Πανεπιστήμιο Θεσσαλίας
Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών & Δικτύων

Τίτλος Διπλωματικής Εργασίας :
Απόκρυψη ευαίσθητων προτύπων σε τροχιές κινούμενων αντικειμένων

της
Καραμάρκου Χρυσούλας

Επιβλέπων Καθηγητής : Βερύκιος Βασίλειος

Βόλος 2009

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Ύστερα από την επανάσταση του Διαδικτύου και του Παγκόσμιου Ιστού (World WideWeb), γεννήθηκε το ερώτημα ποιο θα ήταν το επόμενο βήμα στην εξέλιξη του Παγκόσμιου Ιστού. Παρατηρώντας την σύγκλιση που λαμβάνει χώρα τα τελευταία χρόνια μεταξύ των τεχνολογιών πληροφορικής και τηλεπικοινωνιών, την απάντηση στο παραπάνω ερώτημα φαίνεται να δίνει ο κινητός Ιστός (mobile Web). Ο κινητός Ιστός έρχεται να καλύψει τις ανάγκες για πρόσβαση στην πληροφορία χωρίς χρονικούς ή χωρικούς περιορισμούς (anywhere- anytime access). Ο κινητός Ιστός σήμερα γίνεται πραγματικότητα με την πρόοδο που παρατηρείται στις ασύρματες και κινητές επικοινωνίες. Πολλές εφαρμογές για χρήστες που βρίσκονται εν κινήσει έχουν ήδη αναπτυχθεί και διατίθενται μέσω κινητών συσκευών. Φαίνεται ωστόσο πως αναζητούνται εφαρμογές και υπηρεσίες οι οποίες θα προκαλέσουν την ‘έκρηξη’ ζήτησης δίνοντας την απαιτούμενη ώθηση στην ανάπτυξη του κινητού Ιστού. Οι σύγχρονες τάσεις μας οδηγούν στην κατεύθυνση των υπηρεσιών που εξαρτώνται από την τοποθεσία .

Τα τελευταία χρόνια έχουν εμφανιστεί μια σειρά από μη παραδοσιακές εφαρμογές στο χώρο της πληροφορικής, των οποίων το κύριο χαρακτηριστικό είναι η συνύπαρξη χωρικών με παραδοσιακούς τύπους δεδομένων (π.χ. αριθμητικά ή αλφαριθμητικά δεδομένα). Αυτό σε συνδυασμό με την χρήση ασύρματων συσκευών και συστημάτων εντοπισμού γεωγραφικής θέσης (π.χ. GPS), έδωσε ιδιαίτερη ώθηση στην ανάπτυξη ειδικών εφαρμογών, όπως είναι οι υπηρεσίες θέσης και τα συστήματα

παρακολούθησης οχημάτων, οι οποίες υποστηρίζουν «παρακολούθηση», «αποστολή» και «αποδοχή» πληροφοριών, από οχήματα ή/και φυσικά πρόσωπα τα οποία βρίσκονται εν κινήσει. Η παρακολούθηση και η αποτελεσματική διαχείριση αυτών και γενικότερα των επονομαζόμενων «κινούμενων αντικειμένων» συνθέτουν ένα καινοτομικό ερευνητικό και τεχνολογικό πεδίο, το οποίο σταδιακά ασκεί μεγάλη επιρροή στον τρόπο που οι «εν κινήσει» επαγγελματίες οργανώνουν τις δραστηριότητές τους. Επίσης, δημιουργεί για τις εταιρείες (όπως είναι οι μεταφορικές, τηλεπικοινωνιακές, αεροπορικές, εταιρείες ταχυμεταφορών, χρηματαποστολών, ταξί, κ.ο.κ.) μοναδικές ευκαιρίες ανάπτυξης νέων προϊόντων και υπηρεσιών.

Από την άλλη πλευρά, αξιοσημείωτη είναι η έλλειψη εφαρμογών διαχείρισης δεδομένων κίνησης, οι οποίες θα μπορούσαν να αναλύσουν την κίνηση των χρηστών και να παράγουν γνώση, προκειμένου να ληφθούν ευεργετικά μέτρα για το κοινωνικό σύνολο (π.χ. εύρεση περιοχών κυκλοφοριακής συμφόρησης και ρύθμιση της κυκλοφορίας, λήψη αποφάσεων σχετικά με την επικείμενη δόμηση και την κατασκευή δρόμων κ.α.). Τα υπάρχοντα συστήματα διαχείρισης δεδομένων ενώ είναι εξαιρετικά στη διαχείριση παραδοσιακών τύπων και μπορούν να διαχειριστούν χωρικά δεδομένα, δεν μπορούν να συνδυάσουν ικανοποιητικά τη χρονική και χωρική πληροφορία που συνθέτει η κίνηση των αντικειμένων. Επιπρόσθετα, τα υπάρχοντα συστήματα δεν έχουν αποδείξει ότι είναι ικανά να αντιμετωπίσουν τον τεράστιο όγκο δεδομένων που προκύπτουν ως αποτέλεσμα των εφαρμογών αυτών.

Οι πράγματι ελκυστικές προσφερόμενες υπηρεσίες στο χώρο προϋποθέτουν αφενός ένα ισχυρό σύστημα διαχείρισης της χωροχρονικής πληροφορίας που σχετίζεται με αυτές, κάτι που τα σημερινά εμπορικά συστήματα βάσεων δεδομένων (π.χ. Oracle, IBM DB2, κλπ.), δεν ικανοποιούν στον επιθυμητό βαθμό, αφετέρου ένα σύστημα ικανό να εξασφαλίζει την προστασία της ιδιωτικότητας των ατόμων, των οποίων οι τροχιές καταγράφονται. Το γεγονός αυτό περιορίζει τις δυνατότητες χρήσης αλλά και βελτίωσης των συστημάτων διαχείρισης κινούμενων αντικειμένων και των υπηρεσιών θέσης που μπορούν να προσφερθούν στους τελικούς αποδέκτες.

Στην παρούσα διπλωματική προσεγγίζουμε τη λύση στο προαναφερόμενο πρόβλημα προτείνοντας ένα εργαλείο για την ανάλυση δεδομένων κίνησης (τροχιών) το οποίο συντελεί στην ανακάλυψη χρήσιμης γνώσης και παράλληλα διασφαλίζει την ιδιωτικότητα των ατόμων των οποίων η κίνηση καταγράφεται. Συγκεκριμένα, θεωρούμε μια βάση δεδομένων στην οποία αποθηκεύονται οι τροχιές των διαφόρων κινούμενων αντικειμένων, μαζί ίσως με άλλες πληροφορίες που αφορούν τα άτομα αυτά (π.χ. αναγνωριστικό, όνομα, ηλικία κλπ.) και οι οποίες είναι γνωστές εκ των προτέρων. Η καταγραφή των δεδομένων κίνησης των διαφόρων χρηστών εγείρει σημαντικά ζητήματα ιδιωτικότητας, αναφορικά με το ποιός επιτρέπεται να έχει πρόσβαση στην ευαίσθητη πληροφορία, σε τι βαθμό, καθώς επίσης και τι είδους γνώση επιτρέπεται να αποκομίσει από το σύστημα. Στην περίπτωση που τα δεδομένα κίνησης πρόκειται να αναλυθούν από τρίτους, μη έμπιστους φορείς, είναι εξαιρετικά σημαντικό να υπάρχει πλήρης έλεγχος της γνώσης που μπορούν να αποκομίσουν οι φορείς αυτοί από την ανάλυση των δεδομένων, σε συνδυασμό με την προστασία της ιδιωτικότητας των χρηστών των οποίων η κίνηση έχει καταγραφεί. Η λύση που παρουσιάζουμε (QUERY ENGINE) υιοθετεί τη λογική ενός έμπιστου ενδιάμεσου

σταθμού: πρόκειται για ένα σύστημα που προσφέρει στο χρήστη μια διεπαφή μέσω της οποίας μπορεί να υποβάλει ερωτήματα στη βάση δεδομένων και να αντλήσει γνώση από αυτή, ενώ παράλληλα αναλαμβάνει τον έλεγχο της γνώσης που παρέχεται στο χρήστη βασιζόμενη στο βαθμό εξουσιοδότησής του στο σύστημα.

Η QUERY ENGINE βασίζεται στο σύστημα HERMES, το οποίο αποτελεί μια πρωτοποριακή λύση στο πρόβλημα της διαχείρισης των δεδομένων που αφορούν κίνηση, όπως οι τροχιές των κινούμενων αντικειμένων, παρέχοντας μηχανισμούς αποδοτικής διαχείρισης αυτών. Αναλυτικότερα τα υπάρχοντα σχεσιακά ή αντικειμενο-σχεσιακά συστήματα βάσεων δεδομένων, όπως αυτά που προαναφέρθηκαν, έχουν περιορισμένες δυνατότητες διαχείρισης των τροχιών των αντικειμένων (στο χώρο και στο χρόνο) καθώς με τη σημερινή τεχνολογία οι τροχιές καταγράφονται τμηματικά και χωριστά ως προς τη χωρική και χρονική συνιστώσα. Αντίθετα, το σύστημα HERMES αντιμετωπίζει την κάθε τροχιά ως μια ενιαία οντότητα, δίνοντας έτσι τη δυνατότητα αποδοτικής αποθήκευσης και ανάκτησης αυτής, επιτυγχάνοντας την δυνατότητα βελτιστοποιημένων υπηρεσιών για συστήματα διαχείρισης κινούμενων αντικειμένων.

Η δομή της εργασίας είναι η εξής. Στο Κεφάλαιο 2 παρουσιάζεται το υπόβαθρο γνώσης που πρέπει να διαθέτει κανείς προκειμένου να παρακολουθήσει την παρούσα εργασία. Στο Κεφάλαιο 3 παρουσιάζεται το σύστημα HERMES. Πιο συγκεκριμένα, αναλύονται οι τύποι δεδομένων που υποστηρίζει αλλά και οι διάφορες εφαρμογές που μας παρέχει, μέσω των οποίων δίνεται η δυνατότητα αποθήκευσης, επεξεργασίας και ελέγχου των τροχιών των αντικειμένων. Στο Κεφάλαιο 4 παρουσιάζουμε συνοπτικά

το εργαλείο QUERY ENGINE, παραθέτοντας τα είδη επερωτήσεων που υποστηρίζει και περιγράφοντας τους αλγορίθμους - με χρήση παραδειγμάτων- που υλοποιούν αυτές τις επερωτήσεις. Στο Κεφάλαιο 5 παρουσιάζονται τα είδη των επιθέσεων που απειλούν την ιδιωτικότητα των χρηστών της QUERY ENGINE, αλλά και τους αλγορίθμους που χρησιμοποιεί αυτή προκειμένου να επιτύχει την προστασία τους. Στο Κεφάλαιο 6, παρουσιάζονται οι ψευδοκώδικες των αλγορίθμων για τους οποίους έγινε λόγος στο Κεφάλαιο 4. Στο Κεφάλαιο 7, αναλύεται ο τρόπος με τον οποίο έγιναν κάποιες πειραματικές μετρήσεις στο σύστημα μας και παρουσιάζονται τα αποτελέσματα που πήραμε από τα πειράματα αυτά. Τέλος στο Κεφάλαιο 8, παραθέτουμε τις γενικές παρατηρήσεις μας για την λειτουργία της QUERY ENGINE, όπως επίσης και προτάσεις για μελλοντική επέκταση της.

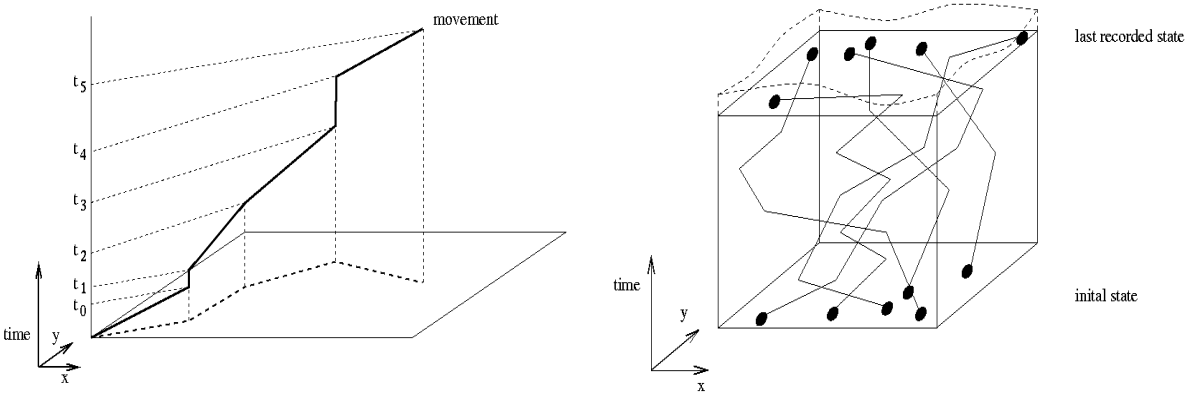
Κεφάλαιο 2

ΥΠΟΒΑΘΡΟ

Στο κεφάλαιο αυτό παρατίθενται όλες οι βασικές γνώσεις που πρέπει να κατέχει ένας αναγνώστης προκειμένου να αντιληφθεί με απόλυτη ευκρίνεια το περιεχόμενο της παρούσας εργασίας.

2.1 Ορισμός κινούμενου αντικειμένου

Όταν επιχειρείται ενοποίηση της χρονικής και χωρικής διάστασης αντικειμένων μιλάμε για γεωμετρίες μεταλλασσόμενες στο χρόνο. Τέτοιου τύπου γεωμετρίες μπορούν να αλλάζουν/κινούνται τόσο σε διακριτά όσο και σε συνεχόμενα στάδια. Στην περίπτωση που μας ενδιαφέρει μόνο η θέση των γεωμετριών στο χώρο μιλάμε για κινούμενα αντικείμενα/σημεία. Όπως έχει διατυπωθεί από τους Pfoser et al. (1), δεδομένα από κινούμενα αντικείμενα μοιάζουν με μία αλληλουχία σημείων στον 3-διάστατο χώρο, δηλαδή το 2-διάστατο επίπεδο συν τη διάσταση του χρόνου (βλ. Εικόνα 2.1).



Εικόνα 2. 1 Κινούμενα αντικείμενα α) τροχιά β) συλλογή τροχιών

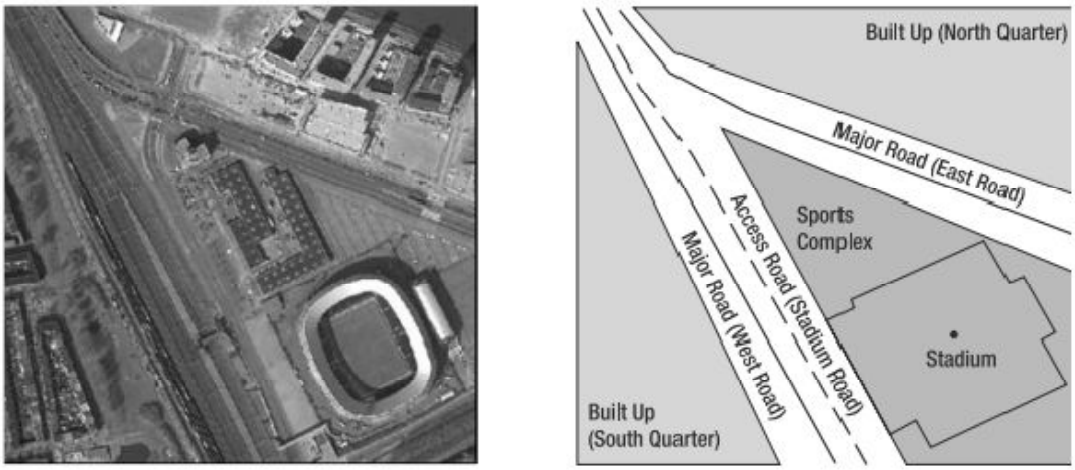
2.2 Αποθήκευση χωρικών δεδομένων στην Oracle Spatial

Η Oracle Spatial είναι ένα υποσύστημα του Συστήματος Διαχείρισης Βάσεων Δεδομένων της Oracle και αποτελεί ένα σύνολο συναρτήσεων και διεργασιών που προσφέρουν την δυνατότητα αποθήκευσης, ανάκτησης, αναπροσαρμογής και δημιουργίας επερωτήσεων (queries) πάνω σε χωρικά δεδομένα μιας βάσης. Η Oracle Spatial είναι σχεδιασμένη έτσι ώστε να διευκολύνει τη διαχείριση γεωγραφικών δεδομένων σε εφαρμογές όπως για παράδειγμα ένα Γεωγραφικό Σύστημα Πληροφοριών (GIS) . Μόλις αποθηκευτούν αυτά τα στοιχεία σε μια βάση δεδομένων της Oracle, μπορούν εύκολα να χειριστούν, να ανακτηθούν καθώς και να συσχετιστούν με όλα τα άλλα στοιχεία της βάσης.

Ορισμένα από τα στοιχεία που παρέχει η Oracle Spatial είναι τα εξής :

- Ένα σχήμα που ορίζει την αποθήκευση, τη σύνταξη, και τη σημασιολογία των γεωμετρικών τύπων δεδομένων που υποστηρίζονται.
- Ένα μηχανισμό δεικτοδότησης χωρικών δεδομένων με χρήση ευρετηρίων.
- Ένα σύνολο τελεστών (operators) και συναρτήσεων (functions) για την διατύπωση χωρικών επερωτήσεων (queries).
- Μια ειδική γλώσσα επερωτήσεων που υποστηρίζει επερωτήσεις που συσχετίζουν τα χωρικά δεδομένα.
- Ένα μοντέλο δικτύου δεδομένων στο οποίο τα αντικείμενα μοντελοποιούνται ως κόμβοι και ακμές ενός χωρικού δικτύου.

Ένα συνηθισμένο παράδειγμα των χωρικών δεδομένων μπορεί να αναπαρασταθεί σε έναν οδικό χάρτη. Ένας οδικός χάρτης είναι ένα δισδιάστατο αντικείμενο που περιέχει τα σημεία, τις γραμμές, και τα πολύγωνα που μπορούν να αντιπροσωπεύουν τις πόλεις, τους δρόμους, και τις επαρχίες αντίστοιχα. Γενικότερα τα χωρικά δεδομένα της Oracle παρέχουν μια απεικόνιση των αντικειμένων ενδιαφέροντος, δημιουργώντας μια απλουστευμένη μορφή της πραγματικότητας. Στην παρακάτω εικόνα (Εικόνα 2.2), τα διάφορα αντικείμενα ενδιαφέροντος έχουν απεικονιστεί με τη βοήθεια σημείων, γραμμών και πολυγώνων. Η εικόνα που αποθηκεύεται στη βάση είναι μια αφαίρεση της πραγματικότητας.



Εικόνα 2.2 : Απεικόνιση σε χωρικά δεδομένα

2.3 Αποθήκευση χωροχρονικών δεδομένων στην Oracle μέσω του εργαλείου HERMES

Το εργαλείο HERMES είναι μία μηχανή υποστήριξης «βάσεων δεδομένων κινούμενων αντικειμένων» (MOD) η οποία χειρίζεται τους τύπους δεδομένων και τους σχετικούς αλγόριθμους για τον ορισμό, την αποθήκευση και την ανάκτηση χωροχρονικών δεδομένων. Ειδικότερα, το αντικειμενο-σχεσιακό σύστημα βάσεων δεδομένων της Oracle το οποίο είναι η πλατφόρμα πάνω στην οποία στηρίζεται η υλοποίηση των εν λόγω καινοτομικών τεχνικών, διαχειρίζεται χωρικά και χωροχρονικά δεδομένα, τα οποία περιέχουν εκτός από στατικά γεωγραφικά στοιχεία (π.χ. δίκτυα δρόμων, κτίρια, ψηφιακούς χάρτες) και ειδικούς τύπους αναπαράστασης κινουμένων αντικειμένων (π.χ. τροχιές ατόμων).

Η QUERY ENGINE βασίζεται στο σύστημα HERMES (2), (3), το οποίο αποτελεί μια πρωτοποριακή λύση στο πρόβλημα της διαχείρισης των δεδομένων που αφορούν κίνηση, όπως οι τροχιές των κινούμενων αντικειμένων, παρέχοντας μηχανισμούς αποδοτικής διαχείρισης αυτών. Αναλυτικότερα τα υπάρχοντα σχεσιακά ή αντικειμενο-σχεσιακά συστήματα βάσεων δεδομένων, έχουν περιορισμένες δυνατότητες διαχείρισης των τροχιών των αντικειμένων (στο χώρο και στο χρόνο) καθώς με τη σημερινή τεχνολογία οι τροχιές καταγράφονται τμηματικά και χωριστά ως προς τη χωρική και χρονική συνιστώσα. Αντίθετα, το σύστημα HERMES αντιμετωπίζει την κάθε τροχιά ως μια ενιαία οντότητα, δίνοντας έτσι τη δυνατότητα αποδοτικής αποθήκευσης και ανάκτησης αυτής, επιτυγχάνοντας την δυνατότητα βελτιστοποιημένων υπηρεσιών για συστήματα διαχείρισης κινούμενων αντικειμένων.

Στο Κεφάλαιο 3 παρουσιάζονται αναλυτικότερα οι τύποι δεδομένων και οι λειτουργίες που υποστηρίζει το σύστημα HERMES.

2.4 Ιδιωτικότητα χωροχρονικών δεδομένων

Οι τεχνολογικές εξελίξεις στον τομέα των δικτύων και των συστημάτων γεωγραφικού εντοπισμού επιτρέπουν την ανάπτυξη εξατομικευμένων εφαρμογών υψηλού κοινωνικού και οικονομικού ενδιαφέροντος. Εντούτοις, οι υπηρεσίες αυτές είναι πολύπλοκες και απαιτούν τη γνώση πολλών χαρακτηριστικών των χρηστών. Πολλές από τις καθημερινές δραστηριότητες των ανθρώπων πραγματοποιούνται τα τελευταία χρόνια χρησιμοποιώντας φορητές συσκευές, όπως κινητά τηλέφωνα, φορητοί υπολογιστές και υπολογιστές παλάμης. Παρόλο που τα δεδομένα αυτά έχουν τεράστιες απαιτήσεις σε χώρο για τη συλλογή τους, η πρόοδος των τεχνολογιών και των βάσεων δεδομένων επιτρέπει τη συνεχή αποθήκευση τέτοιων πληροφοριών. Τα δεδομένα τα οποία συλλέγονται από τις φορητές συσκευές χαρακτηρίζονται ως «ευαίσθητα» εάν μπορούν να αποκαλύψουν τη θέση των χρηστών σε συγκεκριμένες χρονικές στιγμές. Αυτό έχει ως αποτέλεσμα όταν συνδυάζονται με προηγούμενα αποθηκευμένα δεδομένα για κάποιο συγκεκριμένο χρήστη να δημιουργούν μία πλήρη ηλεκτρονική εικόνα για τις κινήσεις του. Ως εκ τούτου, η γνώση τέτοιου είδους πληροφοριών ελλοχεύει πολλούς κινδύνους για θεμελιώδη δικαιώματα και ελευθερίες της ιδιωτικότητας της προσωπικής ζωής των ανθρώπων.

Η ιδιωτικότητα αποτελεί ένα μείζον θέμα στον τομέα των τηλεπικοινωνιών το οποίο έχει κεντρίσει την προσοχή τόσο του νομοθετικού όσο και του τεχνολογικού και ερευνητικού τομέα.

Η ιδιωτικότητα αποτελεί ένα από τα πιο σημαντικά ζητήματα που συνδέονται με τις τεχνολογίες πληροφοριών. Κατά τη διάρκεια της συλλογής και ανταλλαγής δεδομένων προκύπτουν πολλά ερωτήματα τα οποία απαιτούν απαντήσεις. Ειδικότερα, πώς μπορούν τα ευαίσθητα προσωπικά δεδομένα να αποθηκευτούν και να υποβληθούν σε επεξεργασία χωρίς να παραβιαστούν τα προσωπικά δικαιώματα και η ελευθερία των ατόμων. Πώς μπορούν τα προσωπικά δεδομένα να χρησιμοποιούνται αποκλειστικά από εξουσιοδοτημένες οντότητες με τέτοιο τρόπο ώστε να βοηθούν τα άτομα και όχι να τα βλάπτουν, παραδείγματος χάριν στιγματίζοντάς τα κοινωνικά. Αυτά τα ερωτήματα τα οποία απαιτούν απαντήσεις σε ένα συνδυασμένο τεχνικό, νομικό και κοινωνικό επίπεδο, εξετάζουν ένα πολύ κρίσιμο ζήτημα, τόσο από εθνική όσο και από κοινωνική σκοπιά.

Στη διπλωματική αυτή θα παρουσιάσουμε τεχνικές με τις οποίες θα προσπαθήσουμε να διαφυλάξουμε την ιδιωτικότητα αυτή σε όσο το δυνατόν μεγαλύτερο βαθμό.

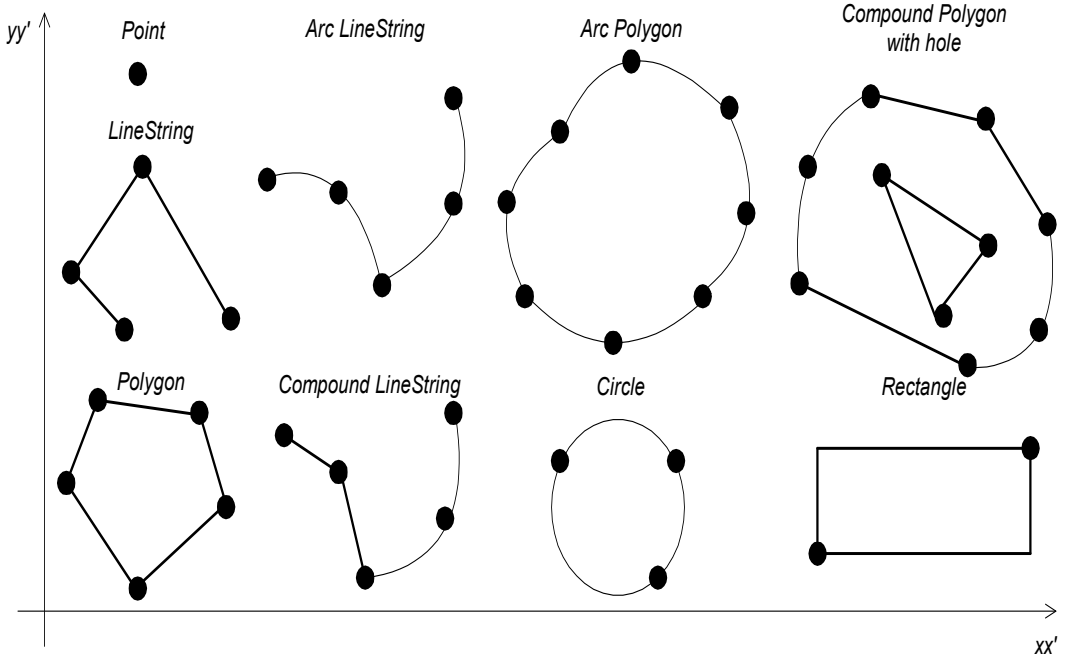
Κεφάλαιο 3

HERMES

Το σύστημα HERMES, ως εργαλείο επερωτήσεων και αποτελεσματικής αποθήκευσης και ανάκτησης τροχιών κινούμενων αντικειμένων, αποτελεί την «καρδιά» του προτεινόμενου συστήματος για την προστασία της ιδιωτικότητας τέτοιου τύπου δεδομένων. Βλέποντας τις λειτουργίες του, θα μπορούσε να πει κανείς ότι είναι σχεδιασμένο έτσι ώστε να βοηθάει ένα προγραμματιστή-διαχειριστή χωροχρονικών εφαρμογών σε όλα τα στάδια των δραστηριοτήτων του. Οι εργασίες αυτές ποικίλουν από την μοντελοποίηση, την δημιουργία, και την υποβολή ερωτήσεων σε μία βάση δεδομένων αντικειμένων που κινούνται στο χώρο σε διακριτά στάδια ή συνεχόμενα, μέχρι την ανάπτυξη ειδικών εφαρμογών (π.χ. γεωγραφικά πληροφοριακά συστήματα) που στηρίζονται σε αυτή τη βάση.

Η επιστημονική μεθοδολογία η οποία ακολουθήθηκε για την ανάπτυξη αυτού του εργαλείου είναι η επέκταση του υπάρχοντος αντικειμενο-σχεσιακού συστήματος διαχείρισης δεδομένων της Oracle 10g (4), με τις χωροχρονικές λειτουργίες που περιγράφηκαν σε αδρές γραμμές προηγουμένως και που θα αναλυθούν λεπτομερώς παρακάτω. Το εργαλείο αυτό έχει σχεδιαστεί με τέτοιο τρόπο ούτως ώστε να μπορεί να χρησιμοποιηθεί είτε ως ένα σύστημα διαχείρισης στατικών γεωμετριών, είτε ως ένα σύστημα υποστήριξης χρονικών λειτουργιών σε παραδοσιακούς τύπους δεδομένων, ενώ η κύρια λειτουργία του είναι η μοντελοποίηση και η διαχείριση «κινούμενων αντικειμένων». Όλες αυτές οι διακριτές λειτουργίες ορίζονται, υλοποιούνται και παρέχονται υπό τη μορφή τριών διακριτών αλλά συνεργαζόμενων

υποσυστημάτων («θηκών δεδομένων») χρησιμοποιώντας την διεπαφή επεκτασιμότητας της Oracle 10g. Στις ενότητες που ακολουθούν εξετάζουμε διεξοδικά κάθε ένα από τα υποσυστήματα αυτά. Το υποσύστημα διαχείρισης χωρικών δεδομένων έχει αναπτυχθεί από την Oracle και παρέχει στον HERMES ένα ολοκληρωμένο σύνολο συναρτήσεων και διαδικασιών με τις οποίες γίνεται εύκολη και αποτελεσματική η αποθήκευση, η προσπέλαση και η ανάλυση χωρικών δεδομένων. Οι γεωμετρικές λειτουργίες αντιμετωπίζουν τα δεδομένα στατικά. Με άλλα λόγια δεν ενυπάρχει η έννοια του χρόνου συνδεδεμένη με τα γεωμετρικά αντικείμενα. Το γεωμετρικό μοντέλο δεδομένων που ακολουθείται από το υποσύστημα αυτό είναι μία ιεραρχική δομή αποτελούμενη από «στοιχεία», «γεωμετρίες» και «στρώματα». Τα «στρώματα» αποτελούνται από ομοιογενείς ή ετερογενείς συλλογές από «γεωμετρίες» οι οποίες με τη σειρά τους αποτελούνται από «στοιχεία». Η Εικόνα 3.1 παρουσιάζει γραφικά τα υποστηριζόμενα γεωμετρικά «στοιχεία».



Εικόνα 3.1 Υποστηριζόμενα γεωμετρικά «στοιχεία»

Το υποσύστημα διαχείρισης χρονικών λειτουργιών αποτελεί τη λύση που προσφέρει ο HERMES υπό το περιβάλλον της Oracle 10g σε θέματα που συσχετίζονται με τη διάσταση του χρόνου. Ο γενικής χρήσεως σχεδιασμός του υποσυστήματος αυτού επιτρέπει την εκμετάλλευσή του σε οποιονδήποτε τομέα ανάλυσης παραδοσιακών δεδομένων (π.χ. ανάλυση χρονοσειρών), αλλά στα πλαίσια του HERMES η μεγάλη του χρησιμότητα έγκειται στο σχεδιασμό και την υλοποίηση του συστήματος διαχείρισης κινούμενων αντικειμένων. Από ερευνητικής και τεχνολογικής απόψεως, το υποσύστημα διαχείρισης χρονικών λειτουργιών είναι η αντιστοίχιση του οντοκεντρικού χρονικού μοντέλου TAU (5), στο αντικείμενο-σχεσιακό μοντέλο της Oracle 10g. Έτσι, το υποσύστημα αυτό επεκτείνει τους τέσσερις χρονικούς τύπους δεδομένων που ορίζονται από το ODMG μοντέλο (6): Ημερομηνία, Χρόνος, Χρονο-αποτύπωμα και Διάστημα, με τρεις νέους χρονικούς τύπους αντικειμένων: Χρονοσημείο, Περίοδος και Χρονικό Στοιχείο.

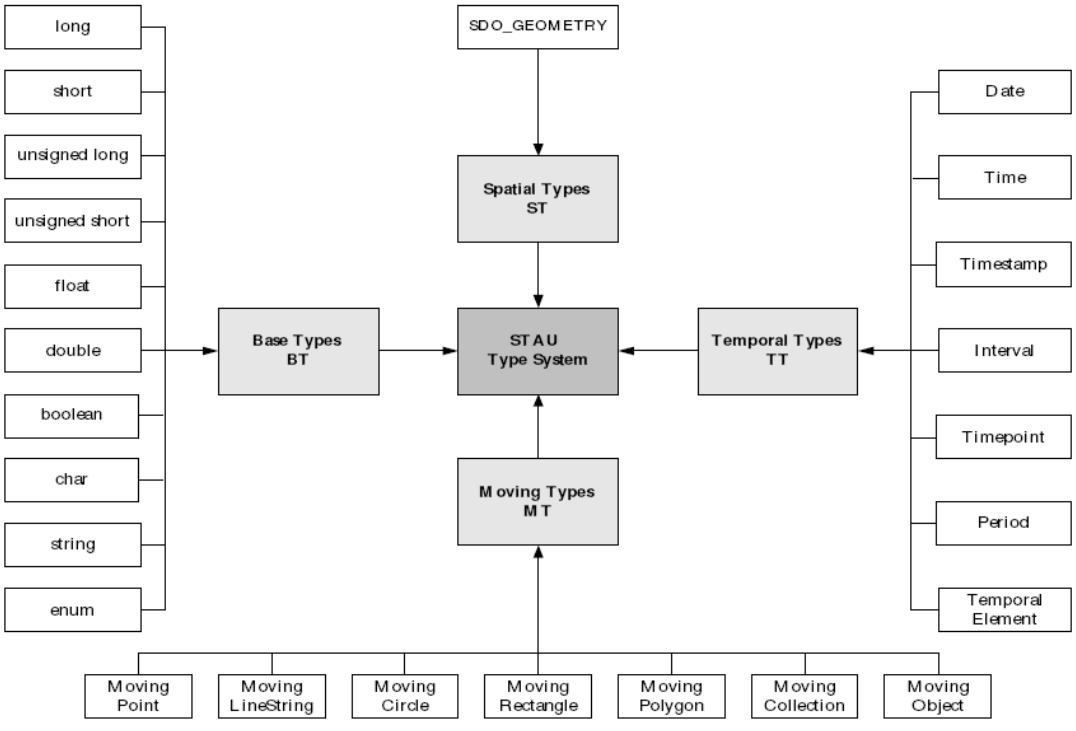
Η σχεδιαστική αρχή πάνω στην οποία βασίζεται ο HERMES είναι η υποστήριξη αντικειμένων των οποίων η θέση μεταβάλλεται διαρκώς στο χώρο. Κλιμακωτές αλλαγές, οι οποίες υποστηρίζονται από παραδοσιακά συστήματα, λαμβάνονται επίσης υπόψη. Για την υλοποίηση ενός τέτοιου συστήματος διαχείρισης δεδομένων είναι απαραίτητοι όλοι οι τύποι γεωμετρικών και χρονικών αντικειμένων μαζί με τις λειτουργίες αυτών, όπως ορίζονται στα δύο προηγούμενα υποσυστήματα. Βασιζόμενοι σε αυτούς τους τύπους δεδομένων και εκμεταλλευόμενοι την ιδέα της «τμηματικής απεικόνισης» των κινούμενων αντικειμένων (7), (8), οδηγούμαστε στον ορισμό ενός συνόλου από χρονικά μεταβαλλόμενες γεωμετρίες, οι τιμές των οποίων

είναι συναρτήσεις που συνδέουν οποιαδήποτε χρονική στιγμή με μία γεωμετρική τιμή.

Παρακάτω παρουσιάζεται ο βασικός τύπος δεδομένων που χρησιμοποιήθηκε από το σύστημά μας, έτσι ώστε να αναπαρασταθούν οι χρονικά μεταβαλλόμενες τροχιές των χρηστών μας.

3.1 Τύποι του συστήματος HERMES

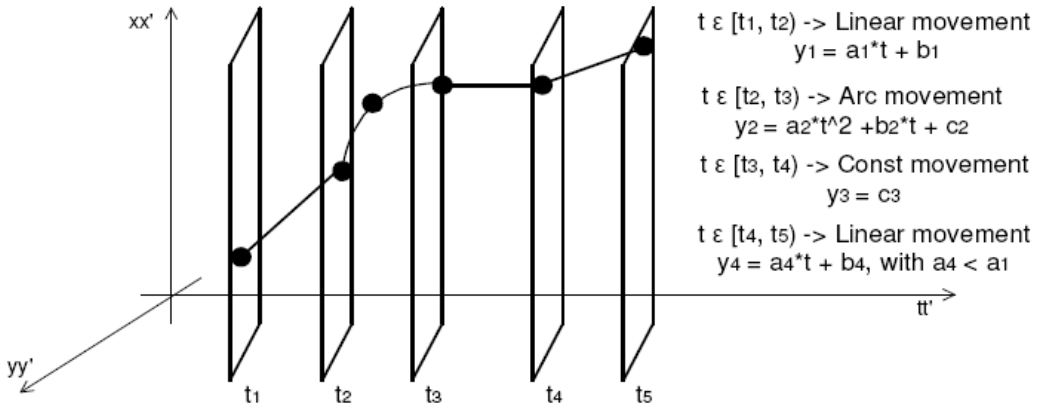
Οι τύποι του συστήματος HERMES (9), χωρίζονται σε τρείς κατηγορίες: τους Base Types (Βασικοί τύποι), τους αμιγώς Temporal Types(Χρονικοί τύποι), τους αμιγώς Spatial Types(Χωρικοί τύποι) και τους Moving Types (κινούμενους τύπους) MT. Για παράδειγμα HERMES = BT U TT U ST U MT. Στην Εικόνα 3.2, εμφανίζονται όλοι οι τύποι του συστήματος HERMES.



Εικόνα 3.2 Τύποι του συστήματος HERMES

3.1.1 Moving Type (Κινούμενος Τύπος)

Ο συγκεκριμένος τύπος που παρέχεται από το σύστημα HERMES είναι αυτός που μας ενδιαφέρει τελικά, καθώς έχει τη δυνατότητα να διαχειριστεί τα δεδομένα μας κάνοντας χρήση τόσο των χωρικών όσο και των χρονικών ιδιοτήτων που έχουν αυτά. Η μορφή απεικόνισης των τροχιών των αντικειμένων γίνεται σε κομμάτια, η βασική ιδέα της οποίας είναι η αποσύνθεση της χρονικής ανάπτυξης μιας κινούμενης τιμής σε τμήματα που ονομάζονται 'slices'(κομμάτια), έτσι ώστε μέσα στα κομμάτια αυτά να περιγράφεται η εξέλιξη με μια σειρά 'απλών' συναρτήσεων. Αυτό απεικονίζεται στην Εικόνα 3.3, για ένα κινούμενο αντικείμενο (moving point).



Εικόνα 3.3 Κινούμενο αντικείμενο (moving point) με διάφορους τύπους κίνησης.

Η 'απεικόνιση σε κομμάτια' χρησιμοποιείται στην εφαρμογή του HERMES-MDC. Για να χρησιμοποιηθεί αυτού του είδους η απεικόνιση στον ορισμό ενός κινούμενου τύπου θα πρέπει να γίνει αποσύνθεση του ορισμού κάθε κινούμενου αντικειμένου σε διάφορους ορισμούς , ένας για κάθε κομμάτι, που αντιστοιχεί σε μια απλή συνάρτηση. Στη συνέχεια, όλοι αυτοί οι υπο-ορισμοί, θα πρέπει να συντεθούν ώστε να οριστεί ο κινούμενος τύπος. Κάθε ένας από αυτούς τους υπο-ορισμούς αντιστοιχεί στο λεγόμενο unit moving point.

Για να οριστεί ένας unit moving type, είναι αναγκαίο να συσχετιστεί μια χρονική περίοδος με την περιγραφή μιας απλής συνάρτησης, η οποία αναπαριστά τη συμπεριφορά του κινούμενου αντικειμένου στη συγκεκριμένη χρονική περίοδο. Με βάση αυτή τη θεωρία, δύο έννοιες χρησιμοποιούνται, η 'χρονική περίοδος' και η 'απλή συνάρτηση'. Η πρώτη ιδέα παρουσιάζεται ως τύπος αντικειμένου μοντέλου από το TAU-TLL (με βάση την ορολογία TAU-TLL ονομάζεται D_Period_Sec). Η δεύτερη ιδέα, είναι ένας τύπος αντικειμένου που λέγεται Unit_Function, και αφορά στις διαφορετικές αναπαραστάσεις απλών συναρτήσεων, που έχουν ως κύριο χαρακτηριστικό, στον ορισμό τους κάθε πραγματικό αριθμό που αντιστοιχεί σε ένα χρονικό σημείο 'μέσα' σ' αυτό που πριν αναφέραμε ως 'χρονική περίοδο', κι έχει τιμές κάθε πραγματικό αριθμό.

$f(t) : t \in \Re \bigcap [t_1, t_2) \longrightarrow \Re$ **,όπου $[t_1, t_2)$ είναι η χρονική περίοδος**

Συνδυάζοντας αυτούς τους δύο τύπους αντικειμένων, ορίζεται ο πιο απλός unit object type, ο οποίος ονομάζεται Unit_Moving_Point. Αυτός είναι ένας βασικός τύπος αφού όλοι οι μεταγενέστεροι τύποι της ίδιας ομάδας ορίζονται με βάση αυτόν.

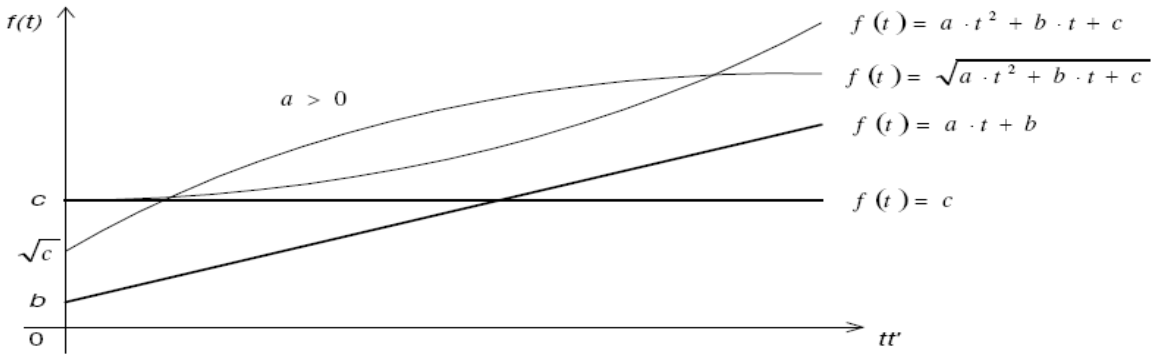
3.1.2 Φυσική αναπαράσταση του Moving Type

Στην ενότητα αυτή, παρουσιάζεται η αναπαράσταση του κινούμενου τύπου που ορίσαμε παραπάνω στη φυσική δομή Moving_Point ώστε να αποθηκεύονται, συνεχόμενα και ξεχωριστά γεωμετρικά δεδομένα που σχετίζονται με το χρόνο, σε μια βάση δεδομένων Oracle 10g.

Moving_Point

Ένας Moving_Point τύπος αντικειμένου ορίζεται ως συλλογή από Unit_Moving_Point αντικείμενα, τα οποία τελικά ορίζονται ως αντικείμενα που αποτελούνται από 3 ιδιότητες. Η πρώτη, είναι η χρονική περίοδος κατά την οποία ορίζονται οι άλλες δύο ιδιότητες. Η χρονική περίοδος εκφράζεται ως D_Period_Sec αντικείμενο που εφαρμόζεται στο TAU-TLL ενώ οι άλλες δύο ιδιότητες είναι οι τύποι του Unit_Function, που χαρακτηρίζεται από μια σειρά πραγματικών αριθμών στο διάστημα (t1,t2), όπου t1 είναι το σημείο έναρξης της περιόδου και t2 το σημείο λήξης της. Υπάρχουν δύο τέτοιες ιδιότητες, μια για καθένα από τους x,y στην καρτεσιανή επιφάνεια.

Το Unit_Function αποτελείται από μια σειρά τριών πραγματικών αριθμών (a,b,c) που αντιπροσωπεύουν διαφορετικές συναρτήσεις και ένα σήμα που δείχνει τον τύπο της απλής συνάρτησης. Υπάρχουν 4 τύποι λειτουργιών: πολυώνυμο πρώτου και δευτέρου βαθμού, η τετραγωνική ρίζα του πολυωνύμου δευτέρου βαθμού και η συνεχής συνάρτηση. Στην Εικόνα 3.4 φαίνεται ο τύπος της κίνησης καθεμιάς από τις μαθηματικές συναρτήσεις.



Εικόνα 3.4 Γραφική αναπαράσταση των τύπων της Unit_Function

Σχεδιάζεται, λοιπόν, ένας κινούμενος τύπος που μεταβάλλεται, για μια χρονική περίοδο, θέτοντας όλα τα Unit_Function αντικείμενα του αντίστοιχου unit-moving τύπου, ώστε να γίνουν συνεπείς συναρτήσεις. Εξαιτίας του γεγονότος ότι οι μεταβλητές που εκπροσωπούνται από τα αντικείμενα αυτά δεν αλλάζουν στη συγκεκριμένη χρονική περίοδο, μπορούμε αντίστοιχα να πάρουμε κάποιο στιγμιότυπο από την κινούμενη γεωμετρία, η οποία είναι έγκυρη και καλύπτει ολόκληρη την περίοδο. Αν έστω και μία από αυτές τις συναρτήσεις δεν είναι συνεχής, τότε η αλλαγή του κινούμενου τύπου συνεχίζεται για όλη τη χρονική περίοδο.

3.2 Λειτουργίες του συστήματος HERMES

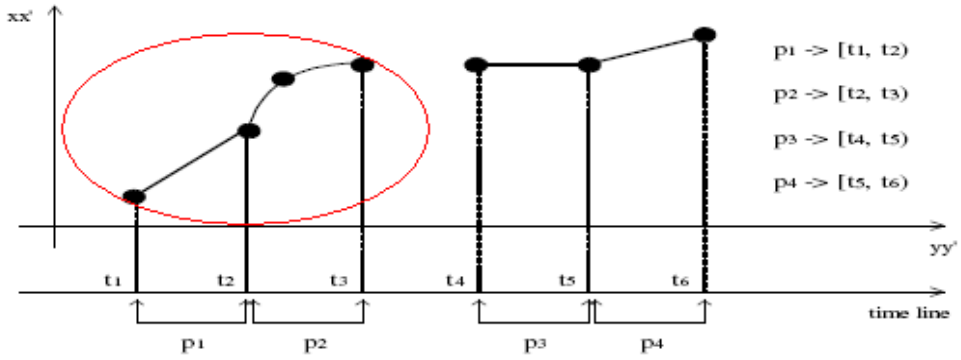
Κατάλληλες συναρτήσεις και διαδικασίες ορίζονται στους τρείς τύπους δεδομένων που αναφέρθηκαν στην Ενότητα 3.1, ώστε να επιτευχθεί η αποτελεσματική διαχείριση κινούμενων αντικειμένων. Οι κατηγορίες λειτουργιών που υποστηρίζονται είναι οι παρακάτω:

- ***Ελεγκτικές*** - Λειτουργίες υπεύθυνες για τη διατήρηση της βάσης σε συνεπή κατάσταση.
- ***Κατηγορήματα*** - Λειτουργίες που περιγράφουν τυπολογικές ή άλλου τύπου σχέσεις μεταξύ κινούμενων αντικειμένων (π.χ. όχημα κινούμενο εσωτερικά ή εξωτερικά μίας περιοχής).
- ***Προβολή και διεπαφή με το χωρικό ή χρονικό πεδίο ορισμού*** - Λειτουργίες οι οποίες περιορίζουν ή/και προβάλουν κινούμενα αντικείμενα στο χώρο και στο χρόνο (π.χ. η θέση ενός οχήματος τη χρονική στιγμή t).
- ***Αριθμητικές*** - Λειτουργίες που υπολογίζουν κάποια αριθμητική τιμή (π.χ. το μήκος της τροχιάς που έχει διαγράψει ένα όχημα έως τη χρονική στιγμή t).
- ***Απόστασης και κατεύθυνσης*** - Μέθοδοι που διευκολύνουν, για παράδειγμα, τον υπολογισμό της ελάχιστης απόστασης ανάμεσα σε κινούμενα αντικείμενα.
- ***Λειτουργίες συνόλων*** - Για παράδειγμα η ένωση δύο κινούμενων οχημάτων σε μία οντότητα.
- ***Ρυθμού αλλαγής*** - Λειτουργίες που περιγράφουν χαρακτηριστικά κίνησης (π.χ. ταχύτητα ή επιτάχυνση ενός οχήματος).

Για την υλοποίηση της Query Engine χρησιμοποιήθηκαν 3 λειτουργίες από την κατηγορία “Προβολή και διεπαφή με το χωρικό ή χρονικό πεδίο ορισμού”, οι οποίες παρουσιάζονται παρακάτω:

at_period (D_Period_Sec): Η μέθοδος επιστρέφει ένα αντικείμενο Moving_Point για τα σημεία του οποίου ισχύει η χρονική περίοδος που ορίζεται από το αντικείμενο D_Period_Sec. Δηλαδή, χρησιμοποιώντας αυτή τη συνάρτηση ο χρήστης μπορεί να οριοθετήσει τη χρονική περίοδο που είναι σημαντική για το σχεδιασμό του κινούμενου αντικειμένου στον τομέα του χώρου. Πιο συγκεκριμένα, η χρονική περίοδος που αποτελεί όρισμα της μεθόδου συγκρίνεται με όλα τα D_Period_sec αντικείμενα που χαρακτηρίζουν τα Moving_Point που μας ενδιαφέρουν. Αν η παράμετρος της περιόδου δεν συμπίπτει με τη συγκρινόμενη περίοδο, τότε το αντίστοιχο κινούμενο Moving_Point παραλείπεται. Αν συμπίπτει, τότε το Moving_Point επιστρέφεται σαν αποτέλεσμα της μεθόδου

Για παράδειγμα, στην Εικόνα 3.6 παρουσιάζεται η γεωμετρία ενός Moving_Point , το οποίο έχει οριστεί στην χρονική περιοχή [t1,t6). Εάν κληθεί η λειτουργία ***at_period*** για τη χρονική περίοδο [t1,t3), σαν αποτέλεσμα θα λάβουμε το κομμάτι του Moving_Point που βρίσκεται μέσα στην κόκκινη περιοχή.



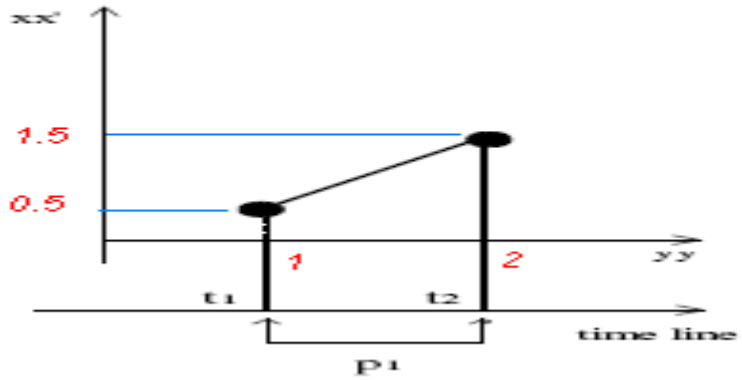
Εικόνα 3.6 Προβολή ενός Moving_Point σε μια χρονική περιοχή

f_trajectory(): Η λειτουργία αυτή μετατρέπει ένα Moving_Point αντικείμενο σε Sdo_Geometry. Πιο συγκεκριμένα, ο σχεδιασμός της κίνησης του Moving_Point στην Καρτεσιανή επιφάνεια γίνεται με τη χαρτογράφηση των χρονικά εξαρτημένων μεταβλητών του αντικειμένου στην αρχή, το τέλος και μια τυχαία ενδιάμεση χρονική στιγμή καθεμιάς από τις περιόδους που προσδιορίζει τα αντικείμενα Unit_Moving_Point τα οποία συνθέτουν το Moving_Point. Η f_trajectory() εξετάζει εάν οι σχεδιασμένες ενδιάμεσες συντεταγμένες 'πέφτουν' πάνω' στη γραμμή που δημιουργείται από τα άλλα δύο ζεύγη συντεταγμένων. Ανάλογα με το αποτέλεσμα, επιστρέφει ένα γραμμικό ή τοξωτό τμήμα που ενώνει τις αρχικές και τελικές σχεδιασμένες συντεταγμένες.

get_time_point (x, y): Η λειτουργία χρησιμοποιείται σε αντικείμενα Moving_Point που αποτελούνται από ένα Unit_Moving_Point. Δέχεται ως ορίσματα τις συντεταγμένες ενός σημείου x, y και ελέγχει εάν το σημείο αυτό συμπίπτει με το

σημείο έναρξης του Unit_Moving_Point του αντικειμένου Moving_Point. Εάν ισχύει αυτό, επιστρέφει τη χρονική στιγμή, που περιγράφεται από το αντικείμενο d_timepoint_sec,που αντιστοιχεί σε αυτό.

Για παράδειγμα, εάν έχω το Moving_Point της Εικόνας 3.7 και καλέσω για αυτό τη get_time_point (0.5, 1), η λειτουργία θα ελέγξει κατά πόσο το σημείο έναρξης του Moving_Point ταυτίζεται με το σημείο (0.5, 1), και εφόσον αυτό ισχύει θα μας επιστρέψει τη χρονική στιγμή t1.



Εικόνα 3.7 Προβολή ενός Moving_Point με ένα Unit_Moving_Point

ΚΕΦΑΛΑΙΟ 4

ΤΟ ΕΡΓΑΛΕΙΟ QUERY ENGINE

Σε αυτό το κεφάλαιο θα δούμε πώς με τη χρήση κάποιων αλγορίθμων, μπορούμε να κάνουμε επερωτήσεις στη βάση δεδομένων μας και ταυτόχρονα να διατηρήσουμε την ιδιωτικότητα των δεδομένων αυτών. Πιο συγκεκριμένα θα περιγράψουμε τους αλγορίθμους που υλοποιούν τις επερωτήσεις που υποστηρίζονται από την Query Engine μας. Τα υποστηριζόμενα είδη επερωτήσεων είναι τα ακόλουθα:

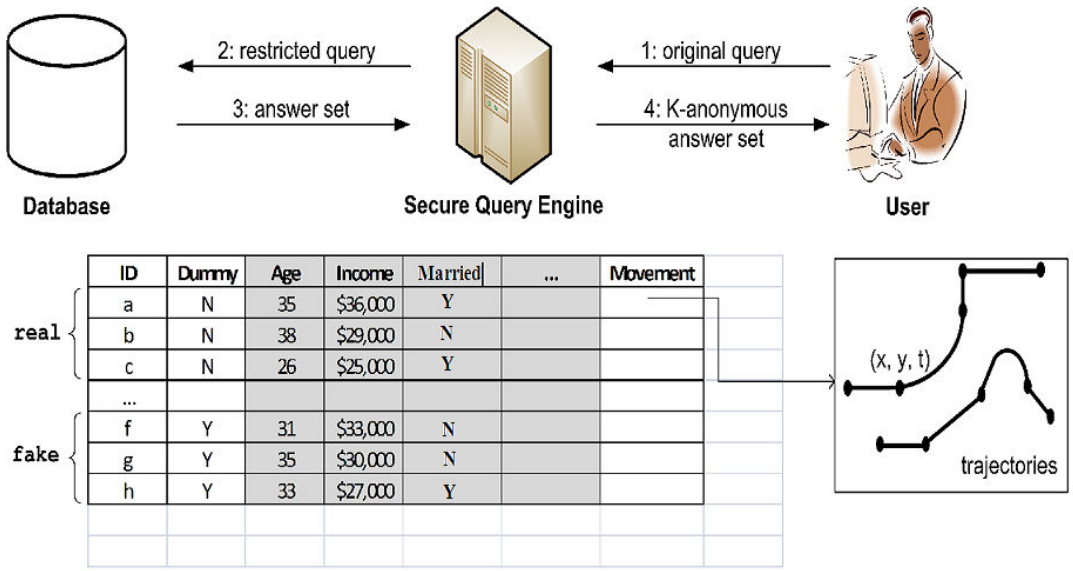
1. **Μη-χωρικές, μη-χρονικές επερωτήσεις**
   - επερωτήσεις καταμέτρησης (count queries),
   - συναθροιστικές επερωτήσεις (queries for aggregate statistics), και
   - κατηγορηματικές επερωτήσεις (queries for predicative data).

2. **Χωρο-χρονικές επερωτήσεις**
   - επερωτήσεις κάλυψης (range queries),
     - επερωτήσεις απόστασης (distance queries),
   - επερωτήσεις εγγύτερου γείτονα (nearest- neighbor queries),
   - επερωτήσεις σημείων ενδιαφέροντος (landmark queries),
   - επερωτήσεις συγκεκριμένων διαδρομών (route queries), και
   - επερωτήσεις κάλυψης-χρόνου (range-time queries).

4.1 Περιγραφή της Query Engine

Έστω ένα σύστημα που συλλέγει δεδομένα κίνησης για ένα μεγάλο πληθυσμό χρηστών. Κάθε χρήστης είναι εξοπλισμένος με μια συσκευή εντοπισμού γεωγραφικής θέσης (π.χ. GPS), η οποία σε συχνά χρονικά διαστήματα μεταδίδει τη θέση του (ενημέρωση θέσης) σε κάποιον έμπιστο εξυπηρετητή. Στον εξυπηρετητή υπάρχουν υλοποιημένες τεχνικές ανακατασκευής τροχιάς, με τις οποίες επιτυγχάνεται η προσέγγιση της πραγματικής τροχιάς που ακολούθησε ο χρήστης από την προηγούμενη ενημέρωση θέσης μέχρι την παρούσα θέση του. Έχοντας ανακατασκευάσει την τροχιά στην οποία κινήθηκε ο κάθε χρήστης, ο έμπιστος εξυπηρετητής ενημερώνει κατάλληλα τη βάση δεδομένων. Εκτός από τα δεδομένα κίνησης (τροχιές) των διαφόρων αντικειμένων, θεωρούμε ότι στη βάση δεδομένων βρίσκονται αποθηκευμένες και άλλες πληροφορίες που αφορούν τους χρήστες του συστήματος, όπως π.χ. ηλικία, εισόδημα, κ.α. Λόγω κάποιας κοινής συμφωνίας, ο κάτοχος των δεδομένων επιθυμεί να παρέχει πρόσβαση στη βάση δεδομένων σε τρίτους, μη έμπιστους φορείς. Προκειμένου να το πετύχει αυτό, αρχικά δημιουργεί μια προβολή της βάσης στην οποία περιέχονται μόνο οι εγγραφές και τα γνωρίσματα που θέλει να κοινοποιήσει. Για παράδειγμα, εάν τα δεδομένα πρόκειται να χρησιμοποιηθούν από κάποια διαφημιστική εταιρεία, η πληροφορία που αφορά το εισόδημα των ατόμων θα πρέπει να διατηρηθεί μυστική. Από την άλλη πλευρά, πληροφορία που αφορά την ηλικία των ατόμων των οποίων τα δεδομένα έχουν συλλεχθεί στη βάση μπορεί να αποδειχθεί εξαιρετικά σημαντική στο να επιτρέψει αποφάσεις σχετικά με το είδος των διαφημιζόμενων προϊόντων και άρα μπορεί να δοθεί. Ένα παράδειγμα προβολής βάσης που μπορεί να δημιουργηθεί από τον κάτοχο των δεδομένων για να εξυπηρετήσει συγκεκριμένους φορείς, παρουσιάζεται στην Εικόνα 4.1. Κάθε εγγραφή στη βάση δεδομένων εμπεριέχει πληροφορία που αφορά

ένα συγκεκριμένο χρήστη, μοναδικά προσδιοριζόμενο μέσω ενός ID. Η κίνηση του χρήστη καταγράφεται ως η συλλογή των τροχιών του στο σύστημα και αποθηκεύεται με τη βοήθεια της υποδομής που προσφέρει το σύστημα HERMES, ως ένα γνώρισμα στη βάση.



Εικόνα 4.1 Επισκόπηση της προτεινόμενης αρχιτεκτονικής για την Query Engine

Προκειμένου να χρησιμοποιήσει κανείς την QUERY ENGINE πρέπει πρώτα να εγγραφεί στο σύστημα και κατόπιν να συνδεθεί με τα στοιχεία του. Κατά τη σύνδεσή του, έχει τη δυνατότητα να υποβάλλει ένα σύνολο επερωτήσεων πάνω στην προβολή της βάσης που του έχει παραχωρηθεί από τον κάτοχο των δεδομένων κίνησης. Κατά την πρώτη είσοδο στο σύστημα, ο πίνακας των δεδομένων των χρηστών (όπως παρουσιάζεται στην Εικόνα 4.1), επεκτείνεται με ένα νέο γνώρισμα (αρχικοποιημένο στο «N» για όλες τις εγγραφές) που διατηρεί πληροφορία για το εάν η εκάστοτε εγγραφή του πίνακα αντιστοιχεί σε πραγματικό ή σε πλασματικό χρήστη. Επιπροσθέτως, ο κάτοχος των δεδομένων καθορίζει μία τιμή K που είναι

συνδεδεμένη με την προβολή της βάσης που έχει δοθεί και η οποία καθορίζει το ελάχιστο επιτρεπόμενο πλήθος εγγραφών που θα πρέπει να αντιστοιχούν στο αποτέλεσμα κάθε επερώτησης του χρήστη στη βάση προκειμένου να διασφαλιστεί η K– ανωνυμία. Ένα σύνολο διαδικασιών είναι διαθέσιμες στο χρήστη προκειμένου να θέσει επερωτήσεις στη βάση δεδομένων Όταν ο χρήστης υποβάλλει μια ερώτηση στην QUERY ENGINE μέσω της παρεχόμενης διεπαφής, το ερώτημα εξετάζεται αρχικά ως προς τη συμβατότητά του απέναντι σε ένα σύνολο από κανόνες που στοχεύουν στην προστασία των χρηστών που είναι αποθηκευμένοι στη βάση δεδομένων από επιθέσεις ταυτοποίησης ή/και παρακολούθησης. Σε περίπτωση που κάποιος από τους κανόνες δεν τηρείται, το σύστημα αρνείται την εξυπηρέτηση της επερώτησης. Διαφορετικά, εάν η επερώτηση είναι συμβατή με όλους τους κανόνες, το σύστημα αναζητά στη βάση δεδομένων την απάντηση στην επερώτηση που έθεσε ο χρήστης, διασφαλίζοντας ότι η απάντηση που θα επιστραφεί θα πληρεί τις απαιτήσεις της K– ανωνυμίας.

Επερωτήσεις καταμέτρησης καθώς επίσης και επερωτήσεις που αφορούν μη-χωρικά, μη-χρονικά δεδομένα (π.χ. «μέση ηλικία του καταγεγραμμένου πληθυσμού») απαντώνται με τη βοήθεια στρατηγικών περιορισμού και K– ανωνυμίας σε σχεσιακά δεδομένα. Από την άλλη πλευρά, σε επερωτήσεις που αφορούν χωρικά/χρονικά και χωροχρονικά δεδομένα, η QUERY ENGINE ανακαλύπτει αρχικά τις εγγραφές στη βάση δεδομένων που αποτελούν την πραγματική απάντηση. Στη συνέχεια, εάν το μέγεθος του συνόλου αυτού (γνωστό ως σύνολο ανωνυμίας) R είναι μικρότερο του K, τότε το σύστημα προχωρά στην κατασκευή K – R πλασματικών εγγραφών τις οποίες και αποθηκεύει στη βάση. Με τις πλασματικές εγγραφές το σύστημα επιτυγχάνει την διασφάλιση της K– ανωνυμίας σε δεδομένα τροχιών αφού για κάθε πλασματική

εγγραφή έχει οριστεί μια (αντίστοιχη) πλασματική πορεία κίνησης για τον «χρήστη». Η απάντηση στην επερώτηση του χρήστη παρέχεται ακολούθως από την συνένωση των πραγματικών με τις πλασματικές εγγραφές, έτσι ώστε το σύνολο απάντησης να περιέχει τουλάχιστον K εγγραφές (να είναι K– ανώνυμο). Από τη στιγμή της κατασκευής τους, οι πλασματικές εγγραφές διατηρούνται στη βάση δεδομένων μαζί με τις πραγματικές εγγραφές των χρηστών για λόγους συνέπειας της πληροφορίας: μελλοντικές επερωτήσεις στη βάση θα πρέπει να παράγουν απαντήσεις που θα είναι συμβατές με αυτές από προηγούμενες επερωτήσεις. Από την άλλη πλευρά, εάν το πλήθος των εγγραφών που αποτελούν την απάντηση σε κάποια επερώτηση είναι επαρκές για την παροχή K– ανωνυμίας (δηλ. εάν $R \geq K$) τότε η απάντηση επιστρέφεται στο χρήστη χωρίς την επαύξηση του συνόλου με πλασματική πληροφορία. Σε κάθε περίπτωση, παράλληλα με την απάντηση στην επερώτηση του χρήστη, το QUERY ENGINE παρέχει πληροφορία σχετικά με το ποσοστό των πραγματικών εγγραφών στην απάντηση που επιστρέφεται. Το γεγονός αυτό δίνει τη δυνατότητα στον χρήστη να ενημερωθεί για το πόσο ακριβή είναι τα αποτελέσματα που έλαβε.

4.2 Υποστηριζόμενοι τύποι επερωτήσεων σε μη χωρο-χρονικά δεδομένα.

4.2.1 Επερωτήσεις καταμέτρησης (count queries)

Στις επερωτήσεις καταμέτρησης το ζητούμενο είναι να ανακαλυφθεί ο αριθμός των χρηστών που ικανοποιούν ένα συγκεκριμένο περιορισμό. Παρακάτω παραθέτουμε ορισμένα παραδείγματα επερωτήσεων καταμέτρησης:

Q1: «Να βρεθεί ο αριθμός των χρηστών, των οποίων η ηλικία είναι μεγαλύτερη των 30 χρόνων»

Q2: «Να βρεθεί ο αριθμός των χρηστών, των οποίων το ετήσιο εισόδημα είναι μεταξύ 30-50 χιλιάδες ευρώ»

Περιγραφή αλγορίθμου

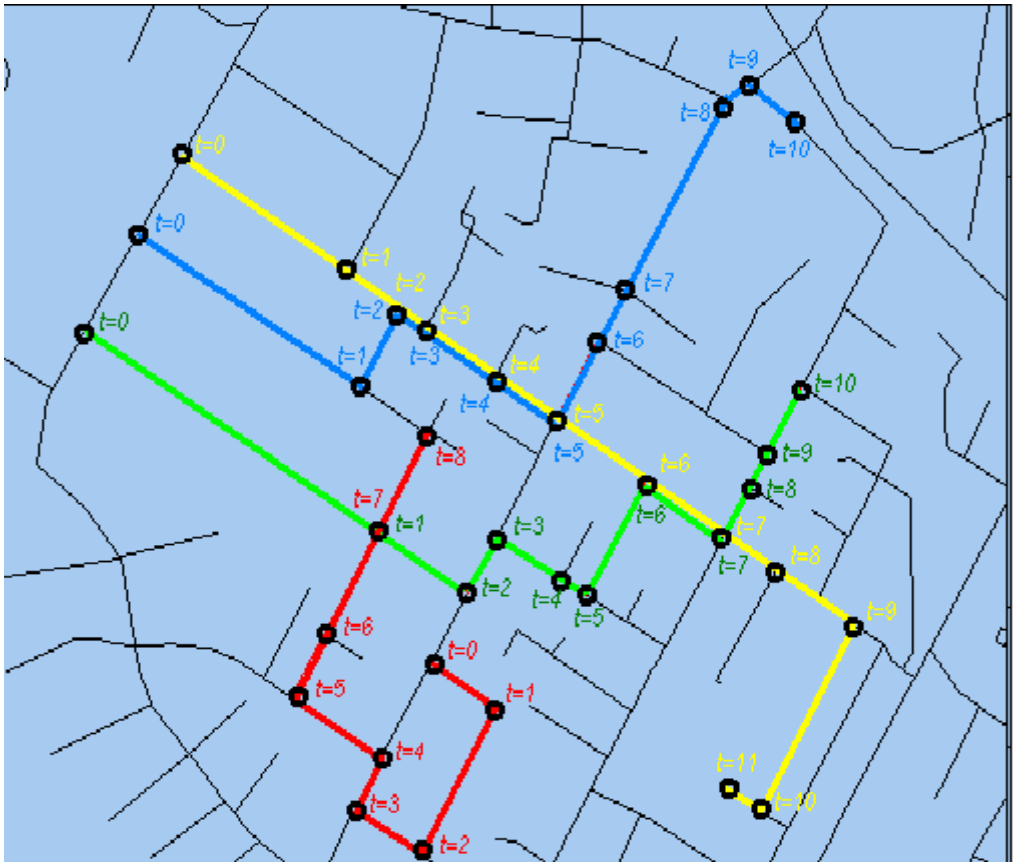
Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

- Ο περιορισμός **cond** με βάση τον οποίο θα γίνει η επιλογή των χρηστών που τον ικανοποιούν.
- Η σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας, δηλαδή τον αριθμό των χρηστών που θα πρέπει να βρίσκονται εντός των περιορισμών, ώστε ο αλγόριθμος να επιτύχει. Η σταθερά αυτή καθορίζεται από το χρήστη και από αυτή εξαρτάται το πόσο ασφαλής θα είναι κάποια αίτηση. Όσο μεγαλύτερη είναι η τιμή του **K** τόσο μεγαλύτερη θα είναι και η ασφάλεια της ιδιωτικότητας του αιτούντα.

Ο αλγόριθμος ζητά από τη βάση δεδομένων να του επιστρέψει τον αριθμό των χρηστών που ικανοποιούν τον περιορισμό που δώσαμε. Εάν ο αριθμός που επιστρέφεται είναι μικρότερος από την σταθερά K, δηλαδή η K– ανωνυμία δεν μπορεί να επιτευχθεί, το σύστημα αρνείται την απάντηση της επερώτησης. Σε αντίθετη περίπτωση ο αλγόριθμος επιστρέφει κανονικά τον ζητούμενο αριθμό.

Παράδειγμα εκτέλεσης αλγορίθμου

Θεωρούμε ότι στη βάση δεδομένων μας έχουμε αποθηκευμένες τις τροχιές τεσσάρων χρηστών (βλ. Εικόνα 4.2). Στην τροχιά με κίτρινο χρώμα απεικονίζεται η κίνηση του χρήστη με id=1, με μπλε χρώμα η κίνηση του χρήστη με id=2, με πράσινο χρώμα η κίνηση του χρήστη με id=3, και με κόκκινο χρώμα η κίνηση του χρήστη με id=4. Πάνω σε κάθε τροχιά σημειώνεται με κύκλο ο κόμβος, και η χρονική στιγμή που πέρασε από αυτόν ο χρήστης. Για λόγους ευκολίας ο χρόνος δεν σημειώνεται με τη μορφή "Έτος-Μήνας-Ημέρα Ώρα:Λεπτά:Δευτερόλεπτα", αλλά θεωρούμε ότι η στιγμή t=0 αντιστοιχεί στη χρονική στιγμή "2009-08-26 08:00:00" , η στιγμή t=1 στη χρονική στιγμή "2009-08-26 08:01:00" κ.ο.κ.



Εικόνα 4.2 Τροχιές των χρηστών

Η προβολή της βάσης για τα μη χωροχρονικά δεδομένα παρουσιάζεται στην Εικόνα 4.3.

ID	Dummy	Age	Income	Married
1	N	35	40.000 €	N
2	N	25	14.000 €	N
3	N	46	51.000 €	Y
4	N	40	25.000 €	Y

Εικόνα 4.3 Προβολή βάσης για τα μη χωροχρονικά δεδομένα

Έστω ότι ένας χρήστης κάνει την επερώτηση: «Ποιος είναι ο αριθμός των χρηστών, των οποίων η ηλικία είναι μεγαλύτερη των 25 χρόνων». Και έστω K=3, ο βαθμός ανωνυμίας που θέλει να έχει. Αρχικά ο αλγόριθμος ζητά από την βάση τον αριθμό των χρηστών με ηλικία μεγαλύτερη του 25, και λαμβάνει ως απάντηση, όπως μπορεί να συμπεράνει κανείς και από την Εικόνα 4.3, το 3. Έπειτα ελέγχει εάν το 3 είναι μεγαλύτερο ή ίσο του K, και αφού αυτό ισχύει επιστρέφει ως απάντηση το 3.Εάν όμως ο βαθμός ανωνυμίας ήταν 4, ο αλγόριθμος θα επέστρεφε το μήνυμα "K-anonymity violation", καθώς υπάρχουν μόνο 3 χρήστες που πληρούν τις προϋποθέσεις.

4.2.2 Συναθροιστικές επερωτήσεις (queries for aggregate statistics)

Στις συναθροιστικές επερωτήσεις το ζητούμενο είναι να ανακαλυφθεί ένα στατιστικό δεδομένο (AVG,SUM,MAX,MIN) των γνωρισμάτων των χρηστών, που ικανοποιούν ένα συγκεκριμένο περιορισμό. Παρακάτω παραθέτουμε ορισμένα παραδείγματα συναθροιστικών επερωτήσεων:

Q1: «Να βρεθεί ο μέσος όρος του εισοδήματος των χρηστών με ηλικία μεγαλύτερη των 40 ετών»

Q2: «Να βρεθεί ο μικρότερος ηλικιακά χρήστης με ετήσιο εισόδημα 30000 ευρώ»

Περιγραφή αλγορίθμου

Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

- Το αλφαριθμητικό **func**, το οποία προσδιορίζει μία SQL aggregate function (**AVG,SUM,MAX,MIN**).
- Το αλφαριθμητικό **attr**, το οποίο προσδιορίζει το γνώρισμα για το οποίο θα γίνει η κλήση της SQL aggregate function.
- Τον περιορισμό **cond** με βάση τον οποίο θα γίνει η επιλογή των χρηστών που τον ικανοποιούν.
- Την σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας, δηλαδή τον αριθμό των χρηστών που θα πρέπει να βρίσκονται εντός των περιορισμών, ώστε ο αλγόριθμος να επιτύχει. Η σταθερά αυτή καθορίζεται από το χρήστη και από αυτή εξαρτάται το πόσο ασφαλής θα είναι κάποια αίτηση. Όσο μεγαλύτερη είναι η τιμή του **K** τόσο μεγαλύτερη θα είναι και η ασφάλεια της ιδιωτικότητας του αιτούντα.

Ο αλγόριθμος ζητά από τη βάση δεδομένων να του επιστρέψει ένα στατιστικό δεδομένο πάνω σε κάποιο γνώρισμα του χρήστη με βάση έναν περιορισμό, και τον αριθμό των χρηστών που ικανοποιούν αυτόν τον περιορισμό. Εάν ο αριθμός των χρηστών που επιστρέφεται είναι μικρότερος από την σταθερά K, δηλαδή η K–ανωνυμία δεν μπορεί να επιτευχθεί, το σύστημα αρνείται την απάντηση της

επερώτησης. Σε αντίθετη περίπτωση ο αλγόριθμος επιστρέφει κανονικά το ζητούμενο στατιστικό δεδομένο.

Παράδειγμα εκτέλεσης αλγορίθμου

Θεωρούμε ότι στη βάση μας έχουμε αποθηκευμένες τις τροχιές τεσσάρων χρηστών, όπως αυτές φαίνονται στην Εικόνα 4.2 και ότι η προβολής της βάσης για τα μη χωροχρονικά δεδομένα είναι αυτή που παρουσιάζεται στην Εικόνα 4.3.

Έστω ότι ένας χρήστης κάνει την επερώτηση: «Ποιος είναι ο μεγαλύτερος ηλικιακά χρήστης με ετήσιο εισόδημα μεγαλύτερο των 30000 ευρώ». Και έστω K=2, ο βαθμός ανωνυμίας που θέλει να έχει. Αρχικά ο αλγόριθμος ζητά από την βάση τον αριθμό των χρηστών που έχουν εισόδημα μεγαλύτερο των 30000 ευρώ, και την μέγιστη ηλικία αυτών. Το αποτέλεσμα που επιστρέφεται είναι 2 και 46 αντίστοιχα. Εφόσον το 2 είναι ίσο του K, επιστρέφεται η απάντηση 46. Στην περίπτωση όμως που το K ήταν μεγαλύτερο του 2, ο αλγόριθμος θα επέστρεφε το μήνυμα "K- anonymity violation"

4.2.3 Κατηγορηματικές επερωτήσεις (queries for predicative data)

Στις κατηγορηματικές επερωτήσεις το ζητούμενο είναι να ανακαλυφθεί ο αριθμός των ατόμων που έχουν μια συγκεκριμένη τιμή σε κάποιο κατηγορηματικό δεδομένο της βάσης. Παρακάτω παραθέτουμε ορισμένα παραδείγματα επερωτήσεων καταμέτρησης:

Q1: «Να βρεθεί ο αριθμός των χρηστών που είναι παντρεμένοι»

Q2: «Να βρεθεί ο αριθμός των χρηστών που είναι ελεύθεροι»

Περιγραφή αλγορίθμου

Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

- Ο περιορισμός **cond** με βάση τον οποίο θα γίνει η επιλογή των χρηστών που τον ικανοποιούν.
- Η σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας, δηλαδή τον αριθμό των χρηστών που θα πρέπει να βρίσκονται εντός των περιορισμών, ώστε ο αλγόριθμος να επιτύχει. Η σταθερά αυτή καθορίζεται από το χρήστη και από αυτή εξαρτάται το πόσο ασφαλής θα είναι κάποια αίτηση. Όσο μεγαλύτερη είναι η τιμή του **K** τόσο μεγαλύτερη θα είναι και η ασφάλεια της ιδιωτικότητας του αιτούντα.

Ο αλγόριθμος ζητά από τη βάση δεδομένων να του επιστρέψει τον αριθμό των χρηστών που ικανοποιούν τον περιορισμό που δώσαμε. Εάν ο αριθμός που επιστρέφεται είναι μικρότερος από την σταθερά K, δηλαδή η K– ανωνυμία δεν μπορεί να επιτευχθεί, το σύστημα αρνείται την απάντηση της επερώτησης. Σε αντίθετη περίπτωση ο αλγόριθμος επιστρέφει κανονικά τον ζητούμενο αριθμό.

Παράδειγμα εκτέλεσης αλγορίθμου

Θεωρούμε ότι στη βάση μας έχουμε αποθηκευμένες τις τροχιές τεσσάρων χρηστών, όπως αυτές φαίνονται στην Εικόνα 4.2 και ότι η προβολής της βάσης για τα μη χωροχρονικά δεδομένα είναι αυτή που παρουσιάζεται στην Εικόνα 4.3.

Έστω ότι ένας χρήστης κάνει την επερώτηση: «Να βρεθεί ο αριθμός των χρηστών που είναι παντρεμένοι». Και έστω K=2, ο βαθμός ανωνυμίας που θέλει να έχει. Αρχικά ο αλγόριθμος ζητά από την βάση τον αριθμό των χρηστών που έχουν στο γνώρισμα Married την τιμή 'Y'. Το αποτέλεσμα που επιστρέφεται είναι 2. Εφόσον το 2 είναι ίσο του K, επιστρέφεται ως απάντηση. Στην περίπτωση όμως που το K ήταν μεγαλύτερο του 2, ο αλγόριθμος θα επέστρεφε το μήνυμα "K- anonymity violation"

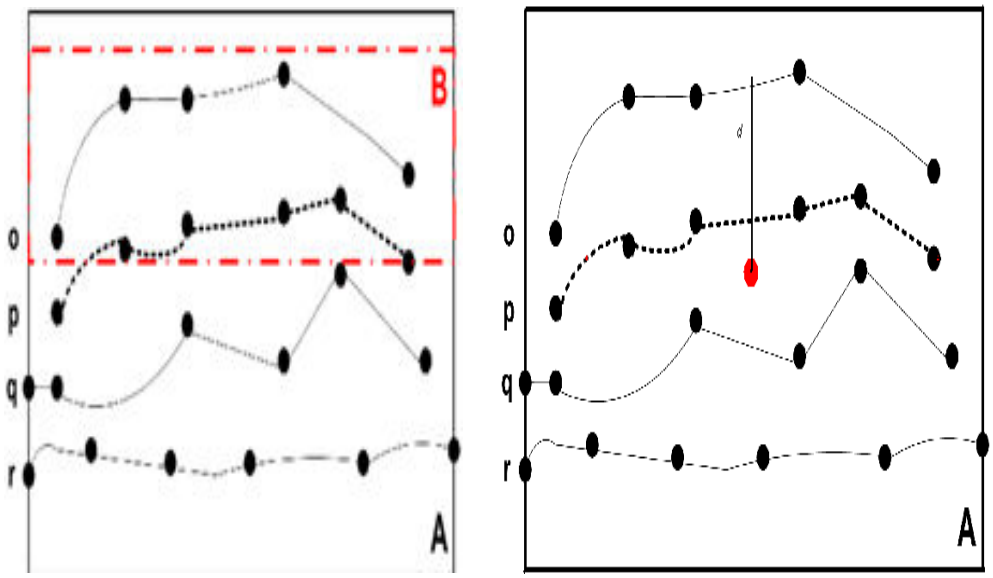
4.3 Υποστηριζόμενοι τύποι επερωτήσεων σε χωρο-χρονικά δεδομένα.

4.3.1 Επερωτήσεις κάλυψης (range queries)

Στις επερωτήσεις κάλυψης το ζητούμενο είναι να ανακαλυφθούν οι τροχιές των ατόμων που είναι αποθηκευμένες στη βάση, οι οποίες είτε (α) βρίσκονται εντός μιας προκαθορισμένης χωρο-χρονικής περιοχής, είτε (β) είναι εντός προκαθορισμένης απόστασης d από κάποιο σημείο αναφοράς. Παρακάτω παραθέτουμε ορισμένα παραδείγματα επερωτήσεων κάλυψης (βλ. Εικόνα 4.3):

Q1: «Να βρεθούν όλες οι τροχιές των χρηστών που διέσχισαν το πεδίο του Άρεως αυτή τη Δευτέρα, από τις 6πμ μέχρι τις 10πμ»

Q2: «Να βρεθούν όλες οι τροχιές των χρηστών που είναι το πολύ σε απόσταση ενός χιλιομέτρου από το Τμήμα Πληροφορικής του Πανεπιστημίου Ιωαννίνων αυτή τη στιγμή»



Range_query Distance_quey

Εικόνα 4.3 Η Rang_query και Distance_query πάνω σε δεδομένα κίνησης – τροχιές.

Περιγραφή αλγορίθμου range_query

Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

- Το ψευδώνυμο **Id** του αιτούντα.
- Την περιοχή **Sgeo** που μας ενδιαφέρει.
- Τη χρονιά **YearStart**, μήνα **MonthStart**, ημέρα **DayStart**, ώρα **HourStart**, λεπτό **MinStart**, και δευτερόλεπτο **SecStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Τη χρονιά **YearEnd**, μήνα **MonthEnd**, ημέρα **DayEnd**, ώρα **HourEnd**, λεπτό **MinEnd**, και δευτερόλεπτο **SecEnd** λήξης της περιόδου που μας ενδιαφέρει.

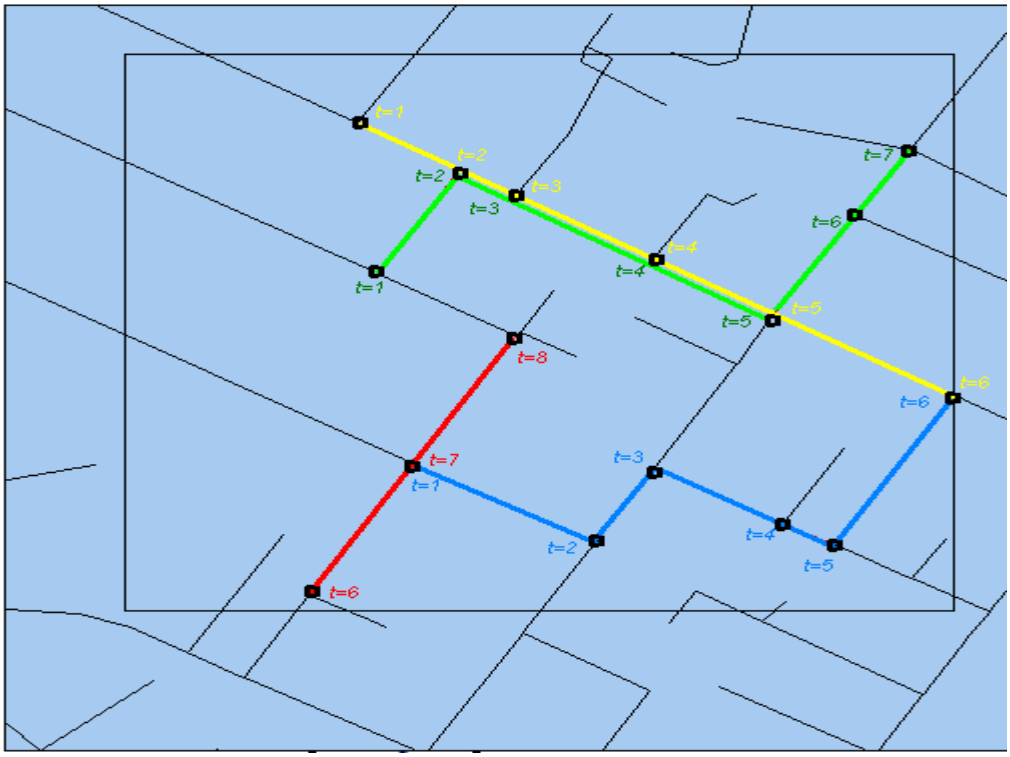
• Την σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας, δηλαδή τον αριθμό των χρηστών που θα πρέπει να βρίσκονται εντός των περιορισμών, ώστε ο αλγόριθμος να επιτύχει.

Αρχικά ο αλγόριθμος ελέγχει εάν η επερώτηση αποτελεί στοχευμένη επίθεση κάποιου χρήστη. Για να το πετύχει αυτό, καλεί την συνάρτηση In_Hist (αναλυτική περιγραφή στην Ενότητα 5.3), η οποία ελέγχει εάν ο συγκεκριμένος χρήστης έχει κάνει στο παρελθόν επερωτήσεις στις οποίες οι παλαιότερες περιοχές ενδιαφέροντος επικαλύπτονται ή γειτνιάζουν με την τωρινή περιοχή ενδιαφέροντος. Αυτό επιτυγχάνεται διατηρώντας σε έναν πίνακα hist το ψευδώνυμο, την περιοχή και το χρονικό διάστημα ενδιαφέροντος κάθε χρήστη. Εάν ανακαλυφθεί ότι αποτελεί στοχευμένη επίθεση ο αλγόριθμος επιστρέφει το μήνυμα “Privacy threat” και τερματίζει τη λειτουργία του. Διαφορετικά ζητά από τη βάση δεδομένων να του επιστρέψει τις τροχιές των χρηστών οι οποίες διασχίζουν την περιοχή που μας ενδιαφέρει. Στη συνέχεια ελέγχεται εάν οι τροχιές που επιστρέφονται αντιστοιχούν σε αριθμό χρηστών μεγαλύτερο ή ίσο του K. Εάν ισχύει αυτό, παρουσιάζονται στον χρήστη οι τροχιές. Εάν ο αριθμός των χρηστών, έστω R, είναι μικρότερος του K, καλείται η διαδικασία Fake_Gen η οποία παράγει K-R πλασματικές τροχιές. Έπειτα ανανεώνονται τα δεδομένα που υπάρχουν μέσα στον πίνακα hist και ο αλγόριθμος παρουσιάζει τις τροχιές στο χρήστη (πλασματικές και μη).

Παράδειγμα εκτέλεσης αλγορίθμου

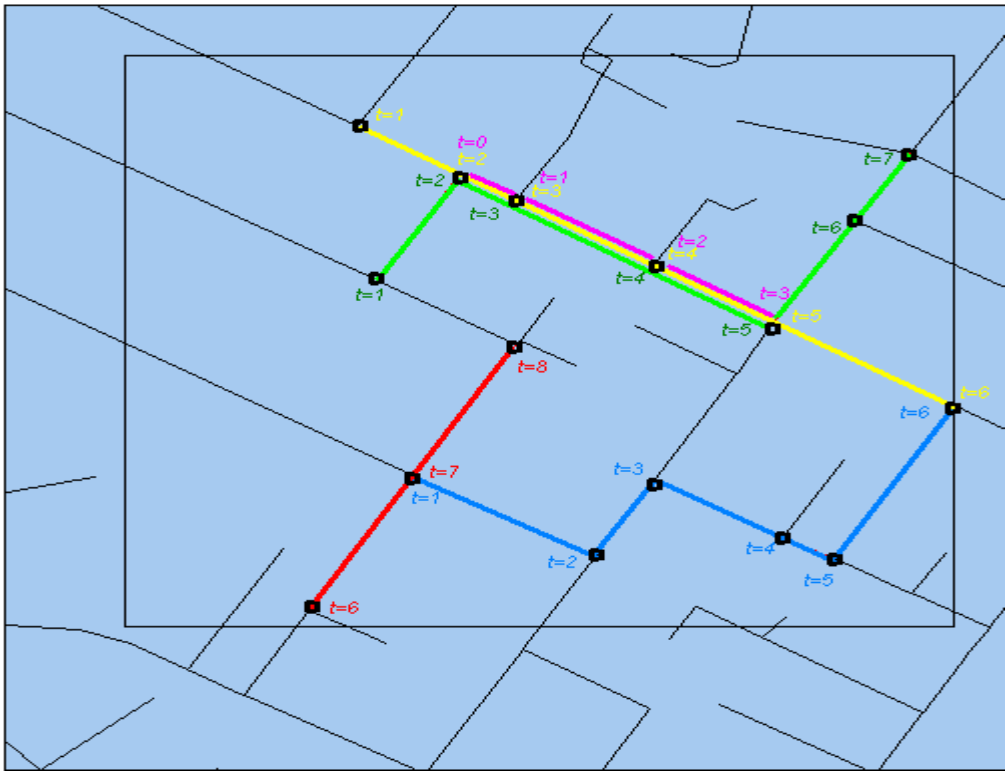
Θεωρούμε ότι στη βάση μας έχουμε αποθηκευμένες τις τροχιές τεσσάρων χρηστών, όπως αυτές φαίνονται στην Εικόνα 4.2 και ότι η προβολής της βάσης για τα μη χωροχρονικά δεδομένα είναι αυτή που παρουσιάζεται στην Εικόνα 4.3.

Έστω ότι ο χρήστης με id=3 κάνει την επερώτηση: «Να βρεθούν όλες οι τροχιές των χρηστών που διέσχισαν το πεδίο του Άρεως την Τετάρτη 26 Αυγούστου 2009, από τις 7πμ μέχρι τις 10πμ». Και έστω K=4, ο βαθμός ανωνυμίας που θέλει να έχει. Στην Εικόνα 4.4 σημειώνεται με ένα ορθογώνιο η περιοχή που θεωρούμε ως πεδίο του ‘Άρεως. Αρχικά ο αλγόριθμος ελέγχει εάν ο χρήστης με id=3 έχει κάνει παλαιότερη επερώτηση, η περιοχή ενδιαφέροντος της οποίας γειτνιάζει ή επικαλύπτεται με την περιοχή του πεδίου του Άρεως. Επειδή κάτι τέτοιο δεν ισχύει, ζητά από τη βάση το τμήμα των τροχιών των χρηστών που πέρασαν από το πεδίο του Άρεως την Τετάρτη 26 Αυγούστου του 2009 μεταξύ 7πμ και 10πμ. Έπειτα συγκρίνει τον αριθμό των χρηστών, στους οποίους αντιστοιχούν αυτά τα τμήματα των τροχιών, με τη σταθερά K. Ο αριθμός είναι 4, δηλαδή ίσος του K, άρα ο αλγόριθμος επιστρέφει στον χρήστη τις “υπο-τροχιές” φαίνονται στην Εικόνα 4.4 .



Εικόνα 4.4 Αποτέλεσμα range query για K=4

Εάν ο χρήστης όμως απαιτούσε K- ανωνυμία 5, ο αριθμός των "υπο-τροχιών" – που όπως είπαμε είναι 4 – θα ήταν μικρότερος του K. Σε αυτή την περίπτωση ο αλγόριθμος καλεί την συνάρτηση Fake_Gen(M), με M=1, καθώς πρέπει να παραχθεί μία πλασματική τροχιά για να καλυφθεί ο περιορισμός της K- ανωνυμίας. Έπειτα γίνεται ενημέρωση του πίνακα που διατηρεί το ιστορικό κάθε χρήστη για τις επερωτήσεις που έχει κάνει στο σύστημα, και έτσι εισάγεται σε αυτόν μία νέα εγγραφή με περιοχή ενδιαφέροντος το πεδίο του Άρεως και id χρήστη το 3. Τέλος, ένα πιθανό αποτέλεσμα που θα παρουσιαζόταν στον χρήστη φαίνεται στην Εικόνα 4.5 (η πλασματική τροχιά αναπαρίσταται με ροζ χρώμα).



Εικόνα 4.5 Αποτέλεσμα range query για K=5, με μια πλασματική τροχιά

Περιγραφή αλγορίθμου distance_query

Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

- Το ψευδώνυμο **Id** του αιτούντα.
- Την οριζόντια συντεταγμένη **Xp** του προκαθορισμένου σημείου ενδιαφέροντος.
- Την κατακόρυφη συντεταγμένη **Yp** του προκαθορισμένου σημείου ενδιαφέροντος.
- Την μέγιστη απόσταση **d** που θέλουμε να απέχουν οι τροχιές από το προκαθορισμένο σημείο ενδιαφέροντος.

•Τη χρονιά **YearStart**, μήνα **MonthStart**, ημέρα **DayStart**, ώρα **HourStart**, λεπτό **MinStart**, και δευτερόλεπτο **SecStart** έναρξης της περιόδου που μας ενδιαφέρει.

•Τη χρονιά **YearEnd**, μήνα **MonthEnd**, ημέρα **DayEnd**, ώρα **HourEnd**, λεπτό **MinEnd**, και δευτερόλεπτο **SecEnd** λήξης της περιόδου που μας ενδιαφέρει.

•Τη σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας, δηλαδή τον αριθμό των χρηστών που θα πρέπει να βρίσκονται εντός των περιορισμών, ώστε ο αλγόριθμος να επιτύχει.

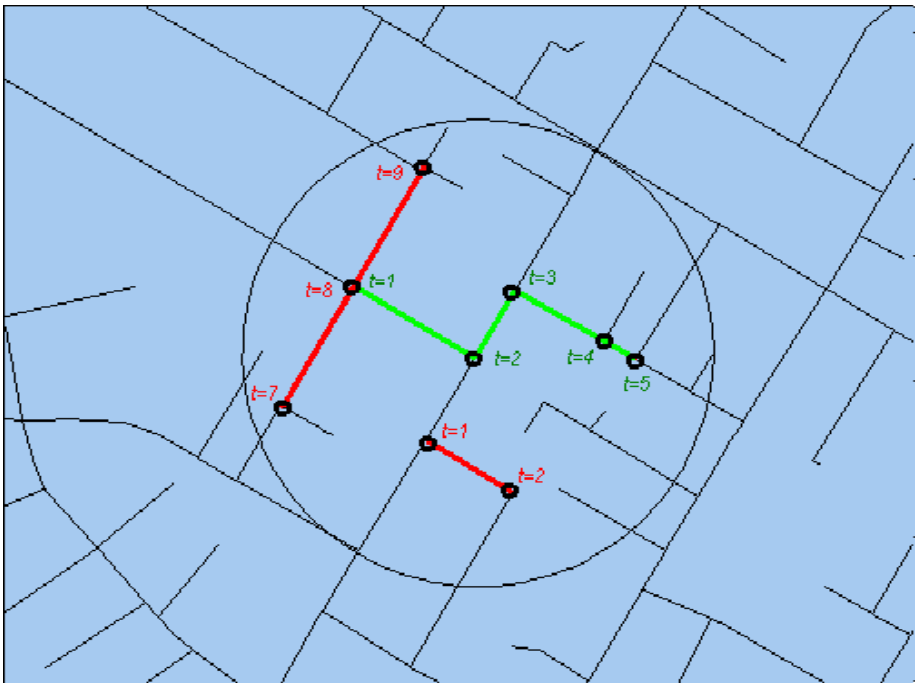
Ο αλγόριθμος αφού δημιουργήσει έναν κύκλο με κέντρο τα σημεία Xp, Yp και ακτίνα την απόσταση d, καλεί τον αλγόριθμο range_qurey για να του βρει τις τροχιές που διασχίζουν την περιοχή του κύκλου αυτού.

Παράδειγμα εκτέλεσης αλγορίθμου

Θεωρούμε ότι στη βάση μας έχουμε αποθηκευμένες τις τροχιές τεσσάρων χρηστών, όπως αυτές φαίνονται στην Εικόνα 4.2 και ότι η προβολής της βάσης για τα μη χωροχρονικά δεδομένα είναι αυτή που παρουσιάζεται στην Εικόνα 4.3.

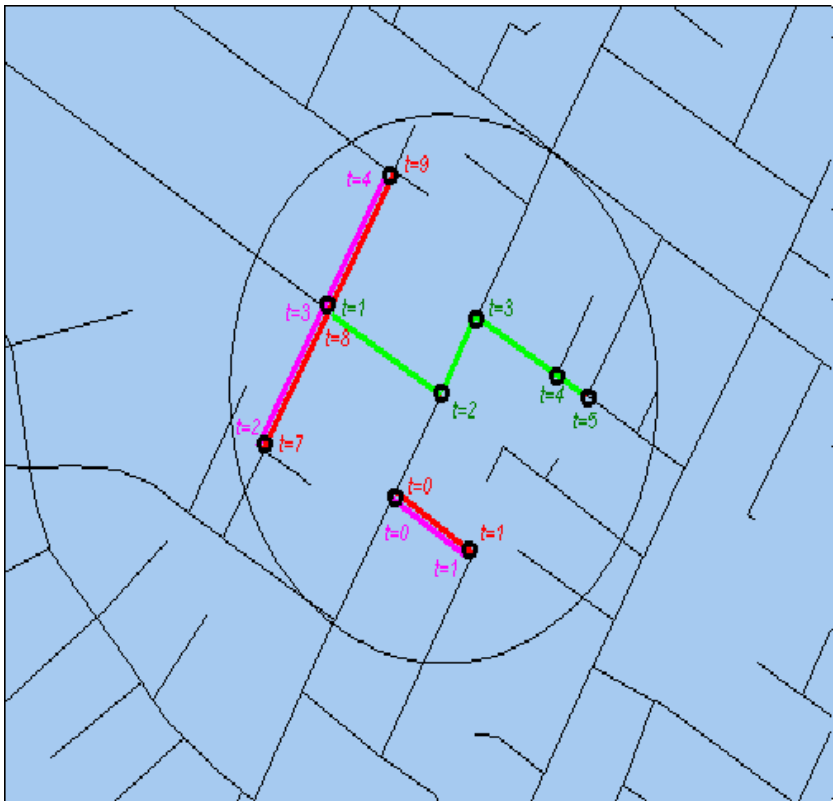
Έστω ότι ο χρήστης με id=1 κάνει την επερώτηση «Να βρεθούν όλες οι τροχιές των χρηστών που ήταν το πολύ σε απόσταση τριών χιλιομέτρων από το Τμήμα Πληροφορικής του Πανεπιστημίου Ιωαννίνων την Τετάρτη 26 Αυγούστου 2009, από τις 7πμ μέχρι τις 10πμ». Και έστω K=2, ο βαθμός ανωνυμίας που θέλει να έχει. Στην

Εικόνα 4.6 σημειώνεται με κύκλο η περιοχή που θεωρούμε ότι βρίσκεται εντός τριών χιλιομέτρων από το Τμήμα Πληροφορικής του Πανεπιστημίου Ιωαννίνων. Πρώτο βήμα του αλγορίθμου είναι να δημιουργήσει έναν κύκλο με κέντρο τις συντεταγμένες (x, y) του Τμήματος Πληροφορικής του Πανεπιστημίου Ιωαννίνων , και ακτίνα τα 3 χιλιόμετρα. Έπειτα καλείται η συνάρτηση Range Query με περιοχή ενδιαφέροντος τον κύκλο που δημιουργήθηκε. Η συνάρτηση Range Query εκτελείται όπως και παραπάνω. Αρχικά ο αλγόριθμος ελέγχει εάν ο χρήστης με id=1 έχει κάνει παλαιότερη επερώτηση, η περιοχή ενδιαφέροντος της οποίας γειτνιάζει ή επικαλύπτεται με τον κύκλο που δημιουργήσαμε. Επειδή κάτι τέτοιο δεν ισχύει, ζητά από τη βάση το τμήμα των τροχιών των χρηστών που πέρασαν από τον κύκλο την Τετάρτη 26 Αυγούστου του 2009 μεταξύ 7πμ και 10πμ. Έπειτα συγκρίνει τον αριθμό των χρηστών, στους οποίους αντιστοιχούν αυτά τα τμήματα των τροχιών, με τη σταθερά K. Ο αριθμός είναι 2, δηλαδή ίσος του K, άρα ο αλγόριθμος επιστρέφει στον χρήστη τις "υπο-τροχιές" φαίνονται στην Εικόνα 4.6 .



Εικόνα 4.6 Αποτέλεσμα distance query για K=2

Εάν ο χρήστης όμως απαιτούσε K- ανωνυμία 3, ο αριθμός των “υπο-τροχιών” – που όπως είπαμε είναι 2 –θα ήταν μικρότερος του K. Σε αυτή την περίπτωση ο αλγόριθμος καλεί την συνάρτηση Fake_Gen(M), με M=1, καθώς πρέπει να παραχθεί μία πλασματική τροχιά για να καλυφθεί ο περιορισμός της K- ανωνυμίας. Έπειτα γίνεται ενημέρωση του πίνακα που διατηρεί το ιστορικό κάθε χρήστη για τις επερωτήσεις που έχει κάνει στο σύστημα, και έτσι εισάγεται σε αυτόν μία νέα εγγραφή με περιοχή ενδιαφέροντος τον παραγόμενο κύκλο και id χρήστη το 1. Τέλος, ένα πιθανό αποτέλεσμα που θα παρουσιαζόταν στον χρήστη φαίνεται στην Εικόνα 4.7 (η πλασματική τροχιά αναπαρίσταται με ροζ χρώμα).



Εικόνα 4.7 Αποτέλεσμα distance query για K=3, με μια πλασματική τροχιά

4.3.2 Επερωτήσεις εγγύτερου γείτονα (nearest- neighbor queries),

Στις επερωτήσεις εγγύτερου γείτονα, ο στόχος είναι η εύρεση των κ πλησιέστερων (K–NN) γειτόνων για μια χρονική περίοδο με βάση κάποιο σημείο αναφοράς. Ένα παράδειγμα τέτοιου είδους επερώτησης είναι η ακόλουθη:

Q1: «Να βρεθούν οι πέντε τροχιές που είναι πλησιέστερες στο πεδίο του Άρεως αυτή τη στιγμή»

Περιγραφή αλγορίθμου KNN_query

Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

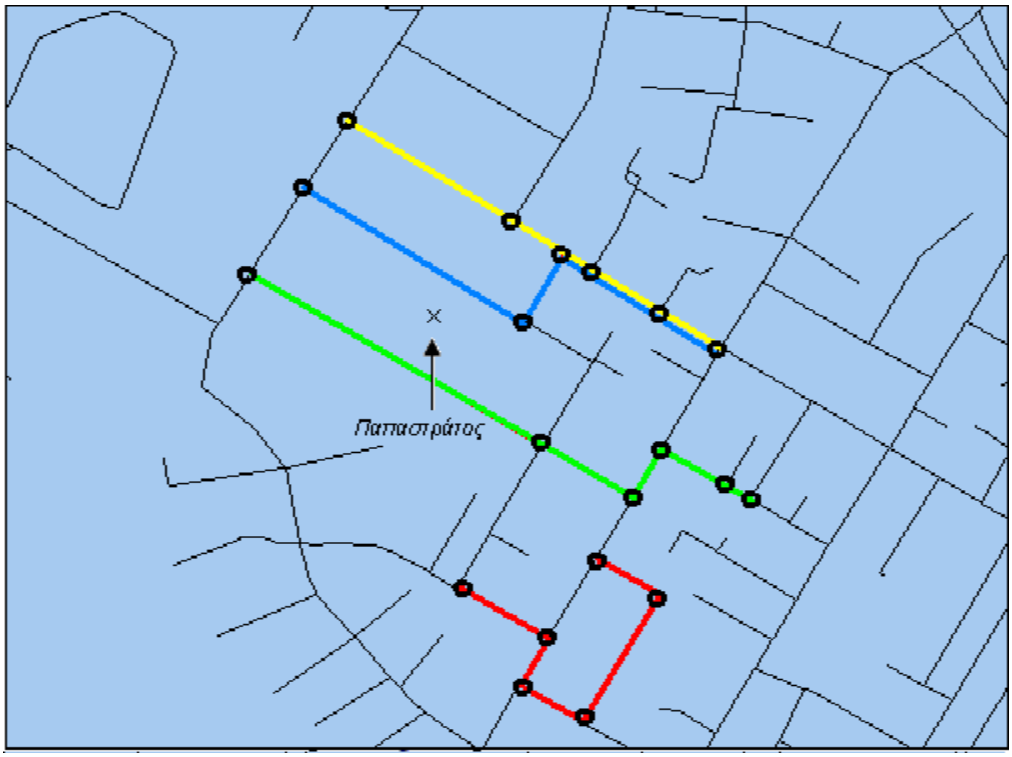
- Το ψευδώνυμο **Id** του αιτούντα.
- Την σημείο αναφοράς **P**.
- Τη χρονιά **YearStart**, μήνα **MonthStart**, ημέρα **DayStart**, ώρα **HourStart**, λεπτό **MinStart**, και δευτερόλεπτο **SecStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Τη χρονιά **YearEnd**, μήνα **MonthEnd**, ημέρα **DayEnd**, ώρα **HourEnd**, λεπτό **MinEnd**, και δευτερόλεπτο **SecEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Την σταθερά **k** που καθορίζει τον αριθμό των γειτόνων.
- Τη σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας, δηλαδή τον αριθμό των χρηστών που θα πρέπει να βρίσκονται εντός των περιορισμών, ώστε ο αλγόριθμος να επιτύχει.

Ο αλγόριθμος ζητά από τη βάση δεδομένων να του επιστρέψει τις τροχιές των κ κοντινότερων χρηστών με βάση κάποιο σημείο αναφοράς. Στη συνέχεια ελέγχεται εάν οι τροχιές που επιστρέφονται αντιστοιχούν σε αριθμό χρηστών μεγαλύτερο ή ίσο του K. Εάν ισχύει αυτό, παρουσιάζονται στον χρήστη οι τροχιές. Εάν ο αριθμός των χρηστών, έστω R, είναι μικρότερος του K, καλείται η διαδικασία Fake_Gen η οποία παράγει K-R πλασματικές τροχιές. Έπειτα ελέγχει εάν η επερώτηση αποτελεί στοχευμένη επίθεση κάποιου χρήστη. Για να το πετύχει αυτό, βρίσκει το κουτί οριοθέτησης (MBR) των τροχιών που επιστράφηκαν και καλεί την συνάρτηση In_Hist. Η συνάρτηση In_Hist ελέγχει εάν ο συγκεκριμένος χρήστης έχει κάνει στο παρελθόν επερωτήσεις στις οποίες οι παλαιότερες περιοχές ενδιαφέροντος επικαλύπτονται ή γειτνιάζουν με το MBR που υπολογίσαμε. Αυτό επιτυγχάνεται διατηρώντας σε έναν πίνακα hist το ψευδώνυμο, την περιοχή και το χρονικό διάστημα ενδιαφέροντος κάθε χρήστη. Εάν ανακαλυφθεί ότι αποτελεί στοχευμένη επίθεση ο αλγόριθμος επιστρέφει το μήνυμα “Privacy threat” και τερματίζει τη λειτουργία του. Διαφορετικά ελέγχει εάν οι τροχιές που επιστράφηκαν αντιστοιχούν σε αριθμό χρηστών μεγαλύτερο ή ίσο του K. Εάν ισχύει αυτό, παρουσιάζονται στον χρήστη οι τροχιές. Εάν ο αριθμός των χρηστών, έστω R, είναι μικρότερος του K, καλείται η διαδικασία Fake_Gen η οποία παράγει K-R πλασματικές τροχιές. Έπειτα ανανεώνονται τα δεδομένα που υπάρχουν μέσα στον πίνακα hist και ο αλγόριθμος παρουσιάζει τις τροχιές των χρηστών (πλασματικές και μη).

Παράδειγμα εκτέλεσης αλγορίθμου

Θεωρούμε ότι στη βάση μας έχουμε αποθηκευμένες τις τροχιές τεσσάρων χρηστών, όπως αυτές φαίνονται στην Εικόνα 4.2 και ότι η προβολής της βάσης για τα μη χωροχρονικά δεδομένα είναι αυτή που παρουσιάζεται στην Εικόνα 4.3.

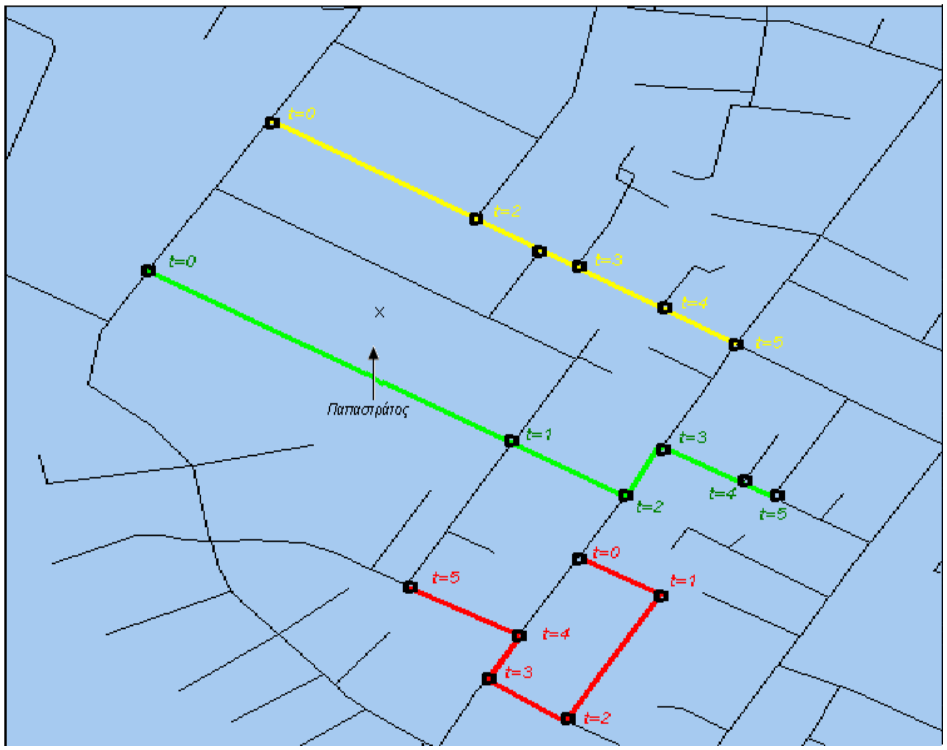
Έστω ότι ο χρήστης με id=2 κάνει την επερώτηση «Να βρεθούν οι τρείς τροχιές που ήταν πλησιέστερες στο Πανεπιστήμιο Θεσσαλίας, κτήριο Παπαστράτος,
την Τετάρτη 26 Αυγούστου 2009, από τις 8πμ μέχρι τις 8:05πμ». Και έστω K=4, ο βαθμός ανωνυμίας που θέλει να έχει και κ=4 ο αριθμός των κοντινότερων γειτόνων. Στην Εικόνα 4.8 σημειώνεται με ✕ το σημείο που θεωρούμε ότι βρίσκεται το κτήριο Παπαστράτος του Πανεπιστημίου Θεσσαλίας. Αρχικά ο αλγόριθμος ζητά από τη βάση το τμήμα των τροχιών των τεσσάρων κοντινότερων χρηστών που γειτνίαζαν με το κτήριο Παπαστράτος την Τετάρτη 26 Αυγούστου του 2009 μεταξύ 8πμ και 8:05πμ. Έπειτα συγκρίνει τον αριθμό των χρηστών, στους οποίους αντιστοιχούν αυτά τα τμήματα των τροχιών, με τη σταθερά K. Ο αριθμός είναι 4, δηλαδή ίσος του K, άρα ο αλγόριθμος επιστρέφει στον χρήστη τις "υπο-τροχιές" φαίνονται στην Εικόνα 4.8 .



Εικόνα 4.8 Αποτέλεσμα knn query για K=4 και κ=4

Εάν ο χρήστης όμως απαιτούσε K- ανωνυμία 5 και 3 γείτονες, ο αριθμός των "υποτροχιών" – που όπως είπαμε είναι 4 – θα ήταν μικρότερος του K. Σε αυτή την περίπτωση ο αλγόριθμος ελέγχει εάν η επερώτηση αποτελεί στοχευμένη επίθεση κάποιου χρήστη. Για να το πετύχει αυτό, βρίσκει το κουτί οριοθέτησης (MBR) των τροχιών που επιστράφηκαν και ελέγχει εάν ο χρήστης με id=2 έχει κάνει παλαιότερη επερώτηση, η περιοχή ενδιαφέροντος της οποίας γειτνιάζει ή επικαλύπτεται με το κουτί οριοθέτησης. Επειδή η συγκεκριμένη επερώτηση δεν αποτελεί στοχευμένη επίθεση, ο αλγόριθμος καλεί την συνάρτηση Fake_Gen(M), με M=1, καθώς πρέπει να παραχθεί μία πλασματική τροχιά για να καλυφθεί ο περιορισμός της K- ανωνυμίας. Έπειτα γίνεται ενημέρωση του πίνακα που διατηρεί το ιστορικό κάθε χρήστη για τις επερωτήσεις που έχει κάνει στο σύστημα, και έτσι εισάγεται σε αυτόν μία νέα εγγραφή με περιοχή ενδιαφέροντος το κουτί οριοθέτησης και id χρήστη το 2. Τέλος,

ένα πιθανό αποτέλεσμα που θα παρουσιαζόταν στον χρήστη φαίνεται στην Εικόνα 4.9 .

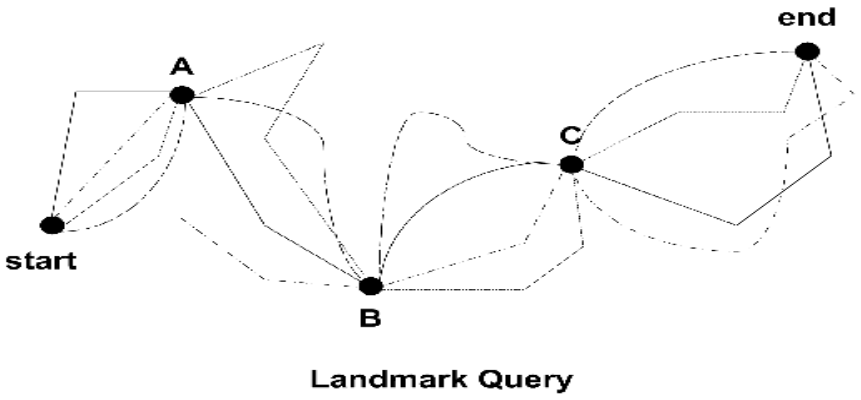


Εικόνα 4.9 Αποτέλεσμα knn query για K=5 και κ=3

4.3.3 Επερωτήσεις σημείων ενδιαφέροντος (landmark queries)

Στις επερωτήσεις για σημεία ενδιαφέροντος, το ζητούμενο είναι η εύρεση των τροχιών που είναι καταγεγραμμένες στη βάση δεδομένων και οι οποίες έχουν προκαθορισμένο σημείο εκκίνησης/κατάληξης και/ή διέρχονται από ένα ή περισσότερα προκαθορισμένα ενδιάμεσα σημεία ενδιαφέροντος (βλ. Εικόνα 4.10). Οι επερωτήσεις για σημεία ενδιαφέροντος είναι χωροχρονικές. Παρακάτω δίνουμε ένα παράδειγμα:

Q1: «Να βρεθούν οι πορείες όλων όσων επισκέφτηκαν την καφετέρια «REEF» την προηγούμενη Τρίτη περίπου στις 8πμ και κατόπιν την Εθνική Τράπεζα περίπου στις 9πμ»



Εικόνα 4.10 Η Landmark_query πάνω σε δεδομένα κίνησης – τροχιές.

Περιγραφή αλγορίθμου Landmark_query

Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

•Το ψευδώνυμο **Id** του αιτούντα.

•Τη χρονιά **YearStart**, μήνα **MonthStart**, ημέρα **DayStart**, ώρα **HourStart**, λεπτό **MinStart**, και δευτερόλεπτο **SecStart** έναρξης της περιόδου που μας ενδιαφέρει.

•Τη χρονιά **YearEnd**, μήνα **MonthEnd**, ημέρα **DayEnd**, ώρα **HourEnd**, λεπτό **MinEnd**, και δευτερόλεπτο **SecEnd** λήξης της περιόδου που μας ενδιαφέρει.

•Τη σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας, δηλαδή τον αριθμό των χρηστών που θα πρέπει να βρίσκονται εντός των περιορισμών, ώστε ο αλγόριθμος να επιτύχει.

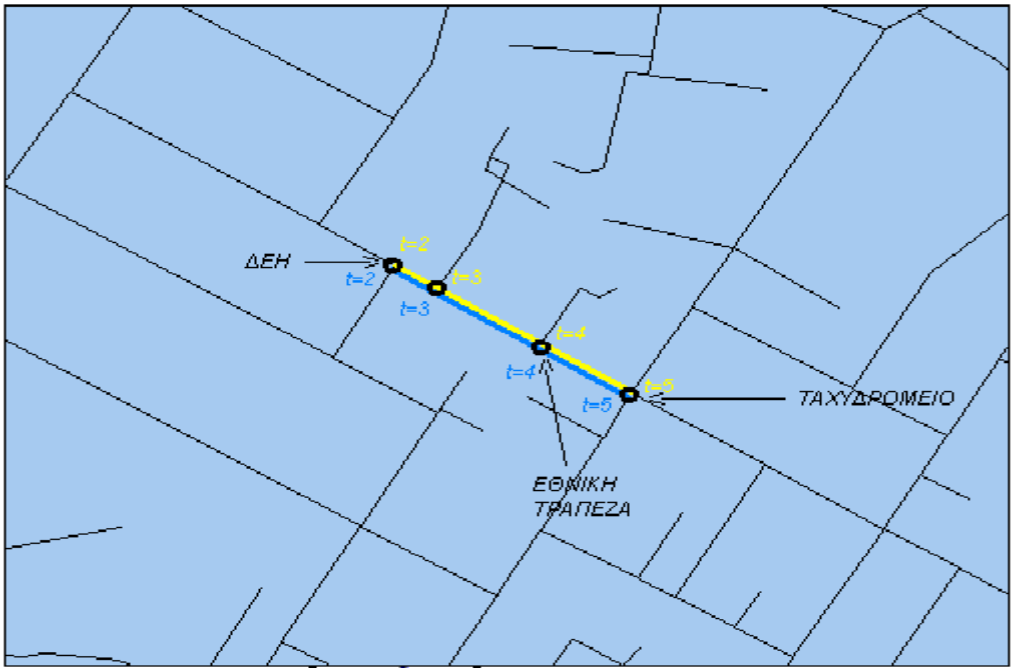
•Την σημεία ενδιαφέροντος **R={P1,P2,…,PN)**.

Ο αλγόριθμος ζητά από τη βάση δεδομένων να βρει όλες τις τροχιές που περνούν από όλα τα σημεία ενδιαφέροντος. Έπειτα από αυτές τις τροχιές επιλέγει τα τμήματα που έχουν ως σημείο έναρξης το σημείο P1 και ως σημείο λήξης το σημείο PN. Στη συνέχεια ελέγχεται εάν οι υπό-τροχιές που επιστρέφονται αντιστοιχούν σε αριθμό χρηστών μεγαλύτερο ή ίσο του K. Εάν ισχύει αυτό, παρουσιάζονται στον χρήστη οι τροχιές. Εάν ο αριθμός των χρηστών, έστω R, είναι μικρότερος του K, καλείται η διαδικασία Fake_Gen η οποία παράγει K-R πλασματικές τροχιές. Έπειτα ελέγχει εάν η επερώτηση αποτελεί στοχευμένη επίθεση κάποιου χρήστη. Για να το πετύχει αυτό, βρίσκει το κουτί οριοθέτησης (MBR) των υπο-τροχιών που επιστράφηκαν και καλεί την συνάρτηση In_Hist. Η συνάρτηση In_Hist ελέγχει εάν ο συγκεκριμένος χρήστης έχει κάνει στο παρελθόν επερωτήσεις στις οποίες οι παλαιότερες περιοχές ενδιαφέροντος επικαλύπτονται ή γειτνιάζουν με το MBR που υπολογίσαμε. Αυτό επιτυγχάνεται διατηρώντας σε έναν πίνακα hist το ψευδώνυμο, την περιοχή και το χρονικό διάστημα ενδιαφέροντος κάθε χρήστη. Εάν ανακαλυφθεί ότι αποτελεί στοχευμένη επίθεση ο αλγόριθμος επιστρέφει το μήνυμα “Privacy threat” και τερματίζει τη λειτουργία του. Διαφορετικά ελέγχει εάν οι υπο-τροχιές που επιστράφηκαν αντιστοιχούν σε αριθμό χρηστών μεγαλύτερο ή ίσο του K. Εάν ισχύει αυτό, παρουσιάζονται στον χρήστη οι τροχιές. Εάν ο αριθμός των χρηστών, έστω R, είναι μικρότερος του K, καλείται η διαδικασία Fake_Gen η οποία παράγει K-R πλασματικές τροχιές. Έπειτα ανανεώνονται τα δεδομένα που υπάρχουν μέσα στον πίνακα hist και ο αλγόριθμος παρουσιάζει τις τροχιές των χρηστών (πλασματικές και μη).

Παράδειγμα εκτέλεσης αλγορίθμου

Θεωρούμε ότι στη βάση μας έχουμε αποθηκευμένες τις τροχιές τεσσάρων χρηστών, όπως αυτές φαίνονται στην Εικόνα 4.2 και ότι η προβολής της βάσης για τα μη χωροχρονικά δεδομένα είναι αυτή που παρουσιάζεται στην Εικόνα 4.3.

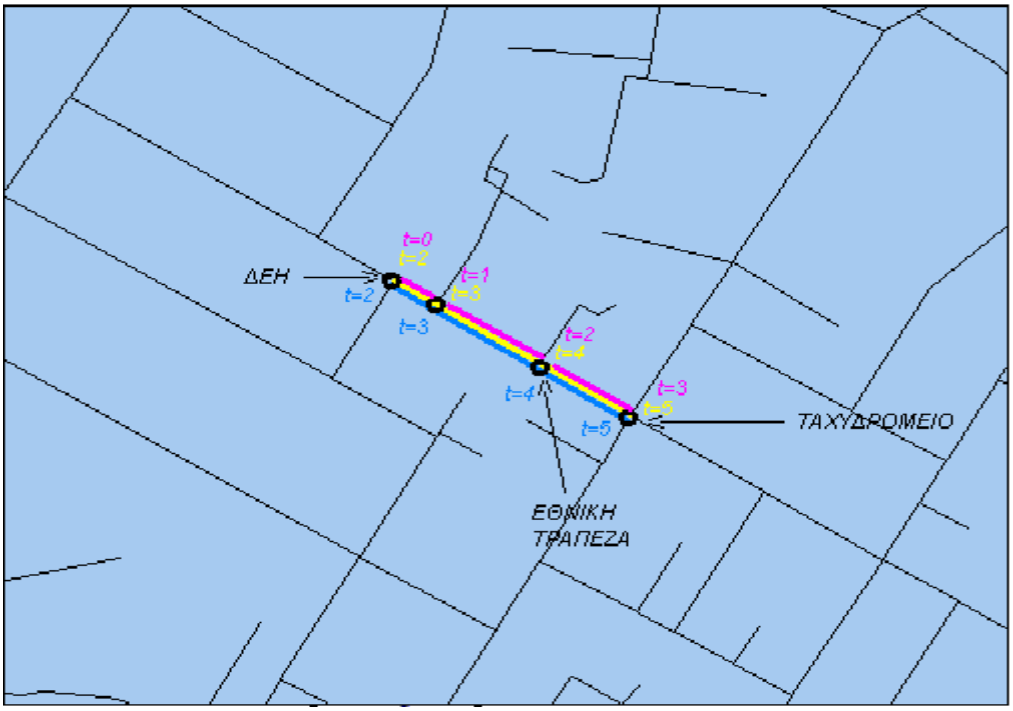
Έστω ότι ο χρήστης με id=4 κάνει την επερώτηση «Να βρεθούν τροχιές των χρηστών που πέρασαν από το κεντρικό κτήριο της ΔΕΗ Βόλου την Τετάρτη 26 Αυγούστου 2009 κατά τις 7πμ , έπειτα ίσως πέρασαν από την Εθνική Τράπεζα Ελλάδος κατά τις 9πμ και τέλος επισκέφτηκαν το κτήριο του Ταχυδρομείου κατά τις 10πμ». Και έστω K=2, ο βαθμός ανωνυμίας που θέλει να έχει. Στην Εικόνα 4.11 μπορεί να δει κανείς τα σημεία αυτά. Αρχικά ο αλγόριθμος βρίσκει τις τροχιές των χρηστών που έχουν περάσει από τη ΔΕΗ , το Ταχυδρομείο και την Εθνική Τράπεζα. Στη συνέχεια από αυτές τις τροχιές , ο αλγόριθμος κρατά τα τμήματα που ξεκινούν από τη ΔΕΗ καταλήγουν στο Ταχυδρομείο και/ή περνούν από την Εθνική Τράπεζα. Έπειτα συγκρίνει τον αριθμό των χρηστών, στους οποίους αντιστοιχούν αυτά τα τμήματα των τροχιών, με τη σταθερά K. Ο αριθμός είναι 2, δηλαδή ίσος του K, άρα ο αλγόριθμος επιστρέφει στον χρήστη τις “υπο-τροχιές” που φαίνονται στην εικόνα 4.11.



Εικόνα 4.11 Αποτέλεσμα landmark query για K=2

Εάν ο χρήστης όμως απαιτούσε K- ανωνυμία 3 (ή μεγαλύτερη), ο αριθμός των "υποτροχιών" – που όπως είπαμε είναι 2 – θα ήταν μικρότερος του K. Σε αυτή την περίπτωση ο αλγόριθμος ελέγχει εάν η επερώτηση αποτελεί στοχευμένη επίθεση κάποιου χρήστη. Για να το πετύχει αυτό, βρίσκει το κουτί οριοθέτησης (MBR) των τροχιών που επιστράφηκαν και ελέγχει εάν ο χρήστης με id=4 έχει κάνει παλαιότερη επερώτηση, η περιοχή ενδιαφέροντος της οποίας γειτνιάζει ή επικαλύπτεται με το κουτί οριοθέτησης. Θεωρούμε ότι η συγκεκριμένη επερώτηση δεν αποτελεί στοχευμένη επίθεση. Ο αλγόριθμος, λοιπόν, καλεί την συνάρτηση Fake_Gen(M), με M=1, καθώς πρέπει να παραχθεί μία πλασματική τροχιά για να καλυφθεί ο περιορισμός της K- ανωνυμίας. Έπειτα γίνεται ενημέρωση του πίνακα που διατηρεί το ιστορικό κάθε χρήστη για τις επερωτήσεις που έχει κάνει στο σύστημα, και έτσι εισάγεται σε αυτόν μία νέα εγγραφή με περιοχή ενδιαφέροντος το κουτί οριοθέτησης και id χρήστη το 4. Τέλος, ένα πιθανό αποτέλεσμα που θα παρουσιαζόταν στον

χρήστη φαίνεται στην Εικόνα 4.12 (η πλασματική τροχιά αναπαρίσταται με ροζ χρώμα).



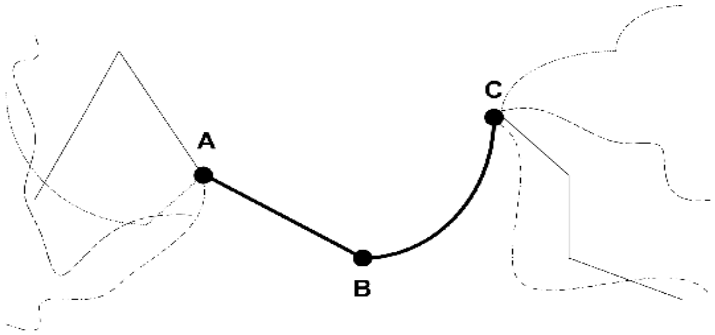
Εικόνα 4.12 Αποτέλεσμα landmark query για K=3, με μια πλασματική τροχιά

4.3.4 Επερωτήσεις συγκεκριμένων διαδρομών (route queries)

Στις επερωτήσεις για συγκεκριμένες διαδρομές, στόχος είναι η εύρεση των τροχιών που ταυτίζονται με κάποια προκαθορισμένη πορεία. Σε αντίθεση με τις επερωτήσεις για σημεία ενδιαφέροντος, στις επερωτήσεις για συγκεκριμένες διαδρομές όλα τα σημεία της διαδρομής πρέπει να συμπεριλαμβάνονται στις τροχιές(βλ. Εικόνα 4.13).

Παρακάτω δίνουμε ένα παράδειγμα:

Q1: «Να βρεθούν οι πορείες όλων των αυτοκινήτων που μπήκαν στη γέφυρα (διασταύρωση Α) περίπου στις 8:30πμ σήμερα, συνέχισαν στον αυτοκινητόδρομο μέχρι το σημείο Β και κατόπιν βγήκαν από τον αυτοκινητόδρομο (έξοδος Γ) περίπου στις 8:40πμ»



Εικόνα 4.13 Η Route_query πάνω σε δεδομένα κίνησης – τροχιές.

Περιγραφή αλγορίθμου Route_query

Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

• Το ψευδώνυμο **Id** του αιτούντα.

• Τη χρονιά **YearStart**, μήνα **MonthStart**, ημέρα **DayStart**, ώρα **HourStart**, λεπτό **MinStart**, και δευτερόλεπτο **SecStart** έναρξης της περιόδου που μας ενδιαφέρει.

• Τη χρονιά **YearEnd**, μήνα **MonthEnd**, ημέρα **DayEnd**, ώρα **HourEnd**, λεπτό **MinEnd**, και δευτερόλεπτο **SecEnd** λήξης της περιόδου που μας ενδιαφέρει.

• Τη σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας, δηλαδή τον αριθμό των χρηστών που θα πρέπει να βρίσκονται εντός των περιορισμών, ώστε ο αλγόριθμος να επιτύχει.

• Την διαδρομή **R**.

Ο αλγόριθμος ζητά από τη βάση δεδομένων να βρει όλες τις τροχιές που περνούν μόνο από τα σημεία της δοθείσας διαδρομής. Εάν κάποιες από τις τροχιές περνούν και από άλλα σημεία απορρίπτονται. Στη συνέχεια ελέγχεται εάν οι τροχιές που επιστρέφονται αντιστοιχούν σε αριθμό χρηστών μεγαλύτερο ή ίσο του K. Εάν ισχύει αυτό, παρουσιάζονται στον χρήστη οι τροχιές. Εάν ο αριθμός των χρηστών, έστω R, είναι μικρότερος του K, καλείται η διαδικασία Fake_Gen η οποία παράγει K-R πλασματικές τροχιές. Έπειτα ελέγχει εάν η επερώτηση αποτελεί στοχευμένη επίθεση κάποιου χρήστη. Για να το πετύχει αυτό, βρίσκει το κουτί οριοθέτησης (MBR) των τροχιών που επιστράφηκαν και καλεί την συνάρτηση In_Hist. Η συνάρτηση In_Hist ελέγχει εάν ο συγκεκριμένος χρήστης έχει κάνει στο παρελθόν επερωτήσεις στις οποίες οι παλαιότερες περιοχές ενδιαφέροντος επικαλύπτονται ή γειτνιάζουν με το MBR Που υπολογίσαμε. Αυτό επιτυγχάνεται διατηρώντας σε έναν πίνακα hist το ψευδώνυμο, την περιοχή και το χρονικό διάστημα ενδιαφέροντος κάθε χρήστη. Εάν ανακαλυφθεί ότι αποτελεί στοχευμένη επίθεση ο αλγόριθμος επιστρέφει το μήνυμα “Privacy threat” και τερματίζει τη λειτουργία του. Διαφορετικά ελέγχει εάν οι τροχιές που επιστράφηκαν αντιστοιχούν σε αριθμό χρηστών μεγαλύτερο ή ίσο του K. Εάν ισχύει αυτό, παρουσιάζονται στον χρήστη οι τροχιές. Εάν ο αριθμός των χρηστών, έστω R, είναι μικρότερος του K, καλείται η διαδικασία Fake_Gen η οποία παράγει K-R πλασματικές τροχιές. Έπειτα ανανεώνονται τα δεδομένα που υπάρχουν μέσα στον πίνακα hist και ο αλγόριθμος παρουσιάζει τις τροχιές των χρηστών (πλασματικές και μη).

Παράδειγμα εκτέλεσης αλγορίθμου

Θεωρούμε ότι στη βάση μας έχουμε αποθηκευμένες τις τροχιές τεσσάρων χρηστών, όπως αυτές φαίνονται στην Εικόνα 4.2 και ότι η προβολής της βάσης για τα μη χωροχρονικά δεδομένα είναι αυτή που παρουσιάζεται στην Εικόνα 4.3.

Έστω ότι ο χρήστης με id=4 κάνει την επερώτηση «Να βρεθούν τροχιές των χρηστών που πέρασαν από το κεντρικό κτήριο της ΔΕΗ Βόλου την Τετάρτη 26 Αυγούστου 2009 κατά τις 7πμ , έπειτα πέρασαν από την Εθνική Τράπεζα Ελλάδος, στη συνέχεια πήγαν στην καφετέρια «reef», πέρασαν από το κτήριο Παπαστράτος, αμέσως μετά πήγαν στο κατάστημα «Γερμανος» και τέλος επισκέφτηκαν το κτήριο του Ταχυδρομείου κατά τις 10πμ». Και έστω K=1, ο βαθμός ανωνυμίας που θέλει να έχει. Στην Εικόνα 4.14 τα σημεία αυτά συμβολίζονται με ένα κύκλο. Αρχικά ο αλγόριθμος βρίσκει τις τροχιές των χρηστών που έχουν περάσει από ένα ή περισσότερα σημεία. Στη συνέχεια από αυτές τις τροχιές , ο αλγόριθμος κρατά τα τμήματα που ξεκινούν από τη ΔΕΗ και καταλήγουν στο Ταχυδρομείο. Έπειτα συγκρίνει τον αριθμό των χρηστών, στους οποίους αντιστοιχούν αυτά τα τμήματα των τροχιών, με τη σταθερά K. Ο αριθμός είναι 1, δηλαδή ίσος του K, άρα ο αλγόριθμος επιστρέφει στον χρήστη τις “υπο-τροχιές” που φαίνονται στην Εικόνα 4.14 .

Εικόνα 4.14 Αποτέλεσμα route query για K=1

Εάν ο χρήστης όμως απαιτούσε K- ανωνυμία 2 (ή μεγαλύτερη), ο αριθμός των "υπο-τροχιών" – που όπως είπαμε είναι 1 –θα ήταν μικρότερος του K. Σε αυτή την περίπτωση ο αλγόριθμος ελέγχει εάν η επερώτηση αποτελεί στοχευμένη επίθεση κάποιου χρήστη. Για να το πετύχει αυτό, βρίσκει το κουτί οριοθέτησης (MBR) των τροχιών που επιστράφηκαν και ελέγχει εάν ο χρήστης με id=4 έχει κάνει παλαιότερη επερώτηση, η περιοχή ενδιαφέροντος της οποίας γειτνιάζει ή επικαλύπτεται με το κουτί οριοθέτησης. Θεωρούμε ότι η συγκεκριμένη επερώτηση δεν αποτελεί στοχευμένη επίθεση. Ο αλγόριθμος, λοιπόν, καλεί την συνάρτηση Fake_Gen(M), με M=1, καθώς πρέπει να παραχθεί μία πλασματική τροχιά για να καλυφθεί ο περιορισμός της K-ανωνυμίας. Έπειτα γίνεται ενημέρωση του πίνακα που διατηρεί το ιστορικό κάθε χρήστη για τις επερωτήσεις που έχει κάνει στο σύστημα, και έτσι εισάγεται σε αυτόν μία νέα εγγραφή με περιοχή ενδιαφέροντος το κουτί οριοθέτησης και id χρήστη ίσον με 4. Τέλος, ένα πιθανό αποτέλεσμα που θα παρουσιαζόταν στον

χρήστη φαίνεται στην Εικόνα 4.15 (η πλασματική τροχιά αναπαρίσταται με ροζ χρώμα).

Εικόνα 4.15 Αποτέλεσμα route query για K=2, με μια πλασματική τροχιά

4.3.5 Επερωτήσεις κάλυψης-χρόνου (range-time queries)

Στις επερωτήσεις κάλυψης-χρόνου το ζητούμενο είναι να ανακαλυφθούν οι τροχιές των ατόμων που είναι αποθηκευμένες στη βάση, οι οποίες βρίσκονται εντός μιας προκαθορισμένης χωρικής περιοχής, και δημιουργούνται κατά τη διάρκεια μίας συγκεκριμένης ημέρας ή ενός συγκεκριμένου μήνα ή ενός συγκεκριμένου έτους. Παρακάτω παραθέτουμε ορισμένα παραδείγματα επερωτήσεων κάλυψης-χρόνου:

Q1: «Να βρεθούν όλες οι τροχιές των χρηστών που διέσχισαν το κέντρο του Βόλου σήμερα»

Q2: «Να βρεθούν όλες οι τροχιές των χρηστών που πέρασαν από το Πήλιο τον περασμένο μήνα»

Περιγραφή αλγορίθμου range-time_query

Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

- Το ψευδώνυμο **Id** του αιτούντα
- Την περιοχή **Sgeo** που μας ενδιαφέρει.
- Τη χρονιά **YearStart**, μήνα **MonthStart**, και την ημέρα **DayStart**, της περιόδου που μας ενδιαφέρει.
- Την σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας, δηλαδή τον αριθμό των χρηστών που θα πρέπει να βρίσκονται εντός των περιορισμών, ώστε ο αλγόριθμος να επιτύχει.

Αρχικά ο αλγόριθμος θέτει στις τοπικές μεταβλητές που συμβολίζουν την αρχική και τελική τιμή των δευτερολέπτων, την αρχική και τελική τιμή των λεπτών και την αρχική και τελική τιμή της ώρας τις τιμές αντίστοιχα ,HourSTART=0, HourEND=23 MinSTART=0, MinEND=59, SecSTART=0, SecEND=59. Επίσης οι τελική τιμή του έτους θα έχει την ίδια τιμή με την αρχική τιμή. Έπειτα ελέγχει εάν κάποια από τις τιμές των DayStart, MonthStart, ή YearStart είναι μηδενική. Εάν η μεταβλητή DayStart έχει μηδενική τιμή τότε σημαίνει πως μας ενδιαφέρουν οι τροχιές ενός συγκεκριμένου μήνα ή χρόνου. Σε αντίθετη περίπτωση ενδιαφερόμαστε για τις τροχιές μιας συγκεκριμένης ημέρας, οπότε οι μεταβλητές για την τελική τιμή του μήνα και τις ημέρας παίρνουν τις τιμές αντίστοιχα MonEND, MonSTART, DayEND, DaySTART. Εάν η μεταβλητή DayStart έχει μηδενική τιμή και η μεταβλητή MonSTART έχει επίσης μηδενική τιμή, αυτό σημαίνει ότι μας ενδιαφέρουν οι τροχιές ενός συγκεκριμένου έτους. Άρα οι τιμές των μεταβλητών που συμβολίζουν την αρχική και τελική τιμή του μήνα και την αρχική και τελική τιμή της ημέρας παίρνουν

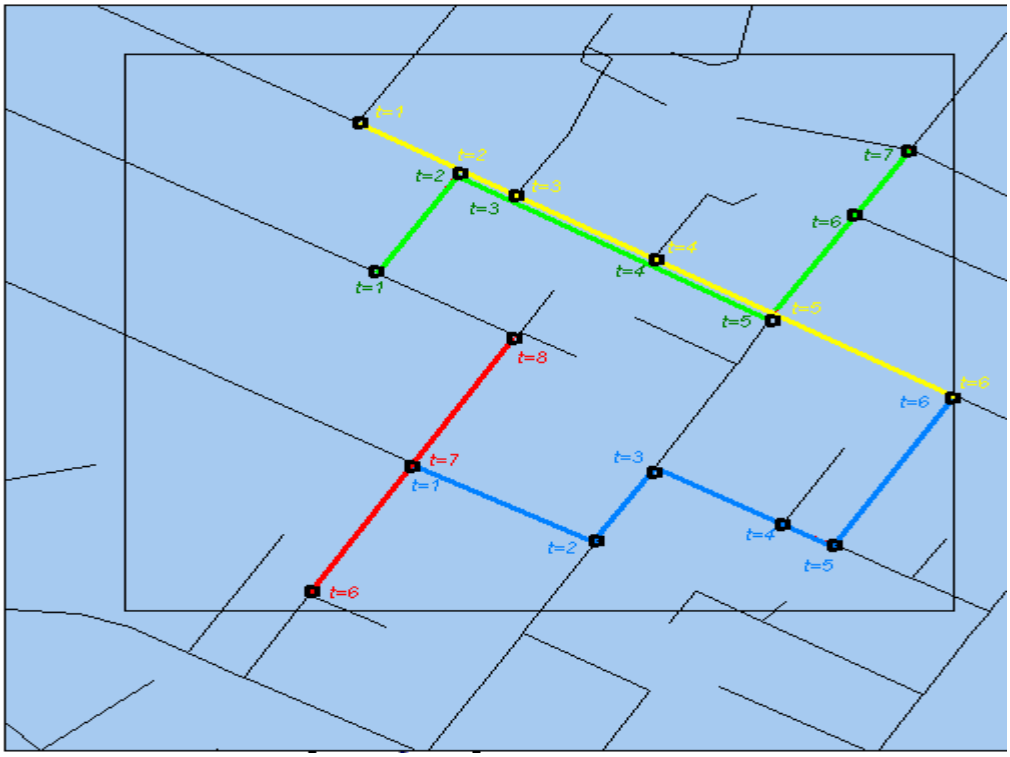
τις τιμές αντίστοιχα, MonSTART=1, DaySTART=1, MonEND=12; DayEND=31. Εάν η μεταβλητή DayStart έχει μηδενική τιμή και η μεταβλητή MonSTART δεν έχει μηδενική τιμή, αυτό σημαίνει ότι μας ενδιαφέρουν οι τροχιές ενός συγκεκριμένου μήνα. Άρα οι τιμές των μεταβλητών MonEND, DaySTART, και DayEND, θα είναι αντίστοιχα MonSTART, 1, και 31.Στη συνέχεια, αφού τεθούν τιμές στις μεταβλητές για τον χρόνο, ο αλγόριθμος ελέγχει εάν η επερώτηση αποτελεί στοχευμένη επίθεση κάποιου χρήστη. Για να το πετύχει αυτό, καλεί την συνάρτηση In_Hist (αναλυτική περιγραφή στην Ενότητα 6.3), η οποία ελέγχει εάν ο συγκεκριμένος χρήστης έχει κάνει στο παρελθόν επερωτήσεις στις οποίες οι παλαιότερες περιοχές ενδιαφέροντος επικαλύπτονται ή γειτνιάζουν με την τωρινή περιοχή ενδιαφέροντος. Αυτό επιτυγχάνεται διατηρώντας σε έναν πίνακα hist το ψευδώνυμο, την περιοχή και το χρονικό διάστημα ενδιαφέροντος κάθε χρήστη. Εάν ανακαλυφθεί ότι αποτελεί στοχευμένη επίθεση ο αλγόριθμος επιστρέφει το μήνυμα “Privacy threat” και τερματίζει τη λειτουργία του. Διαφορετικά ζητά από τη βάση δεδομένων να του επιστρέψει τις τροχιές των χρηστών οι οποίες διασχίζουν την περιοχή που μας ενδιαφέρει. Στη συνέχεια ελέγχεται εάν οι τροχιές που επιστρέφονται αντιστοιχούν σε αριθμό χρηστών μεγαλύτερο ή ίσο του K. Εάν ισχύει αυτό, παρουσιάζονται στον χρήστη οι τροχιές. Εάν ο αριθμός των χρηστών, έστω R, είναι μικρότερος του K, καλείται η διαδικασία Fake_Gen η οποία παράγει K-R πλασματικές τροχιές. Έπειτα ανανεώνονται τα δεδομένα που υπάρχουν μέσα στον πίνακα hist και ο αλγόριθμος παρουσιάζει τις τροχιές (πλασματικές και μη).

Παράδειγμα εκτέλεσης αλγορίθμου

Θεωρούμε ότι στη βάση μας έχουμε αποθηκευμένες τις τροχιές τεσσάρων χρηστών, όπως αυτές φαίνονται στην Εικόνα 4.2 και ότι η προβολής της βάσης για τα μη χωροχρονικά δεδομένα είναι αυτή που παρουσιάζεται στην Εικόνα 4.3.

Έστω ότι ο χρήστης με id=3 κάνει την επερώτηση: «Να βρεθούν όλες οι τροχιές των χρηστών που διέσχισαν το πεδίο του Άρεως τον Αύγουστου του 2009». Και έστω K=3, ο βαθμός ανωνυμίας που θέλει να έχει. Στην Εικόνα 4.16 σημειώνεται με ένα ορθογώνιο η περιοχή που θεωρούμε ως πεδίο του ‘Άρεως.

Εφ’ όσον θέλουμε τις τροχιές για έναν συγκεκριμένο μήνα οι μεταβλητές που είναι απαραίτητες για να ορίσουν την περίοδο παίρνουν τις τιμές: HourSTART=0, HourEND=23, MinSTART=0, MinEND=59, SecSTART=0, SecEND=59, YearEND=YearSTART, MonEND=MonSTART, DaySTART=1, DayEND=31. Έπειτα ο αλγόριθμος ελέγχει εάν ο χρήστης με id=3 έχει κάνει παλαιότερη επερώτηση, η περιοχή ενδιαφέροντος της οποίας γειτνιάζει ή επικαλύπτεται με την περιοχή του πεδίου του Άρεως. Θεωρούμε ότι κάτι τέτοιο δεν ισχύει, οπότε ο αλγόριθμος ζητά από τη βάση το τμήμα των τροχιών των χρηστών που πέρασαν από το πεδίο του Άρεως τον Αύγουστου του 2009. Έπειτα συγκρίνει τον αριθμό των χρηστών, στους οποίους αντιστοιχούν αυτά τα τμήματα των τροχιών, με τη σταθερά K. Ο αριθμός είναι 3, δηλαδή ίσος του K, άρα ο αλγόριθμος επιστρέφει στον χρήστη τις “υπο-τροχιές” φαίνονται στην Εικόνα 4.16.



Εικόνα 4.16 Αποτέλεσμα range-time query για K=3

Κεφάλαιο 5

ΑΛΓΟΡΙΘΜΟΙ ΠΡΟΣΤΑΣΙΑΣ ΙΔΙΩΤΙΚΟΤΗΤΑΣ

Όπως έχει αναφερθεί και προηγουμένως, στόχος της παρούσας διπλωματικής είναι η εύρεση ενός συνόλου προτύπων κίνησης που επιτρέπουν την εξαγωγή συμπερασμάτων, ενώ ταυτόχρονα θωρακίζουν την ιδιωτικότητα των ατόμων των οποίων οι τροχιές αναλύονται. Απαραίτητο στοιχείο λοιπόν των χωροχρονικών επερωτήσεων που παρουσιάσαμε στο προηγούμενο κεφάλαιο, είναι η παροχή κάλυψης από εξειδικευμένες επιθέσεις κακόβουλων χρηστών. Στην παρούσα ενότητα εξετάζονται δύο είδη πιθανών επιθέσεων που μπορούν να γίνουν από τέτοιους χρήστες με στόχο να ανακαλύψουν συγκεκριμένα άτομα στη βάση δεδομένων ή/και να παρακολουθήσουν την κίνηση συγκεκριμένων χρηστών, όπως αυτή έχει καταγραφεί από το σύστημα. Για καθένα από τα δύο είδη επιθέσεων παρουσιάζεται μια ολοκληρωμένη μεθοδολογία που επιτυγχάνει την επαρκή θωράκιση της Query Engine.

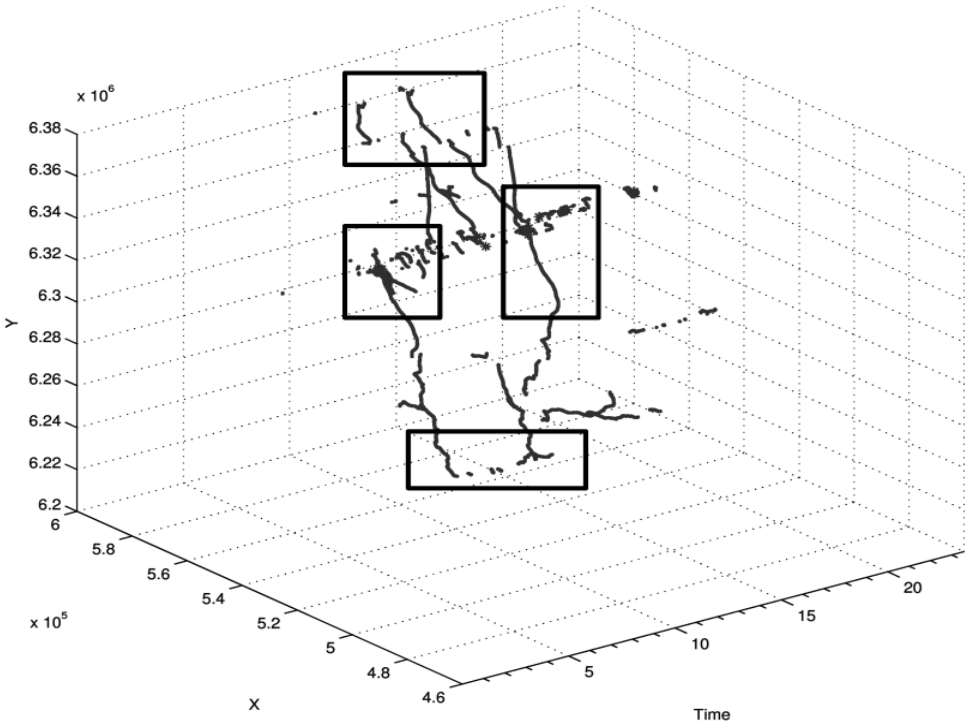
5.1 Επίθεση ταυτοποίησης χρήστη

Στην επίθεση ταυτοποίησης χρήστη, ο επίδοξος εισβολέας κατασκευάζει ένα σύνολο από στοχευμένες επερωτήσεις που παρουσιάζουν αλληλοεπικάλυψη στο σύνολο των γνωρισμάτων-τιμών, με στόχο να αυξήσει την εμπιστοσύνη του σχετικά με την πραγματική ταυτότητα κάποιου χρήστη στο σύστημα. Στο παρόν κεφάλαιο ενδιαφερόμαστε για την αντιμετώπιση αυτού του είδους της επίθεσης σε

χωροχρονικές βάσεις που διατηρούν πληροφορία κίνησης αντικειμένων. Ας θεωρήσουμε την επερώτηση κάλυψης που παρουσιάστηκε στην Εικόνα 4.3 και η οποία επιστρέφει τις τροχιές που βρίσκονται στην περιοχή Α. Υποθέτοντας 4–ανωνυμία, η Query Engine παράγει μια πλασματική τροχιά p. Οι πραγματικές τροχιές o, q και r μαζί με την πλασματική τροχιά p ικανοποιούν την αρχή της 4–ανωνυμίας στην περιοχή Α. Ας υποθέσουμε τώρα ότι ο επίδοξος εισβολέας θέλει να ανακαλύψει εάν η τροχιά ο είναι πραγματική ή πλασματική. Προκειμένου να το καταφέρει, επιλέγει μια περιοχή Β που είναι τμήμα της περιοχής Α και η οποία περιέχει την τροχιά ο. Στη συνέχεια, θέτει μια επερώτηση στη βάση δεδομένων (μέσω της διεπαφής της Query Engine) αναζητώντας τις τροχιές στην περιοχή Β. Εάν η επερώτηση αυτή γίνει δεκτή από το σύστημα και εξυπηρετηθεί, τότε θα επιστραφεί στον χρήστη η τροχιά ο και μέρος της τροχιάς p, μαζί με δύο πλασματικές τροχιές προκειμένου να εξασφαλιστεί η 4–ανωνυμία στην περιοχή Β. Ως εκ τούτου, και καθώς ο επίδοξος εισβολέας γνωρίζει τις τροχιές στην περιοχή Α, η εμπιστοσύνη του αναφορικά με την πορεία ο θα αυξηθεί δραματικά από 1/4 (περιοχή Α) σε 1/2 (περιοχή Β). Όπως γίνεται εμφανές, μια τέτοια επίθεση μπορεί αποτελεσματικά να καταργήσει την Κ– ανωνυμία που προσφέρεται από το σύστημα.

Προκειμένου να αποφευχθεί η επίθεση ταυτοποίησης χρήστη, η Query Engine λαμβάνει υπόψη τον μηχανισμό ελέγχου επερωτήσεων που εφαρμόζεται στα σχεσιακά και τον επεκτείνει σε χωροχρονικά δεδομένα και σε τροχιές αντικειμένων. Συγκεκριμένα, ο μηχανισμός ελέγχου επερωτήσεων διατηρεί ενημερωμένα αρχεία καταγραφής (ιστορικό επερωτήσεων) για όλες τις επερωτήσεις που έχει υποβάλλει κάποιος χρήστης στο σύστημα και ελέγχει κάθε νέα επερώτηση του χρήστη με βάση το ιστορικό του για να ανακαλύψει τυχόν προσπάθειες για επίθεση στη βάση

δεδομένων. Στην περίπτωση που εξετάζουμε εμείς, η Query Engine χρειάζεται να διατηρεί το ιστορικό των επερωτήσεων που έχει υποβάλλει κάθε χρήστης στο σύστημα, ως το σύνολο των χωροχρονικών περιοχών που καλύπτονται στις απαντήσεις των επερωτήσεων του χρήστη. Έτσι, όταν ο χρήστης υποβάλλει ένα νέο ερώτημα στη βάση, το σύστημα θα ελέγξει το ιστορικό του για να εξετάσει εάν η χωροχρονική περιοχή που καλύπτεται στην τωρινή απάντηση έχει αλληλοεπικάλυψη με κάποια από τις περιοχές που έχει καλυφθεί σε προηγούμενες επερωτήσεις του χρήστη. Στην περίπτωση που κάτι τέτοιο ισχύει, το σύστημα θα αρνηθεί να απαντήσει στο χρήστη και θα καταγράψει το γεγονός αυτό σε ένα αρχείο καταγραφής συμβάντων. Διαφορετικά, το σύστημα θα απαντήσει στην επερώτηση του χρήστη και ακολούθως θα αποθηκεύσει την χωροχρονική περιοχή της απάντησης στο ιστορικό επερωτήσεων του χρήστη.



Εικόνα 5.1 Χωροχρονικός έλεγχος για τη θωράκιση από την επίθεση ταυτοποίησης χρήστη

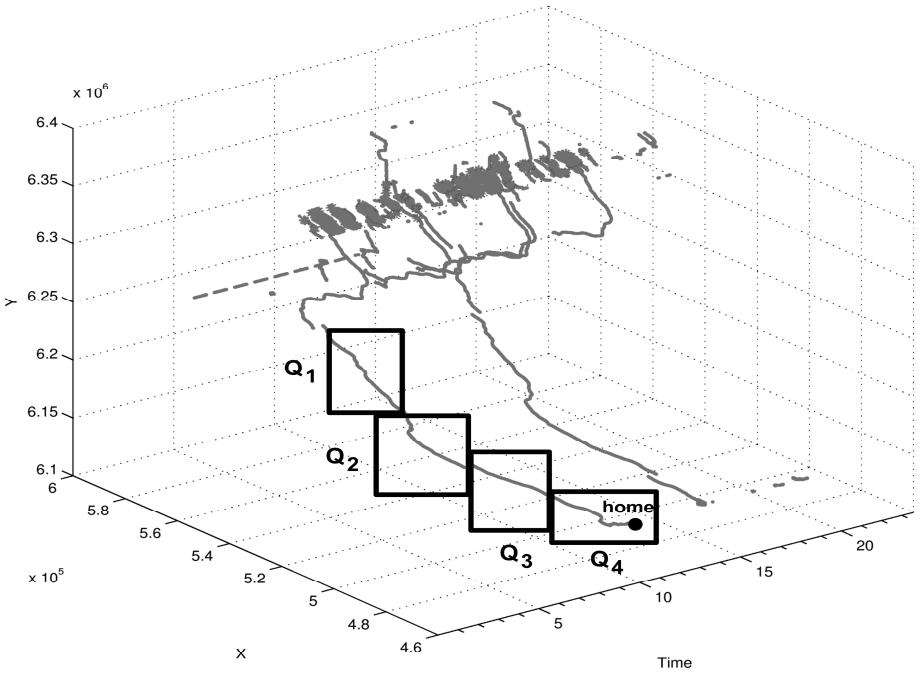
Ένα παράδειγμα εφαρμογής της τεχνικής αυτής παρουσιάζεται στην Εικόνα 5.1 (10), όπου υπάρχουν 4 περιοχές καταγεγραμμένες στο ιστορικό επερωτήσεων του χρήστη κάτι που σημαίνει ότι μελλοντικές επερωτήσεις που αφορούν τις περιοχές αυτές δεν πρόκειται να εξυπηρετηθούν από το σύστημα.

5.2 Επίθεση παρακολούθησης χρήστη

Κατά την επίθεση παρακολούθησης χρήστη, ο επίδοξος εισβολέας προσπαθεί να «ακολουθήσει» την τροχιά ενός συγκεκριμένου χρήστη στη βάση δεδομένων. Προκειμένου να το επιτύχει αυτό, κατασκευάζει ένα σύνολο από στοχευμένες επερωτήσεις σε χωροχρονικές περιοχές που είναι γειτονικές μεταξύ τους ή έχουν μερική αλληλοεπικάλυψη. Εάν δεν ληφθούν ειδικά μέτρα για τη θωράκιση του συστήματος από αυτού του είδους την επίθεση τότε ο επίδοξος εισβολέας θα κατορθώσει (α) να αυξήσει την εμπιστοσύνη του σχετικά με το εάν κάποια συγκεκριμένη τροχιά είναι πραγματική ή πλασματική, και (β) να βεβαιωθεί ότι κάποια συγκεκριμένη πορεία που «ακολουθεί» στο σύστημα, ανήκει σε πραγματικό και όχι σε πλασματικό χρήστη.

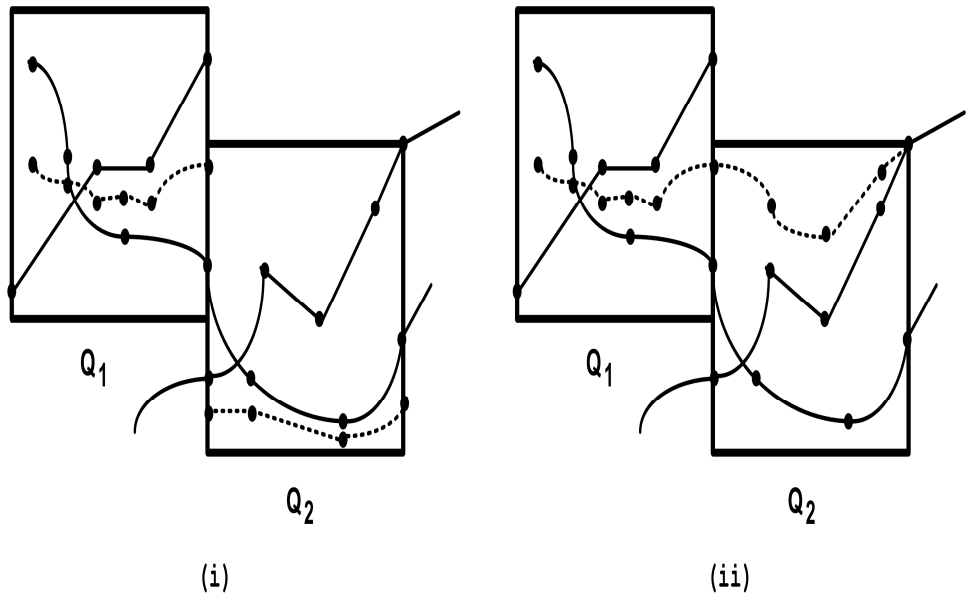
Θεωρήστε τις επερωτήσεις κάλυψης Q1 – Q4 που παρουσιάζονται στη Εικόνα 5.2. Όταν κάποιος επίδοξος εισβολέας θέσει την επερώτηση Q1 στη βάση δεδομένων, αυτή θα επιστρέψει την τροχιά που απεικονίζεται στην Εικόνα 5.2 μαζί με άλλες K–1 τροχιές προκειμένου να παρασχεθεί K– ανωνυμία. Εν συνεχεία, έστω ότι ο εισβολέας θέσει την επερώτηση Q2, αναζητώντας κατά τον τρόπο αυτό τις πορείες που υπάρχουν σε μια περιοχή γειτονική της Q1. Η πραγματική πορεία που μετείχε στην επερώτηση Q1 και απεικονίζεται στην Εικόνα 5.2 (10), θα εμφανιστεί ομοίως στην

περιοχή Q2 και η κίνηση θα συνεχίζει ομαλά, όπως άλλωστε είναι αναμενόμενο. Προκειμένου να παρασχεθεί K– ανωνυμία στην περιοχή Q2, το σύστημα θα ανακτήσει όλες τις τροχιές από τη βάση δεδομένων που αντιστοιχούν στην Q2 και θα δημιουργήσει (εάν χρειάζεται) ένα πλήθος από πλασματικές τροχιές. Ένας απλοϊκός όμως αλγόριθμος κατασκευής πλασματικών τροχιών, θα παράγει τροχιές λαμβάνοντας υπόψη του μόνο την περιοχή Q2 και όχι την Q1. Ως αποτέλεσμα, ο επίδοξος εισβολέας θα μπορέσει εύκολα να διακρίνει τις πραγματικές από τις πλασματικές τροχιές στην περιοχή Q2 και ακολούθως να βεβαιωθεί για το εάν ο χρήστης που παρακολουθεί είναι πραγματικός ή όχι. Το ίδιο ισχύει για όλη τη διαδρομή από το Q1 ως το Q4 και ως αποτέλεσμα ο επίδοξος εισβολέας μπορεί εύκολα να «ακολουθήσει» τον χρήστη στον τελικό του προορισμό, όπως φαίνεται στο Q4.



Εικόνα 5.2 Χωροχρονικός έλεγχος για τη θωράκιση από την επίθεση παρακολούθησης χρήστη

Η κατάσταση που μόλις περιγράφηκε απεικονίζεται καλύτερα στην Εικόνα 5.3 (10), όπου θεωρούμε 3–ανωνυμία. Οι τροχιές που εμφανίζονται με συνεχόμενη γραμμή αφορούν πραγματικούς χρήστες, ενώ αυτές με διακεκομμένη γραμμή αντιστοιχούν σε πλασματικούς χρήστες. Όπως μπορεί κανείς να παρατηρήσει, η απάντηση του Q2 αποκαλύπτει την πλασματική τροχιά στην περιοχή Q1, καθώς η κίνηση του πλασματικού χρήστη στην περιοχή Q1 διακόπτεται απότομα στην Q2. Ως αποτέλεσμα, η εμπιστοσύνη του επίδοξου εισβολέα αναφορικά με τις πραγματικές τροχιές στην περιοχή Q1 αυξάνεται από 1/3 σε 1/2. Επιπροσθέτως, εάν ο επίδοξος εισβολέας έχει γνώση των αλγορίθμων παρεμβολής που χρησιμοποιούνται στην Query Engine , τότε μπορεί να είναι βέβαιος ότι η μοναδική τροχιά που επεκτείνεται από την περιοχή Q1 στην Q2 αντιστοιχεί σε πραγματικό χρήστη.



Εικόνα 5.3 Προστασία της βάσης δεδομένων από την επίθεση παρακολούθησης χρήστη

Προκειμένου να αποφευχθεί η παρακολούθηση της τροχιάς των χρηστών από επίδοξους εισβολείς, χρησιμοποιούμε την τεχνική του χωροχρονικού ελέγχου, όπως και νωρίτερα, μόνο που στην περίπτωση αυτή ανακαλύπτουμε επερωτήσεις που αφορούν χωροχρονικές περιοχές που γειτνιάζουν (ή έχουν μερική αλληλοεπικάλυψη) με αυτές από το ιστορικό επερωτήσεων του χρήστη. Στην περίπτωση που ανακαλυφθεί μια τέτοια επερώτηση ακολουθούμε την προσέγγιση που παρουσιάζεται στην Εικόνα 5.3(ii). Συγκεκριμένα, η προτεινόμενη τεχνική στοχεύει στο να δημιουργήσει σύγχυση στον επίδοξο εισβολέα αναφορικά με τους πραγματικούς και τους πλασματικούς χρήστες στη βάση δεδομένων, με το να επεκτείνει την «κίνηση» των πλασματικών χρηστών σε γειτονικές περιοχές. Παρόλο που υπάρχουν εναλλακτικές λύσεις για να αντιμετωπιστεί το συγκεκριμένο πρόβλημα, θεωρούμε ότι η προτεινόμενη προσέγγιση είναι ταυτόχρονα απλή και αποτελεσματική. Αρχικά, η εμπιστοσύνη του επιτιθέμενου αναφορικά με τις πραγματικές τροχιές στην περιοχή Q1 παραμένει αμετάβλητη καθώς δεν μπορεί να ανακαλύψει ποιές τροχιές στην Q1 είναι πραγματικές και ποιές πλασματικές. Επιπλέον, βάσει του αλγόριθμου κατασκευής των πλασματικών τροχιών, ο επίδοξος εισβολέας δεν μπορεί να είναι βέβαιος ότι ο χρήστης που προσπαθεί να παρακολουθήσει είναι πραγματικός και όχι πλασματικός. Συμπερασματικά, δοθείσης μιας λογικής τιμής του K στην K–ανωνυμία, ο επίδοξος εισβολέας θα αποθαρρυνθεί από το να προσπαθήσει να «ακολουθήσει» την τροχιά κάποιου χρήστη που είναι αποθηκευμένη στη βάση δεδομένων, καθώς η πιθανότητα να ακολουθεί πραγματικό χρήστη θα είναι αρκετά μικρή.

5.3 Ιστορικό επερωτήσεων

Το σύστημά μας προκειμένου να ανακαλύψει τυχόν προσπάθειες για επίθεση στη βάση δεδομένων, πρέπει να διατηρεί ενημερωμένα αρχεία καταγραφής για όλες τις επερωτήσεις που έχει υποβάλλει κάποιος χρήστης στο σύστημα και να ελέγχει κάθε νέα επερώτηση του χρήστη με βάση το ιστορικό του. Οι αλγόριθμοι των IN_HIST και UPDATE_HIST που παρουσιάζονται παρακάτω, υλοποιούν τις λειτουργίες που αναφέρθηκαν.

Περιγραφή αλγορίθμου IN_HIST

Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

- Το ψευδώνυμο **Id** του αιτούντα
- Την περιοχή ενδιαφέροντος **Sgeo**.
- Τη χρονοσφραγίδα έναρξης **tstart** της περιόδου ενδιαφέροντος .
- Τη χρονοσφραγίδα έναρξης **tend** της περιόδου ενδιαφέροντος .

Ο αλγόριθμος ελέγχει εάν για τον συγκεκριμένο χρήστη υπάρχει εγγραφή μέσα στον πίνακα hist, η περιοχή ενδιαφέροντος της οποίας επικαλύπτεται ή γειτνιάζει με την περιοχή Sgeo. Εάν ισχύει αυτό ο αλγόριθμος επιτυγχάνει, διαφορετικά όχι.

Ψευδοκώδικας IN_HIST

Είσοδος :

- Το ψευδώνυμο **Id** του χρήστη.

- Η περιοχή ενδιαφέροντος **Sgeo.**

- Η χρονοσφραγίδα έναρξης **tstart** της περιόδου ενδιαφέροντος.

- Η χρονοσφραγίδα έναρξης **tend** της περιόδου ενδιαφέροντος.

Έξοδος :

- Μία Boolean μεταβλητή ,η οποία προσδιορίζει την επιτυχία ή όχι του αλγορίθμου.

Σώμα Αλγορίθμου :

1) Βρες τον αριθμό των εγγραφών οι τροχιές των οποίων επικαλύπτονται με την περιοχή Sgeo ή γειτνιάζουν και ανήκουν στον χρήστη με ψευδώνυμο **Id** .

- Εάν ο αριθμός είναι διαφορετικός του μηδέν, επέστρεψε true.

- Αλλιώς επέστρεψε false.

Περιγραφή αλγορίθμου UPDATE_HIST

Σαν είσοδο ο αλγόριθμος δέχεται τις εξής παραμέτρους :

- Το ψευδώνυμο **Id** του αιτούντα
- Την περιοχή ενδιαφέροντος **Sgeo**.
- Τη χρονοσφραγίδα έναρξης **tstart** της περιόδου ενδιαφέροντος .
- Τη χρονοσφραγίδα έναρξης **tend** της περιόδου ενδιαφέροντος .

Ο αλγόριθμος εισάγει στον πίνακα hist (ο οποίος διατηρεί το ιστορικό των επερωτήσεων κάθε χρήστη) μία νέα εγγραφή για τον χρήστη με ψευδώνυμο Id, με περιοχή ενδιαφέροντος Sgeo και χρονική περίοδο tstart - tend.

Ψευδοκώδικας UPDATE_HIST

Είσοδος :

- Το ψευδώνυμο **Id** του χρήστη.

- Η περιοχή ενδιαφέροντος **Sgeo.**

- Η χρονοσφραγίδα έναρξης **tstart** της περιόδου ενδιαφέροντος.

- Η χρονοσφραγίδα έναρξης **tend** της περιόδου ενδιαφέροντος.

Σώμα Αλγορίθμου :

1) Εισήγαγε στον πίνακα hist μία νέα εγγραφή για τον χρήστη με ψευδώνυμο **Id, με** περιοχή ενδιαφέροντος **Sgeo** γι**α** τη χρονική περίοδο **tstart – tend.**

5.4 Πλασματικές τροχιές

Η Query Engine παράγει πλασματικές εγγραφές εφαρμόζοντας τεχνικές παρεμβολής σε πραγματικές εγγραφές. Στη συνέχεια εξετάζουμε δύο είδη παρεμβολής: (α) παρεμβολή αριθμητικών δεδομένων και (β) παρεμβολή τροχιών κινούμενων αντικειμένων. Και στις δύο περιπτώσεις θεωρούμε ότι K – R πλασματικές εγγραφές χρειάζεται να παραχθούν έτσι ώστε μαζί με τις R πραγματικές εγγραφές να διαμορφώσουν ένα σύνολο απάντησης που να βασίζεται σε K χρήστες.

Παρεμβολή αριθμητικών δεδομένων

Η παρεμβολή αριθμητικών δεδομένων αφορά την επιλογή των κατάλληλων τιμών για την ενημέρωση των αριθμητικών γνωρισμάτων μιας πλασματικής εγγραφής. Οι επιλεγμένες τιμές θα πρέπει να βρίσκονται κοντά στις πραγματικές, ενώ ταυτόχρονα θα πρέπει να υπακούουν και στις βασικές στατιστικές ιδιότητες του συνόλου απάντησης: ελάχιστη τιμή, μέγιστη τιμή, μέσος όρος. Έστω ότι θέλουμε να

καθορίσουμε τις τιμές ενός αριθμητικού γνωρίσματος σε ένα σύνολο από πλασματικές εγγραφές. Αρχικά παράγουμε K – R άδειες εγγραφές και τις ομαδοποιούμε, με τυχαίο τρόπο, σε ζεύγη. Στη συνέχεια υπολογίζουμε τον μέσο όρο m των R πραγματικών εγγραφών για το συγκεκριμένο γνώρισμα και ανακαλύπτουμε την (προσημασμένη) μικρότερη d_{min} και τη μεγαλύτερη d_{max} απόκλιση των R εγγραφών από τον μέσο όρο. Για κάθε ζεύγος πλασματικών εγγραφών επιλέγουμε τυχαία δύο τιμές $\pm d \in [d_{min}, d_{max}]$ και θέτουμε την τιμή του γνωρίσματος ίση με m – d για τη μία εγγραφή και m + d για την άλλη. Τέλος, στην περίπτωση που το πλήθος των πλασματικών εγγραφών είναι περιττός αριθμός, θέτουμε την τιμή του γνωρίσματος για την ασύζευκτη εγγραφή ίση με m.

Παρεμβολή τροχιών κινούμενων αντικειμένων

Η παρεμβολή της τροχιάς των κινούμενων αντικειμένων λαμβάνει χώρα ως μια επαναληπτική διαδικασία. Η προτεινόμενη τεχνική επιλέγει τυχαία σε κάθε βήμα ένα ζεύγος τροχιών που μετέχει στο σύνολο απάντησης και ακολούθως σαρώνει τον άξονα x για να συλλέξει τις ενημερώσεις θέσης των δύο τροχιών. Για κάθε ενημέρωση θέσης, ο αλγόριθμος ανακαλύπτει το αντίστοιχο (x, y, t) σημείο στην συζυγή τροχιά, το οποίο έχει την ίδια x συντεταγμένη. Αυτό επιτυγχάνεται με την αξιοποίηση της γνώσης σχετικά με την τεχνική ανακατασκευής που χρησιμοποιήθηκε αρχικά. Έχοντας ανακαλύψει τα δύο σημεία θέσης, ο αλγόριθμος παρεμβολής υπολογίζει ένα νέο σημείο θέσης (x, y’, t), όπου y’ είναι ο μέσος όρος των δύο y συντεταγμένων των ενημερώσεων θέσης. Ακολουθώντας την ίδια διαδικασία επαναληπτικά, ο αλγόριθμος παράγει ένα σύνολο πλασματικών ενημερώσεων θέσης με βάση τις ενημερώσεις θέσης του ζεύγους των τροχιών. Στη συνέχεια, ο αλγόριθμος

«ανακατασκευάζει» την πλασματική τροχιά χρησιμοποιώντας την ίδια μέθοδο με αυτή που εφαρμόζεται στις πραγματικές τροχιές των χρηστών. Ως αποτέλεσμα, με τη βοήθεια των πλασματικών ενημερώσεων θέσης, δημιουργείται μια νέα τροχιά η οποία μοιάζει πολύ με τις υπόλοιπες τροχιές του συνόλου απάντησης. Το τελευταίο βήμα του αλγορίθμου παρεμβολής αφορά την εισαγωγή της πλασματικής τροχιάς στην πλασματική εγγραφή του «χρήστη» και την επανάληψη της ίδιας διαδικασίας έως ότου το σύνολο απάντησης να είναι K– ανώνυμο. Εάν δεν υπάρχουν κοινά x, χρησιμοποιείται η πρώτη τροχιά από το ζεύγος που επιλέχθηκε τυχαία ως πρότυπο για την πλασματική τροχιά αλλάζοντας τα αριθμητικά δεδομένα αυτής.

Ψευδοκώδικας FAGE_GEN

Είσοδος :

- Ο αριθμός M των πλασματικών τροχιών που θέλουμε να δημιουργήσουμε.

- Η έναρξη της χρονικής περιόδου tstart που μας ενδιαφέρει.

- Η λήξη της χρονικής περιόδου tend που μας ενδιαφέρει.

Σώμα Αλγορίθμου :

1) Μετέτρεψε τα tstart, tend από timestamp σε long integer ts1, te2 αντίστοιχα

2) Για k=1, k<=M ,k++ επανέλαβε τις γραμμές 3-9

3) Επέλεξε τυχαία δύο εγγραφές i, j από τον πίνακα T

4) Σάρωσε τον άξονα x των δύο εγγραφών

5) Εάν xi=xj και ts1<=te2 θέσε

- fi.x= xi
- fi.y= yi+yj/2
- fi.t = ts1
- ts1= ts1+100

6) Εάν M%2=1 τότε

- για κάθε αριθμητικό γνώρισμα α θέσε f1.α = AVG(α,T)

Διαφορετικά, για κάθε αριθμητικό γνώρισμα α θέσε

- dmin= AVG(α,T) – MIN(α,Tα)
- dmax= MAX(α,T) - AVG(α,T)

7) Για κάθε ζευγάρι fj,fk θέσε
- d= τυχαία τιμή μεταξύ [0, MIN(dmin,dmax)]
- fj.α = AVG(α,T) – d
- fk.α = AVG(α,T) + d

8) Εισήγαγε τη νέα εγγραφή στον πίνακα T.
9) Εισήγαγε τη νέα εγγραφή στον πίνακα Data.
10) Εάν δεν υπάρχει xi=xj θέσε
- fi.x= xi
- fi.y= yi
- fi.t = ti

11) Εάν M%2=1 τότε

- για κάθε αριθμητικό γνώρισμα α θέσε f1.α = AVG(α,T)

Διαφορετικά, για κάθε αριθμητικό γνώρισμα α θέσε

- dmin= AVG(α,T) – MIN(α,Tα)
- dmax= MAX(α,T) - AVG(α,T)

12) Για κάθε ζευγάρι fj,fk θέσε
- d= τυχαία τιμή μεταξύ [0, MIN(dmin,dmax)]
- fj.α = AVG(α,T) – d
- fk.α = AVG(α,T) + d

13) Εισήγαγε τη νέα εγγραφή στον πίνακα T.
14) Εισήγαγε τη νέα εγγραφή στον πίνακα Data.

Κεφάλαιο 6

ΑΛΓΟΡΙΘΜΟΙ ΤΗΣ QUERY ENGINE ΜΕ ΔΥΝΑΤΟΤΗΤΑ ΠΡΟΣΤΑΣΙΑΣ ΤΗΣ ΙΔΩΤΙΚΟΤΗΤΑΣ

Στο Κεφάλαιο 4 είδαμε τα είδη των επερωτήσεων που υποστηρίζει η Query Engine μας, όπως επίσης παρουσιάστηκε και μια μικρή περιγραφή των αλγορίθμων που υλοποιούν τις επερωτήσεις αυτές. Στο προηγούμενο κεφάλαιο συνοψίσαμε τις πιθανές επιθέσεις που μπορεί να έχει η βάση μας μέσω αυτών των επερωτήσεων και παρουσιάσαμε τους αλγορίθμους που επιφέρουν προστασία στις επιθέσεις αυτές και κατά συνέπεια και στην ιδιωτικότητα των χρηστών. Σε αυτό το κεφάλαιο παρουσιάζεται ο ψευδοκώδικας των αλγορίθμων που είδαμε στο Κεφάλαιο 4, οι οποίοι κάνουν χρήση των μεθόδων προστασίας που είδαμε στο Κεφάλαιο 5.

6.1 Αλγόριθμος count query

Όπως έχουμε προαναφέρει, ο αλγόριθμος της **count query** επιστρέφει τον αριθμό των χρηστών που ικανοποιούν ένα συγκεκριμένο περιορισμό, ο οποίος δίνεται από τον χρήστη που κάνει την επερώτηση. Στην γραμμή 1, λοιπόν, υπολογίζεται ο αριθμός αυτός με βάση τον περιορισμό που έδωσε ο χρήστης. Επειδή όμως στην επερώτηση μας θέλουμε να παρέχουμε και κάποιο βαθμό ανωνυμίας, ο αλγόριθμος μας , όπως φαίνεται και στη γραμμή 2, ελέγχει εάν ο αριθμός αυτός που βρήκαμε καλύπτει τον βαθμό ανωνυμίας που επιθυμεί ο χρήστης. Εάν ο βαθμός καλύπτεται

επιστρέφεται ο αριθμός των χρηστών που βρήκαμε στη γραμμή 1, διαφορετικά επιστρέφεται ένα μήνυμα παραβίασης της ανωνυμίας.

Αλγόριθμος count query

Είσοδος :

- Το αλφαριθμητικό **cond,** το οποίο προσδιορίζει τον περιορισμό

- Η σταθερά **K**

Έξοδος :

-Η Boolean τιμή της μεταβλητής **ans** που δηλώνει την επιτυχία ή αποτυχία του αλγορίθμου

Σώμα Αλγορίθμου :

1) Βρες στον πίνακα data τον αριθμό των χρηστών για τους οποίους ισχύει ο περιορισμός **cond**

2) Εάν ο αριθμός των χρηστών είναι μεγαλύτερος ή ίσο της σταθεράς K, θέσε 'ans = TRUE'. Διαφορετικά επέστρεψε το μήνυμα "K-anonymity violation".

6.2 Αλγόριθμος aggregate query

Όπως έχουμε προαναφέρει, ο αλγόριθμος της aggregate query επιστρέφει ένα στατιστικό δεδομένο (AVG,SUM,MAX,MIN) κάποιου γνωρίσματος των χρηστών, που ικανοποιούν ένα συγκεκριμένο περιορισμό. Στην γραμμή 1, λοιπόν, υπολογίζεται ο αριθμός των χρηστών που ικανοποιούν τον περιορισμό που έδωσε ο χρήστης, όπως επίσης και το στατιστικό δεδομένο που επιθυμούμε για κάποιο γνώρισμα αυτών. Επειδή όμως στην επερώτηση μας θέλουμε να παρέχουμε και κάποιο βαθμό ανωνυμίας, ο αλγόριθμος μας , όπως φαίνεται και στη γραμμή 2, ελέγχει εάν ο αριθμός των χρηστών που βρήκαμε καλύπτει τον βαθμό ανωνυμίας που επιθυμεί ο χρήστης. Εάν ο βαθμός καλύπτεται επιστρέφεται το στατιστικό δεδομένο που

βρήκαμε στη γραμμή 1, διαφορετικά επιστρέφεται ένα μήνυμα παραβίασης της ανωνυμίας.

Αλγόριθμος aggregate query

Είσοδος :

- Το αλφαριθμητικό **func,** το οποίο προσδιορίζει την SQL aggregate function

- Το αλφαριθμητικό **attr,** το οποίο προσδιορίζει το γνώρισμα για το οποίο θα γίνει η κλήση της SQL aggregate function

- Το αλφαριθμητικό **cond,** το οποίο προσδιορίζει τον περιορισμό

- Η σταθερά **K**

Έξοδος :

-Η integer τιμή της μεταβλητής **ans** που δηλώνει το ζητούμενο στατιστικό δεδομένο.

Σώμα Αλγορίθμου :

1) Βρες στον πίνακα data το στατιστικό δεδομένο για το επιθυμητό γνώρισμα των χρηστών για τους οποίους ισχύει ο περιορισμός **cond,** καθώς και τον αριθμό αυτών

2) Εάν ο αριθμός των χρηστών είναι μεγαλύτερος ή ίσος της σταθεράς K, θέσε 'ans = στατιστικό δεδομένο'. Διαφορετικά επέστρεψε το μήνυμα "K-anonymity violation".

6.3 Αλγόριθμος predicative data query

Όπως έχουμε προαναφέρει, ο αλγόριθμος της predicative data query επιστρέφει τον αριθμό των χρηστών που έχουν μια συγκεκριμένη τιμή σε κάποιο κατηγορηματικό δεδομένο της βάσης. Στην γραμμή 1, λοιπόν, υπολογίζεται ο αριθμός των χρηστών που ικανοποιούν τον περιορισμό που έδωσε ο χρήστης. Επειδή όμως στην επερώτηση μας θέλουμε να παρέχουμε και κάποιο βαθμό ανωνυμίας, ο αλγόριθμος μας , όπως φαίνεται και στη γραμμή 2, ελέγχει εάν ο αριθμός των χρηστών που βρήκαμε

καλύπτει τον βαθμό ανωνυμίας που επιθυμεί ο χρήστης. Εάν ο βαθμός καλύπτεται επιστρέφεται το ο αριθμός των χρηστών που βρήκαμε στη γραμμή 1, διαφορετικά επιστρέφεται ένα μήνυμα παραβίασης της ανωνυμίας.

Αλγόριθμος predicative data query

Είσοδος :

- Το αλφαριθμητικό **cond,** το οποίο προσδιορίζει τον περιορισμό για ένα κατηγορηματικό γνώρισμα

- Η σταθερά **K**

Έξοδος :

-Η Boolean τιμή της μεταβλητής **ans** που δηλώνει την επιτυχία ή αποτυχία του αλγορίθμου

Σώμα Αλγορίθμου :

1) Βρες στον πίνακα data τον αριθμό των χρηστών για τους οποίους ισχύει ο περιορισμός **cond**

2) Εάν ο αριθμός των χρηστών είναι μεγαλύτερος ή ίσος της σταθεράς K, θέσε 'ans = TRUE'. Διαφορετικά επέστρεψε το μήνυμα "K-anonymity violation".

6.4 Αλγόριθμος range query

Όπως έχουμε προαναφέρει, ο αλγόριθμος της range query επιστρέφει τις τροχιές των ατόμων που είναι αποθηκευμένες στη βάση, οι οποίες βρίσκονται εντός μιας προκαθορισμένης χωρο-χρονικής περιοχής. Αποτελείται από τρία τμήματα. Το πρώτο τμήμα (γραμμή 1) περιλαμβάνει τον έλεγχο για πιθανή στοχευμένη επίθεση. Για να το πετύχει αυτό ο αλγόριθμος καλεί τη συνάρτηση In_Hist (η λειτουργία της περιγράφηκε στην Ενότητα 5.3).Στη γραμμή 2, δημιουργείται ένας βοηθητικός πίνακας tmp στον οποίο αποθηκεύονται όλες οι τροχιές σε μορφή Moving point, που ικανοποιούν την

χρονική περίοδο που έδωσε ο χρήστης. Στη γραμμή 3, κατασκευάζεται ένας ακόμα βοηθητικός πίνακας tmp1, προκειμένου να μετατραπούν οι τροχιές που βρέθηκαν στη γραμμή 2 σε μορφή sdo_trajectory. Η μετατροπή αυτή είναι απαραίτητη για την επεξεργασία των δεδομένων μας. Στη γραμμή 4, ενημερώνονται τα μεταδεδομένα και κατασκευάζεται ευρετήριο για τον πίνακα που δημιουργήθηκε στη γραμμή 3. Με τη δημιουργία ευρετηρίου έχουμε τη δυνατότητα να καλέσουμε χωρικές συναρτήσεις της Oracle για τα χωρικά δεδομένα μας. Στη γραμμή 5, χρησιμοποιούμε τη συνάρτηση της Oracle SDO_GEOM_RELATE, με μάσκα COVERS και INSIDE, προκειμένου να βρούμε τις τροχιές που περιέχονται μέσα στην προκαθορισμένη από το χρήστη περιοχή. Στη γραμμή 6, εισάγουμε σε ένα πίνακα T τις τροχιές των χρηστών που βρήκαμε σε μορφή moving_point. Για να το πετύχουμε αυτό βρίσκουμε τις τροχιές του πίνακα tmp που αντιστοιχούν στις τροχιές του tmp1. Στο τρίτο τμήμα, (γραμμή 7) διασφαλίζουμε την K ανωνυμία που ζήτησε ο χρήστης. Για να το πετύχουμε αυτό ελέγχουμε εάν ο αριθμός των χρηστών στους οποίους αντιστοιχούν οι τροχιές που είναι αποθηκευμένες στον πίνακα T είναι ίσος ή μεγαλύτερος από τον αριθμό K που προσδιορίζει τον βαθμό ανωνυμίας. Εάν ισχύει αυτό, παρουσιάζονται στο χρήστη τα δεδομένα του πίνακα T, διαφορετικά καλείται η συνάρτηση Fake_Gen (η λειτουργία της περιγράφηκε στην Ενότητα 5.4) για να παράγει τις πλασματικές τροχιές που είναι απαραίτητες για να καλυφθεί ο βαθμός ανωνυμίας. Οι πλασματικές αυτές τροχιές εισάγονται στον πίνακα T και παρουσιάζονται μαζί με τις αληθινές στο χρήστη. Τέλος στη γραμμή 8, ο πίνακας T αδειάζει από δεδομένα.

Αλγόριθμος range query

Είσοδος :

- Το ψευδώνυμο **Id** του χρήστη

- Η περιοχή **Sgeo** που μας ενδιαφέρει

- Η χρονιά **YearStart** έναρξης της περιόδου που μας ενδιαφέρει.

- Ο μήνας **MonthStart** έναρξης της περιόδου που μας ενδιαφέρει.

- Η ημέρα **DayStar**t έναρξης της περιόδου που μας ενδιαφέρει.

- Η ώρα **HourStart** έναρξης της περιόδου που μας ενδιαφέρει.

- Το λεπτό **MinStart** έναρξης της περιόδου που μας ενδιαφέρει.

- Το δευτερόλεπτο **SecEnd** έναρξης της περιόδου που μας ενδιαφέρει.

- Η χρονιά **YearEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Ο μήνας **MonthEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Η ημέρα **DayEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Η ώρα **HourEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Το λεπτό **MinEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Το δευτερόλεπτο **SecEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Η σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας.

Έξοδος :

- Οι τροχιές των χρηστών που βρίσκονται μέσα στην ορισμένη περιοχή.

Σώμα Αλγορίθμου :

1) Κάλεσε την συνάρτηση In_Hist με μεταβλητές το ψευδώνυμο του χρήστη, την περιοχή και το χρονικό διάστημα ενδιαφέροντος, για να ελέγξει εάν στον πίνακα hist υπάρχει ο ίδιος χρήστης με επερώτηση σε περιοχή που επικαλύπτεται ή γειτνιάζει με την καινούργια περιοχή ενδιαφέροντος.

- Εάν υπάρχει, επέστρεψε το μήνυμα “Privacy threat” και τερμάτισε την λειτουργία.

- Εάν όχι, πήγαινε στο βήμα 2

2) Δημιούργησε έναν βοηθητικό πίνακα tmp στον οποίο αποθήκευσε τις τροχιές του πίνακα data που ικανοποιούν τη χρονική περίοδο που μας ενδιαφέρει.

3) Κατασκεύασε έναν δεύτερο βοηθητικό πίνακα tmp1, στον οποίο μετέτρεψε τα moving_points που περιγράφουν την τροχιά των χρηστών του πίνακα tmp, σε sdo_geometry δεδομένα μέσω της συνάρτησης f_trajectory().

4) Ενημέρωσε τα μεταδεδομένα και κατασκεύασε χωρικό ευρετήριο

5) Βρες τους χρήστες στον πίνακα tmp1, για τις τροχιές των οποίων η συνάρτηση SDO_GEOM_RELATE με μάσκα COVERS ή η συνάρτηση SDO_GEOM_RELATE με μάσκα INSIDE ισχύει.

6) Βρες στον πίνακα tmp,τις τροχιές των χρηστών που βρήκες παραπάνω εισήγαγε τες στον πίνακα T.

7)Έλεγξε εάν ο αριθμός R των χρηστών, των οποίων οι τροχιές βρίσκονται αποθηκευμένες στον πίνακα T, είναι μεγαλύτερος ή ίσος του K.

- Εάν ναι, επέστρεψε τις τροχιές των χρηστών που βρίσκονται στον πίνακα T.

- Εάν όχι, εισήγαγε στον πίνακα T ,μέσω της Fake_Gen, K-R πλασματικές τροχιές και ανανέωσε τον πίνακα hist με μεταβλητές το ψευδώνυμο του χρήστη, την περιοχή και το χρονικό διάστημα ενδιαφέροντος. Τέλος, επέστρεψε τις τροχιές των χρηστών που βρίσκονται στον πίνακα T.

8) Διέγραψε τα δεδομένα του πίνακα T.

6.5 Αλγόριθμος distance query

Ο αλγόριθμος της distance query επιστρέφει τις τροχιές των χρηστών που είναι αποθηκευμένες στη βάση, οι οποίες είναι εντός προκαθορισμένης απόστασης d από κάποιο σημείο αναφοράς. Αποτελείται από δύο τμήματα. Το πρώτο τμήμα (γραμμή 1) περιλαμβάνει τη δημιουργία ενός κύκλου με κέντρο το σημείο αναφοράς που δόθηκε από το χρήστη και ακτίνα την προκαθορισμένη απόσταση που επιθυμούμε από το σημείο αυτό. Με αυτό τον τρόπο ορίζουμε ουσιαστικά την περιοχή κάλυψης στην οποία θέλουμε να βρούμε τις τροχιές των χρηστών. Για αυτό τον λόγο στο δεύτερο τμήμα (γραμμή 2) καλούμε τον αλγόριθμο range query με περιοχή ενδιαφέροντος τον κύκλο που δημιουργήσαμε. Όπως είναι κατανοητό, η του αλγορίθμου συνεχίζεται όπως περιγράφεται παραπάνω στον αλγόριθμο range query.

Αλγόριθμος distance query

Είσοδος :

- Το ψευδώνυμο **Id** του χρήστη

- Η οριζόντια συντεταγμένη **Xp**

- Η κατακόρυφη συντεταγμένη **Yp**

- Η απόσταση **d**

- Η χρονιά **YearStart** έναρξης της περιόδου που μας ενδιαφέρει.

- Ο μήνας **MonthStart** έναρξης της περιόδου που μας ενδιαφέρει.

- Η ημέρα **DayStar**t έναρξης της περιόδου που μας ενδιαφέρει.

- Η ώρα **HourStart** έναρξης της περιόδου που μας ενδιαφέρει.

- Το λεπτό **MinStart** έναρξης της περιόδου που μας ενδιαφέρει.

- Το δευτερόλεπτο **SecEnd** έναρξης της περιόδου που μας ενδιαφέρει.

- Η χρονιά **YearEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Ο μήνας **MonthEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Η ημέρα **DayEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Η ώρα **HourEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Το λεπτό **MinEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Το δευτερόλεπτο **SecEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Η σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας.

Έξοδος :

- Οι τροχιές των χρηστών που βρίσκονται μέσα στον κύκλο που ορίσαμε.

Σώμα Αλγορίθμου :

1) Δημιούργησε έναν κύκλο με κέντρο το σημείο (Xp,Yp) και ακτίνα d.

2) Κάλεσε τη συνάρτηση range_query και θέσε ως μεταβλητή περιοχής ενδιαφέροντος τον κύκλο που δημιούργησες στο βήμα 1.

6.6 Αλγόριθμος KNN query

Ό αλγόριθμος της KNN query επιστρέφει τις τροχιές των κ πλησιέστερων (K–NN) γειτόνων για μια χρονική περίοδο με βάση κάποιο σημείο αναφοράς. Αποτελείται από τρία τμήματα. Στο πρώτο τμήμα (γραμμή 1), δημιουργείται ένας βοηθητικός πίνακας tmp στον οποίο αποθηκεύονται όλες οι τροχιές σε μορφή Moving point, που ικανοποιούν την χρονική περίοδο που έδωσε ο χρήστης. Στη γραμμή 2, κατασκευάζεται ένας ακόμα βοηθητικός πίνακας tmp1, προκειμένου να μετατραπούν οι τροχιές που βρέθηκαν στη γραμμή 2 σε μορφή sdo_trajectory. Η μετατροπή αυτή είναι απαραίτητη για την επεξεργασία των δεδομένων μας. Στη γραμμή 3, ενημερώνονται τα μεταδεδομένα και κατασκευάζεται ευρετήριο για τον πίνακα που δημιουργήθηκε στη γραμμή 2. Με τη δημιουργία ευρετηρίου έχουμε τη δυνατότητα να καλέσουμε χωρικές συναρτήσεις της Oracle για τα χωρικά δεδομένα μας. Στη γραμμή 4, χρησιμοποιούμε τη συνάρτηση της Oracle SDO_NN, προκειμένου να βρούμε τις τροχιές των κ κοντινότερων γειτόνων. Στη γραμμή 5, εισάγουμε σε ένα πίνακα T, σε μορφή moving_point, τις τροχιές των χρηστών που βρήκαμε. Για να το πετύχουμε αυτό βρίσκουμε τις τροχιές του πίνακα tmp που αντιστοιχούν στις τροχιές του tmp1. Το δεύτερο τμήμα (γραμμή 5-6) περιλαμβάνει τον έλεγχο για πιθανή στοχευμένη επίθεση. Για να το πετύχει αυτό ο αλγόριθμος κατασκευάζει ο κουτί οριοθέτησης(MBR) των τροχιών που περιλαμβάνονται στον πίνακα T (γραμμή 5), και καλεί τη συνάρτηση In_Hist (η λειτουργία της περιγράφηκε στην ενότητα 5.3) με τιμή για το όρισμα της περιοχής ενδιαφέροντος το κουτί αυτό. Στο τρίτο τμήμα, (γραμμή 7) διασφαλίζουμε την K ανωνυμία που ζήτησε ο χρήστης. Για να το πετύχουμε αυτό ελέγχουμε εάν ο αριθμός των χρηστών στους οποίους αντιστοιχούν οι τροχιές που είναι αποθηκευμένες στον πίνακα T είναι ίσος ή μεγαλύτερος από τον αριθμό K που

προσδιορίζει τον βαθμό ανωνυμίας. Εάν ισχύει αυτό, παρουσιάζονται στο χρήστη τα δεδομένα του πίνακα T, διαφορετικά καλείται η συνάρτηση Fake_Gen (η λειτουργία της περιγράφηκε στην Ενότητα 5.4) για να παράγει τις πλασματικές τροχιές που είναι απαραίτητες για να καλυφθεί ο βαθμός ανωνυμίας. Οι πλασματικές αυτές τροχιές εισάγονται στον πίνακα T και παρουσιάζονται μαζί με τις αληθινές στο χρήστη. Τέλος στη γραμμή 8, ο πίνακας T αδειάζει από δεδομένα.

Αλγόριθμος KNN query

Είσοδος :

- Το ψευδώνυμο **Id** του χρήστη.
- Το σημείο αναφοράς **P** που μας ενδιαφέρει.
- Η χρονιά **YearStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Ο μήνας **MonthStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Η ημέρα **DayStar**t έναρξης της περιόδου που μας ενδιαφέρει.
- Η ώρα **HourStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Το λεπτό **MinStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Το δευτερόλεπτο **SecEnd** έναρξης της περιόδου που μας ενδιαφέρει.
- Η χρονιά **YearEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Ο μήνας **MonthEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Η ημέρα **DayEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Η ώρα **HourEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Το λεπτό **MinEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Το δευτερόλεπτο **SecEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Η σταθερά **κ** που καθορίζει τον αριθμό τον γειτόνων.
- Η σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας.

Έξοδος :

- Οι τροχιές των κ κοντινότερων γειτόνων

Σώμα Αλγορίθμου :

1) Δημιούργησε έναν βοηθητικό πίνακα tmp στον οποίο αποθήκευσε τις τροχιές του πίνακα data που ικανοποιούν τη χρονική περίοδο που μας ενδιαφέρει.

2) Κατασκεύασε έναν δεύτερο βοηθητικό πίνακα tmp1, στον οποίο μετέτρεψε τα moving_points που περιγράφουν την τροχιά των χρηστών στον πίνακα tmp,σε sdo_geometry δεδομένα μέσω της συνάρτησης f_trajectory().

3) Ενημέρωσε τα μεταδεδομένα και κατασκεύασε χωρικό ευρετήριο

4) Βρες τις τροχιές των κ κοντινότερων χρηστών στη ζητούμενη χρονική περιοχή, και εισήγαγε τες στον πίνακα T.

5) Βρες το κουτί οριοθέτησης των τροχιών που βρήκες στο βήμα 5.

6) Κάλεσε την συνάρτηση In_Hist με μεταβλητές το ψευδώνυμο του χρήστη, το κουτί οριοθέτησης και το χρονικό διάστημα ενδιαφέροντος, για να ελέγξει εάν στον πίνακα hist υπάρχει ο ίδιος χρήστης με επερώτηση σε περιοχή που επικαλύπτεται ή γειτνιάζει με την με το κουτί οριοθέτησης.

- Εάν υπάρχει, επέστρεψε το μήνυμα “Privacy threat” και τερμάτισε την λειτουργία.

- Εάν όχι, πήγαινε στο βήμα 7.

7) Έλεγξε εάν ο αριθμός R των χρηστών, των οποίων οι τροχιές βρίσκονται αποθηκευμένες στον πίνακα T, είναι μεγαλύτερος ή ίσος του K.

- Εάν ναι, επέστρεψε τις τροχιές των χρηστών που βρίσκονται στον πίνακα T.

- Εάν όχι, εισήγαγε στον πίνακα T ,μέσω της Fake_Gen, K-R πλασματικές τροχιές και ανανέωσε τον πίνακα hist με μεταβλητές το ψευδώνυμο του χρήστη, την περιοχή και το χρονικό διάστημα ενδιαφέροντος. Τέλος, επέστρεψε τις τροχιές των χρηστών που βρίσκονται στον πίνακα T.

8) Διέγραψε τα δεδομένα του πίνακα T.

6.7 Αλγόριθμος Landmark query

Ό αλγόριθμος της Landmark query επιστρέφει τις τροχιές των χρηστών που είναι καταγεγραμμένες στη βάση δεδομένων και οι οποίες έχουν προκαθορισμένο σημείο εκκίνησης/κατάληξης και/ή διέρχονται από ένα ή περισσότερα προκαθορισμένα ενδιάμεσα σημεία ενδιαφέροντος. Αποτελείται από τρία τμήματα. Στο πρώτο τμήμα (γραμμή 1), δημιουργείται ένας βοηθητικός πίνακας tmp στον οποίο αποθηκεύονται όλες οι τροχιές σε μορφή Moving point, που ικανοποιούν την χρονική περίοδο που έδωσε ο χρήστης. Στη γραμμή 2, κατασκευάζεται ένας ακόμα βοηθητικός πίνακας tmp1, προκειμένου να μετατραπούν οι τροχιές που βρέθηκαν στη γραμμή 2 σε μορφή sdo_trajectory. Η μετατροπή αυτή είναι απαραίτητη για την επεξεργασία των δεδομένων μας. Στη γραμμή 3, ενημερώνονται τα μεταδεδομένα και κατασκευάζεται ευρετήριο για τον πίνακα που δημιουργήθηκε στη γραμμή 2. Με τη δημιουργία ευρετηρίου έχουμε τη δυνατότητα να καλέσουμε χωρικές συναρτήσεις της Oracle για τα χωρικά δεδομένα μας. Στη γραμμή 4, χρησιμοποιούμε τη συνάρτηση της Oracle SDO_ANYINTERACT, προκειμένου να βρούμε τις τροχιές των χρηστών του πίνακα tmp1 που περιέχουν όλα ή κάποια από τα σημεία ενδιαφέροντος. Στη γραμμή 5, χρησιμοποιούμε τη συνάρτηση της Oracle SDO_ANYINTERACT για τις τροχιές που βρήκαμε στη γραμμή 4, προκειμένου να κρατήσουμε τις τροχιές των χρηστών του που περιέχουν το σημείο έναρξης και το σημείο τερματισμού. Στη γραμμή 6, εισάγουμε σε ένα πίνακα T, σε μορφή moving_point, τις τροχιές των χρηστών που βρήκαμε στη γραμμή 5. Για να το πετύχουμε αυτό βρίσκουμε τις τροχιές του πίνακα tmp που αντιστοιχούν στις τροχιές του tmp1. Το δεύτερο τμήμα (γραμμή 7-8) περιλαμβάνει τον έλεγχο για πιθανή στοχευμένη επίθεση. Για να το πετύχει αυτό ο αλγόριθμος κατασκευάζει ο κουτί οριοθέτησης(MBR) των τροχιών που

περιλαμβάνονται στον πίνακα Τ (γραμμή 7), και καλεί τη συνάρτηση In_Hist (η λειτουργία της περιγράφηκε στην Ενότητα 5.3) με τιμή για το όρισμα της περιοχής ενδιαφέροντος το κουτί αυτό (γραμμή 8). Στο τρίτο τμήμα, (γραμμή 9) διασφαλίζουμε την Κ ανωνυμία που ζήτησε ο χρήστης. Για να το πετύχουμε αυτό ελέγχουμε εάν ο αριθμός των χρηστών στους οποίους αντιστοιχούν οι τροχιές που είναι αποθηκευμένες στον πίνακα Τ είναι ίσος ή μεγαλύτερος από τον αριθμό Κ που προσδιορίζει τον βαθμό ανωνυμίας. Εάν ισχύει αυτό, παρουσιάζονται στο χρήστη τα δεδομένα του πίνακα Τ, διαφορετικά καλείται η συνάρτηση Fake_Gen (η λειτουργία της περιγράφηκε στην Ενότητα 5.4) για να παράγει τις πλασματικές τροχιές που είναι απαραίτητες για να καλυφθεί ο βαθμός ανωνυμίας. Οι πλασματικές αυτές τροχιές εισάγονται στον πίνακα Τ και παρουσιάζονται μαζί με τις αληθινές στο χρήστη. Τέλος στη γραμμή 10, ο πίνακας Τ αδειάζει από δεδομένα.

Αλγόριθμος Landmark query

Είσοδος :

- Το ψευδώνυμο **Id** του χρήστη.
- Η χρονιά **YearStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Ο μήνας **MonthStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Η ημέρα **DayStar**t έναρξης της περιόδου που μας ενδιαφέρει.
- Η ώρα **HourStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Το λεπτό **MinStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Το δευτερόλεπτο **SecEnd** έναρξης της περιόδου που μας ενδιαφέρει.
- Η χρονιά **YearEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Ο μήνας **MonthEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Η ημέρα **DayEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Η ώρα **HourEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Το λεπτό **MinEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Το δευτερόλεπτο **SecEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Τα σημεία **geom={P1, P2, …, PN}** που μας ενδιαφέρουν.

- Η σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας.

Έξοδος :

- Οι τροχιές των χρηστών που ξεκινούν/καταλήγουν από/σε ένα συγκεκριμένο σημείο και διέρχονται από ένα ή περισσότερα προκαθορισμένα ενδιάμεσα σημεία ενδιαφέροντος.

Σώμα Αλγορίθμου :

1) Δημιούργησε έναν βοηθητικό πίνακα tmp στον οποίο αποθήκευσε τις τροχιές του πίνακα data που ικανοποιούν τη χρονική περίοδο που μας ενδιαφέρει.

2) Κατασκεύασε έναν δεύτερο βοηθητικό πίνακα tmp1, στον οποίο μετέτρεψε τα moving_points που περιγράφουν την τροχιά των χρηστών στον πίνακα tmp,σε sdo_geometry δεδομένα μέσω της συνάρτησης f_trajectory().

3) Ενημέρωσε τα μεταδεδομένα και κατασκεύασε χωρικό ευρετήριο

4) Βρες στον πίνακα tmp1, τους χρήστες, για τις τροχιές των οποίων ισχύει η συνάρτηση SDO_ANYINTERACT για όλα τα σημεία .

5) Από αυτούς, βρες στον πίνακα tmp1 τους χρήστες για τις τροχιές των οποίων ισχύει η συνάρτηση SDO_ANYINTERACT για τα σημεία P1, PN.

6)Βρες στον πίνακα tmp, τις τροχιές των χρηστών που βρήκες στο βήμα 5 με σημείο έναρξης το P1 και σημείο προορισμού το PN , και εισήγαγε τες στον πίνακα T

7) Βρες το κουτί οριοθέτησης των τροχιών που βρήκες στο βήμα 5.

8) Κάλεσε την συνάρτηση In_Hist με μεταβλητές το ψευδώνυμο του χρήστη, το κουτί οριοθέτησης και το χρονικό διάστημα ενδιαφέροντος, για να ελέγξει εάν στον πίνακα hist υπάρχει ο ίδιος χρήστης με επερώτηση σε περιοχή που επικαλύπτεται ή γειτνιάζει με την με το κουτί οριοθέτησης.

- Εάν υπάρχει, επέστρεψε το μήνυμα “Privacy threat” και τερμάτισε την λειτουργία.

- Εάν όχι, πήγαινε στο βήμα 7.

9) Έλεγξε εάν ο αριθμός R των χρηστών, των οποίων οι τροχιές βρίσκονται αποθηκευμένες στον πίνακα T, είναι μεγαλύτερος ή ίσος του K.

- Εάν ναι, επέστρεψε τις τροχιές των χρηστών που βρίσκονται στον πίνακα T.

- Εάν όχι, εισήγαγε στον πίνακα T ,μέσω της Fake_Gen, K-R πλασματικές τροχιές και ανανέωσε τον πίνακα hist με μεταβλητές το ψευδώνυμο του χρήστη, την περιοχή και το χρονικό διάστημα ενδιαφέροντος. Τέλος, επέστρεψε τις τροχιές των χρηστών που βρίσκονται στον πίνακα T.

10) Διέγραψε τα δεδομένα του πίνακα T.

6.8 Αλγόριθμος Route query

Ό αλγόριθμος της Route query επιστρέφει τις τροχιές των χρηστών που ταυτίζονται με κάποια προκαθορισμένη πορεία. Αποτελείται από τρία τμήματα. Στο πρώτο τμήμα (γραμμή 1), δημιουργείται ένας βοηθητικός πίνακας tmp στον οποίο αποθηκεύονται όλες οι τροχιές σε μορφή Moving point, που ικανοποιούν την χρονική περίοδο που έδωσε ο χρήστης. Στη γραμμή 2, κατασκευάζεται ένας ακόμα βοηθητικός πίνακας tmp1, προκειμένου να μετατραπούν οι τροχιές που βρέθηκαν στη γραμμή 2 σε μορφή sdo_trajectory. Η μετατροπή αυτή είναι απαραίτητη για την επεξεργασία των δεδομένων μας. Στη γραμμή 3, ενημερώνονται τα μεταδεδομένα και κατασκευάζεται ευρετήριο για τον πίνακα που δημιουργήθηκε στη γραμμή 2. Με τη δημιουργία ευρετηρίου έχουμε τη δυνατότητα να καλέσουμε χωρικές συναρτήσεις της Oracle για τα χωρικά δεδομένα μας. Στη γραμμή 4, χρησιμοποιούμε τη συνάρτηση της Oracle SDO_ANYINTERACT, προκειμένου να βρούμε τις τροχιές που περιέχουν τα σημεία της πορείας που μας ενδιαφέρει. Στη γραμμή 5, εισάγουμε σε ένα πίνακα T, σε μορφή moving_point, τις τροχιές των χρηστών που βρήκαμε στη γραμμή 4, για τις οποίες όμως τα σημεία είναι ακριβώς τα ίδια με την πορεία ενδιαφέροντος. Για να το πετύχουμε αυτό βρίσκουμε τις τροχιές του πίνακα tmp που αντιστοιχούν στις τροχιές του tmp1. Το δεύτερο τμήμα (γραμμή 6-7) περιλαμβάνει τον έλεγχο για πιθανή στοχευμένη επίθεση. Για να το πετύχει αυτό ο αλγόριθμος κατασκευάζει ο κουτί οριοθέτησης(MBR) των τροχιών που περιλαμβάνονται στον πίνακα T (γραμμή 6), και

καλεί τη συνάρτηση In_Hist (η λειτουργία της περιγράφηκε στην Ενότητα 5.3) με τιμή για το όρισμα της περιοχής ενδιαφέροντος το κουτί αυτό (γραμμή 7). Στο τρίτο τμήμα, (γραμμή 8) διασφαλίζουμε την K ανωνυμία που ζήτησε ο χρήστης. Για να το πετύχουμε αυτό ελέγχουμε εάν ο αριθμός των χρηστών στους οποίους αντιστοιχούν οι τροχιές που είναι αποθηκευμένες στον πίνακα T είναι ίσος ή μεγαλύτερος από τον αριθμό K που προσδιορίζει τον βαθμό ανωνυμίας. Εάν ισχύει αυτό, παρουσιάζονται στο χρήστη τα δεδομένα του πίνακα T, διαφορετικά καλείται η συνάρτηση Fake_Gen (η λειτουργία της περιγράφηκε στην Ενότητα 5.4) για να παράγει τις πλασματικές τροχιές που είναι απαραίτητες για να καλυφθεί ο βαθμός ανωνυμίας. Οι πλασματικές αυτές τροχιές εισάγονται στον πίνακα T και παρουσιάζονται μαζί με τις αληθινές στο χρήστη. Τέλος στη γραμμή 9, ο πίνακας T αδειάζει από δεδομένα.

Αλγόριθμος Route query

Είσοδος :

- Το ψευδώνυμο **Id** του χρήστη.
- Η χρονιά **YearStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Ο μήνας **MonthStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Η ημέρα **DayStar**t έναρξης της περιόδου που μας ενδιαφέρει.
- Η ώρα **HourStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Το λεπτό **MinStart** έναρξης της περιόδου που μας ενδιαφέρει.
- Το δευτερόλεπτο **SecEnd** έναρξης της περιόδου που μας ενδιαφέρει.
- Η χρονιά **YearEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Ο μήνας **MonthEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Η ημέρα **DayEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Η ώρα **HourEnd** λήξης της περιόδου που μας ενδιαφέρει.
- Το λεπτό **MinEnd** λήξης της περιόδου που μας ενδιαφέρει.

- Το δευτερόλεπτο **SecEnd** λήξης της περιόδου που μας ενδιαφέρει..

- Η σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας.

- Τα διαδρομή **R** ενδιαφέροντος.

Έξοδος :

- Οι τροχιές των χρηστών που ταυτίζονται με την δοθείσα διαδρομή.

Σώμα Αλγορίθμου :

1) Δημιούργησε έναν βοηθητικό πίνακα tmp στον οποίο αποθήκευσε τις τροχιές του πίνακα data που ικανοποιούν τη χρονική περίοδο που μας ενδιαφέρει.

2) Κατασκεύασε έναν δεύτερο βοηθητικό πίνακα tmp1, στον οποίο μετέτρεψε τα moving_points που περιγράφουν την τροχιά των χρηστών στον πίνακα tmp,σε sdo_geometry δεδομένα μέσω της συνάρτησης f_trajectory().

3) Ενημέρωσε τα μεταδεδομένα και κατασκεύασε χωρικό ευρετήριο

4) Βρες τις τροχιές των χρηστών που έχουν οποιαδήποτε σχέση με τα σημεία της διαδρομής R.

5) Βάλε στον πίνακα T μόνος τις τροχιές που βρήκες στο βήμα 4,οι οποίες ταυτίζονται με τη διαδρομή R.

5) Βρες το κουτί οριοθέτησης των τροχιών που βρήκες στο βήμα 5.

6) Κάλεσε την συνάρτηση In_Hist με μεταβλητές το ψευδώνυμο του χρήστη, το κουτί οριοθέτησης και το χρονικό διάστημα ενδιαφέροντος, για να ελέγξει εάν στον πίνακα hist υπάρχει ο ίδιος χρήστης με επερώτηση σε περιοχή που επικαλύπτεται ή γειτνιάζει με την με το κουτί οριοθέτησης.

- Εάν υπάρχει, επέστρεψε το μήνυμα “Privacy threat” και τερμάτισε την λειτουργία.

- Εάν όχι, πήγαινε στο βήμα 7.

7) Έλεγξε εάν ο αριθμός R των χρηστών, των οποίων οι τροχιές βρίσκονται αποθηκευμένες στον πίνακα T, είναι μεγαλύτερος ή ίσος του K.

- Εάν ναι, επέστρεψε τις τροχιές των χρηστών που βρίσκονται στον πίνακα T.

- Εάν όχι, εισήγαγε στον πίνακα T ,μέσω της Fake_Gen, K-R πλασματικές τροχιές και ανανέωσε τον πίνακα hist με μεταβλητές το ψευδώνυμο του χρήστη, την περιοχή και το χρονικό διάστημα ενδιαφέροντος. Τέλος, επέστρεψε τις τροχιές των χρηστών που βρίσκονται στον πίνακα T.

8) Διέγραψε τα δεδομένα του πίνακα T.

6.4 Αλγόριθμος range-time query

Όπως έχουμε προαναφέρει, ο αλγόριθμος της range query επιστρέφει τις τροχιές των χρηστών που είναι αποθηκευμένες στη βάση, οι οποίες βρίσκονται εντός μιας προκαθορισμένης χωρικής περιοχής, και ισχύουν για τη διάρκεια μίας συγκεκριμένης ημέρας ή ενός συγκεκριμένου μήνα ή ενός συγκεκριμένου έτους. Ο αλγόριθμος στη γραμμή 1-2 θέτει στις μεταβλητές που ορίζουν τη χρονική περίοδο, τις απαραίτητες τιμές ανάλογα με το εάν θέλουμε να επιλέξουμε τις τροχιές για μια συγκεκριμένη μέρα, μήνα ή έτος. Στη γραμμή 3 γίνεται έλεγχος για πιθανή στοχευμένη επίθεση. Για να το πετύχει αυτό ο αλγόριθμος καλεί τη συνάρτηση In_Hist (η λειτουργία της περιγράφηκε στην Ενότητα 5.3).Στη γραμμή 4, δημιουργείται ένας βοηθητικός πίνακας tmp στον οποίο αποθηκεύονται όλες οι τροχιές σε μορφή Moving point, που ικανοποιούν την χρονική περίοδο που έδωσε ο χρήστης. Στη γραμμή 5, κατασκευάζεται ένας ακόμα βοηθητικός πίνακας tmp1, προκειμένου να μετατραπούν οι τροχιές που βρέθηκαν στη γραμμή 4 σε μορφή sdo_trajectory. Η μετατροπή αυτή είναι απαραίτητη για την επεξεργασία των δεδομένων μας. Στη γραμμή 5, ενημερώνονται τα μεταδεδομένα και κατασκευάζεται ευρετήριο για τον πίνακα που δημιουργήθηκε στη γραμμή 4. Με τη δημιουργία ευρετηρίου έχουμε τη δυνατότητα να καλέσουμε χωρικές συναρτήσεις της Oracle για τα χωρικά δεδομένα μας. Στη γραμμή 6, χρησιμοποιούμε τη συνάρτηση της Oracle SDO_GEOM_RELATE, με μάσκα COVERS και INSIDE, προκειμένου να βρούμε τις τροχιές που περιέχονται μέσα στην προκαθορισμένη από το χρήστη περιοχή. Στη γραμμή 7, εισάγουμε σε ένα πίνακα T τις τροχιές των χρηστών που βρήκαμε σε μορφή moving_point. Για να το πετύχουμε αυτό βρίσκουμε τις τροχιές του πίνακα tmp που αντιστοιχούν στις τροχιές του tmp1. Στη γραμμή 8, διασφαλίζουμε την K ανωνυμία που ζήτησε ο χρήστης. Για να το πετύχουμε αυτό

ελέγχουμε εάν ο αριθμός των χρηστών στους οποίους αντιστοιχούν οι τροχιές που είναι αποθηκευμένες στον πίνακα T είναι ίσος ή μεγαλύτερος από τον αριθμό K που προσδιορίζει τον βαθμό ανωνυμίας. Εάν ισχύει αυτό, παρουσιάζονται στο χρήστη τα δεδομένα του πίνακα T, διαφορετικά καλείται η συνάρτηση Fake_Gen (η λειτουργία της περιγράφηκε στην Ενότητα 5.4) για να παράγει τις πλασματικές τροχιές που είναι απαραίτητες για να καλυφθεί ο βαθμός ανωνυμίας. Οι πλασματικές αυτές τροχιές εισάγονται στον πίνακα T και παρουσιάζονται μαζί με τις αληθινές στο χρήστη. Τέλος στη γραμμή 9, ο πίνακας T αδειάζει από δεδομένα.

Αλγόριθμος range-time query

Είσοδος :

- Το ψευδώνυμο **Id** του χρήστη

- Η περιοχή **Sgeo** που μας ενδιαφέρει

- Η χρονιά **YearStart** έναρξης της περιόδου που μας ενδιαφέρει.

- Ο μήνας **MonthStart** έναρξης της περιόδου που μας ενδιαφέρει.

- Η ημέρα **DayStar**t έναρξης της περιόδου που μας ενδιαφέρει.

- Η σταθερά **K** που καθορίζει το βαθμό της ανωνυμίας.

Έξοδος :

- Οι τροχιές των χρηστών που βρίσκονται μέσα στην ορισμένη περιοχή.

Σώμα Αλγορίθμου :

1)Αρχικοποίηση τιμών: HourSTART=0, MinSTART=0, SecSTART=0, HourEND=23, MinEND=59, SecEND=59, YearEND=YearSTART, MonEND=0, DayEND=0.

2)Εάν η τιμή της μεταβλητής DaySTART είναι ίση με 0,

-και η τιμή της μεταβλητής MonSTART είναι ίση με 0, θέσε τις εξής τιμές στις παρακάτω μεταβλητές: MonSTART=1, DaySTART=1, MonEND=12, DayEND=31.

-και η τιμή της μεταβλητής MonSTART δεν είναι ίση με 0, θέσε τις εξής τιμές στις παρακάτω μεταβλητές: MonEND=MonSTART, DaySTART=1, DayEND=31.

Διαφορετικά, θέσε τις εξής τιμές στις παρακάτω μεταβλητές: MonEND=MonSTART, DayEND=DaySTART.

3) Κάλεσε την συνάρτηση In_Hist με μεταβλητές το ψευδώνυμο του χρήστη, την περιοχή και το χρονικό διάστημα ενδιαφέροντος, για να ελέγξει εάν στον πίνακα hist υπάρχει ο ίδιος χρήστης με επερώτηση σε περιοχή που επικαλύπτεται ή γειτνιάζει με την καινούργια περιοχή ενδιαφέροντος.

- Εάν υπάρχει, επέστρεψε το μήνυμα "Privacy threat" και τερμάτισε την λειτουργία.

- Εάν όχι, πήγαινε στο βήμα 2

4) Δημιούργησε έναν βοηθητικό πίνακα tmp στον οποίο αποθήκευσε τις τροχιές του πίνακα data που ικανοποιούν τη χρονική περίοδο που μας ενδιαφέρει.

5) Κατασκεύασε έναν δεύτερο βοηθητικό πίνακα tmp1, στον οποίο μετέτρεψε τα moving_points που περιγράφουν την τροχιά των χρηστών του πίνακα tmp, σε sdo_geometry δεδομένα μέσω της συνάρτησης f_trajectory().

6) Ενημέρωσε τα μεταδεδομένα και κατασκεύασε χωρικό ευρετήριο

7) Βρες τους χρήστες στον πίνακα tmp1, για τις τροχιές των οποίων η συνάρτηση SDO_GEOM_RELATE με μάσκα COVERS ή η συνάρτηση SDO_GEOM_RELATE με μάσκα INSIDE ισχύει.

8) Βρες στον πίνακα tmp,τις τροχιές των χρηστών που βρήκες παραπάνω εισήγαγε τες στον πίνακα T.

Έλεγξε εάν ο αριθμός R των χρηστών, των οποίων οι τροχιές βρίσκονται αποθηκευμένες στον πίνακα T, είναι μεγαλύτερος ή ίσος του k.

- Εάν ναι, επέστρεψε τις τροχιές των χρηστών που βρίσκονται στον πίνακα T.

- Εάν όχι, εισήγαγε στον πίνακα T ,μέσω της Fake_Gen, k-R πλασματικές τροχιές και ανανέωσε τον πίνακα hist με μεταβλητές το ψευδώνυμο του χρήστη, την περιοχή και το χρονικό διάστημα ενδιαφέροντος. Τέλος, επέστρεψε τις τροχιές των χρηστών που βρίσκονται στον πίνακα T.

9) Διέγραψε τα δεδομένα του πίνακα T.

Κεφάλαιο 7

ΠΕΙΡΑΜΑΤΑ – ΑΠΟΤΕΛΕΣΜΑΤΑ

Στην παρούσα ενότητα παρουσιάζουμε μια ποιοτική ανάλυση των προτεινόμενων μεθοδολογιών οι οποίες και συνθέτουν τη λειτουργικότητα που προσφέρεται την σύστημα Query Engine. Η ανάλυσή μας κινείται σε μία βασική κατεύθυνση, στην παραποίηση της βάσης λόγω της εισαγωγής πλασματικών εγγραφών.

7.1 Εισαγωγή χωροχρονικών δεδομένων στη βάση με χρήση ειδικών προγραμμάτων

Για να καταφέρουμε να εξομοιώσουμε με πιστικό τρόπο την λειτουργία της Query Engine μας θα πρέπει να κατασκευάσουμε μία βάση δεδομένων από κινούμενα αντικείμενα, τα οποία θα αντιπροσωπεύουν τους διάφορους χρήστες που κινούνται σε ένα συγκεκριμένο οδικό δίκτυο. Αυτοί οι χρήστες αποτελούν τα χωρο-χρονικά αντικείμενα της βάσης μας. Όπως θα δούμε στη συνέχεια τέτοιου είδους αντικείμενα μπορούν να παραχθούν με τη βοήθεια ειδικών προγραμμάτων παραγωγής δεδομένων (Generators).

7.1.1Παραγωγή χώρο-χρονικών αντικείμενων με την χρήση Generators

Ένα από τα πιο γνωστά προγράμματα για την παραγωγή χώρο-χρονικών δεδομένων, έχει προταθεί από τον Thomas Brinkhoff. Αυτό το πρόγραμμα χρησιμοποιείται ευρέως από πολλούς επιστήμονες για ερευνητικές εργασίες. Αυτό είναι και το πρόγραμμα που θα χρησιμοποιηθεί και στα πλαίσια αυτής της εργασίας, προκειμένου να παραχθούν τα χώρο-χρονικά αντικείμενα της εφαρμογής που θα αναπτυχθεί. Σαν είσοδο αυτό το πρόγραμμα δέχεται ένα δίκτυο, το οποίο είναι σε κατάλληλη μορφή-κωδικοποίηση. Στην παρούσα εργασία δώσαμε ως είσοδο τα δύο αρχεία του δικτύου της πόλης Oldenburg που κατασκευάσαμε. Ο τρόπος παραγωγής των χώρο-χρονικών αντικειμένων, που παράγονται από το πρόγραμμα αυτό, βασίζεται κυρίως σε μετρήσεις που έχουν γίνει πάνω σε πραγματικά συστήματα. Για παράδειγμα σε ένα σύστημα GPS έχουν γίνει μετρήσεις σχετικά με την θέση από όπου οι χρήστες κάνουν τις αιτήσεις τους, καθώς και για τα μεσοδιαστήματα μεταξύ δυο διαδοχικών αιτήσεων. Με βάση αυτά τα στοιχεία, το πρόγραμμα προσπαθεί να παράγει κινούμενα αντικείμενα των οποίων τα χαρακτηριστικά προσεγγίζουν τον πραγματικό κόσμο. Πέρα από αυτά τα πειραματικά δεδομένα, τα οποία έχουν προκύψει από μετρήσεις, χρησιμοποιούνται και κάποιες βασικές κατανομές από την θεωρία πιθανοτήτων, όπως η Gaussian.

Όσον αφορά τις μετρήσεις που έχουν γίνει, θα χρησιμοποιήσουμε ένα παράδειγμα για να δείξουμε πως αυτές βοηθούν έτσι ώστε τα δεδομένα που παράγονται να είναι ρεαλιστικά, όσον αφορά την προσομοίωση της κίνησης των αντικειμένων. Έστω ότι έχουμε ορίσει ένα δίκτυο το οποίο αναπαριστά μια πόλη και τα κινούμενα αντικείμενα αντιπροσωπεύουν οχήματα. Το κάθε όχημα δεν επιλέγει τυχαία την

πορεία του, αλλά η πορεία του καθορίζεται με βάση τον προορισμό του. Έτσι με βάση κάποια κριτήρια όπως:

- Ποιος είναι ο πιο σύντομος δρόμος για να φτάσει στον προορισμό του ένα όχημα
- Αν από την διαδρομή που θα ακολουθήσει θα διασχίσει έναν δρόμο ταχείας κυκλοφορίας και έτσι θα αναπτύξει μεγαλύτερη ταχύτητα,(το πόσο ταχύτητα μπορεί να αναπτύξει ένα όχημα σε ένα συγκεκριμένο δρόμο-ακμή του δικτύου μπορεί να καθορίζεται από ένα κατώφλι που έχει οριστεί για κάθε ακμή)
- Αν σε αυτό το μονοπάτι από την αφετηρία προς τον προορισμό υπάρχει μεγάλη κίνηση και άρα δεν είναι δυνατόν το όχημα να αναπτύξει μεγάλη ταχύτητα(η κίνηση μπορεί να προσδιοριστεί από τη χωρητικότητα της κάθε ακμής)
- Εξωγενείς παράγοντες, όπως για παράδειγμα οι συνθήκες καιρού που επικρατούν, οι οποίες επηρεάζουν την κίνηση των οχημάτων.

Αυτοί και άλλοι πολλοί παράγοντες μπορούν να ληφθούν υπόψη, έτσι ώστε σε μια προσομοίωση ενός συστήματος, το πρόγραμμα (generator) να παράγει κινούμενα αντικείμενα των οποίων η κίνηση να προσεγγίζει όσον το δυνατόν περισσότερο την πραγματικότητα.

Ένας από τους πιο κύριους αλγόριθμους που χρησιμοποιεί το πρόγραμμα του Thomas Brinkhoff, είναι ο **GSTD algorithm** (**Generate SpatioTemporal Data**). Αυτός ο αλγόριθμος αρχίζει την παραγωγή των δεδομένων με βάση την Gaussian κατανομή. Στην συνέχεια, αυτά τα δεδομένα τροποποιούνται με την χρήση κάποιων τυχαίων συναρτήσεων, οι οποίες είναι παραμετροποιημένες κατάλληλα με βάση τις μετρήσεις

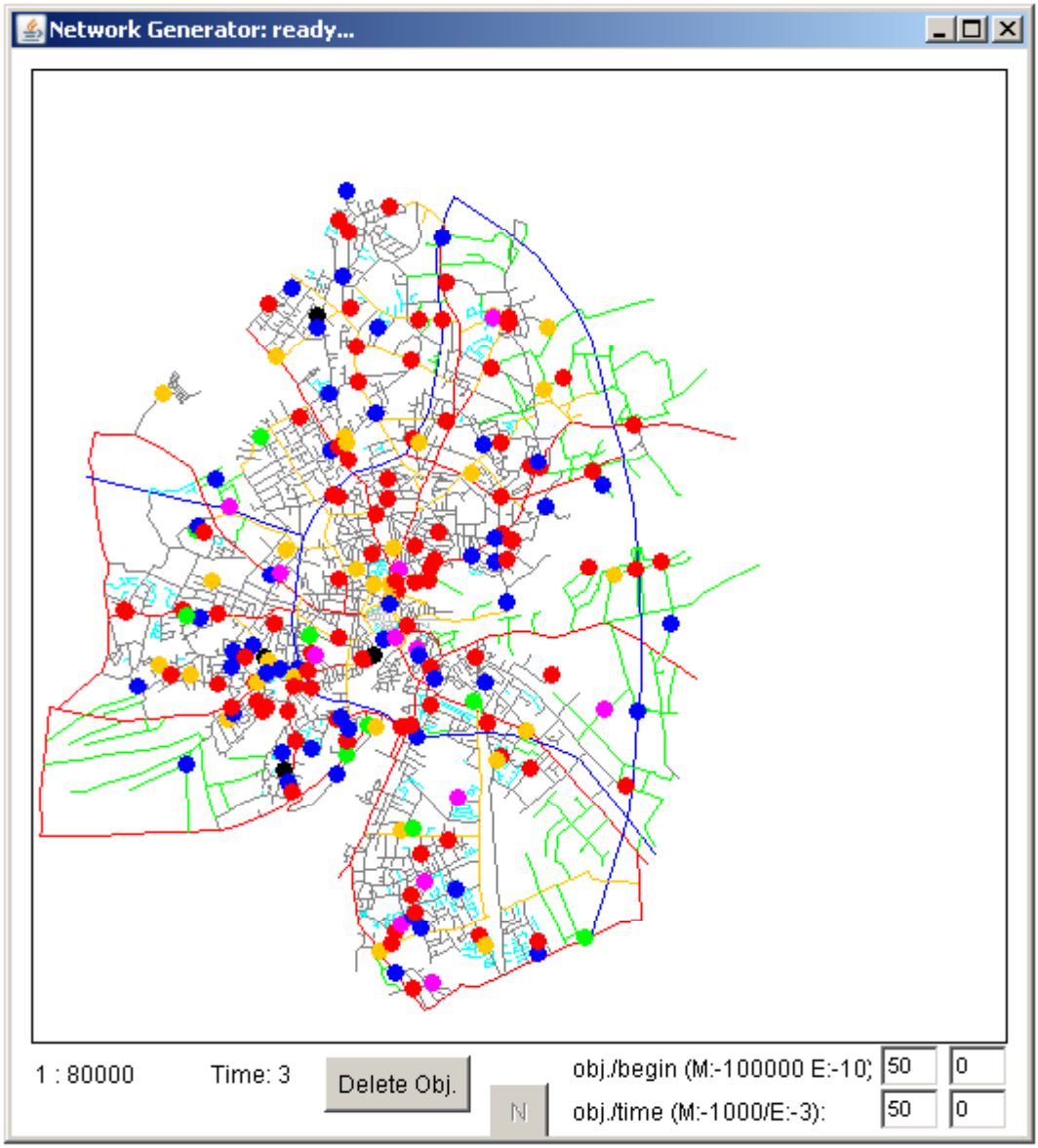
από τα πειράματα. Τέλος αν ένα αντικείμενο έχει κινηθεί έξω από τα προκαθορισμένα όρια θεωρείται άκυρο και απορρίπτεται. Το ίδιο γίνεται και αν ένα αντικείμενο έχει κινηθεί εκτός των χωρικών ορίων, δηλαδή αν έχει για κάποια χρονική στιγμή συντεταγμένες που δεν είναι επιτρεπτές με βάση τον χάρτη που έχουμε ορίσει.

Είναι αναγκαίο να τονίσουμε ότι η ενσωμάτωση στοιχείων από την στατιστική είναι αναγκαία, διότι αν βασιζόμαστε μόνο σε πειραματικά δεδομένα από ένα συγκεκριμένο σύστημα, τα δεδομένα τα οποία θα παράγονταν για την δική μας εφαρμογή ίσως να μην ήταν ικανοποιητικά. Έτσι το πρόγραμμα συνδυάζει τα πειραματικά δεδομένα- μετρήσεις με κάποια στοιχεία από την στατιστική και προσπαθεί να παράγει δεδομένα, τα οποία να είναι κατάλληλα για κάθε είδους εφαρμογή και παράλληλα να είναι ρεαλιστικά.

7.1.2Εισαγωγή των Χωρο-Χρονικών Αντικειμένων στη Βάση μας

Αφού είδαμε πως παράγονται τα χωρο-χρονικά αντικείμενα με τη χρήση Generators, και πιο συγκεκριμένα, τον Generator του Brinkhoff, σε αυτήν την ενότητα θα δούμε πως μπορούμε να εισάγουμε τις κινήσεις των αντικειμένων πάνω στη βάση δεδομένων μας. Όπως αναφέραμε προηγουμένως για την παραγωγή των αντικειμένων αυτών θα πρέπει να δώσουμε ως είσοδο στον Generator τα αρχεία του δικτύου. Έτσι δίνουμε ως είσοδο τα δύο αρχεία (.node και .edge) του δικτύου της πόλης του Oldenburg. Μπορούμε επίσης να καθορίσουμε τον αριθμό των χρηστών που επιθυμούμε να κινούνται πάνω στο δίκτυο, καθώς και την περίοδο κατά την οποία θα γίνονται αυτές οι κινήσεις. Στη συνέχεια ο Generator αρχίζει και παράγει

κινούμενα αντικείμενα πάνω στο δίκτυο με τους τρόπους που είδαμε προηγουμένως (Εικόνα 7.1).



Εικόνα 7.1 Κινούμενα αντικείμενα πάνω στο δίκτυο Oldenburg

Ο Generator τελικά θα εξάγει ένα αρχείο κειμένου (OldenburgGen.dat) στο οποίο είναι καταγεγραμμένες οι τροχιές που ακολούθησαν όλοι οι χρήστες του δικτύου. Έτσι το αρχείο αυτό περιέχει πληροφορίες όπως το αναγνωριστικό κάθε χρήστη (USER_ID), το σημείο (x, y) πάνω στο δίκτυο, στο οποίο βρίσκεται και την χρονική στιγμή (t) κατά την οποία βρίσκεται σε αυτό το σημείο.

Για τους σκοπούς της εργασίας κατασκευάσαμε ένα αρχείο (createPHL.java) για την αποθήκευση των δεδομένων που μας δίνει ogenerator. Πιο συγκεκριμένα:

- Αρχικά δημιουργήσαμε ένα πίνακα (phl) στην Oracle με τη χρήση SQL, για την αποθήκευση των location updates των χρηστών.
- Χρησιμοποιήσαμε το αρχείο OldenburgGen.dat ως αρχείο εισόδου για ανάγνωση και αποθήκευση στη Java.
- Μετατρέψαμε το αρχείο αυτό σε μορφή τέτοια ώστε να το αναγνωρίζει η Oracle, δηλαδή σε δεδομένα που μπορούν να εισαχθούν στους πίνακες του ΣΔΒΔ της Oracle. Ειδικά για την μετατροπή των συντεταγμένων σε γεωμετρίες της Oracle Spatial, χρησιμοποιήσαμε μια από τις δομές του Network Data Model Java Interface (JGeometry) η οποία μεταφέρει δεδομένα γεωμετρίας από τη Java στην Oracle και το αντίστροφο.
- Εισήγαμε με SQL τα μετασχηματισμένα δεδομένα στον πίνακα phl.

Για την κατασκευή των Moving Points δημιουργήσαμε το αρχείο createMovingPoints.java. Πιο συγκεκριμένα:

- Αρχικά δημιουργήσαμε ένα πίνακα (data) στην Oracle με χρήση SQL.
- Ομαδοποιήσαμε τα δεδομένα μας με βάση το User_id τους. Η ενέργεια αυτή ήταν απαραίτητη καθώς ο generator μας επιστρέφει τις θέσεις των χρηστών όχι σε συνεχόμενες χρονικές στιγμές αλλά ομαδοποιημένες με βάση τον χρόνο.
- Δεδομένου ότι σε ένα Moving Point ο χρόνος ορίζεται με τη μορφή "Έτος, Μήνας, Ημέρα, Ώρα, Λεπτά, Δευτερόλεπτα" ενώ ο generator μας παράγει διακριτές χρονικές στιγμές t, για τους τομείς Έτος ,Μήνας, Ημέρα, Ώρα

χρησιμοποιήσαμε τα χρονικά δεδομένα της στιγμής που έτρεξε ο αλγόριθμός μας. Για τομέα Λεπτά χρησιμοποιήσαμε τη χρονική στιγμή t που μας δίνει ο generator, ενώ τον τομέα Δευτερόλεπτα τον ορίσαμε μηδενικό.

- Εισάγαμε τις τροχιές των χρηστών στον πίνακα data μαζί με τα id τους, ενώ τα αριθμητικά δεδομένα (age,income) και το κατηγορηματικό δεδομένο (married) τα παράγουμε τυχαία μέσω κατάλληλης συνάρτησης. Θεωρήσαμε ότι η τιμή του age κυμαίνεται μεταξύ των 18-80 ετών, και η τιμή του income μεταξύ 5-100 χιλιάδες ευρώ.

Πλέον στη βάση μας είναι αποθηκευμένο το πλήρες ιστορικό των κινήσεων όλων των χρηστών που βρίσκονται στο δίκτυο μας, καθώς και οι υπόλοιπες πληροφορίες για αυτόν (age,income). Πάνω σε αυτά θα βασιστούμε στη συνέχεια για την υλοποίηση των αλγορίθμων μας.

7.2 ΠΟΙΟΤΙΚΗ ΑΝΑΛΥΣΗ

Για το σκοπό των πειραμάτων δημιουργήσαμε ένα αρχείο κειμένου (config.txt) το οποίο δέχεται τις διάφορες παραμέτρους, για την εκτέλεση των αλγορίθμων. Χρησιμοποιήσαμε επίσης 3 διαφορετικά σύνολα δεδομένων (datasets), τα οποία προέκυψαν από 3 διαφορετικές εκτελέσεις του Generator του Brinkhoff, για το δίκτυο Oldenburg. Για κάθε dataset εισήγαμε διαφορετικό αριθμό χρηστών που κινούνται πάνω στο δίκτυο καθώς και διαφορετικά χρονικά όρια κίνησης :

• Dataset 1 : 3000 χρήστες, T=[0,30]

• Dataset 2 : 5000 χρήστες, T=[0,20]

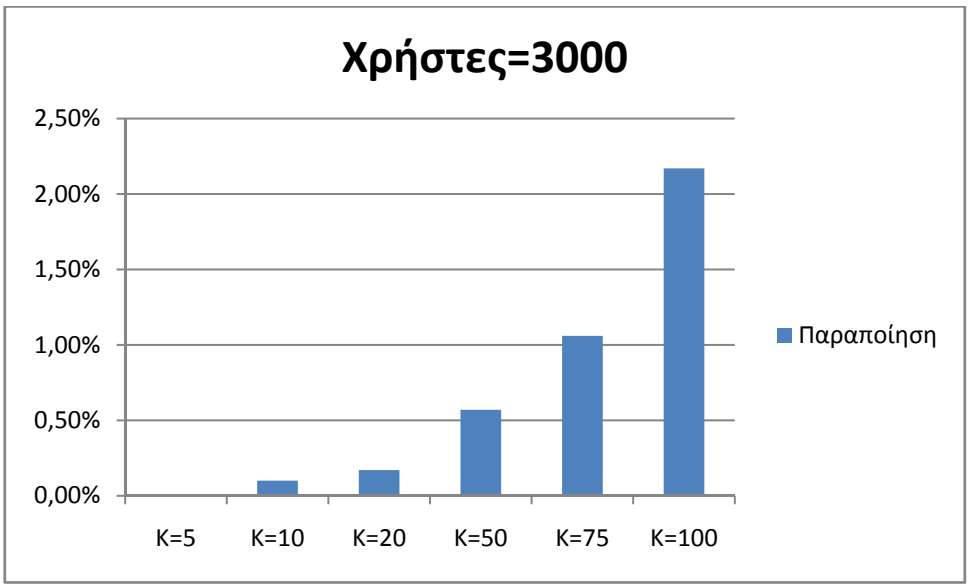
• Dataset 3 : 10000 χρήστες, T=[0,15]

Οι παράμετροι για τις οποίες τρέξαμε τους αλγορίθμους είναι οι εξής :

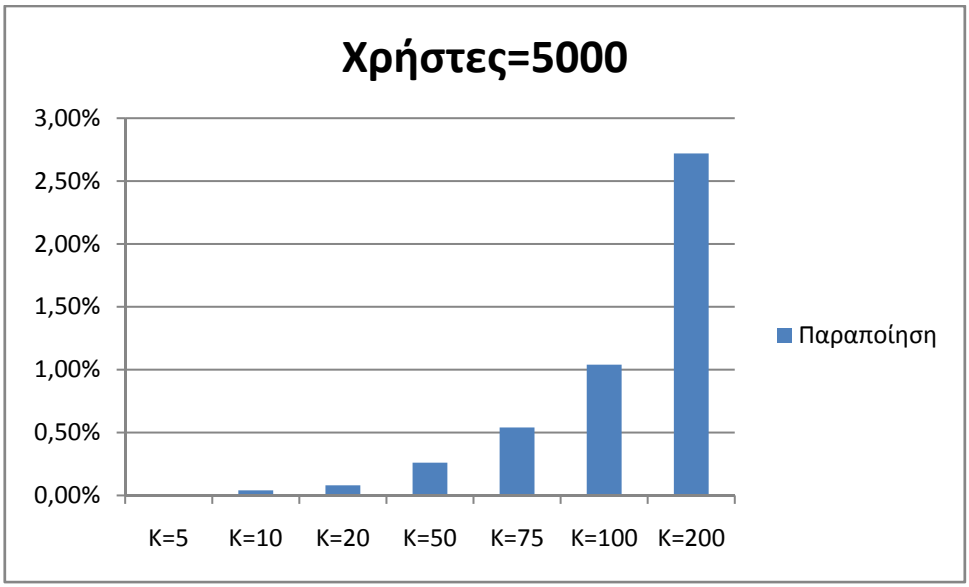
- Αριθμός Αιτήσεων : Ο συνολικός αριθμός των αιτήσεων για τις οποίες τρέχει κάθε πείραμα.
- Βαθμός K : Ο βαθμός της ανωνυμίας που επιθυμεί ο χρήστης να έχει.
- Το έτος YearStart: Το έτος έναρξης της χρονικής περιόδου (π.χ. 2009) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Ο μήνας MonthStart: Ο μήνας έναρξης της χρονικής περιόδου (π.χ. 8) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Η ημέρα DayStart: Η ημέρα έναρξης της χρονικής περιόδου (π.χ. 26) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Η ώρα HourStart: Η ώρα έναρξης της χρονικής περιόδου (π.χ. 17) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Το λεπτό MinStart:Το λεπτό έναρξης της χρονικής περιόδου (π.χ. 0) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Το δευτερόλεπτο SecStart: Το δευτερόλεπτο έναρξης της χρονικής περιόδου (π.χ. 0) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.

- Το έτος YearEnd: Το έτος λήξης της χρονικής περιόδου (π.χ. 2009) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Ο μήνας MonthEnd: Ο μήνας λήξης της χρονικής περιόδου (π.χ. 8) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Η ημέρα DayEnd: Η ημέρα λήξης της χρονικής περιόδου (π.χ. 26) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Η ώρα HourEnd: Η ώρα λήξης της χρονικής περιόδου (π.χ. 18) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Το λεπτό MinStart: Το λεπτό λήξης της χρονικής περιόδου (π.χ. 9) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Το δευτερόλεπτο SecEnd: Το δευτερόλεπτο λήξης της χρονικής περιόδου (π.χ. 45) κατά την διάρκεια της οποίας θέλουμε να λάβουμε τις απαντήσεις-τροχιές κάθε αλγορίθμου.
- Ο αριθμός των γειτόνων neighbors: Ο αριθμός συμπληρώνεται μόνο για την knn-query, διαφορετικά παίρνει μηδενική τιμή, και συμβολίζει των αριθμό των γειτονικών τροχιών που επιθυμούμε να μας επιστρέψει ο αλγόριθμος.
- Το είδος επερώτησης κάλυψης: Εάν επιθυμούμε την range query δίνουμε μηδενική τιμή, ενώ εάν επιθυμούμε την distance δίνουμε τιμή ίση με 1.

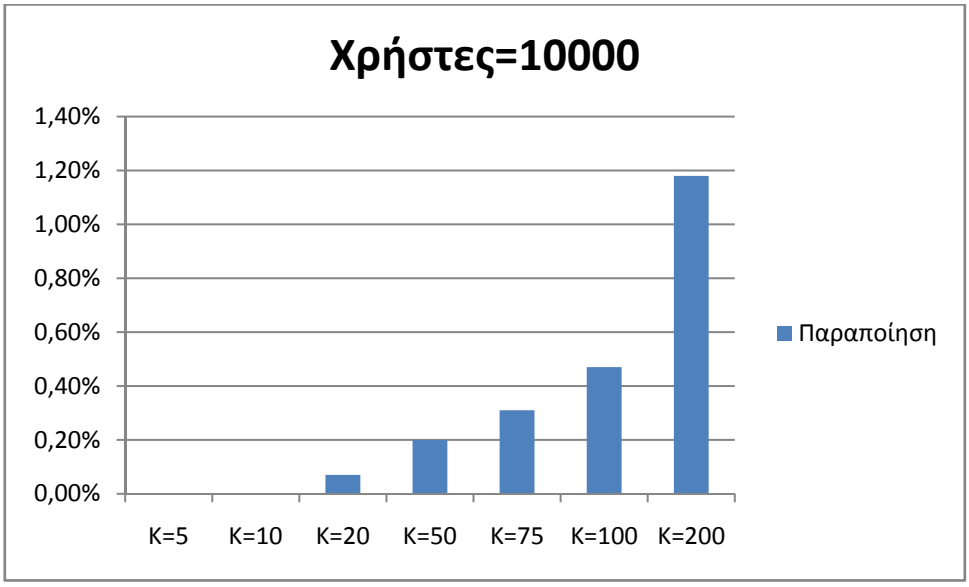
Οι Εικόνες 7.2, 7.3, 7.4 παρουσιάζουν την παραποίηση που υφίσταται η βάση δεδομένων λόγω της εισαγωγής πλασματικών εγγραφών. Η παραποίηση μετράται ως το ποσοστό των πλασματικών εγγραφών στη βάση σε κάποια θεωρούμενη χρονική στιγμή. Όπως εύκολα μπορεί κανείς να παρατηρήσει, μικρότερες τιμές του K οδηγούν σε μικρότερη παραποίηση της βάσης καθώς λιγότερες πλασματικές εγγραφές απαιτούνται για την παροχή της K–ανωνυμίας. Επιπλέον, όσους περισσότερους χρήστες έχουμε στη βάση μας τόσο μικρότερο είναι το ποσοστό παραποίησης.



Εικόνα 7.2 Ποσοστό παραποίησης βάσης με 3000 χρήστες

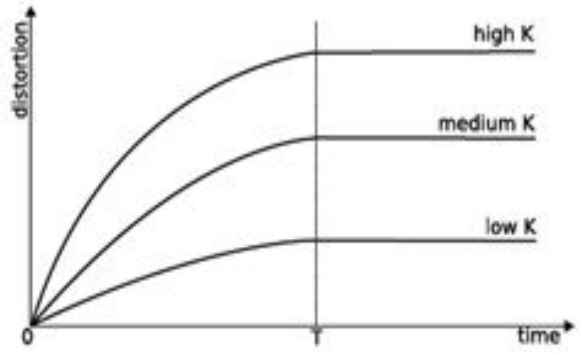


Εικόνα 7.3 Ποσοστό παραποίησης βάσης με 5000 χρήστες



Εικόνα 7.4 Ποσοστό παραποίησης βάσης με 10000 χρήστες

Παρατηρήθηκε επίσης πως για κάθε τιμή του K υπάρχει κάποια χρονική στιγμή T κατά την οποία οι πλασματικές εγγραφές που έχουν εισαχθεί στη βάση δεδομένων επαρκούν για την απάντηση ενός μεγάλου πλήθους επερωτήσεων, χωρίς να απαιτείται η δημιουργία επιπλέον πλασματικών εγγραφών. Από το σημείο εκείνο και έπειτα, η βάση δεδομένων παραποιείται με πολύ μικρότερο ρυθμό από ότι νωρίτερα.



Εικόνα 7.5 Παραποίηση της βάσης δεδομένων λόγω εισαγωγής πλασματικών εγγραφών

Οι τιμές που δώσαμε στις παραμέτρους για διαφορετικές εκτελέσεις, φαίνονται στον Πίνακα 7.1. Ο συνδυασμός των διαφόρων τιμών, μας έδωσε ένα μεγάλο σε εύρος πλήθος πειραμάτων, ικανά για να βγάλουμε χρήσιμα συμπεράσματα σχετικά με την αξιολόγηση της τεχνικής που υλοποιήσαμε.

Ανάλογα, με τις τιμές των παραμέτρων, μπορούμε να ορίσουμε και διαφορετικά επίπεδα ασφαλείας, κατά την υποβολή αιτήσεων. Έτσι κάποιος χρήστης, όταν κάνει μια αίτηση, μπορεί να επιλέξει τον βαθμό ασφαλείας που θέλει να έχει (Χαμηλό, Μεσαίο, Υψηλό). Για παράδειγμα, κάποιος που θέλει να έχει υψηλό βαθμό ασφαλείας, θα επιλέξει μεγάλο βαθμό K και μεγάλους χωρο-χρονικούς περιορισμούς.

Δίκτυο Oldenburg	DATASET 1	DATASET 2	DATASET 3
Αριθμός Χρηστών	3000	5000	10000
Χρονικά Όρια	[0,30]	[0,20]	[0,15]
Βαθμός Ανωνυμίας K	{5,10,20,50,75, 100 }	{5,10,20 ,50,75, 100,200}	{5,10,20,50,75 100,200}

Πίνακας 7.1 : Τιμές Παραμέτρων

Επίπεδο Ασφαλείας/Παράμετροι	K
Χαμηλό	5,10
Μεσαίο	20,50,75
Υψηλό	100,200

Πίνακας 7.2 : Επίπεδα Ασφαλείας

Κεφάλαιο 8

ΣΥΜΠΕΡΑΣΜΑΤΑ- ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Στο παρούσα διπλωματική παρουσιάσαμε αναλυτικά ένα σύγχρονο σύστημα προστασίας της ιδιωτικότητας κινούμενων χρηστών κατά την καταγραφή και την ανάλυση δεδομένων κίνησης. Το προτεινόμενο σύστημα επιτρέπει στα δεδομένα να μένουν αποθηκευμένα σε έναν κεντρικό, έμπιστο εξυπηρετητή, ενώ με τη βοήθεια του συστήματος HERMES, υποστηρίζει ένα μεγάλο πλήθος επερωτήσεων που αφορούν χωρικά/χρονικά και χωροχρονικά δεδομένα. Μέσω ενός συνόλου ειδικά σχεδιασμένων τεχνικών, οι απαντήσεις που παρέχονται στο χρήστη διασφαλίζουν την K– ανωνυμία των ατόμων των οποίων η κίνηση έχει καταγραφεί στη βάση δεδομένων, ενώ παράλληλα προσφέρουν κάλυψη από εξειδικευμένες επιθέσεις. Τέλος, μέσω των διάφορων πειραμάτων που έγιναν στους αλγορίθμους που παρουσιάσαμε, η Query Engine αποδεικνύεται ικανό για την προστασία της βάσης δεδομένων από επίδοξους εισβολείς, χωρίς να δημιουργεί παράλληλα μεγάλη παραποίηση της βάσης δεδομένων.

Ο μηχανισμός της Query Engine επαρκεί για τον καθορισμό επερωτήσεων που αφορούν συγκεκριμένα σημεία ενδιαφέροντος ή χωρικές περιοχές με κοινό επίπεδο ανάλυσης Επιπλέον, μπορεί να διαχειριστεί αποδοτικά επερωτήσεις που αφορούν χρονικά δεδομένα στη μορφή χρονόσημων ή χρονικών περιόδων εντός κάποιας θεωρούμενης ημερολογιακής μέρας. Οι ιδιότητες αυτές του μηχανισμού υποβολής

επερωτήσεων, εάν συνδυαστούν, επιτρέπουν τον ορισμό χωροχρονικών επερωτήσεων σε κοινή ανάλυση. Παρόλα αυτά, υπάρχουν πολλές περιπτώσεις κατά τις οποίες ο χρήστης που υποβάλλει κάποιο ερώτημα θέλει να λάβει γνώση σε διαφορετικά επίπεδα ανάλυσης, πιθανώς πιο αδρά από αυτά που παρουσιάστηκαν, ενδιαφερόμενος κυρίως για στατιστικά δεδομένα. Προς τον τομέα αυτό θα μπορούσε να κινηθεί κανείς για μια μελλοντική επέκταση της υπάρχουσας Query Engine.

Βιβλιογραφία

1. **Dieter Pfoser, Christian S. Jensen, Yannis Theodoridis.** Novel Approaches in Query Processing for Moving Object Trajectories. *VLDB.* 2000. σσ. 395-406.

2. **Nikos Pelekis, Yannis Theodoridis.** Boosting location-based services with a moving object database engine. *MobiDE.* 2006. σσ. 3-10.

3. **Nikos Pelekis, Elias Frentzos, Nikos Giatrakos, Yannis Theodoridis.** HERMES: aggregative LBS via a trajectory DB engine. *SIGMOD Conference.* 2008. σσ. 1255-1258.

4. Oracle Corp. Oracle® Database Documentation Library 10g Release 2 (10.2). *http://www.oracle.com/pls/db102/homepage.* [Ηλεκτρονικό] 2009.

5. **I.Kakoudakis.** TAU: Towards a Unified Temporal Information Management Framework. 2001.

6. **R. G. G. Cattell, Douglas K. Barry.** The Object Data Standard: ODMG 3.0. 2000 : Morgan Kaufmann.

7. **Ralf Hartmut Güting, Michael H. Böhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos, Markus Schneider, Michalis Vazirgiannis.** A Foundation for Representing and Querying Moving Objects. 2000. σσ. 1-42.

8. **Luca Forlizzi, Ralf Hartmut Güting, Enrico Nardelli, Markus Schneider.** Data Model and Data Structures for Moving Objects Databases. *SIGMOD Conference.* 2000. σσ. 319-330.

9. **Nikos Pelekis, Yannis Theodoridis.** An Oracle Data Cartridge for Moving Objects. 2007.

10. **Aris Gkoulalas-Divanis, Vassilios S. Verykios.** A privacy-aware trajectory tracking query engine. *SIGKDD Explorations.* 2008. 40-49.