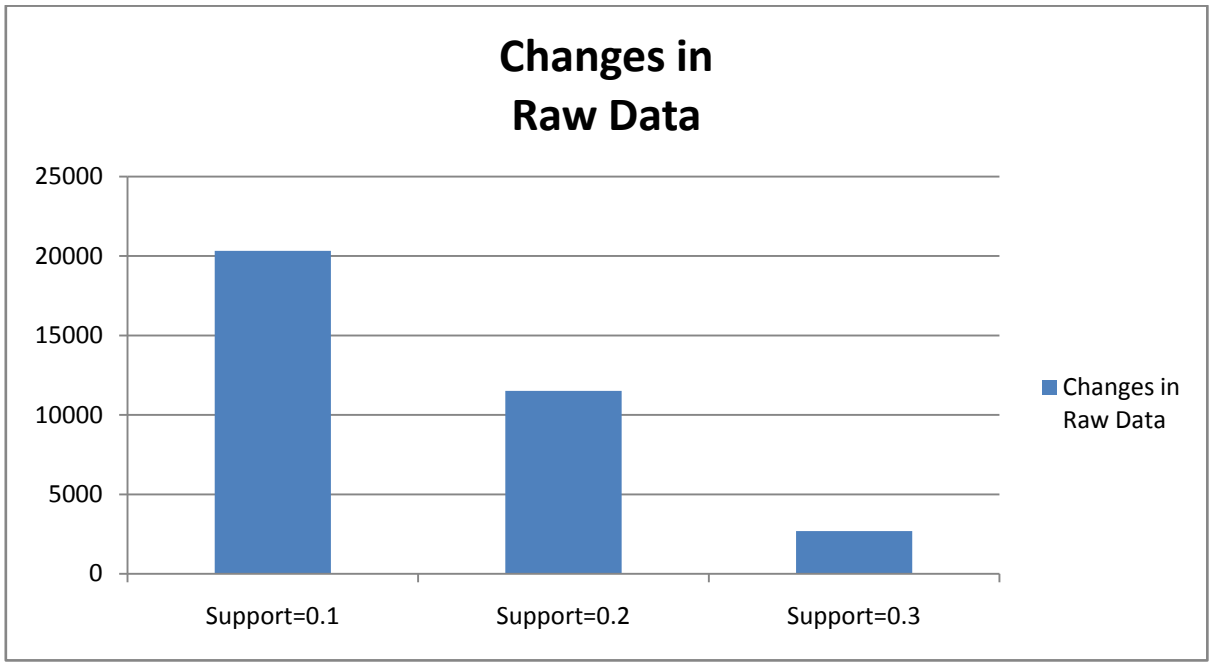


Σχήμα 5.14: Η μετρική Side Effects ανά αλγόριθμο σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (a) με την Β.Δ. Retail. Το αποτέλεσμα είναι μηδενικές τιμές Side Effects λόγω των ελάχιστων (σχετικά με το μέγεθος της Β.Δ.) συχνών Σ.Σ.

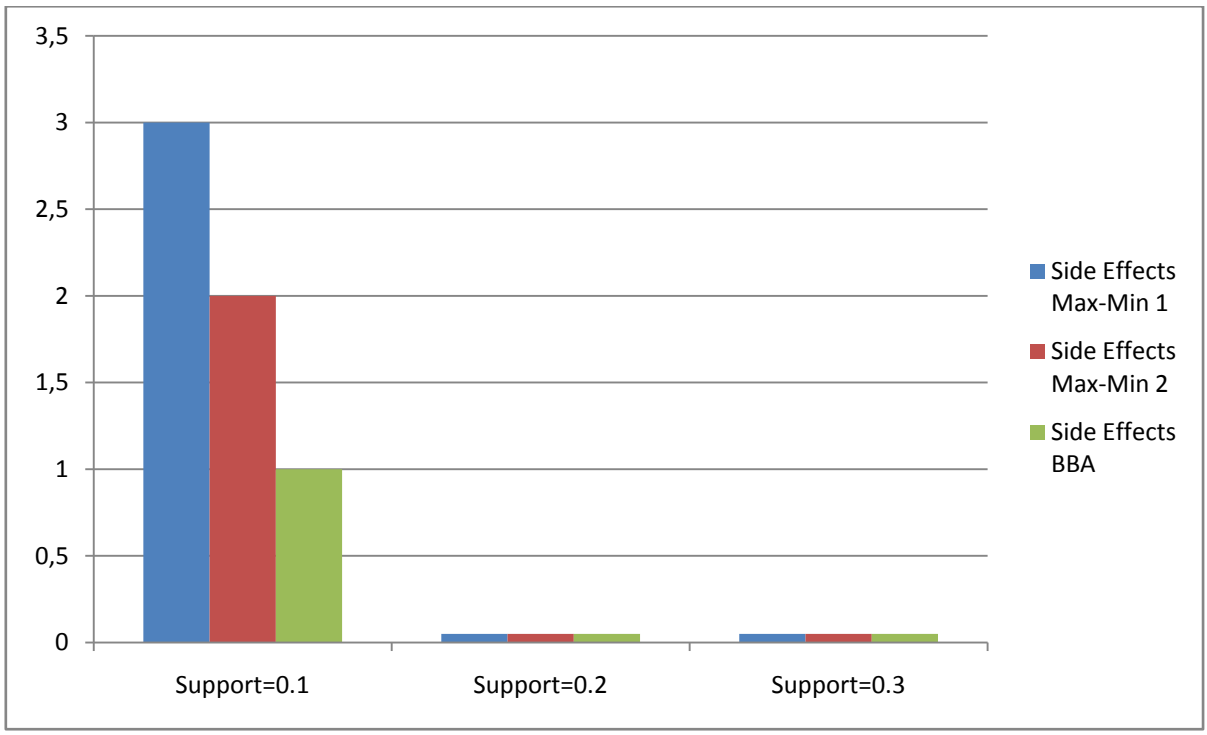


Σχήμα 5.12: Η μετρική CPU Time ανά αλγόριθμο σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (a) με την Β.Δ. Retail

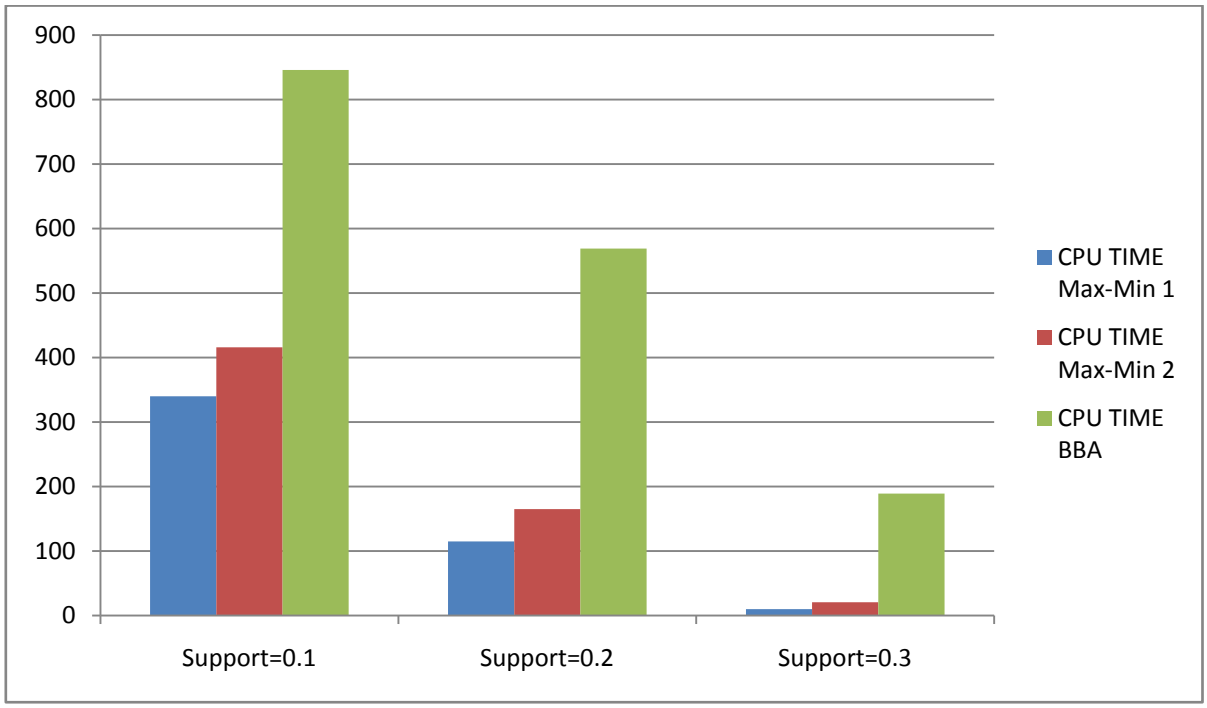
(β) κατηγορία πειραμάτων



Σχήμα 5.13: Η μετρική Changes in Raw Data σαν συνάρτηση του μήκους του ευαίσθητου Σ.Σ. για την κατηγορία των πειραμάτων (b) με την Β.Δ. Retail. Σε αυτή τη περίπτωση αριθμός των αλλαγών στην Β.Δ. είναι πολύ μεγάλος.



Σχήμα 5.14: Η μετρική Side Effects σαν συνάρτηση της υποστήριξης για την κατηγορία των πειραμάτων (b) με την Β.Δ. Retail. Αν και ο αριθμός των αλλαγών στην Β.Δ. είναι πολύ μεγάλος (όπως φαίνεται στο προηγούμενο σχήμα), ο αριθμός των Side Effects είναι μηδενικός, κάτι το οποίο ωφείλεται στα χαρακτηριστικά της Retail.



Σχήμα 5.15: Η μετρική CPU Time σαν συνάρτηση της υποστήριξης για την κατηγορία των πειραμάτων (b) με την Β.Δ. Retail.

5.4 Συμπεράσματα

Τα πειράματα που διεξήχθηκαν παραπάνω έδειξαν την δυνατότητα του λογισμικού που αναπτύχθηκε στα πλαίσια της παρούσης Μεταπτυχιακής διατριβής, να λειτουργήσει και σε μεγάλες Β.Δ. εξάγοντας χρήσιμα συμπεράσματα. Πράγματι, ο χρήστης του εργαλείου μπορεί να πειραματισθεί με οποιαδήποτε Β.Δ. επιθυμεί, αρκεί αυτή να ακολουθεί την κατάλληλη δομή, όπως περιγράφηκε στο κεφάλαιο 4, να σχεδιάσει και να εκτελέσει διάφορα πειράματα εισάγοντας πολλαπλά ευαίσθητα Σ.Σ. υπό μορφή αρχείου, διαφορετικού μήκους το καθένα, για διάφορες τιμές υποστήριξης και να μελετήσει την συμπεριφορά των αλγορίθμων μέσω της μεταβολής των δεδομένων, λαμβάνοντας τα αποτελέσματα που παράγει το εργαλείο σε μορφή καμπυλών ή απλού κειμένου.

5.4.1 Αποτελέσματα Σύγκρισης των Αλγορίθμων

Όπως αναφέρθηκε και παραπάνω, στόχος του παρόντος κεφαλαίου δεν ήταν κυρίως η συγκριτική απόδοση των ενσωματωμένων στο εργαλείο αλγορίθμων. Άλλωστε ένα από τα συμπεράσματα που προκύπτουν μετά την εκτέλεση πολλών πειραμάτων είναι ότι ο ίδιος ο τρόπος σχεδίασης των πειραμάτων μπορεί να αποτελεί σημαντικό παράγοντα που επηρεάζει την απόδοση των αλγορίθμων. Στα πειράματα που διεξαγάγαμε, με τον τρόπο που περιγράφηκε αναλυτικά παραπάνω, ο αλγόριθμος BBA απέδωσε καλύτερα αποτελέσματα αναφορικά με την μετρική Side Effects, είχε όμως την χειρότερη απόδοση αναφορικά με τον χρόνο εκτέλεσης του, κάνοντας πολλές φορές προβληματική την χρήση του λόγω του μεγάλου χρόνου αναμονής των αποτελεσμάτων. Ο Max-Min 1 είναι ο πιο γρήγορος, με υψηλότερο όμως βαθμό παρενεργειών, ενώ ο Max-Min 2 φαίνεται να είναι μια βέλτιστη λύση για περιορισμένο αριθμό Side Effects με χαμηλό, σχετικά χρόνο εκτέλεσης. Τέλος, αναφορικά με την μετρική Changes in Raw Data και οι τρείς αλγόριθμοι έδωσαν σχεδόν τα ίδια αποτελέσματα.

5.4.2 Σχετικές Εργασίες

Για την σύγκριση του παρόντος Εργαλείου με εργασίες σχετικές με σύγκριση μεθόδων απόκρυψης Κ.Σ. ή ανάπτυξη εφαρμογών Α.Κ.Σ. ή συχνών Σ. Σ που τυχόν έχουν υλοποιηθεί από την επιστημονική κοινότητα, ανατρέξαμε στο διαδίκτυο προς αναζήτηση εφαρμογών.

Ενδιαφέρουσα εργασία αποτελεί η μεταπτυχιακή διατριβή του κ. Baris Yildiz [17], με τίτλο "Impacts Of Frequent Itemset Hiding Algorithms On Privacy Preserving Data Mining", Ινστιτούτο Τεχνολογίας Σμύρνης. Στην εργασία του ο κ. Baris Yildiz συγκρίνει δύο αλγορίθμους εξόρυξης, τους FP-Growth και Matrix Apriori. Ο λόγος που αναφέρεται σε αυτούς τους συγκρίνει είναι ότι

κανείς από τους δύο δεν βασίζεται στην δημιουργία υποψήφιων Σ.Σ. όπως ο κλασσικός Apriori και συνεπώς είναι ταχύτεροι καθώς αποφεύγουν το πολλαπλό scan της Β.Δ. Σύμφωνα με τον συγγραφέα, η σύγκριση ανάμεσα στους δύο (FP-Growth και Matrix Apriori) αναδεικνύει νικητή τον Matrix Apriori. Στην συνέχεια ο συγγραφέας εξετάζει τέσσερις αλγορίθμους απόκρυψης (spmaxFI, spminFI, lpmaxFI, lpminFI) οι οποίοι βασίζονται στον Matrix Apriori (είναι τροποποιημένοι Matrix Apriori ώστε να κάνουν και data hiding και είναι παρόμοιοι μεταξύ τους) και το πλεονέκτημά τους είναι ότι δεν βασίζονται στην λογική του pre-mining, δηλαδή ότι θα πρέπει πρώτα να γίνει εξόρυξη των Σ.Σ. και υπολογισμός των αντίστοιχων τιμών υποστήριξης. Με αυτόν τον τρόπο επιτυγχάνεται μεγαλύτερη απόδοση των αλγορίθμων απόκρυψης κ.τ.λ. Τα πειράματα γίνονται με συνθετικές βάσεις δεδομένων που δημιουργήθηκαν με το RTool dataset generator SOFTWARE (Cristofor 2002) [05]. Στο τελευταίο παράρτημα, ο συγγραφέας παρουσιάζει τρία screenshots, ένα για κάθε μία εφαρμογή που έχει αναπτύξει. Η κάθε μια εφαρμογή υλοποιεί έναν αλγόριθμο και περιλαμβάνει ένα απλό U.I. με το οποίο γίνεται η εισαγωγή του αρχείου που περιέχει τη Β.Δ., το support entry form και δύο text monitors για την προβολή των αποτελεσμάτων. Με την πρώτη εφαρμογή τρέχει τον αλγόριθμο εξόρυξης FP-Growth, με την δεύτερη τον Matrix Apriori και με τον τρίτο τον spmaxFI. Δεν αναφέρονται λεπτομέρειες για τον τρόπο που έχει δημιουργηθεί το Software παρά μόνο ότι έγινε με χρήση του Lazarus IDE (http://www.lazarus.freepascal.org/) .

Συμπερασματικά μπορούμε να πο ύμε ότι εργασία του κ. Baris Yildiz είναι ενδιαφέρουσα διότι ασχολείται με αλγορίθμους που κάνουν απόκρυψη χωρίς pre-mining. Ωστόσο, δεν δημιουργεί ένα ολοκληρωμένο περιβάλλον μέσα από το οποίο να μπορεί κανείς να επιλέξει και να τρέξει τους διάφορους αλγορίθμους απόκρυψης ή εξόρυξης. Αντίθετα, κάθε αλγόριθμος είναι και μία απλή stand-alone εφαρμογή, με το δικό της U. I., η οποία εμφανίζει υπό μορφή text τα αποτελέσματα, δεν έχει δυνατότητα σύγκρισης αλγορίθμων και δεν έχει δυνατότητα απεικόνισης των αποτελεσμάτων υπό την μορφή γράφων.

5.4.3 Μελλοντικές Επεκτάσεις

Αν και το Εργαλείο πραγματοποιεί ακριβώς αυτό που υπόσχεται, είναι προφανές ότι υπό την παρούσα μορφή έχει τους δικούς του περιορισμούς και αδυναμίες. Παρακάτω, δίνουμε μια σειρά από δυνατότητες λειτουργικών επεκτάσεων του Εργαλείου, οι οποίες μπορούν να πραγματοποιηθούν χωρίς να απαιτείται κάποια δομική αλλαγή της αρχιτεκτονικής του.

1. Ενσωμάτωση επιπλέον αλγορίθμων στο Εργαλείο.

2. Επέκταση των μετρικών αξιολόγησης των αλγορίθμων με βάση όσα αναφέρθηκαν στην παράγραφο 5.1.

3. Προσθήκη ενός εργαλείου δημιουργίας συνθετικής Β.Δ. βάση των χαρακτηριστικών που επιθυμεί ο χρήστης (σε παρόμοιο εργαλείο αναφερθήκαμε στην προηγούμενη παράγραφο).

4. Στατιστική επεξεργασία της Β.Δ. που εισάγει ο χρήστης, με εύρεση των στοιχείων που απαρτίζουν τις συναλλαγές (- 1 itemset), εύρεση μέσων τιμών, τυπικών αποκλίσεων κ.τ.λ.

5. Στατιστική επεξεργασία των αποτελεσμάτων σύγκρισης των αλγορίθμων.

6. Επιλογή του τρόπου της γραφικής απεικόνισης των αποτελεσμάτων σύγκρισης (ράβδοι, γραμμές κ. τ. λ.)

7. Το Εργαλείο και όλο το λογισμικό που αναπτύχθηκε, υλοποιήθηκε σε Python v3.3 και μπορεί να λειτουργήσει μόνο σε αυτό το περιβάλλον. Μια χρήσιμη, πιστεύουμε επέκταση θα ήταν η δημιουργία μιας έκδοσης ικανής να λειτουργήσει και σε Python v2.7 ή ακόμη και σε προγενέστερη.

Η προσθήκη των παραπάνω επεκτάσεων πιστεύουμε ότι θα καταστήσει το Εργαλείο χρήσιμο βοήθημα του μελετητή του επιστημονικού πεδίου της Απόκρυψης Κ. Σ/ συχνών Σ.Σ.

Κεφάλαιο 6
Επίλογος

Στην παρούσα μεταπτυχιακή διατριβή, ασχοληθήκαμε με το ζήτημα της διατήρησης της ιδιωτικότητας κατά την εξόρυξη δεδομένων. Διερευνήσαμε τις πτυχές του προβλήματος της Απόκρυψης Κανόνων Συσχέτισης και ειδικότερα ασχοληθήκαμε με την παραλλαγή της απόκρυψης συχνών Σ.Σ.. Είδαμε τις βασικές έννοιες της Α.Κ.Σ. και κάναμε ειδική μνεία στην Θεωρία Συνόρων, στην οποία έχουν στηριχθεί και οι τρεις αλγόριθμοι που έχουν μελετηθεί και υλοποιηθεί στην παρούσα μεταπτυχιακή διατριβή. Στη συνέχεια, παρουσιάσαμε αναλυτικά την αρχιτεκτονική και τη λειτουργικότητα του Εργαλείου, ενός ολοκληρωμένου περιβάλλοντος πειραματισμού και αξιολόγησης αλγορίθμων απόκρυψης που δημιουργήσαμε, για το οποίο δώσαμε και λειτουργικά παραδείγματα. Τέλος, υποβάλαμε το Εργαλείο σε μια σειρά από δοκιμασίες με μεγάλες και καλά τεκμηριωμένες Β.Δ., που αντλήσαμε από το διαδίκτυο, συγκεντρώσαμε και παρουσιάσαμε τα αποτελέσματα ανά αλγόριθμο.

Φυσικά, το θέμα της Απόκρυψης Κανόνων Συσχέτισης και ειδικότερα της δημιουργίας λογισμικού γύρω από αυτό το ερευνητικό πεδίο είναι μεγάλο, εμπλουτίζεται συνεχώς με νέες επιστημονικές εργασίες και βεβαίως δεν είναι δυνατόν, να καλυφθεί στα πλαίσια μια Μεταπτυχιακής διατριβής. Για τον λόγο αυτό δώσαμε και μία σειρά από χρήσιμες επεκτάσεις που θα μπορούσαν να υλοποιηθούν στο μέλλον για να καταστήσουν το Εργαλείο περισσότερο λειτουργικό και βιώσιμο σε μεγαλύτερο βάθος χρόνου.

Βιβλιογραφία

[01] R. Agrawal, R. Srikant, «Fast Algorithms for Mining Association Rules in Large Databases». In Proceedings of the 20th International Conference on Very Large Data Bases: 487-499, 1994.

[02] R. Agrawal, R. Srikant, «Privacy-Preserving Data Mining: Models and Algorithms», Proc. of the ACM SIGMOD Conference on Management of Data, Dallas, TX, May 2000.

[03] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, V. Verykios, «Disclosure Limitation of Sensitive Rules» In Proceedings of the 1999 IEEE Workshop on Knowledge and Data Engineering Exchange (KDEX): 45-52, 1999.

[04] E. Bertino, I. Nai Fovino, L. Parasiliti Provenza, «A Framework for Evaluating Privacy Preserving Data Mining Algorithms», Sprinnger, pages 121-154, 2005.

[05] Cristofor, L. 2002. ARtool Project. http://www.cs.umb.edu/~laur/ARtool/ (last access May 13 2010)

[06] E. Dasseni., V. S. Verykios, A. K. Elmagarmid, and E. Bertino, «Hiding Association Rules by Using Confidence and Support», In Proceedings of the 4th International Workshop on Information Hiding, pages 369–383, 2001.

[07] A. Gkoulalas-Divanis and V. S. Verykios, «Association Rule Hiding for Data Mining», Springer, 2010

[08] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery, 1 (3): 241–258, 1997

[09] G. V. Moustakides and V. S. Verykios, «A maxmin approach for hiding frequent itemsets», Data and Knowledge Engineering, 65: 75–89, 2008

[10] S. R. M. Oliveira, and O. R. Zaïane, «Protecting Sensitive Knowledge by Data Sanitization», In Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM), pages 211–218, 2003.

[11] E. Pontikakis, Y. Theodoridis, A. Tsitsonis, L. Chang, and V. S. Verykios, «A Quantitative and Qualitative Analysis of Blocking in Association Rule Hiding», In Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society (WPES), pages 29–30, 2004.

[12] E. D. Pontikakis, A. A. Tsitsonis, and V. S. Verykios, «An Experimental Study of Distortion–based Techniques for Association Rule Hiding», In Proceedings of the 18th Conference on Database Security (DBSEC), pages 325–339, 2004.

[13] X. Sun, P. Yu, «A Border-Based Approach for Hiding Sensitive Frequent Itemsets In Proceedings of the 5th IEEE International Conference on Data Mining: 426-433, 2005.

[14] P. N. Tan, M. Steinbach, and V. Kumar (Επιμέλεια Μετάφρασης: Βασίλειος Σ. Βερύκιος, Μετάφραση: Σταύρος Σουραβλάς), «Εισαγωγή στην Εξόρυξη Δεδομένων», Εκδόσεις Τζιόλα, Θεσσαλονίκη 2010

[15] V. Verykios, A. Elmagarmid, E., Bertino, Y. Saygin, E. Dasseni, E.. «Association Rule Hiding», IEEE Transactions on Knowledge and Data Engineering (16): 434-447, 2004.

[16] Y. Saygin, V. S. Verykios, and C. W. Clifton, «Using Unknowns to Prevent Discovery of Association Rules», ACM SIGMOD Record, 30(4):45–54, 2001.

[17] Y. Saygin, V. S. Verykios, and A. K. Elmagarmid, «Privacy Preserving Association Rule Mining», In Proceedings of the 2002 International Workshop on Research Issues in Data Engineering: Engineering E–Commerce/E–Business Systems (RIDE), pages 151–163, 2002.

[18] B. Yildiz, «Impacts Of Frequent Itemset Hiding Algorithms On Privacy Preserving Data Mining», Izmir Institute of Technology, 2010.

[19] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, «New algorithms for fast discovery of association rules. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining»: 283-286, 2005.

Παράρτημα Α
Οδηγίες εγκατάστασης

Η εφαρμογή έχει υλοποιηθεί εξολοκλήρου σε Python v3.3 και έχει δοκιμασθεί σε αυτήν την έκδοση, σε περιβάλλον Windows. Σε διαφορετικές εκδόσεις Python ενδέχεται να δώσει εσφαλμένα αποτελέσματα ή να προκύψει κάποιας μορφής μήνυμα σφάλματος. Συνεπώς, το πρώτο βήμα που πρέπει να κάνει κάποιος προκειμένου να χρησιμοποιήσει το Εργαλείο είναι να εγκαταστήσει στον υπολογιστή του την Python v3.3, κάτι που μπορεί να γίνει εύκολα μέσω του συνδέσμου http://www.python.org/download/releases/3.3.0/

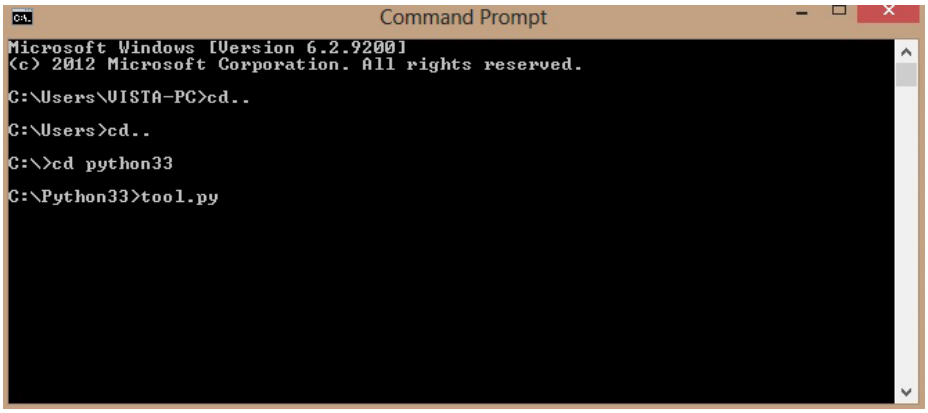
Η εγκατάσταση της Python v3.3 δημιουργεί τον κατάλογο C:\python33. Στη συνέχεια θα πρέπει να δημιουργήσουμε το αντίστοιχο path. Μέσα στον παραπάνω κατάλογο τοποθετούμε τα έξι modules, το αρχείο how_to_use.docx το αρχείο που περιέχει τη Β.Δ. και το αρχείο των προς απόκρυψη ευαίσθητων Σ.Σ.

Στη συνέχεια, οι υποψήφιοι χρήστες του Εργαλείου, θα πρέπει να εγκαταστήσουν το PyWin32, κάτι το οποίο μπορεί να γίνει μέσω του συνδέσμου http://sourceforge.net/projects/pywin32/files/pywin32/Build%20218/. Σε περίπτωση που στον υπολογιστή του χρήστη υπάρχει εγκατεστημένη έκδοση του PyWin32 για

προγενέστερες εκδόσεις της Python, τότε το πιθανότερο είναι να μην υπάρξει κανένα πρόβλημα.

Τέλος, για την ανάγνωση του HELP θα πρέπει να υπάρχει εγκατεστημένο το MS Office 2007.

Η εκκίνηση της εφαρμογής μπορεί να γίνει από το command prompt πληκτρολογώντας Tool.py, όπως δείχνει και η εικόνα Α.1



Εικόνα Α.1: Εκκίνηση της εφαρμογής μέσω του command prompt

Για την αναλυτική παρουσίαση τόσο της λειτουργίας του Εργαλείου, όσο και της δομής των αρχείων Data Set και Sensitive Data Set, με παραδείγματα χρήσης, ο αναγνώστης καλείται να ανατρέξει στο Κεφάλαιο 4.

Παράρτημα Β

Ο Κώδικας σε Python

Στο παράρτημα αυτό παρουσιάζουμε μέρη του κώδικα της εφαρμογής που αναπτύχθηκε. Ωστόσο, καθώς ο κώδικας του λογισμικού που αναπτύχθηκε προσεγγίζει τις δύο χιλιάδες γραμμές (περίπου 100KB), είναι προφανές ότι, για πρακτικούς λόγους, δεν μπορεί να παρατεθεί στο σύνολό του. Έτσι, στο παράρτημα αυτό παρουσιάζουμε τα βασικότερα τμήματά του υπό μορφή συναρτήσεων.

Οι συναρτήσεις για την εύρεση του S_{max}, S_{min}, και Α.Θ.Σ:

```
/* Η συνάρτηση S_max λαμβάνει σαν είσοδο τα σύνολα των συχνών και
ελάχιστων ευαίσθητων Σ.Σ.( Smin) και επιστρέφει το σύνολο Smax.*/

def S_max(f_set,s_min):
    smax=set()
    for x in s_min:
        for y in f_set:
            if x.issubset(y):
                smax.add(y)
    return(smax)

-----------------------------------------------------------------------
```

```
/* Η συνάρτηση Revised_Fd λαμβάνει σαν είσοδο το σύνολο των συχνών
Σ.Σ. και το σύνολο Smax και επιστρέφει το Αναθεωρημένο σύνολο Σ. Σ*/

def Revised_Fd(s_max,f_set):
    Fd_rev=f_set-s_max
    return(Fd_rev)

-----------------------------------------------------------------
/* Η συνάρτηση Revised_pos_border λαμβάνει σαν είσοδο το
Αναθεωρημένο σύνολο Σ.Σ. και επιστρέφει το Αναθεωρημένο θετικό
Σύνορο*/

def Revised_pos_border(rev_fd):
    temp1=[]
    temp2=[]
    positive_border=[]
    for x in rev_fd:
        for y in x:
            if not isinstance(y,float):
                temp1=temp1+[y]
        temp2.append(set(temp1))
        temp1=[]
    for x in temp2:
        positive_border.append(x)
    i=0
    while i<len(temp2):
        j=0
        while j<len(temp2):
            if i!=j and temp2[i].issubset(temp2[j]):
                try:
                    positive_border.remove(temp2[i])
                except:
                    pass
                j=j+1
            else:
                j=j+1
        i=i+1
    temp1=[]
    r_p_b=collections.namedtuple('r_p_b', 'i_set sup')
    for x in positive_border:
        for y in rev_fd:
            if x.issubset(y):
                z=y-x
                for k in z:
                    if isinstance(k,float):
                        w=list(x)
                        w.sort()
                        s=r_p_b(w,k)
                        temp1.append(s)
    p_b_s_list=[]
    p_b_list=[]
    for x in temp1:
        for y in x:
            if isinstance(y,float):
                p_b_s_list.append(y)
            else:
                p_b_list.append(y)
    #print(p_b_s_list)
```

```
    #print(p_b_list)
    return(positive_border,p_b_list,p_b_s_list)

-----------------------------------------------------------------------
/* Η συνάρτηση S_min_sorted λαμβάνει σαν είσοδο τη λίστα των
ευαίσθητων και σύνολο των συχνών Σ.Σ. και επιστρέφει το σύνολο Smin
ταξινομημένο πρώτα κατά το μήκος του Σ.Σ. (σύμφωνα με τις επιταγές
του BBA)*/

def S_min_sorted(all_list,f_col):
    s_set=collections.namedtuple('s_set', 'i_set len_i_set sup')
    all_set=set();temp_1=set();temp_2=set()
    temp_3=set();temp_4=[];S_min_sort=[]
    s_list=[];s_s_list=[]
    i=0;
    while i<len(all_list)-1:
        j=i+1
        while j<len(all_list):
            if (set(all_list[j]).issuperset(set(all_list[i]))):
                temp_1.add(frozenset(all_list[j]))
            elif(set(all_list[i]).issuperset(set(all_list[j]))):
                temp_2.add(frozenset(all_list[i]))
            j=j+1
        i=i+1

    for x in temp_1:
        temp_3.add(x)
    for x in temp_2:
        temp_3.add(x)

    S_min=all_froz-temp_3

    print('S_min=',S_min)
    #print(temp_1)

    for x in f_col:
        for y in S_min:
            Y=list(y)
            Y.sort()
            if Y in x:
                s=s_set(Y,len(y),x.sup)
                temp_4.append(s)
    #print('temp2=',temp_4)
    S_min_sort=sorted(temp_4,key=attrgetter('len_i_set','sup'),
                      reverse=True)

    for x in S_min_sort:
        for y in x:
            if isinstance(y,float):
                s_s_list.append(y)
            elif isinstance(y,int):
                continue
            else:
                s_list.append(y)
    print(S_min_sort)
    return(S_min_sort,s_list,s_s_list)
```

Η Συνάρτηση που υλοποιεί τον Αλγόριθμο Max-Min 1:

```
/* Η συνάρτηση max_min1 λαμβάνει σαν είσοδο τη λίστα των ευαίσθητων
Σ.Σ., το Α.Θ.Σ., τον αύξων αριθμό του Σ.Σ. προς απόκρυψη (k) και το
βήμα επανάληψης του αλγορίθμου για το τρέχον Σ.Σ. Η συνάρτηση
επιστρέφει ένα στοιχείο από τη λίστα των Max-Min items (το πρώτο)*/

def max_min1(sen_list, p_bor_list, sen_sup_list,
p_bor_sup_list,k,n):
    vi=collections.namedtuple('vi', 'tent_v it_set sup')
    v_i_list=[ ]
    #print('The vi-list for the', n+1, 'loop of',sen_list[k],'is:')
    for i,x in enumerate(sen_list):

        #print('v_i_list for', x, 'is:')
        for y in x:
            for j,z in enumerate(p_bor_list):
                if y in z and i==k:
                    v_i=vi(y,z,p_bor_sup_list[j])
                    v_i_list.append(v_i)
    i=0
    while i<len(v_i_list)-1:
        j=i+1
        while j<len(v_i_list):
            if (v_i_list[i].tent_v==v_i_list[j].tent_v):
                if (v_i_list[i].sup < v_i_list[j].sup):
                    v_i_list.remove(v_i_list[j])
                    i=0
                    j=i+1
                    break
                if (v_i_list[i].sup > v_i_list[j].sup):
                    v_i_list.remove(v_i_list[i])
                    i=0
                    j=i+1
            j=j+1
        i=i+1
    min_items=[  ]
    for x in v_i_list:
        min_items.append(x)
    m=0
    for x in min_items:
        c=0
        for y in min_items:
            if x.sup>y.sup:
                min_items.remove(min_items[c])
                c=c+1
            else:
                c=c+1
        m=m+1
    max_min_items=[ ]
    for x in min_items:
        max_min_items.append(x)

    return(max_min_items[0])
```

Η Συνάρτηση που υλοποιεί το Max-Min κριτήριο του Αλγόριθμου Max-Min 2:

```
/* Η συνάρτηση max_min λαμβάνει σαν είσοδο τη λίστα των ευαίσθητων
Σ.Σ., το Α.Θ.Σ., τον αύξων αριθμό του Σ.Σ. προς απόκρυψη (k) και
επιστρέφει το σύνολο των Max-Min items*/

def max_min(sen_list,p_bor_list,sen_sup_list,p_bor_sup_list,k):
    global lines
    vi=collections.namedtuple('vi', 'tent_v it_set sup')
    v_i_list=[]
    #print('The vi-list for',sen_list[k],'is:')
    for i,x in enumerate(sen_list):
        #print('v_i_list for', x, 'is:')
        for y in x:
            for j,z in enumerate(p_bor_list):
                if y in z and i==k:
                    #print(i,y,z,p_bor_sup_list[j])
                    v_i=vi(y,z,p_bor_sup_list[j])
                    v_i_list.append(v_i)
                    #print(y,v_i)

    if v_i_list==[]:
        return(sen_list)
    i=0
    while i<len(v_i_list)-1:
        j=i+1
        while j<len(v_i_list):
            if (v_i_list[i].tent_v==v_i_list[j].tent_v):
                if (v_i_list[i].sup < v_i_list[j].sup):
                    v_i_list.remove(v_i_list[j])
                    i=0
                    j=i+1
                    break
                if (v_i_list[i].sup > v_i_list[j].sup):
                    v_i_list.remove(v_i_list[i])
                    i=0
                    j=i+1
            j=j+1
        i=i+1
    min_items=[]
    for x in v_i_list:
        min_items.append(x)
    print('The min item set is:')
    for x in min_items:
        print(x.tent_v,x.it_set,x.sup)
    #-------------------------------------------------------------
    max_min_items=[]
    max_items=min_items.copy()
    max_min_items.append(max_items[0])
    for x in max_items:
        if x.sup>max_min_items[0].sup:
            for i,y in enumerate(max_min_items):
                max_min_items.remove(max_min_items[i])
            max_min_items.append(x)
        if (x.sup==max_min_items[0].sup):
```

```
            max_min_items.append(x)
    #
    i=0
    while i<len(max_min_items)-1:
        j=i+1
        while j<len(max_min_items):
            if ((max_min_items[i].tent_v == max_min_items[j].tent_v

)and(max_min_items[i].it_set==max_min_items[j].it_set
                    )and(max_min_items[i].sup==
max_min_items[j].sup)):
                max_min_items.remove(max_min_items[j])
                break
            j=j+1
        i=i+1
    #-------------------------------------------------------------
    #print('The Max-Min item set is:')
    #for x in max_min_items:
        #print(x.tent_v,x.it_set,x.sup)
    return(max_min_items)
```

Η Συνάρτηση που υλοποιεί τον Αλγόριθμο Max-Min 2:

/* Η συνάρτηση max_min2 λαμβάνει σαν είσοδο τα Σ.Σ. που απαρτίζουν το Max-Min itemset (m_m_s), το σύνολο ευαίσθητων Σ.Σ. και τον αύξων αριθμό του Σ.Σ. προς απόκρυψη. Η συνάρτηση αναγνωρίζει ποιο από τα τρία σενάρια του αλγορίθμου πρέπει να εφαρμοστεί και επιστρέφει το στοιχείο που πρέπει να διαγραφεί από τη λίστα συναλλαγών και ένα σύνολο που περιέχει αύξοντες αριθμούς συναλλαγών, σε μια από τις οποίες θα γίνει η διαγραφή του στοιχείου. */

```
def max_min_2(m_m_s,Sens_L,N):
    global tid
    #print('m_m_s=',m_m_s)
    tent_vict_item=set()
    v_i_set=set()
    for x in m_m_s:
        #print(x.tent_v)
        tent_vict_item.add(x.tent_v)
        v_i_set.add(tuple(x.it_set))

    Li=set()
    Lu=set()

    ######----FIRST CASE SCENARIO-----######
    if len(tent_vict_item)==1: #and l==False:
        print('first case scenario')
        for m in tent_vict_item:
            Vic_it=m
        #print('tent_vict_item=',tent_vict_item)
        for i,Set in enumerate(tid):
            if set(Sens_L[N]).issubset(Set):
                Li.add(i)
            for item2 in v_i_set:
                if set(item2).issubset(Set):
                    Lu.add(i)

        L=Li-Lu

        if L == set():
            return(Vic_it,Li)
        else:
            return(Vic_it,L)

     ######----SECOND CASE SCENARIO-----######

     else:
        print('second case scenario')
        K=v_i_set.copy()
        for i,Set in enumerate(tid):
            if set(Sens_L[N]).issubset(Set):
                Li.add(i)
        for k in K:
            Lu=set()
            for j,line in enumerate(tid):
                if set(k).issubset(line):
                    Lu.add(j)
```

```
            L=Li-Lu
            #print('Li=',Li)
            #print('Lu=',Lu)
            #print('L=',L)
            if L!=set():
                return(x.tent_v,L)
        Lu1=set()
        Lu2=set()

    #######----SECOND CASE SCENARIO DOES NOT APPLY-----#####

        print('Second Case does not apply')
        for k1 in K:
            for k2 in K:
                if k1!=k2:
                    for j,line in enumerate(tid):
                        if set(k1).issubset(line):
                            Lu1.add(j)
                        if set(k2).issubset(line):
                            Lu2.add(j)

                    L=(Lu1&Li)-(Lu2&Li)
                    if L!=set():
                        return(x.tent_v,L)
                    else:
                        return(x.tent_v,Lu1&Li)
                else:
                    return(x.tent_v,Li)
```

Η Συνάρτηση που βρίσκει τους συντελεστές βάρους για τον Αλγόριθμο BBA:

```
/* Η συνάρτηση  λαμβάνει σαν είσοδο τον αύξων αριθμό του ελάχιστου Σ.Σ. (Smin), τον αύξων αριθμό της συναλλαγής, το ευαίσθητο στοιχείου του οποίου το βάρος ζητούμε και το Α.Θ.Σ. Η συνάρτηση επιστρέφει το βάρος του ευαίσθητου στοιχείου για μία συναλλαγή. Προφανώς, η συνάρτηση επαναλαμβάνεται για κάθε συναλλαγή και ευαίσθητο στοιχείο.*/

def Weight(index,trans_id,sens_item,Rev_pos_bord):
    global TID
    temp=[];w=0
    w_set=collections.namedtuple('w_set', 'index trans item weight')
    for I_S in Rev_pos_bord:
        if (I_S).issubset(TID[trans_id]) :
            temp.append(I_S)

    for x in temp:
        if sens_item in x:
            w=w+1
    weight=w_set(index,trans_id,sens_item,w)
    #print(index,'TID=',trans_id,',',sens_item,',','w=',w)
    return(weight)
```

Η Συνάρτηση που υλοποιεί τον Αλγόρθμο BBA:

/* Η συνάρτηση αυτή λαμβάνει σαν είσοδο το S_{min}, το πλήθος των συναλλαγών, το Α.Θ.Σ. και το κατώφλι υποστήριξης. Η συνάρτηση επιστρέφει το βάρος του ευαίσθητου στοιχείου για μία συναλλαγή. Σκοπός της συνάρτησης αυτής είναι να δημιουργήσει τα δεδομένα εισόδου και να καλέσει την προηγούμενη συνάρτηση (Weight)για τον υπολογισμό των βαρών.*/

```
def BBA(S_min_l,S_min_sup_l,lines,Bd,msup):
    global TID
    W_list=[]
    for Index,Item_set in enumerate(S_min_l):
        #print('C-I for', Item_set, 'is:')
        for transaction_id,Trans_set in enumerate(TID):
            if set(Item_set).issubset(Trans_set):
                for item in Item_set:
                    W=Weight(Index,transaction_id,item,Bd)
                    W_list.append(W)

    Weight_sort=sorted(W_list,key=attrgetter('index','weight'))
    #print(Weight_sort)
    INDEX=0
    while INDEX<len(S_min_l):
        while S_min_sup_l[INDEX]>=msup:
            for i,x in enumerate(Weight_sort):
                if (x.index == INDEX):
                    break
            #print('INDEX=',x.index,',','TID=',

            #      x.trans,',','ITEM=',x.item)

Weight_sort=remove_item_and_TID(x.trans,x.item,Weight_sort)
            S_min_sup_l[INDEX]=S_min_sup_l[INDEX]-1/lines
        INDEX=INDEX+1
```