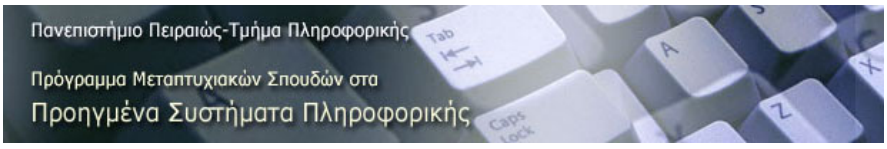
Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Οπτικοποίηση Αποτελεσμάτων Αλγορίθμων Εξόρυξης Γνώσης από Δεδομένα Κίνησης**
Ονοματεπώνυμο Φοιτητή	**Πλεμένος Ανάργυρος του Νικολάου**
Αριθμός Μητρώου	**ΜΠΣΠ/09053**
Κατεύθυνση	**Ευφυείς Τεχνολογίες Αλληλεπίδρασης Ανθρώπου-Υπολογιστή (ΕΤΕΑΥ)**
Επιβλέπων	**Γιάννης Θεοδωρίδης**, **Αναπληρωτής Καθηγητής**



Ημερομηνία Παράδοσης **Οκτώβριος** 2011

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)	(υπογραφή)	(υπογραφή)
Γιάννης Θεοδωρίδης Αναπληρωτής Καθηγητής	Νίκος Πελέκης Λέκτορας	Άγγελος Πικράκης Λέκτορας

ΠΕΡΙΛΗΨΗ

Με την ευρεία διαθεσιμότητα των συσκευών *GPS* και *RFID*, υπάρχει η δυνατότητα να καταγράφονται οι μετακινήσεις των ανθρώπων ή των αντικειμένων σε μεγάλη κλίμακα. Όλα αυτά τα δεδομένα περιέχουν σημαντικές πληροφορίες, οι οποίες μπορούν να αξιοποιηθούν από τους αναλυτές. Παρ' όλα αυτά, ένα εργαλείο που θα παρουσιάζει τα κινούμενα δεδομένα σε ένα χάρτη δεν αρκεί για να υποστηρίξει την ανάλυση της πληροφορίας αλλά χρειάζονται σύγχρονες αλληλεπιδραστικές διεπαφές που να είναι φιλικές προς τον χρήστη, να διευκολύνουν στην εξαγωγή γνώσης με την χρήση έξυπνων διαδραστικών εργαλείων και να δίνει την δυνατότητα σύγκρισης των αποτελεσμάτων. Πάνω σε αυτήν την κατεύθυνση, ο στόχος της μεταπτυχιακής διατριβής είναι να παρουσιάσει ένα οπτικό διαδραστικό εργαλείο (visual analytic tool), το οποίο θα δίνει την δυνατότητα στον χρήστη είτε είναι ένας απλός είτε κάποιος έμπειρος να ανακτά, να επεξεργάζεται και να αποθηκεύει την γνώση από τα κινούμενα δεδομένα με την χρήση διάφορων αλληλεπιδραστικών τεχνικών. Συγκεκριμένα, η εργασία χωρίζεται σε δύο σημαντικές ενότητες. Όσο αφορά την πρώτη ενότητα, αναπτύχθηκε ένας μηχανισμός προοδευτικής ανάλυσης των δεδομένων που ως στόχο έχει την επεξεργασία και την ανάδειξη γνώσης βήμα προς βήμα όπου σε κάθε επίπεδο προχωρούμε σε μεγαλύτερο βάθος και με μεγαλύτερη λεπτομέρεια. Η προοδευτική ανάλυση των δεδομένων δίνει την δυνατότητα στον τελικό χρήστη να χρησιμοποιεί και να συνδυάζει πολλούς αλγόριθμους εξόρυξης γνώσης μεταξύ τους αλλά και τους μηχανισμούς των απλών ερωτημάτων του *HERMES* [21][22][23][24]. Σχετικά με την δεύτερη ενότητα, υλοποιήθηκε πάνω στην πλατφόρμα τεχνικές εξόρυξης γνώσης που διαφυλάσσουν την ιδιωτική πληροφορία των ατόμων. Συγκεκριμένα, η πλατφόρμα περιέχει δύο σημαντικά σημεία, (α) μηχανισμούς απλών ερωτημάτων τόσο για την διαχείριση των δεδομένων κίνησης όσο και την διαφύλαξη των προσωπικών δεδομένων, και (β) αλγόριθμους ανωνυμοποίησης των δεδομένων που μπορούν να αξιολογηθούν με διάφορες τεχνικές εξόρυξης γνώσης. Η πλατφόρμα αυτή παρουσιάζει ένα πλήρες σύνολο των καινοτόμων αλγόριθμων ανωνυμοποίησης κινούμενων δεδομένων όπως είναι ο *NWA* [14][33] και ο *W4M* [15][32] και τεχνικές εξόρυξης γνώσης σε τροχιές, οι οποίες έχουν ενσωματωθεί σε μηχανισμούς απλών ερωτημάτων και σε ερωτήματα ασφάλειας δεδομένων. Τέλος, θα περιγραφούν τα τεχνικά βήματα που ένας προγραμματιστής μπορεί να εντάξει ένα νέο αλγόριθμο καθώς επίσης την διαδικασία κλήσεων των κλάσεων του παρόν συστήματος από την εκτέλεση ενός ερωτήματος μέχρι την οπτικοποίηση των αποτελεσμάτων στο τρισδιάστατο χάρτη.

ABSTRACT

With the widespread availability of GPS and RFID devices, it is possible to record the movement of people or objects on a large scale. All these data contain important information which can be exploited by the analysts. Nevertheless, a framework that represents moving data on a map is not adequate to support completely the analysis of information but it needs novel interactive interfaces that are friendly for users and facilitating the mining of knowledge through the use of visual analytic tools which allow the evaluation of the results. In this MSc thesis, the goal is to present a visual analytic tool that the user either is a simple or an expert, has the ability to retrieve, analyze and store the knowledge of moving data using various interactive techniques. Specifically, the work is divided into two major sections. As regards the first section, a mechanism of progressive querying and mining is the target for the extraction and analysis of knowledge step by step where at each level the user moves in greater depth and more detail. The progressive querying and mining enables the end user to use and combine several data mining algorithms together as well as the queries of the *HERMES* [21][22][23][24]. Concerning the second section, the presented platform uses mining techniques that preserve the privacy of data. In more detail, the platform includes two important engines, (a) a simple query mechanism for both managing mobility data and preserving personal data, and (b) data anonymization algorithms that can be evaluated through various data mining techniques. This platform presents a complete set of the state of the art data anonymization algorithms such as *NWA* [14][33] and *W4M* [15][32], and data mining techniques which have been integrated with a query engine and a privacy query engine. Finally, I describe the technical steps that a developer follows to integrate a new algorithm as well as I depict the process of system calls by running a query to visualizing the output in a 3D globe.

ΕΥΧΑΡΙΣΤΙΕΣ

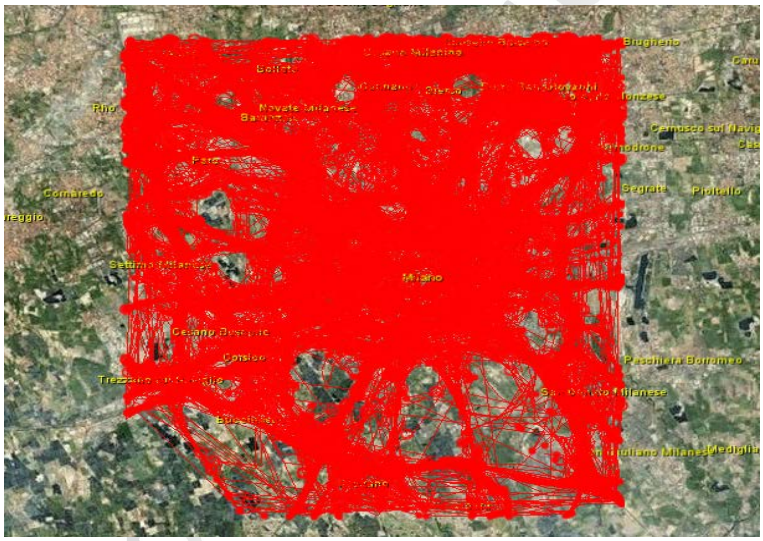
Πρώτα από όλα, θα ήθελα να ευχαριστήσω τον κ. Νικόλαο Πελέκη και τον κ. Γιάννη Θεοδωρίδη για την καθοδήγηση και την βοήθεια τους κατά την διάρκεια της μεταπτυχιακής μου διατριβής. Επίσης, θα ήθελα να ευχαριστήσω τον Μάριο Βόντα για την συμβολή του σε τεχνικά θέματα πάνω στην εργασία καθώς και τα παιδιά του *iSTLab* για τις πολύτιμες συμβουλές τους κατά την διάρκεια ανάπτυξης της εφαρμογής και συγγραφής της παρούσας διατριβής και επιπλέον, τους ευχαριστώ για το χώρο που μου διέθεσαν να εργαστώ. Τέλος, ευχαριστώ την οικογένειά μου για την υποστήριξη και την συμπαράσταση της κατά την διάρκεια των σπουδών μου.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

1. ΕΙΣΑΓΩΓΗ

Με την ευρεία διαθεσιμότητα των συσκευών GPS, έχει γίνει δυνατόν να καταγράφονται οι τροχιές των ανθρώπων ή των αντικειμένων σε μεγάλη κλίμακα (π.χ. μετακινήσεις αυτοκινήτων μέσα σε μια πόλη, δρομολόγια αεροπλάνων από πόλη σε πόλη ή από χώρα σε χώρα, κινήσεις εντόμων, κτλ). Όλα αυτά τα δεδομένα περιέχουν σημαντικές πληροφορίες για τους αναλυτές για να εξάγουν γνώση. Για παράδειγμα, ένας συγκοινωνιολόγος μπορεί να παρατηρήσει τις κινήσεις των ανθρώπων σε μια πόλη από το σπίτι στην δουλειά ή το αντίστροφο, ποιά διαδρομή ακολουθείται, ποιο είναι το πιο δημοφιλές μέρος, ποια δρομολόγια επιβαρύνονται κατά τις ώρες αιχμής, κλπ. Γενικά, η ανάλυση των δεδομένων περιέχει δυο μεγάλες κατηγορίες, την επεξεργασία και την πρόβλεψη. Δηλαδή, από τη μία πλευρά, οι αναλυτές μπορούν να εξετάσουν και να αναλύσουν τα στοιχεία αυτά και από την άλλη πλευρά, προσπαθούν να προβλέψουν ποία θα είναι η επόμενη θέση μιας ομάδας ανθρώπων. Παρ' όλα αυτά, η συλλογή αυτών των πληροφοριών δεν είναι εύκολο να επεξεργαστούν από τους εμπειρογνώμονες από την στιγμή που έχουν να αντιμετωπίσουν ένα τεράστιο όγκο πληροφοριών (βλέπε Εικόνα 1.1). Επιπλέον, υπάρχει έλλειψη από αλγόριθμους εξόρυξης γνώσης, οι οποίοι να αυτοματοποιούν την επεξεργασία και ανάλυση των δεδομένων καθώς επίσης και ελάχιστα εργαλεία οπτικοποίησης και ανάλυσης κινούμενων δεδομένων (visual analytics tools) έχουν προταθεί στην βιβλιογραφία.



Εικόνα 1.1: Οπτική παρουσίαση των κινούμενων δεδομένων σε όλη την βάση του Μιλάνο.

Μερικοί από τους αλγόριθμους εξόρυξης γνώσης που υπάρχουν στην βιβλιογραφία είναι οι αλγόριθμοι συσταδοποίησης όπως ο *T-Optics* [19] και ο *Traclus* [13], οι αλγόριθμοι που εξάγουν πρότυπα όπως είναι ο *T-Pattern* [6], αλγόριθμοι δειγματοληψίας π.χ. ο *T-Sampling* [25] και άλλοι όπως είναι ο *TR-FCM, CenTra, TX-CenTra* [26], κτλ. Όλοι αυτοί οι αλγόριθμοι για να παρουσιάσουν τα αποτελέσματά τους χρειάζεται να ενσωματωθούν σε εργαλεία οπτικοποίησης (*visualization tools*) και να εφαρμοστούν πάνω σε γεωγραφικούς χάρτες (π.χ. *Google maps* http://maps.google.com/maps). Για να ερμηνευτούν, τα αποτελέσματα των αλγορίθμων θα πρέπει να διατυπωθούν και να παρουσιαστούν σωστά στον αναλυτή. Επίσης, σε αυτά τα εργαλεία παίζει πολύ σημαντικό ρόλο οι τεχνικές διεπαφής χρήστη και υπολογιστή [38]. Με λίγα λόγια, ένα εργαλείο που θα παρουσιάζει τις τροχιές σε ένα χάρτη δεν αρκεί για να υποστηρίξει την ανάλυση της πληροφορίας αλλά χρειάζονται σύγχρονες αλληλεπιδραστικές διεπαφές που να είναι φιλικές προς τον χρήστη, να διευκολύνουν στην εξαγωγή γνώσης με την χρήση έξυπνων μηχανισμών και να δίνει την δυνατότητα σύγκρισης των αποτελεσμάτων. Καθ' όσον δύναμαι να γνωρίζω, τέτοια εργαλεία υστερούν στην βιβλιογραφία και ακόμη και σήμερα αποτελούν πηγή έρευνας για τους ερευνητές. Αν και έχουν γίνει σημαντικές προσπάθειες πάνω σε τέτοια εργαλεία, πολύ ελάχιστη συνεισφορά έχει γίνει πάνω σε κινούμενα δεδομένα, το οποίο συμπεριλαμβάνεται μέσα όχι μόνο ο χώρος αλλά και ο χρόνος. Το πρόβλημα με αυτά τα δεδομένα είναι ότι είναι περίπλοκα με αποτέλεσμα η επεξεργασία και εξόρυξη γνώσης από αυτά να δυσκολεύεται περισσότερο. Σε αυτό το πρόβλημα, έρχεται να προστεθεί και η ασφάλεια των δεδομένων. Δηλαδή, η συλλογή και στην συνέχεια η εξαγωγή πληροφορίας, μπορεί πολλές φορές να

προβεί στην γνωστοποίηση προσωπικών δεδομένων, τα οποία να αυξάνουν τον κίνδυνο παραβίασης ιδιωτικής ζωής των ατόμων. Άρα, εκτός του ότι χρειάζονται σύγχρονες οπτικές τεχνικές για την διευκόλυνση της ανάλυσης δεδομένων, θα πρέπει να υπάρχουν και μηχανισμοί που να αποτρέπουν την ανάδυση ευαίσθητης πληροφορίας.

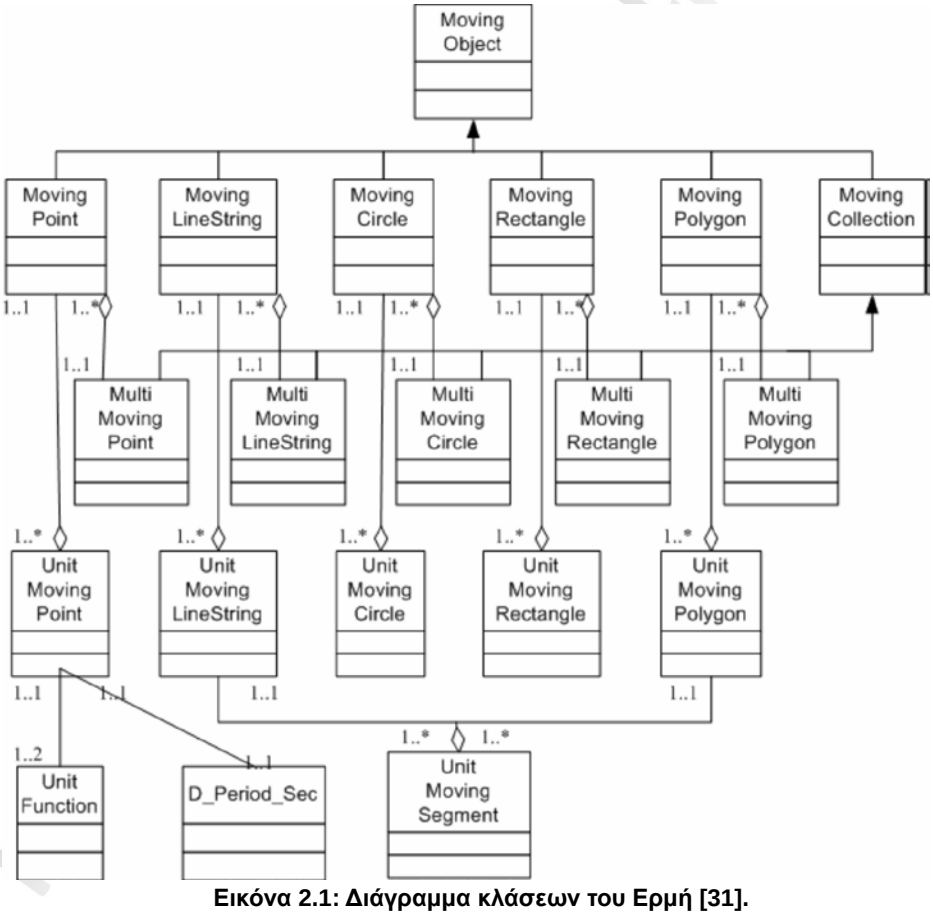
Το επίκεντρο της μεταπτυχιακής διατριβής χωρίζεται σε δύο ενότητες. Όσο αφορά την πρώτη ενότητα, ανέπτυξα ένα μηχανισμό προοδευτικής ανάλυσης των δεδομένων που ως στόχο έχει την επεξεργασία και την ανάδειξη γνώσης βήμα προς βήμα όπου σε κάθε επίπεδο προχωρούμε σε μεγαλύτερο βάθος και με μεγαλύτερη λεπτομέρεια. Η προοδευτική ανάλυση των δεδομένων δίνει την δυνατότητα στον τελικό χρήστη να χρησιμοποιεί και να συνδυάζει πολλούς αλγόριθμους εξόρυξης γνώσης μεταξύ τους αλλά και τους μηχανισμούς των απλών ερωτημάτων του HERMES. Με αυτόν τον τρόπο, ο τελικός χρήστης (π.χ. αναλυτής) θα μπορεί να χρησιμοποιεί τα αποτελέσματα ενός απλού ερωτήματος ως εισαγωγή για ένα ερώτημα αλγορίθμου εξόρυξης γνώσης ή για ένα άλλο απλό ερώτημα ή αντίστροφα και αυτή η διαδικασία να επαναλαμβάνεται έως ότου ο χρήστης να εξάγει το αποτέλεσμα που επιθυμεί. Σχετικά με την δεύτερη ενότητα, υλοποίησα μια ολοκληρωμένη πλατφόρμα για την εφαρμογή τεχνικών εξόρυξης γνώσης διαφυλάσσοντας την ευαίσθητη πληροφορία. Συγκεκριμένα, η πλατφόρμα περιέχει δύο σημεία αναφοράς, (α) μηχανισμούς απλών ερωτημάτων τόσο για την διαχείριση των δεδομένων κίνησης όσο και την διαχείριση των προσωπικών δεδομένων, και (β) αλγόριθμους ανωνυμοποίησης των δεδομένων που μπορούν να αξιολογηθούν με διάφορες τεχνικές εξόρυξης γνώσης. Η πλατφόρμα αυτή υιοθετεί τις παραπάνω ιδέες, προτείνει λύσεις, και παρουσιάζει τα αποτελέσματά της και απ' όσο ξέρω, αυτό είναι το πρώτο έργο που παρουσιάζει ένα πλήρες σύνολο των καινοτόμων αλγόριθμων ανωνυμοποίησης κινούμενων δεδομένων όπως είναι ο NWA και ο W4M και τεχνικές εξόρυξης γνώσης σε τροχιές, οι οποίες έχουν ενσωματωθεί σε μηχανισμούς απλών ερωτημάτων και σε ερωτήματα ασφάλειας δεδομένων.

Το υπόλοιπο της μεταπτυχιακής διατριβής έχει δομηθεί ως εξής. Στο δεύτερο κεφάλαιο, παρουσιάζω το θεωρητικό υπόβαθρο της εργασίας, κάνοντας μια συνοπτική περιγραφή στα συστήματα που βασίστηκε η εφαρμογή. Στο τρίτο κεφάλαιο, περιγράφω την αρχιτεκτονική του συστήματος καθώς επίσης τα βασικά συστατικά μέρη της διεπαφής. Στο τέταρτο κεφάλαιο, επιδεικνύω τις τεχνικές για την προοδευτική ανάλυση των δεδομένων κίνησης, σε μια πλατφόρμα που εφαρμόζει συνδυασμούς σε (α) απλά ερωτήματα με απλά ερωτήματα, (β) απλά ερωτήματα με ερωτήματα εξόρυξης γνώσης, (γ) ερωτήματα εξόρυξης γνώσης με απλά ερωτήματα και (δ) ερωτήματα εξόρυξης γνώσης με ερωτήματα εξόρυξης γνώσης, δημιουργώντας έτσι μια κυκλική ροή. Στο πέμπτο κεφάλαιο, περιγράφω τα οπτικά αποτελέσματα πάνω στον 'Ιδιωτικό ΕΡΜΗ' (Private-HERMES), μια ανεξάρτητη πλατφόρμα που δίνει στους χρήστες τη δυνατότητα (α) να θέτουν απλά ερωτήματα ή ερωτήματα απαραβίαστων δεδομένων, (β) να εφαρμόζουν δημοφιλείς αλγόριθμους ανωνυμοποίησης δεδομένων κίνησης, ενώ να έχουν τη δυνατότητα να συγκρίνουν και να αξιολογούν τα αποτελέσματα μεταξύ των αρχικών και των ανώνυμων δεδομένων μέσα από τεχνικές εξόρυξης γνώσης, και (γ) το σχεδιασμό και την εκτέλεση σημεία σύγκρισης για την αξιολόγηση των επιδόσεων των αλγορίθμων ανωνυμίας, χρησιμοποιώντας διαφορετικούς τύπους ερωτημάτων. Τόσο στο τέταρτο όσο και στο πέμπτο κεφάλαιο, παρουσιάζω την μεθοδολογία, την αρχιτεκτονική του συστήματος, την μελέτη περιπτώσεων σε πραγματικά δεδομένα και την εξαγωγή χρήσιμων συμπερασμάτων όπως επίσης, επισημαίνω διάφορες τεχνικές και μεθόδους που χρησιμοποιούνται στην υλοποίηση για την επίτευξη των αποτελεσμάτων. Στο έκτο κεφάλαιο, αναφέρω την υλοποίηση της πλατφόρμας και περιγράφω τις τεχνολογίες που εφαρμόστηκαν για την ανάπτυξη τους προγράμματος. Τέλος, στο έβδομο κεφάλαιο, συνοψίζω τα συμπεράσματα της μεταπτυχιακής διατριβής και επισημαίνω μερικές μελλοντικές κατευθύνσεις.

2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

2.1. HERMES

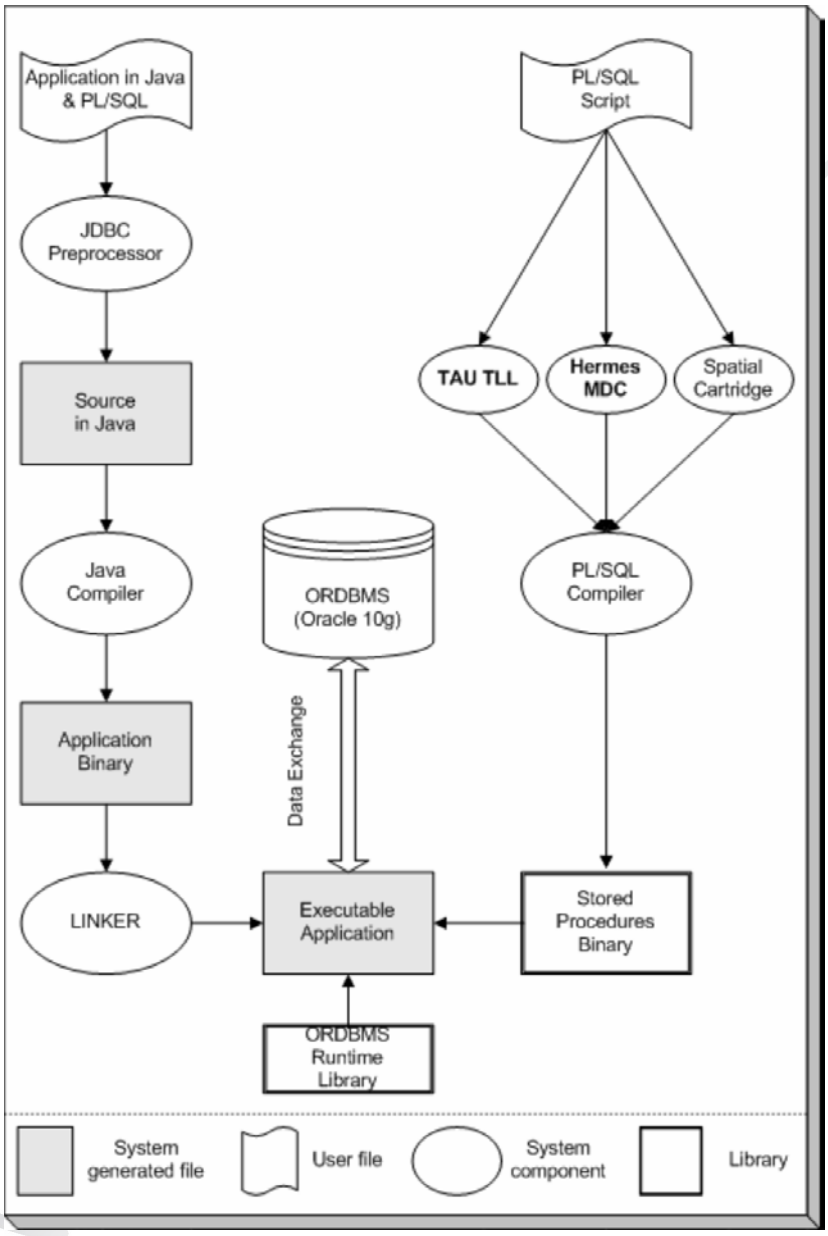
Ο Ερμής [21][22][23][24] (αγγελιοφόρος των Θεών στην ελληνική μυθολογία) είναι ένα ερευνητικό πρωτότυπο για την διαχείριση δεδομένων κίνησης και έχει αναπτυχθεί ως μια επέκταση στην Oracle10g για να παρέχει χώρο χρονικές λειτουργίες. Το σύστημα έχει σχεδιαστεί με τέτοιο τρόπο ώστε να υποστηρίζει είτε χρονικές είτε χωρικές λειτουργίες αλλά η κυριότερη λειτουργία του είναι η μοντελοποίηση, διαχείριση και ανάλυση βάσεων δεδομένων από αντικείμενα που κινούνται και μεταβάλλουν σχήμα ή θέση, στο χώρο και στο χρόνο, σε συνεχόμενα ή σε διακριτά βήματα. Η συλλογή των τύπων δεδομένων και οι λειτουργίες του έχουν προσδιοριστεί και αναπτυχθεί ως Oracle data cartridge όπου το *Hermes Moving Data Cartridge* (*Hermes-MDC*) είναι το κύριο συστατικό του. Το *Hermes-MDC* διαθέτει ένα σύνολο από γεωμετρικά σχήματα που περιέχουν μέσα το χρόνο και δίνει την δυνατότητα στον τελικό χρήστη μέσα από διάφορες μεθόδους και συναρτήσεις να αναλύει και να επεξεργάζεται εύκολα και γρήγορα βάσεις δεδομένων κίνησης. Ο Ερμής για να πραγματοποιηθεί, βασίστηκε στην χωρική βάση της Oracle10g [34] και στην χρονικές λειτουργίες του *TAU Temporal Literal Library Data Cartridge (TAU-TLL)* [29]. Συγκεκριμένα, τα σύνολα των γεωμετρικών σχημάτων που υποστηρίζει ο Ερμής φαίνονται στο παρακάτω διάγραμμα κλάσεων.



Εικόνα 2.1: Διάγραμμα κλάσεων του Ερμή [31].

Ο σκοπός του Hermes είναι να δίνει την δυνατότητα στον χρήστη να κατασκευάζει χώρο χρονικά σχήματα σε μια αντικειμενοστραφής βάση δεδομένων και να αναπτύσσει διεπαφές που να συναλλάσσονται με την βάση αυτή. Για να επιτευχθεί αυτό, ο σχεδιαστής θα πρέπει να γράψει αρχεία σε scripts τα οποία βασίζονται στην σύνταξη των *DDL (Data Definition Language)* και είναι επέκταση της γλώσσας *PL/SQL* της Oracle συμπεριλαμβανομένου χώρο χρονικές λειτουργίες. Στην συνέχεια, ένας προγραμματιστής θα μπορεί να χτίσει εφαρμογές σε Java, τα οποία θα ενσωματώνουν μέσα στα προγράμματα τα scripts ή θα τα καλεί από μια κλάση διασύνδεσης με την βάση δεδομένων με σκοπό να εκτελεί ερωτήματα και να σχεδιάζει αντικείμενα πάνω σε ένα χάρτη. Η διασύνδεση με την Java και της λειτουργίες του

Hermes γίνεται με το *JDBC pre-processor* και μαζί με το *ORDBMS Runtime Library* κάνουν την εφαρμογή εκτελέσιμη. Στην παρακάτω εικόνα, απεικονίζεται η αρχιτεκτονική του Ερμή όπως περιγράφτηκε παραπάνω.



Εικόνα 2.2: Η αρχιτεκτονική του Ερμή [31].

Με τις λειτουργίες του Hermes, ένας αναλυτής ή ακόμα ένα απλός χρήστης μπορεί να πετύχει ενδιαφέροντα συμπεράσματα σε μια πληθώρα συλλογή πληροφοριών από μια χώρο χρονική βάση δεδομένων, κάνοντας χρήση σε διάφορους τύπους ερωτήσεων. Μερικοί τύποι ερωτήσεων είναι οι εξής:

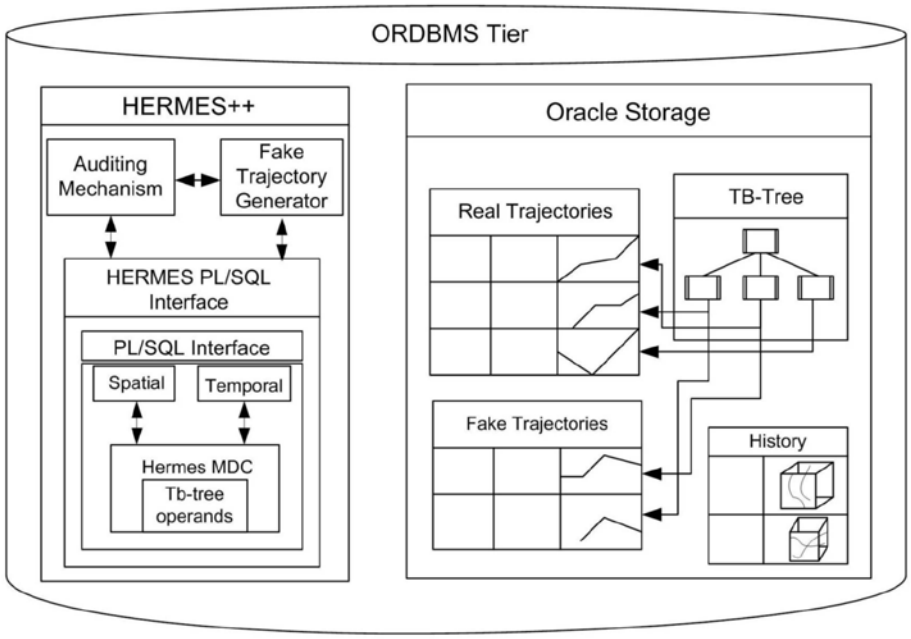
- *Range queries* – βρες όλες τις τροχιές των οχημάτων που διέσχισαν την πόλη του Μιλάνο ή βρες τις τροχιές των οχημάτων την Δευτέρα 31/03/08 ή βρες τις τροχιές των οχημάτων που διέσχισαν την πόλη του Μιλάνο την Δευτέρα 31/03/08 μεταξύ 6:00 και 10:00 το πρωί. Συγκεκριμένα, αυτά τα ερωτήματα χωρίζονται σε χωρικά, χρονικά ή και τα δυο.
- *Nearest Neighbor queries* [4] – βρες τα κοντινότερα βενζινάδικα από το σημείο που βρίσκομαι.
- *Topological queries* – βρες τις τροχιές των οχημάτων που μπήκαν την προηγούμενη ώρα μέσα στην πόλη του Μιλάνο.

- *Directional queries* – ποια φορτηγά κατευθύνθηκαν νότια από την πόλη της Αθήνας για τον μήνα Αύγουστο.
- *Similarity queries* [28] – βρες τις δυο πιο όμοιες τροχιές μεταξύ των οχημάτων σε μια πόλη.

Επίσης, μπορούν να ειπωθούν ερωτήματα που θα αφορούν τις ιδιότητες των τροχιών όπως πόση ήταν η μέση ταχύτητα στην πόλη του Μιλάνο για το βράδυ του Σαββάτου ή πόση ήταν η μέση απόσταση που διανύει ένας Μιλανέζος για να πάει από το σπίτι στην δουλειά του, κτλ. Όλα αυτά τα ερωτήματα για να υποστηριχθούν, θα πρέπει να παρουσιαστούν σε οπτικές εφαρμογές έτσι ώστε ο χρήστης να παρατηρεί το αποτέλεσμα αλλά κυρίως να επεμβαίνει στην ανάλυση του. Ο σκοπός της παρούσας μεταπτυχιακής διατριβής είναι να παρουσιάσει μια τέτοια εφαρμογή.

2.2. HERMES++

Ο *Hermes++* [21] σχεδιάστηκε ως μια επέκταση του *Hermes*, αξιοποιώντας την λειτουργικότητα αποθήκευσης δεδομένων κίνησης σε μια βάση δεδομένων και το μηχανισμό χώρο χρονικών ερωτημάτων του *Hermes* για να παρέχει ανώνυμα ερωτήματα στους χρήστες. Δηλαδή, ο *Hermes* αναπτύχθηκε ως μια επέκταση της *Oracle10g* για να παρέχει χώρο χρονικές λειτουργίες στους χρήστες (βλέπε προηγούμενο κεφάλαιο) και ο *Hermes++* χρησιμοποίησε αυτές τις λειτουργίες για να αποθηκεύει πραγματικές ή εικονικές τροχιές καθώς επίσης, οποιαδήποτε ιστορική πληροφορία των ερωτημάτων που θέτουν οι χρήστες ώστε να χρησιμοποιηθεί αργότερα για την αποτροπή διάφορων ειδών επιθέσεων. Ο *Hermes++* (α) διαθέτει μηχανισμούς ερωτημάτων στους τελικούς χρήστες, που μπλοκάρουν αποτελεσματικά επιθέσεις όπως επιθέσεις ταυτοποίησης χρήστη, επιθέσεις εντοπισμού περιοχών ενδιαφέροντος ή επιθέσεις παρακολούθησης τροχιών [1][21], (β) παράγει 'ρεαλιστικές' εικονικές τροχιές που ενσωματώνονται μαζί με τις πραγματικές στα αποτελέσματα των ερωτήσεων και (γ) μειώνει την διαστρέβλωση της βάσης δεδομένων εμπεριέχοντας μόνο των αριθμό των εικονικών τροχιών που είναι απαραίτητες για την διατήρηση της ασφάλειας των δεδομένων έτσι ώστε να δίνει περισσότερο ακριβής απαντήσεις.



Εικόνα 2.3: Η αρχιτεκτονική του *HERMES++* [21].

Για να επιτευχθούν, ο *Hermes++* έχει εφαρμόσει δυο αλγόριθμους, (α) τον μηχανισμό εικονικών τροχιών και (β) τον μηχανισμό ελέγχου ερωτημάτων. Όσο αφορά το πρώτο, ο μηχανισμός εικονικών τροχιών (*fake trajectory generation*) έχει την ικανότητα να δημιουργεί τροχιές που ακολουθούν την τάση του συνόλου των πραγματικών τροχιών με τέτοιο τρόπο ώστε να ελαχιστοποιεί λιγότερο είτε την παραμόρφωση της βάσης από αυτές τις τροχιές ή την παραβίαση της ιδιωτικής ζωής των ατόμων όταν τα ερωτήματα καταφθάνουν στους τελικούς χρήστες. Σχετικά με το δεύτερο, ο μηχανισμό ελέγχου ερωτημάτων (*query auditing*) χρησιμοποιεί διάφορες τεχνικές ελέγχου για να απορρίψει επιθέσεις κακόβουλων χρηστών ή να

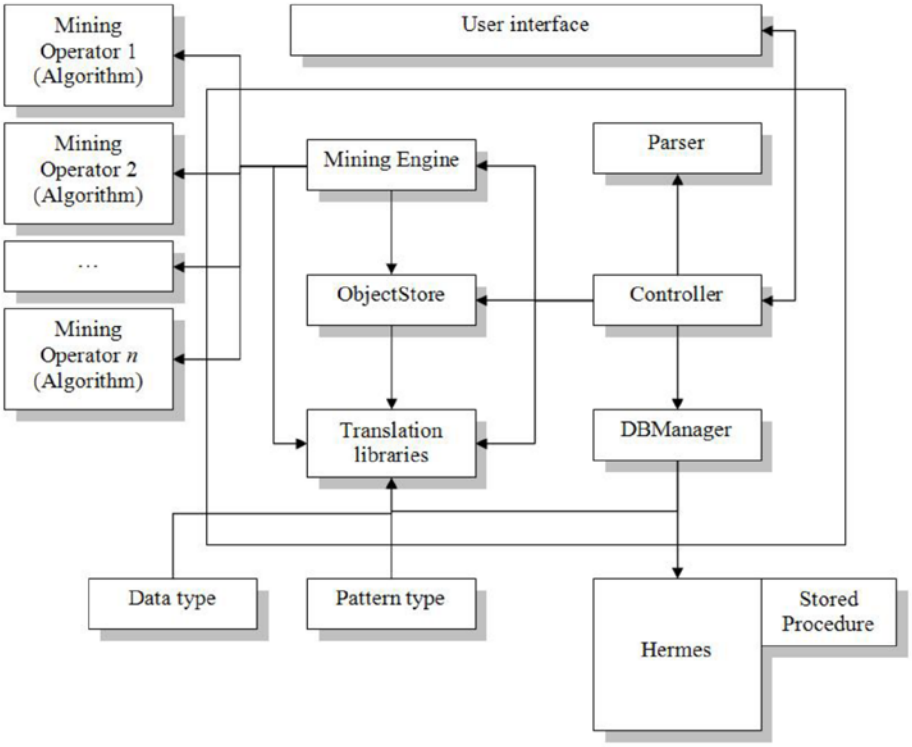
αποκρύψει ευαίσθητη πληροφορία. Όλο το πλαίσιο έχει υλοποιηθεί πάνω στο επίπεδο του *ORDBMS*, που σημαίνει ότι ο χρήστης μπορεί να θέτει ερωτήματα μέσω της γλώσσας *PL/SQL*. Από την αρχιτεκτονική πλευρά, ο *Hermes++* ενεργεί ως ένα πρόσθετο τμήμα μηχανής ερωτημάτων του *Hermes* όπως φαίνεται στην Εικόνα 2.3 και όχι ως ενδιάμεσο τμήμα. Όπως αναφέραμε και προηγουμένως, ο μηχανισμός αυτός μπορεί να προστατεύσει τα δεδομένα των χρηστών από τρείς επιθέσεις και είναι οι εξής:

- Επιθέσεις ταυτοποίησης χρήστη (*user identification attack*): ο επιτιθέμενος προσπαθεί να αναγνωρίσει την ταυτότητα του χρήστη, κάνοντας κάποια 'ειδικά' ερωτήματα. Τέτοια ερωτήματα συνήθως περιέχουν επαναλαμβανόμενες επικαλύψεις μικρών περιοχών σε μια περιοχή ενδιαφέροντος.
- Επιθέσεις εντοπισμού περιοχών ενδιαφέροντος (*sensitive location tracking attack*): ο κακόβουλος χρήστης προσπαθεί να εντοπίσει ένα αρχικό ή τελικό σημείο μιας τροχιάς κάνοντας ταύτιση με σημεία που είναι ήδη γνωστά (π.χ. νοσοκομείο) έτσι ώστε να αναγνωρίσει μερικά χαρακτηριστικά του χρήστη.
- Επιθέσεις παρακολούθησης τροχιών (*sequential tracking attack*): ο επιτιθέμενος προσπαθεί να αποκαλύψει ευαίσθητες περιοχές που ένας χρήστης έχει επισκεφτεί, παρακολουθώντας την πορεία του κινούμενου στόχου. Ο τρόπος που το πετυχαίνει αυτό είναι με διαδοχικά *range queries*, το ένα κοντά στο άλλο.

Οι παραπάνω τύποι των επιθέσεων θα αναλυθούν περισσότερο στο κεφάλαιο 3 και θα παρουσιάσω διάφορα παραδείγματα όπου ο μηχανισμός που περιγράφτηκε θα αναγνωρίζει και θα αποτρέπει τις επιθέσεις αυτές σε μια πραγματική βάση δεδομένων.

2.3. **ΤΟ ΣΥΣΤΗΜΑ** DAEDALUS

Το σύστημα *Daedalus* [36] είναι ένα περιβάλλον ανάλυσης κινούμενων δεδομένων και ουσιαστικά, από τις πρώτες αξιόλογες πλατφόρμες που αναπτύχθηκαν για να υποστηρίζουν μηχανισμούς ερωτημάτων και αλγορίθμων εξόρυξης γνώσης σε υψηλό επίπεδο μέσα στα πλαίσια του ευρωπαϊκού προγράμματος *GeoPKDD* [7][20][5] (*http://www.geopkdd.eu/*). Το *Daedalus* παρέχει το *MO-DMQL*, ένα μηχανισμό γλώσσας ερωτημάτων εξόρυξης γνώσης με σκοπό ο χρήστης να προσδιορίζει και να διατυπώνει ερωτήματα εξόρυξης. Η γλώσσα *MO-DMQL* βασίστηκε σε ένα αλγεβρικό πλαίσιο, καλούμενο *2W Model* [37] με την δυνατότητα να προσαρμόζει και να συνδυάζει μοντέλα εξόρυξης γνώσης σε ένα πολυδιάστατο επίπεδο. Ουσιαστικά, το σύστημα *Daedalus* είναι ένα στρώμα πάνω από τον *Hermes* και η αρχιτεκτονική του φαίνεται στην Εικόνα 2.4. Συγκεκριμένα, το συστατικό *διεπαφή-χρήστη* (*user interface*) είναι αυτό που βλέπει ο τελικός χρήστης, δίνοντας την δυνατότητα να διατυπώνει ερωτήματα σε γλώσσα *MO-DMQL* και να παρατηρεί το αποτέλεσμα πάνω σε ένα γεωγραφικό χάρτη. Ο *Mining Controller* είναι η κεντρική μονάδα για την εκτέλεση αλγορίθμων εξόρυξης γνώσης και είναι υπεύθυνο για την διεκπεραίωση των ερωτημάτων από τους αλγόριθμους αυτούς καθώς επίσης για τον συντονισμό στα καθήκοντα που εκτελούνται από τα άλλα συστατικά που αφορούν το κομμάτι της εξόρυξης γνώσης. Η δήλωση των παραμέτρων των αλγόριθμων επικυρώνεται από τον *Parser*, που στη συνέχεια τις μετατρέπει σε μια Αφηρημένη Γλώσσα Ερωτημάτων *Object Query Language* (OQL), το οποίο είναι χτισμένο για να απαντάει σε αυτές τις δηλώσεις. Ο *DBCManager* παρέχει κεντρική πρόσβαση στο στρώμα των δεδομένων. Βασικά, αλληλεπιδρά με τις αποθηκευμένες διεργασίες του *HERMES*, προκειμένου να εκτελεί χώρο-χρονικά ερωτήματα αλλά και πιο πολύπλοκα ερωτήματα. Με την εκτέλεση αυτών των ερωτημάτων πραγματοποιούνται πολύπλοκοι μετασχηματισμοί από σχεσιακά αντικείμενα του *HERMES* σε αντικείμενα του επιπέδου της εφαρμογής και το αντίστροφο. Αυτό μπορεί να γίνει με την εκμετάλλευση του αντικειμένου *Translation Libraries* που μετατρέπει μια αναπαράσταση σε μια άλλη (π.χ. από Oracle σε Java και Java σε Oracle). Το *Object Store* είναι ένα αντικείμενο που ενεργεί στο να επιταχύνει την επεξεργασία των αλγορίθμων εξόρυξης γνώσης. Ο *Mining Engine* ενεργοποιείται από τον *Controller* για την εκτέλεση των αλγορίθμων εξόρυξης γνώσης. Τα πρότυπα που αποκαλύπτονται από τον αλγόριθμο περνάνε από τον *Object Store* και στην συνέχεια μετατρέπονται σε μια αντικείμενο-σχεσιακή παρουσίαση για να αποθηκευτούν στην βάση. Αυτό μπορεί να γίνει από τον *Controller* αξιοποιώντας την βιβλιοθήκη *Translation Libraries* και αποθηκεύοντας τα αποτελέσματα στην βάση δεδομένων του HERMES μέσω του *DBCManager*. Όλα τα συστατικά του *Daedalus* είναι ανεξάρτητα από το καθένα αλλά είναι εδραιωμένα και ελέγχονται από τον *Controller* όπως φαίνεται και στο παρακάτω σχήμα.



Εικόνα 2.4: Η αρχιτεκτονική του DAEDALUS [36].

Γενικά, ο στόχος του συστήματος ήταν να προσφέρει τέσσερις τουλάχιστον λειτουργίες για την ανάλυση των τροχιών. Η πρώτη λειτουργικότητα αφορούσε την δημιουργία, αποθήκευση και επανάκτηση των τροχιών μέσω χώρο χρονικών διαδικασιών. Η δεύτερη λειτουργικότητα υιοθετούσε την ιδέα Weka-like όπου διέθετε μια βιβλιοθήκη από αλγόριθμους εξόρυξης γνώσης για δεδομένα κίνησης και θα αναπτύσσονταν κατά την διάρκεια του χρόνου. Η τρίτη κατηγορία υποστήριζε μια γλώσσα ερωτημάτων με σκοπό να δώσει την δυνατότητα στον αναλυτή να θέτει ερωτήματα χώρο χρονικά, τα αποτελέσματα να αποθηκεύονται σε μια βάση δεδομένων και να χρησιμοποιεί τα αποτελέσματα των ερωτημάτων αυτών για τον συνδυασμό νέων ερωτημάτων και ούτω καθεξής. Η τέταρτη λειτουργικότητα ήταν η ιδέα να συνδυάσει αναζήτηση με εξόρυξη με τέτοιο τρόπο ώστε να εμπλουτίσει τα δεδομένα με γνώση.

Το σύστημα που θα παρουσιάσουμε στην παρούσα μεταπτυχιακή διατριβή έχει βασιστεί στην αρχιτεκτονική του *Daedalus* και ουσιαστικά είναι μια παρεμφερές πλατφόρμα αλλά με εμπλουτισμένες τεχνικές αλληλεπίδρασης χρήστη-υπολογιστή και με περισσότερους αλγόριθμους εξόρυξης γνώσης. Ξέχωρα από αυτό, υποστηρίζει μηχανισμούς ασφάλειας δεδομένων και αλγόριθμους ανωνυμίας.

2.4. ΑΛΓΟΡΙΘΜΟΙ ΕΞΟΡΥΞΗΣ ΓΝΩΣΗΣ

Οι αλγόριθμοι εξόρυξη γνώσης είναι μια διαδικασία για την εξαγωγή τάσεων ή προτύπων σε μεγάλες βάσεις δεδομένων συνδυάζοντας συνήθως τεχνικές από τα γνωστικά πεδία της στατιστικής και της τεχνητής νοημοσύνης με το σύστημα διαχείρισης βάσης δεδομένων (*DBMS*). Οι αλγόριθμοι εξόρυξη γνώσης επικεντρώνονται σε δυο μεγάλες κατηγορίες, (α) την ανάλυση και (β) την πρόγνωση των δεδομένων. Οι αλγόριθμοι που θα παρουσιαστούν παρακάτω αλλά και η παρούσα μεταπτυχιακή διατριβή θα εστιαστεί στην πρώτη κατηγορία.

Ένας από τους πιο δημοφιλής αλγόριθμος εξόρυξης γνώσης σε τροχιές αντικειμένων είναι ο *Trajectory Pattern* [6], ο οποίος περιγράφει κινήσεις (τάσεις) από ένα σύνολο κινούμενων αντικειμένων. Συγκεκριμένα, τα πρότυπα, που εξάγονται από τον *T-Pattern*, παρουσιάζονται ως ακολουθίες χωρικών περιοχών επισυναπτόμενα από ένα χρονικό διάστημα. Τα αρχεία στα οποία αποθηκεύονται τα πρότυπα αυτά είναι σε w.r.t. και οι ελάχιστοι παράμετροι που πρέπει να περαστούν από τον *T-Pattern* είναι το κατώφλι ελάχιστης συχνότητας προτύπου (*minimum frequency of a pattern*) και το χρονικό κατώφλι (*time threshold* – χρονική ανεκτικότητα για την ένωση χρονικών διαστημάτων). Ένα από τα πιο σημαντικά σημεία του αλγόριθμου είναι να εξάγει περιοχές ενδιαφέροντος (*Regions of Interest-ROI*) από ένα σύνορο μέσα

σε ένα πλέγμα. Οι περιοχές ενδιαφέροντος αποθηκεύονται στο αρχείο '*dense regions w.r.t.*', τα οποία εξάγονται από ένα ορισμένο κατώφλι πυκνότητας (*density threshold*).

Ο *T-Pattern* εξάγει δύο αρχεία, το MiSTA.output και το regions.output. Το πρώτο αρχείο περιέχει συχνά πρότυπα ταξινομημένα με βάση το μήκος και το σχετικό χρονικό διάστημα. Για κάθε πρότυπο, υπάρχει μια γραμμή που περιγράφει το πρότυπο με το '*relative support*' και το '*absolute support*'. Στη συνέχεια, η γραμμή που ακολουθείται από το χρονικό διάστημα (*temporal annotation*) συνδέεται με αυτό το πρότυπο και περιλαμβάνει ένα χρονικό διάστημα ανά γραμμή, το οποίο περιγράφεται από τις χαμηλότερες και υψηλότερες συντεταγμένες ενός ορθογωνίου, και με την αντίστοιχη συχνότητα (ονομάζεται επίσης πυκνότητα). Για παράδειγμα, ένα τμήμα του αρχείου MiSTA.output, συσχετιζόμενο από το πρότυπο '(0) έως (9)' και τους μεταβατικούς χρόνους (*transition times*) είναι το εξής:

(0) (9) : 0.5 [abs:126]

[542.87, 544.6] Density: 76

[545.4, 547.72] Density: 76

Το δεύτερο αρχείο (regions.output) περιέχει τις περιοχές ενδιαφέροντος για το κάθε πρόθεμα (*prefix*) δηλαδή το ID της περιοχής ακολουθούμενο από τις χαμηλότερες και υψηλότερες συντεταγμένες ενός ορθογωνίου σε ένα δισδιάστατο χώρο. Το παρακάτω παράδειγμα δείχνει το περιεχόμενο του αρχείου, περιλαμβάνοντας 3 περιοχές.

18 59 38 59 38

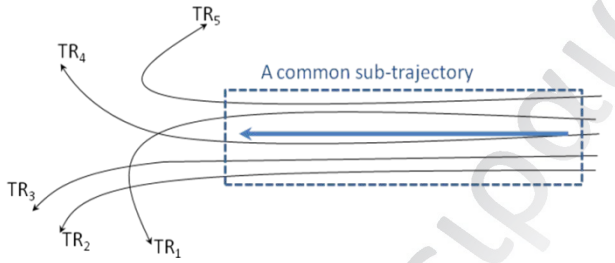
19 58 38 59 39

20 56 38 56 40

Ο αλγόριθμος *T-Optics* [19] είναι ένας από τους πιο σύγχρονους αλγόριθμους συσταδοποίησης (*clustering*) και είναι βασισμένος στην ιδέα του *DBSCAN* [18] αντιμετωπίζοντας όμως μια μεγάλη αδυναμία του, την δυνατότητα να ανιχνεύει συστάδες σε δεδομένα με διαφορετική πυκνότητα μεταξύ τους. Συγκεκριμένα, ο στόχος του είναι η ικανότητα να ανιχνεύει μη σφαιρικές συστάδες (με αφηρημένο σχήμα), κάτι που οι προγενέστεροι αλγόριθμοι όπως ο *K-Means* και η ιεραρχική μέθοδοι δεν κατείχαν. Επίσης, δίνει την δυνατότητα να αντέχει και να μην αλλοιώνει το αποτέλεσμα από τον θόρυβο καθώς επίσης, να ανακαλύπτει ένα αυθαίρετο αριθμό συστάδων ανάλογα με την πηγή δεδομένων. Επιπλέον, η πολυπλοκότητα του *T-Optics* είναι χαμηλή πράγμα που το κάνει γρήγορο στον υπολογισμό και στην εξόρυξη του αποτελέσματος. Συγκεκριμένα, ο *T-Optics* λειτουργεί ως εξής: ένα τυχαίο αντικείμενο επιλέγεται, στην συνέχεια, το επόμενο αντικείμενο που θα επιλεχτεί από την βάση θα είναι αυτό με το πιο χαμηλό *reachability distance* και η διαδικασία συνεχίζεται έως ότου ταξινομηθούν όλα τα αντικείμενα. Το *reachability distance* υπολογίζεται με βάση πόσο κοντά είναι το επιλεγμένο αντικείμενο από το αντικείμενο πυρήνα (*core object*). Το *core object* είναι το αντικείμενο που η περιοχή γύρω του είναι πυκνή και ανήκει σε μια συστάδα και όχι σε θόρυβο. Ο *T-Optics* ως είσοδο δέχεται μια σειρά από τροχιές D = {T1, T2, …, Tn} και προϋποθέτει δυο παραμέτρους για να εκτελεστεί, το *MinPts* και το ε. Το *MinPts* περιγράφει τον αριθμό των τροχιών που απαιτούνται για να δημιουργήσουν μια συστάδα, ενώ η παράμετρος ε χαρακτηρίζει την μέγιστη απόσταση (ακτίνα) που χρειάζεται να έχουν οι τροχιές μεταξύ τους για να συμπεριληφθούν σε μια συστάδα. Το αποτέλεσμα που εξάγει ο αλγόριθμος *T-Optics* είναι ένα δυσδιάστατο διάγραμμα με τις ταξινομημένες τροχιές στον άξονα X και το αντίστοιχο *reachability distance* στον άξονα Y. Αξίζει να σημειωθεί λόγω της πολυπλοκότητας της τροχιάς, χρησιμοποιούνται διάφορα κριτήρια συναρτήσεων απόστασης (*distance functions*) για να υπολογιστούν οι αποστάσεις μεταξύ των κινούμενων δεδομένων. Μερικά *distance functions* [38] είναι η κοινή προέλευση (*common source*) ή ο κοινός προορισμός (*common destination*), τα οποία υπολογίζουν την απόσταση δυο τροχιών στο χώρο μεταξύ των αρχικών ή τελικών σημείων τους, αντίστοιχα. Το *route similarity*, το οποίο σχεδιάστηκε για να ανέχεται μη ολοκληρωμένες τροχιές (π.χ. κάποια αρχικά σημεία ή τελικά της τροχιάς δεν υπολογίζονται) και αναζητάει το κοντινότερο ζευγάρι σημείων μεταξύ δυο τροχιών. Επιπλέον, το *route similarity + dynamics* λαμβάνει υπόψη και τους αντίστοιχους χρόνους στα σημεία των τροχιών. Αυτοί οι χρόνοι χωρίζονται στους σχετικούς (π.χ. αρχικά χρονικά σημεία της τροχιάς ή τελικά χρονικά σημεία, κτλ) και στους απόλυτους χρόνους. Επίσης, τα *distance functions* μπορούν να επεκταθούν στα χαρακτηριστικά του ατόμου κάθε τροχιάς (π.χ. ηλικία, επάγγελμα, κτλ), στα είδη κινήσεων (π.χ. περπάτημα, οδήγηση, ποδηλασία, κτλ) ή στις ιδιότητες του περιβάλλον (π.χ. το είδος των δρόμων, σημεία ενδιαφέροντος, κτλ).

Ένας άλλος αλγόριθμος συσταδοποίησης, ο *Traclus* [13] προτάθηκε από τον Jae-Gil Lee και τους συνεργάτες του. Η ιδέα του *Traclus* είναι ο τεμαχισμός κάθε τροχιάς σε ένα σύνολο από υπό-τροχιές (τμή-

ματα) και στην συνέχεια, η ομαδοποίηση των τμημάτων αυτών σε μια συστάδα. Αναλυτικότερα, ο αλγόριθμος *Traclus* περιέχει δύο στάδια: 1) το στάδιο του διαχωρισμού, ένας αλγόριθμος αναλαμβάνει την τμηματοποίηση της κάθε τροχιάς αν απαιτείται και 2) το στάδιο της ομαδοποίησης, ένας αλγόριθμος συσταδοποίησης που βασίζεται στην πυκνότητα όπως αυτοί που έχουν προταθεί στην βιβλιογραφία αναλαμβάνει την δημιουργία συστάδων. Το μεγαλύτερο πλεονέκτημα αυτού του αλγόριθμου είναι να αναδεικνύει υπό-τροχιές σε μια βάση κινούμενων δεδομένων. Το να ανακαλύπτουμε υποσύνολα τροχιών είναι πολύ χρήσιμο σε πολλές εφαρμογές ειδικά εάν υπάρχουν περιοχές ειδικού ενδιαφέροντος για ανάλυση. Παραδείγματος χάριν, όπως παρατηρούμε στην παρακάτω εικόνα, υπάρχει μια κοινή συμπεριφορά απεικονισμένη από ένα παχύ βέλος μέσα στο μπλε πλαίσιο αν και οι πέντε τροχιές ακολουθούν μια διαφορετική πορεία. Παρ όλα αυτά, εάν εφαρμόζαμε ένα αλγόριθμο συσταδοποίησης σε ολόκληρη την τροχιά, δεν θα ανακαλύπταμε αυτήν την κοινή συμπεριφορά, χάνοντας μια πολύτιμη πληροφορία.



Εικόνα 2.5: Ένα παράδειγμα από ένα κοινό υποσύνολο τροχιών.

Ένας άλλος μηχανισμός συσταδοποίησης είναι αυτό που εισήγαγαν ο Ν. Πελέκης και οι συνεργάτες του [26][27], μια μέθοδο συσταδοποίησης τριών σταδίων για να αντιμετωπίσουν την επίδραση της αβεβαιότητας των δεδομένων σε μια βάση τροχιών (*Trajectory Database - TD*). Πρώτον, οι συγγραφείς πρότειναν μια συμβολική αναπαράσταση (διανύσματα) των τροχιών ως μια ακολουθία περιοχών που περιλαμβάνει την αβεβαιότητα των δεδομένων (π.χ. από που πέρασε μια τροχιά, κτλ). Στην περίπτωση που μια βάση *TD*, περιλαμβάνει μια ακολουθία περιοχών που πιθανόν μια τροχιά να διέρχεται από αυτές τότε απαιτούνται μεγέθη που αναπαριστούν την παρουσία ή όχι των τροχιών σε αυτές τις περιοχές. Για να αξιοποιηθεί η πληροφορία αυτή, ένα αποτελεσματικό μετρικό μέγεθος απόστασης αναλαμβάνει την αντιμετώπιση της αβεβαιότητας αυτής με σκοπό να ενσωματωθεί στον αλγόριθμο *Fuzzy C-means* (*FCM*) όπου θα δημιουργούνται συστάδες υπό συνθήκες αβεβαιότητας. Ο *FCM* έχει την δυνατότητα να ομαδοποιεί τα κέντρα βάρους (*centroids*) των τροχιών χρησιμοποιώντας μεθόδους ομοιότητας. Όμως, για να επιτευχθούν καλύτερα αποτελέσματα δεν χρησιμοποιούνται μεθόδους ομοιότητας σε ολόκληρη την τροχιά αλλά μεταξύ των τμημάτων των τροχιών. Βασιζόμενη σε αυτήν την ιδέα, δεύτερον, ο *Centra*, ένας πρωτότυπος αλγόριθμος αναδεικνύει τα *centroids* των τροχιών από μια ομάδα κινήσεων, λαμβάνοντας υπόψη το πλεονέκτημα της τοπικής ομοιότητα μεταξύ των τμημάτων των τροχιών. Τρίτον, ένας νέος αλγόριθμος συσταδοποίησης, ο *CenTR-I-FCM*, ο οποίος αξιοποιεί τον αλγόριθμο *Centra* σε κάθε στάδιο υπολογισμού του *centroid*, χρησιμοποιεί τους μεθόδους ομοιότητας για να ομαδοποιήσει ολόκληρες τις τροχιές σε ένα υψηλό επίπεδο και βελτιώνει τα αποτελέσματα αυτά με την χρήση τοπικής ομοιότητα μεταξύ των τμημάτων των τροχιών. Τέλος, ο αλγόριθμος *TX-CenTra* βελτιώνει την παρουσίαση του *Centra* 'καλυτερεύοντας' τα διανύσματα αναπαράστασης με σκοπό να αναδείξει πρότυπα στον αναλυτή με ένα πιο διαισθητικό τρόπο.

Γενικά, οι αλγόριθμοι συσταδοποίησης που αναφέραμε προηγουμένως χωρίζονται σε τέσσερις κατηγορίες:

- στους αλγόριθμους που χρησιμοποιούν μεθόδους τμηματοποίησης (π.χ. k-means, k-medoids, bisecting k-medoids, κτλ)
- στους ιεραρχικούς αλγόριθμους (π.χ. BIRCH)
- στους αλγόριθμους βασισμένοι στην πυκνότητα των δεδομένων (π.χ. DBSCAN και OPTICS)
- και στους αλγόριθμους που χρησιμοποιούν μεθόδους πλέγματος (*grid-based methods)* όπως είναι οι αλγόριθμοι STING, CenTR-I-FCM, CenTra, κτλ.

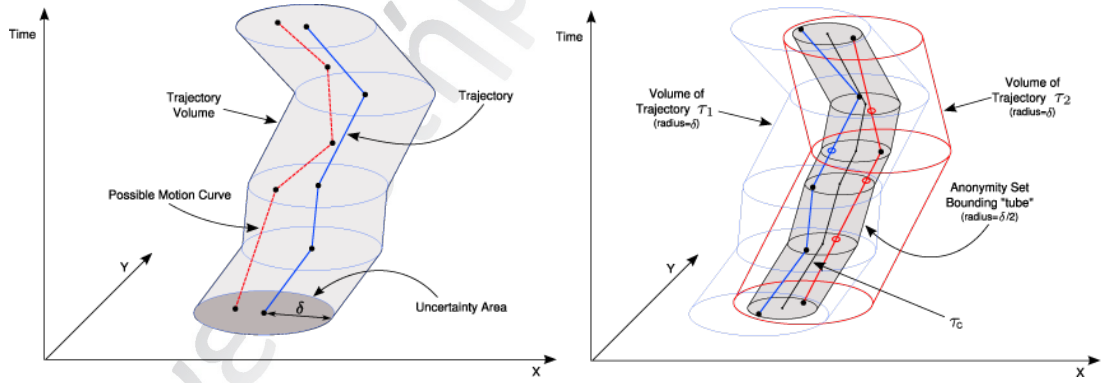
Έκτος από τους αλγόριθμους συσταδοποίησης και τον *T-Pattern* που περιγράψαμε παραπάνω, υπάρχουν στην βιβλιογραφία και μηχανισμοί όπως είναι η δειγματοληψία που ασχολούνται με το στάδιο της προ-επεξεργασίας των δεδομένων και συνήθως, το αποτέλεσμά τους μετά χρησιμοποιείται ως εισαγωγή στους παραπάνω αλγόριθμους. Ένας τέτοιος μηχανισμός είναι ο *T-Sampling* [25] και είναι ένας αλγόριθ-

μος για να παίρνει αντιπροσωπευτικά δείγματα από μια βάση *TD*. Συγκεκριμένα, ο *T-Sampling* λειτουργεί ως εξής: πρώτον, μια συμβολική αναπαράσταση υιοθετείται για να μοντελοποιήσει όλες τις τροχιές σε μια βάση *TD* ως διανύσματα περιλαμβάνοντας τον χρόνο και το χώρο. Δεύτερον, μια μέθοδος παρουσιάζει την κάθε τροχιά ως συνεχόμενη συνάρτηση, η οποία περιγράφει την αντιπροσωπευτικότητα του κάθε τμήματος της τροχιάς σε σχέση με ολόκληρη την βάση. Στην συνέχεια, ένας μηχανισμός αποκαλούμενος *SyTra* βελτιώνει το αντιπροσωπευτικό δείγμα και τρίτον, ο αλγόριθμος ανακτά ένα υποσύνολο των τροχιών με βάση αυτά τα αντιπροσωπευτικά δείγματα. Το πλεονέκτημα του *T-Sampling* είναι ότι λαμβάνει υπόψη όχι μόνο τις πιο πυκνές περιοχές αλλά τις λιγότερο πυκνές, οι οποίες είναι επίσης ενδιαφέροντες σε περιπτώσεις που θέλουμε να ανιχνεύσουμε ακραίες καταστάσεις (*outliers*) ή αραιές συστάδες.

Οι παραπάνω αλγόριθμοι εξόρυξης γνώσης είναι οι μηχανισμοί που χρησιμοποιούνται στην πλατφόρμα που υλοποιήθηκε στα πλαίσια της μεταπτυχιακής διατριβής. Οι αλγόριθμοι αυτοί συνδυάζονται με την βάση *Hermes* σε χαμηλό επίπεδο και σε υψηλό επίπεδο, παρουσιάζουν τα αποτελέσματά τους στον τελικό χρήστη σε τρισδιάστατους χάρτες.

2.5. ΑΛΓΟΡΙΘΜΟΙ ΑΝΩΝΥΜΟΠΟΙΗΣΗΣ ΔΕΔΟΜΕΝΩΝ

Οι βάσεις κινούμενων δεδομένων περιέχουν ευαίσθητη πληροφορία όπως την ταυτότητα του χρήστη (π.χ. η διεύθυνση του σπιτιού, συχνά σημεία επισκέψεως, κτλ), τα οποία παραβιάζουν την ασφάλεια της ιδιωτικής ζωής ενός ατόμου. Αυτή η πληροφορία μπορεί να αποκαλυφθεί από κακόβουλους χρήστες ή να εμφανιστεί τυχαία από διάφορους πειραματισμούς των τελικών χρηστών. Για να αποφθεχτεί αυτό το φαινόμενο, έχουν δημιουργηθεί αλγόριθμοι ανωνυμοποίησης των δεδομένων, οι οποίοι όταν διαπιστώσουν ότι μια ευαίσθητη πληροφορία πάει να αναδειχθεί στον χρήστη, καμουφλάρει τα υπάρχοντα δεδομένα με ψεύτικα. Συγκεκριμένα, οι αλγόριθμοι αυτοί βασίζονται στην θεωρία του *k-anonymity* [35], δηλαδή η θέση ενός αποθηκευμένου ατόμου στην βάση γενικεύεται με τέτοιο τρόπο ώστε να συμπεριλαμβάνεται μέσα σε μια μικρή περιοχή που θα βρίσκονται K άτομα. Έτσι, το άτομο είναι μη αναγνωρίσιμο όταν βρίσκεται ανάμεσα από K άτομα και ακόμα περισσότερο όταν βρίσκεται σε μια πολύ μικρή περιοχή και το αντίστοιχο K είναι πολύ μεγάλο. Σε αυτήν την θεωρία, έχουν βασιστεί και προταθεί πολλοί αλγόριθμοι ανωνυμοποίησης και δυο από τους πιο δημοφιλείς και σύγχρονους είναι ο *Never Walk Alone* (*NWA*) [33] και ο *Wait For Me* (*W4M*) [32].



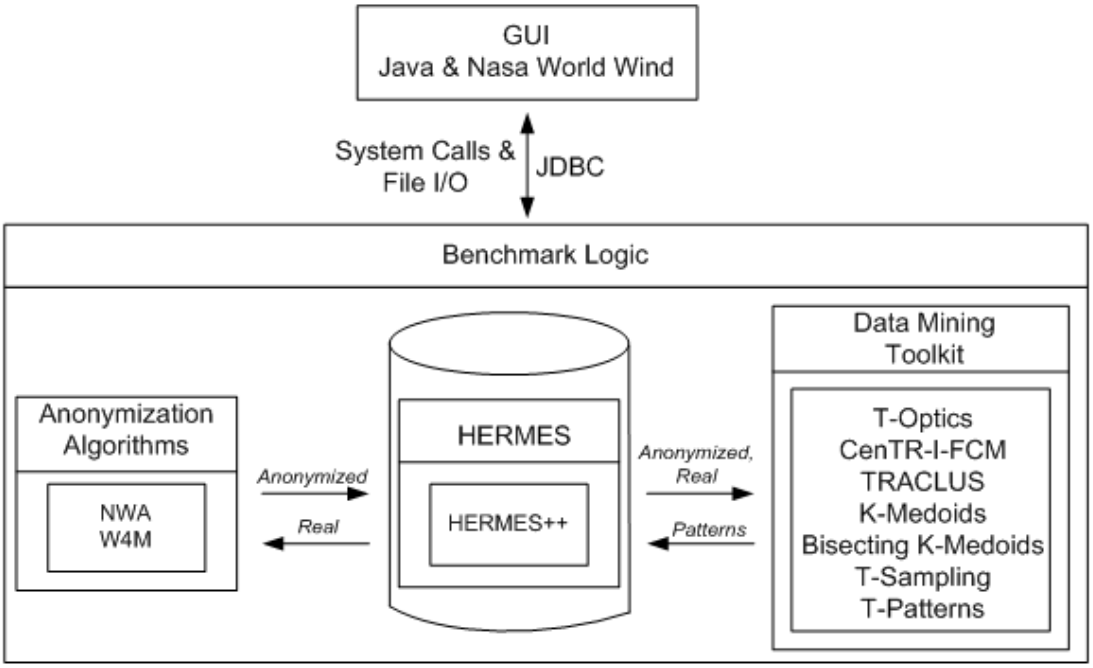
Εικόνα 2.6: α) Ανώνυμη τροχιά, ανώνυμη περιοχή, ο κύλινδρος και η πιθανή θέση της τροχιά, β) δύο ανώνυμες τροχιές με τους αντίστοιχους κυλίνδρους και τον κεντρικό κύλινδρο που περιέχει και τις δύο τροχιές [14].

Η βασική ιδέα του *NWA* είναι ότι μια τροχιά σε ένα τρισδιάστατο χώρο δεν είναι μια συνεχόμενη γραμμή αλλά ένα κύλινδρος *C* που η ακτίνα *δ* αναπαριστά την πιθανή τοποθεσία όπου η τροχιά αυτή γνωρίζουμε ότι βρίσκεται μέσα στον κύλινδρο *C* αλλά δεν ξέρουμε που ακριβώς. Αν μια άλλη τροχιά ή περισσότερες διασχίσουν τον κύλινδρο *C*, τότε οι τροχιές είναι δυσδιάκριτες η μία με την άλλη (βλέπε Εικόνα 2.6) και αυτό οδηγεί στην έννοια του *k-anonymity* όπως αναφέραμε στην προηγούμενη παράγραφο. Συγκεκριμένα, ο *NWA* εκτελείται σε τρία στάδια, στο πρώτο στάδιο γίνεται η προ-επεξεργασία, στοχεύοντας στην τμηματοποίηση της βάσης σε μεγάλες ισοδύναμες ομάδες, τα οποία περιέχουν όλες τις τροχιές που διαθέτουν ίδιους αρχικούς ή τελικούς χρόνους. Στο δεύτερο στάδιο, μια διαδικασία συσταδοποίησης αναλαμβάνει να δημιουργήσει ομάδες τροχιών μέσα σε κ παρόμοιες ομάδες με τέτοιο τρόπο ώστε να κρατάει μικρό το μήκος της ακτίνας των παραγόμενων συστάδων και να μην συμπεριλαμβάνει τα *outliers*. Στο τρίτο στάδιο, ο *NWA* διαταράσσει τις τροχιές μέσα στις συστάδες ώστε να είναι η μια

τροχιά κοντά με την άλλη αλλά μέχρι το όριο αβεβαιότητας. Από την άλλη μεριά, ο *W4M* αναπτύχθηκε για να παράγει ποιοτικότερες ανώνυμες τροχιές από τον προκάτοχό του, *NWA*, και να είναι ανεκτικός στις χρονικές τιμές, το οποίο ο *NWA* δεν διέθετε. Για περισσότερες λεπτομέρειες, οι αλγόριθμοι *NWA* και *W4M* όπως και τα αντίστοιχα άρθρα υπάρχουν στις ιστοσελίδες [14] και [15], αντίστοιχα.

3. ΕΠΙΣΚΟΠΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Ένα από τα βασικά πλεονεκτήματα του προτεινόμενου συστήματος είναι ότι παρέχει στους χρήστης την πραγματοποίηση διαφορετικών διεργασιών όσο αφορά τα κινούμενα δεδομένα, όπως φαίνεται στην Εικόνα 3.1. Ξεκινώντας, η βάση του συστήματος είναι ο *Hermes*, μια επέκταση της Oracle DB για την υποστήριξη μηχανισμών ανάλυσης πάνω στα κινούμενα δεδομένα (όπως αναφέρθηκε στο κεφάλαιο 2.1). Εκτός από τον Hermes, ο Hermes++ είναι ενσωματωμένος πάνω στην βάση, ο οποίος διαθέτει χώρο χρονικές λειτουργίες που διασφαλίζουν την ακεραιότητα των δεδομένων κίνησης.



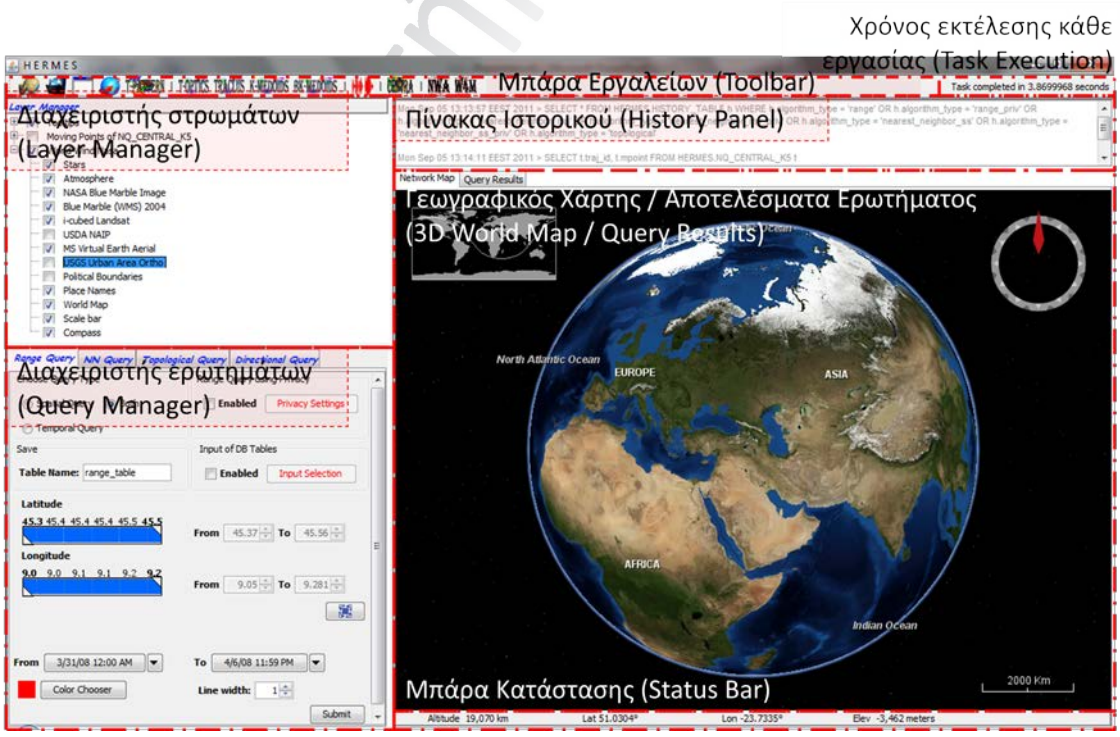
Εικόνα 3.1: Η αρχιτεκτονική του συστήματος.

Συγκεκριμένα, οι βασικές λειτουργικότητες που περιέχει το σύστημα είναι οι εξής:

- *Απλές λειτουργίες*: η πλατφόρμα είναι ικανή να εκτελεί απλά ερωτήματα, τα οποία είναι ενσωματωμένα στον HERMES, όπως ερωτήματα εύρους (*range queries*) και ερωτήματα κοντινότερου γείτονα (*nearest neighbor queries*). Τα ερωτήματα απεικονίζονται σε ένα γραφικό περιβάλλον που παρέχει θεμελιώδης δυνατότητες συμπεριλαμβανομένων της επιλογής ερωτήματος, της ρύθμισης παραμέτρων και της προβολής των αποτελεσμάτων.
- *Ερωτήματα προστασίας προσωπικών δεδομένων από τον HERMES++*: ο χρήστης έχει τη δυνατότητα να τρέξει τα απλά ερωτήματα χρησιμοποιώντας της λειτουργίες του *HERMES++*. Ο μηχανισμός του *HERMES++* έχει την ικανότητα να προστατεύει τους χρήστες που θέτουν ερωτήματα από τις διάφορες επιθέσεις όπως είναι η επίθεση ταυτοποίηση χρήστη, η επίθεση εντοπισμού ευαίσθητης τοποθεσίας και η επίθεση παρακολούθησης τροχιών που εκδίδονται από κακόβουλους χρήστες. Ο μηχανισμός αυτός προϋποθέτει ότι τουλάχιστον ένας αριθμός τροχιών επιστρέφεται στους τελικούς χρήστες, σε απάντηση των ερωτήσεων τους και για όλους τους διαφορετικούς τύπους των ερωτημάτων του *HERMES++*. Παρ' όλα αυτά, αν το αποτέλεσμα δεν διασφαλίζει την προστασία των προσωπικών δεδομένων τότε ένα σύνολο από προσεκτικών επεξεργασμένων ψεύτικων τροχιών προστίθενται στα υπάρχοντα με σκοπό να διατηρήσουν την ανωνυμία των δεδομένων και χωρίς να αλλοιώσουν το πραγματικό αποτέλεσμα.
- *Λειτουργίες Εξόρυξης γνώσης*: η πλατφόρμα διαθέτει μια εκτεταμένη υποστήριξη για την δημιουργία ομάδων και μοντέλων των λειτουργιών εξόρυξης γνώσης μέσω υλοποιήσεων διάφορων αλγόριθμων όπως είναι ο *T-Optics*, *CENTR-I-FCM*, *TRACLUS*, *K-Medoids*, *Bisecting K-Medoids*, *T-Sampling*, και ο *T-Patterns*. Αυτοί οι αλγόριθμοι μπορούν αφενός να αντιμετωπίζονται ως αυτόνομες μονάδες που ομαδοποιούν δεδομένα κίνησης και αφετέρου να συγκρίνουν τα αποτελέσματα τους για τον εντοπισμό του πιο κατάλληλου αλγόριθμου στα δοσμένα δεδομένα μέσω ενός γραφικού περιβάλλοντος.

- *Λειτουργίες αλγορίθμων ανωνυμίας και πειραμάτων*: Στην πλατφόρμα, δύο δημοφιλής αλγόριθμοι, ο *NWA* και ο *W4M*, έχουν ενσωματωθεί για να προστατεύσουν την ανωνυμία των χρηστών. Και οι δύο αλγόριθμοι λαμβάνουν ως είσοδο, τροχιές το οποίο ενδέχεται να έχουν εξαχθεί από ερωτήματα του *Hermes*, τις μετατρέπουν σε ανώνυμες τροχιές και στη συνέχεια, αποθηκεύονται στη βάση κινούμενων δεδομένων. Ένα πλεονέκτημα της πλατφόρμας είναι η δυνατότητα να σχεδιάζει και να εκτελεί πειράματα, τα οποία αξιολογούν τα αποτελέσματα από την πραγματοποίηση των αλγορίθμων ανωνυμίας σε σχέση με την παραμόρφωση που έχουν δεχθεί οι πραγματικές τροχιές. Επίσης, οι ενσωματωμένες τεχνικές εξόρυξης δεδομένων μπορούν να εφαρμοστούν πάνω στα αρχικά δεδομένα έτσι ώστε να συγκριθούν με τα πρότυπα που προκύπτουν από τα ανώνυμα δεδομένα. Αυτό μπορεί να επιτευχθεί με την εκτέλεση ερωτημάτων εξόρυξης στα αρχικά και ανώνυμα δεδομένα και συγκρίνοντας τα αποτελέσματα μεταξύ τους.

Σε υψηλότερο επίπεδο, το σύστημα παρέχει ένα γραφικό αναλυτικό εργαλείο (*visual analytic tool*), το οποίο αξιοποιεί διαδραστικά παράθυρα και τρισδιάστατες αλληλεπιδραστικές απεικονίσεις. Τεχνικά, όλο το έργο είναι υλοποιημένο στην γλώσσα Java και εμπεριέχει εξειδικευμένες εργαλειοθήκες όπως είναι η εργαλειοθήκη *Swing*, *Nasa World Wind*, *Prefuse* και *JFreechart*. Επίσης, χρησιμοποιεί ένα προσαρμοσμένο εργαλείο, το *RadioCheckBoxTree* [2] (βλέπε *Layer Manager* της Εικόνα 3.2), το οποίο επαυξάνει το συστατικό του *Swing*, *JTree* με την ενσωμάτωση των συστατικών *JRadiobutton* και *JCheckbox* μέσα στο εργαλείο ακολουθώντας το μοντέλο MVC. Περισσότερες τεχνικές λεπτομέρειες θα αναφερθούν στο κεφάλαιο 'Υλοποίηση Συστήματος'. Επιπλέον, η διεπαφή συνδέεται στην βάση δεδομένων της Oracle χρησιμοποιώντας το JDBC με σκοπό να ανακτήσει τα δεδομένα που χρειάζονται για την εκτέλεση των λειτουργιών. Αξίζει να σημειωθεί ότι οι λειτουργίες είναι υλοποιημένες με τρεις διαφορετικούς τρόπους, (α) σε εκτελέσιμα αρχεία, (β) σε SQL συναρτήσεις και (γ) σε κλάσεις της Java. Στην πρώτη περίπτωση, το πρόγραμμα τρέχει τα εκτελέσιμα αρχεία με την εφαρμογή κλήσης συστήματος. Τα εκτελέσιμα αρχεία αποθηκεύουν το αποτέλεσμα σε αρχεία κειμένου και κατόπιν, διαβάζονται από το πρόγραμμα και παρουσιάζονται στον τρισδιάστατο χάρτη. Στην δεύτερη περίπτωση, το πρόγραμμα καλεί συναρτήσεις SQL μέσω του JDBC για την εκτέλεση των λειτουργιών που είναι υλοποιημένες πάνω στην *Oracle*. Στην τρίτη περίπτωση, η εφαρμογή εκτελεί τις κλάσεις Java των αλγόριθμων άμεσα από το Virtual Machine της Java. Και στις τρεις περιπτώσεις, η πλατφόρμα παρέχει διαδραστικά παράθυρα για την ρύθμιση και εκτέλεση των λειτουργιών, οπτικοποιεί το αποτέλεσμα στο γεωγραφικό χάρτη και αποθηκεύει τα αποτελέσματα στην βάση δεδομένων κίνησης με σκοπό να αξιοποιηθούν αργότερα για περαιτέρω ανάλυση.



Εικόνα 3.2: Τα κύρια συστατικά μέρη του συστήματος.

Όσο αφορά την διεπαφή, τα κύρια συστατικά μέρη είναι 7 (βλέπε Εικόνα 3.2). Αναλυτικότερα, η Μπάρα Εργαλείων (toolbar), η οποία βρίσκεται στο πάνω αριστερό μέρος της εφαρμογής, περιέχει εικονίδια που αφορούν την εκτέλεση αλγορίθμων εξόρυξης γνώσης (π.χ. *T-Pattern*, *T-Optics*, *Traclus*, κτλ) καθώς και επιπλέον λειτουργίες του συστήματος. Αυτές οι λειτουργίες είναι τον *SQL Plus* που δίνει την δυνατότητα στον χρήστη να γράφει SQL ερωτήματα, τον *DB Connector* όπου ο χρήστης περνάει το username και το password για να συνδεθεί στην βάση δεδομένων, το *Open*, το οποίο ανοίγει τα αποθηκευμένα αποτελέσματα των ερωτημάτων και το *T-Aggregator*, ένας αλγόριθμος συνάθροισης των δεδομένων κίνησης. Επίσης, δίπλα στην Μπάρα Εργαλείων υπάρχει ένα πλαίσιο που δείχνει το χρόνο εκτέλεσης ενός ερωτήματος ή αλγορίθμου εξόρυξης γνώσης (*task execution*) μετά την ολοκλήρωσή του. Προς την αριστερή πλευρά και κάτω από την Μπάρα Εργαλείων τοποθετείται ο Διαχειριστής Στρωμάτων (*Layer Manager*). Ο Διαχειριστής Στρωμάτων είναι υπεύθυνο να ενεργοποιεί ή να απενεργοποιεί όλα τα διαθέσιμα στρώματα που υπάρχουν στο χάρτη. Μερικά από αυτά, μπορεί να ανήκουν στην βιβλιοθήκη της *World Wind Nasa* (π.χ. *Place Names*, *Compass*, κτλ) και άλλα να δημιουργούνται κατά την εκτέλεση των ερωτημάτων και μπορεί να αφορούν τα κινούμενα δεδομένα (π.χ. *Trajectories*) μέχρι τα αποτελέσματα των αλγορίθμων εξόρυξης γνώσης. Ο Διαχειριστής Ερωτημάτων (*Query Manager*), το οποίο βρίσκεται ακριβώς από κάτω από το Διαχειριστή Στρωμάτων, δίνει την δυνατότητα στον χρήστη να εκτελεί ερωτήματα του *Hermes* συμπληρώνοντας μερικά πεδία ή επιλέγοντας κάποιες προεπιλεγμένες τιμές μέσα από διαδραστικά παράθυρα. Ο Πίνακας Ιστορικού (*History Panel*) που τοποθετείται πάνω από τον *3D World Map*, περιέχει όλα τα ερωτήματα που έχουν εκτελεστεί από τον χρήστη σημειώνοντας την ημέρα, την ώρα εκτέλεσης και το αντίστοιχο SQL ερώτημα. Το κυριότερο συστατικό της εφαρμογής είναι ο Γεωγραφικός Χάρτης (*3D World Map*), το οποίο βρίσκεται στο κέντρο της πλατφόρμας και είναι το αντικείμενο που θα «φιλοξενήσει» οπτικά τα κινούμενα δεδομένα ή τα μοντέλα των αλγορίθμων εξόρυξης γνώσης. Επίσης, τα αποτελέσματα των ερωτημάτων μπορούμε να τα παρατηρήσουμε και σε μορφή κειμένου πατώντας στην ετικέτα (tab) *Query Results*. Τέλος, η Μπάρα Κατάστασης (*Status Bar*) περιλαμβάνει μερικά χαρακτηριστικά του χάρτη όπως είναι οι παγκόσμιες συντεταγμένες και το υψόμετρο του συγκεκριμένου σημείου που δείχνει ο χάρτης (τοποθετείται κάτω από το 3D World Map).

4. ΠΡΟΟΔΕΥΤΙΚΗ ΑΝΑΛΥΣΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ ΚΙΝΗΣΗΣ

4.1. ΕΙΣΑΓΩΓΗ

Οι μεγάλες ποσότητες δεδομένων κίνησης που συλλέγονται από τις συσκευές GPS, RFID και κινητά τηλέφωνα έχουν δώσει το κίνητρο στους αναλυτές να χρησιμοποιήσουν τις βάσεις τροχιών για να εξάγουν ωφέλιμη πληροφορία είτε είναι για συγκοινωνιακούς σκοπούς είτε για σκοπούς διαχείρισης είτε για οποιοδήποτε άλλο λόγο. Σε αυτήν την ενότητα, θα επισημανθούν αρχικά μεθοδολογίες για την διαχείριση μεγάλων βάσεων δεδομένων κίνησης όσο αφορά το κομμάτι της προοδευτικής ανάλυσης των δεδομένων αυτών. Καταρχήν, η προοδευτική ανάλυση των δεδομένων ή αλλιώς η προοδευτική εξόρυξη και αναζήτηση επικεντρώνονται στην σταδιακή ανάλυση των δεδομένων είτε αναζητώντας δεδομένα από ερωτήματα είτε εξορύσσοντας γνώση με την χρήση πολύπλοκων μαθηματικών αλγορίθμων. Με άλλα λόγια, ο πιο απλός και διαισθητικός τρόπος για να κάνουμε ανάλυση στα δεδομένα είναι μέσα από μια ακολουθία βημάτων. Όπου σε κάθε βήμα, ένας χρήστης θα επιλέγει ένα ερώτημα ή έναν αλγόριθμο εξόρυξης γνώσης, θα ορίζει τις παραμέτρους που απαιτούνται και θα επιλέγει ένα σύνολο δεδομένων για ανάλυση. Αυτή η διαδικασία θα ακολουθείται βήμα προς βήμα επιλέγοντας κάθε φορά διαφορετικό μηχανισμό ανάλυσης των δεδομένων και διαφορετικό σύνολο δεδομένων. Με αποτέλεσμα, ο αναλυτής προοδευτικά να εξάγει γνώση από τα δεδομένα και να συνεχίζει την διαδικασία με πιο αναλυτικές ερωτήσεις από το αποτέλεσμα της προηγούμενης ανάλυσης. Ο Rinzivillo και οι συνεργάτες του είχαν προτείνει την διαδικασία προοδευτικής συσταδοποίησης [38] όπου ακολουθούσε την παραπάνω διαδικασία αλλά χρησιμοποιώντας μόνο έναν μηχανισμό ανάλυσης, τον αλγόριθμο *T-Optics*. Σε αυτήν την έρευνα, οι συγγραφείς παρουσίασαν μια αλληλεπιδραστική πλατφόρμα που χρησιμοποιούσε οπτικές και διαδραστικές τεχνικές για να υποστηρίξει την διαδικασία. Αξίζει να σημειωθεί ότι οι οπτικές και διαδραστικές τεχνικές παίζουν καθοριστικό ρόλο στην διαχείριση μεγάλων βάσεων δεδομένων διότι βοηθούν στην ερμηνεία των αποτελεσμάτων. Πάνω σε αυτό το πλαίσιο, η εφαρμογή της παρούσας μεταπτυχιακής διατριβής έχει βασιστεί παρουσιάζοντας ένα εργαλείο που θα διευκολύνει τον χρήστη είτε είναι ένα απλό μέλος είτε ένας εμπειρογνώμονας που θα να αναλύει σταδιακά την βάση δεδομένων κίνησης της πόλης του Μιλάνο* μέσα από οπτικές και αλληλεπιδραστικές διεπαφές. Ο κύριος σκοπός της πλατφόρμας είναι ένα εργαλείο για ανάλυση μεγάλων βάσεων δεδομένων κίνησης με την χρήση οπτικών διαδραστικών εργαλείων, το οποίο θα απευθύνεται σε πολλαπλούς χρήστες.

Το υπόλοιπο της ενότητας χωρίζεται ως εξής. Στην επόμενα υποενότητα θα αναφερθούν περιληπτικά σχετικές έρευνες πάνω στην προοδευτική εξόρυξη και αναζήτηση των δεδομένων, στην συνέχεια, θα περιγραφούν αναλυτικότερα η διαδικασία του μηχανισμού αυτού που ακολούθησε η προτεινόμενη πλατφόρμα και θα τεκμηριωθούν με μερικά παραδείγματα χρήσεις. Επίσης, θα επισημανθούν μερικοί αλγόριθμοι οι οποίοι δεν παρουσιάστηκαν στην προηγούμενη υποενότητα καθώς και τα διαδραστικά εργαλεία του συστήματος που συμβάλουν στην ανάλυση της πληροφορίας. Τέλος, θα επισημανθούν συνοπτικά τα αποτελέσματα της ενότητας αναφέροντας και κάποιες μελλοντικές κατευθύνσεις.

4.2. ΣΧΕΤΙΚΗ ΕΡΕΥΝΑ

Στην βιβλιογραφία, έχουν προταθεί κυρίως έρευνες πάνω στην αναζήτηση των δεδομένων (π.χ. *range queries, nearest neighbor queries*, similarity queries, κτλ) ή στην εξόρυξη των δεδομένων κίνησης (π.χ. *T-Pattern, T-Optics*, κτλ) αλλά λίγες προσπάθειες έχουν γίνει όσο αφορά την προοδευτική εξόρυξη και αναζήτηση των δεδομένων κίνησης. Όσο αφορά το τρίτο κομμάτι, οι ερευνητές ανάπτυξαν γλώσσες ερωτημάτων για την εξόρυξη γνώσης (*Data Mining Query Languages - DMQL*) με τέτοιο τρόπο ώστε να υποστηρίζουν συνδυασμούς αναζήτησης και εξόρυξης των τροχιών σε μια βάση δεδομένων κίνησης. Στη πρώτη ερευνητική κατεύθυνση, το σημείο εστίασης ήταν να παρέχουν μια διασύνδεση μεταξύ των αρχικών δεδομένων με τα εξορυγμένα μοντέλα. Κάτω από αυτήν την προοπτική, οι γλώσσες *DMQL* ήταν το

* Η βάση δεδομένων κίνησης που χρησιμοποιήθηκε στην εφαρμογής είναι από την πόλη του Μιλάνο και περιέχει τροχιές των οχημάτων που συλλέχτηκαν από συσκευές GPS την περίοδο 31/03/2008 έως 6/04/2008. Συνολικά, η βάση του Μιλάνο περιέχει 5954 τροχιές.

κύριο μέσο για να προσδιορίσουν τα αρχικά δεδομένα με τα πρότυπα ενδιαφέροντος. Στην δεύτερη ερευνητική κατεύθυνση, οι γλώσσες *DMQL* χρησιμοποιήθηκαν για να παρέχουν ένα ειδικό ρόλο στην ανάλυση των δεδομένων κίνησης. Σε πιο πρόσφατες έρευνες, η ιδέα γενικεύτηκε και οι γλώσσες *DMQL* ενσωματώθηκαν πάνω σε ένα αφηρημένο πλαίσιο ώστε να υποστηρίζουν μια ολοκληρωμένη ανάλυση της γνώσης για οποιαδήποτε δεδομένα καθώς επίσης και για οποιοδήποτε συνδυασμό περίπλοκων οντοτήτων.

Αναλυτικότερα, μια από τις πρώτες ερευνητικές προσπάθειες πάνω στην προοδευτική ανάλυση των δεδομένων είναι *3W Model* [37][36], το οποίο περιέχει τρεις κόσμους για την εξόρυξη γνώσης: τον κόσμο των δεδομένων (*Data-World*), τον εννοιολογικό κόσμο (*Intentional-World*) και τον επεκταμένο κόσμο (*Extensional-World*). Ο *D-World* αναπαριστά τα αρχικά δεδομένα που αναλύονται στα πλαίσια βασικών οντοτήτων της σχεσιακής άλγεβρας. Ο *I-World* περιέχει τα αντικείμενα, τα οποία παρουσιάζονται ως μια ξεχωριστή τάξη των μοντέλων εξόρυξης γνώσης (π.χ. περιοχές που μπορούν να προσδιοριστούν ως συνδυασμοί συνόλων από τις οντότητες του *D-World*). Στο *E-World,* μια περιοχή απλά παρουσιάζεται ως μια απαρίθμηση όλων των οντοτήτων που ανήκουν σε αυτήν την περιοχή. Δηλαδή, οι σχέσεις αυτού του κόσμου αποσπώνται από τον συνδυασμό των σχέσεων των δύο κόσμων που προηγουμένως σχηματίστηκαν και αποτελούν ένα σχήμα που περιέχει μερικές σχέσεις από τον *D-World* και μερικές από τον *I-World.* Το αποτέλεσμα του *3W Model* μπορεί να προσδιοριστεί από το σύνολο των τριών κόσμων. Με λίγα λόγια, η λειτουργία του *I-World* εξάγει περιοχές από αυτόν τον κόσμο από τα δεδομένα του *D-World.* Στην συνέχεια, αυτές οι περιοχές μπορούν περάσουν μια επανάληψη από τον *I-World* στον *I-World* ή ο χειριστής του *E-World* να δημιουργήσει μια σχέση βασιζόμενη σε μερικές περιοχές του *I-World* και του *D-World.* Τέλος, πολυσύνθετα αντικείμενα του *E-World* μπορούν να προβληθούν στους δύο άλλους κόσμους δηλαδή να επιστρέψουν στους κόσμους *D-World* και *I-World* μέσω της επιλογή κατάλληλων χαρακτηριστικών από τον κόσμο *E-World.* Το *3W Model* παρέχει μια όψη της εξόρυξης γνώσης σε αλγεβρικούς όρους δηλαδή η διαδικασία της ανακάλυψης γνώσης είναι μια εφαρμογή από μια ακολουθία χειριστών με σκοπό να μετατρέψει ένα σύνολο πινάκων. Επίσης, οι οντότητες του *3W Model* και η υλοποίηση των χειριστών που διαθέτει είναι τα κύρια κλειδιά στην σχεδίαση ενός δυναμικού εργαλείου για την εξόρυξης των δεδομένων. Παρ' όλα αυτά, το *3W Model* έχει κάποιους βασικούς περιορισμούς όπως στον *D-World,* δεν υπάρχει πιθανότητα να εκφράσει πολύπλοκες σχέσεις λόγω ότι το μοντέλο δεδομένων έχει σταθερό βάθος. Επίσης, στον *I-World*, οι περιοχές απεικονίζονται ως γραμμικά άνισα σύνολα οπότε τα μοντέλα αυτού του κόσμου δεν είναι διατυπώσιμα με αποτέλεσμα οι αναπαραστάσεις του να απαιτούν πολύπλοκες μαθηματικές δομές.

Για να αντιμετωπιστούν οι αδυναμίες του *3W Model,* αναπτύχθηκε το *2W Model* από τον Ortale και τους συνεργάτες του, το οποίο δίνει την δυνατότητα στην περιγραφή περίπλοκων αντικειμένων και στην υποστήριξη εξαγωγής όλων των απαιτούμενων προτύπων από τα αρχικά δεδομένα. Συγκεκριμένα, το *2W Model* παρουσιάζει την διαδικασία της ανάλυσης των δεδομένων μέσα από δύο διαφορετικούς κόσμους, τον κόσμο των δεδομένων (*Data-World*) και τον κόσμο των μοντέλων (*Model-World*). Σε αυτούς τους δύο κόσμους, έχουν αναπτυχθεί τρία είδη αλληλεπίδρασης έτσι ώστε να επιτευχθεί η διαδικασία της εξόρυξης γνώσης μέσα από πολύπλοκες σχέσεις και είναι οι εξής:

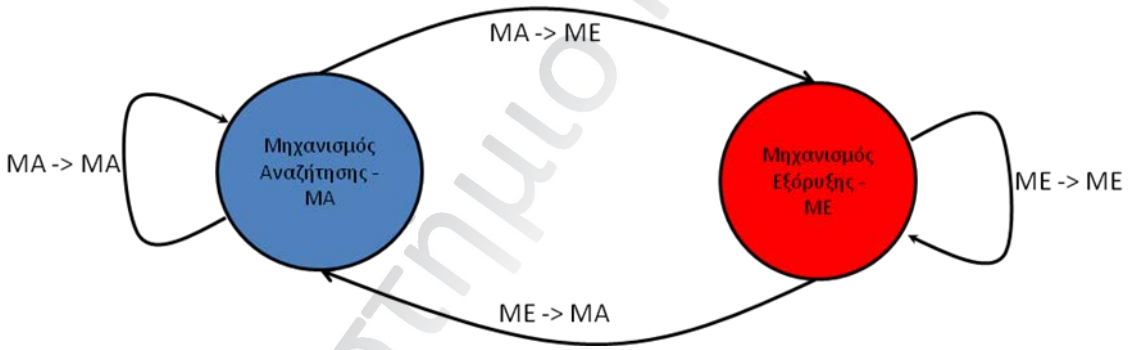
- Συναρτήσεις φιλτραρίσματος παίρνουν ένα σύνολο από οντότητες και παράγουν ένα νέο σύνολο οντοτήτων. Τέτοιοι συνδυασμοί μπορεί να είναι είτε ένα σύνολο δεδομένων να παράγει ένα νέο σύνολο δεδομένων ή ένα σύνολο μοντέλων να δημιουργεί ένα νέο σύνολο μοντέλων.
- Συναρτήσεις εξόρυξης που έχουν ως πηγή δεδομένων τα αρχικά δεδομένα και παράγουν μοντέλα. Στην πραγματικότητα, αυτές οι συναρτήσεις αφορούν τους αλγόριθμους εξόρυξης γνώσης που αναδεικνύουν συνθετικά αντικείμενα (πρότυπα) από ένα δοσμένο σύνολο αρχικών δεδομένων.
- Συναρτήσεις που παράγουν δεδομένα από ένα σύνολο μοντέλων ακολουθώντας μια μετά-επεξεργασία των δεδομένων φιλτράροντας ενδιαφέροντα πρότυπα.

Παρ' όλα αυτά, το *2W Model* υλοποιήθηκε για να υποστηρίζει μόνο εμπειρογνώμονες χρήστες, οι οποίοι θα πρέπει να ξέρουν την γλώσσα που υποστηρίζει για να συνθέσουν και να εκτελέσουν συνθετικά ερωτήματα. Ως αντίθεση με όλα αυτά, η εφαρμογή της παρούσας εργασία βασίστηκε στο *2W Model* αλλά ο σκοπός της είναι να διευκολύνει το χρήστη είτε είναι ένας απλός χρήστης είτε ένας εμπειρογνώμονας να συνθέσει και να θέσει πολύπλοκα ερωτήματα μέσα από την διαδικασία της προοδευτικής ανάλυσης των δεδομένων χρησιμοποιώντας εύχρηστες διεπαφές και αλληλεπιδραστικά εργαλεία ανάλυσης που θα απλοποιούν την εξόρυξη των δεδομένων κίνησης. Στα επόμενα υποκεφάλαια, θα περιγραφούν αναλυτικότερα την διαδικασία της προοδευτικής εξόρυξης και αναζήτησης κινούμενων δεδομένων που εφαρ-

μόζει η πλατφόρμα και θα παρουσιαστούν μερικά παραδείγματα χρήσεων για τους τρόπους που η συγκεκριμένη εφαρμογή εξάγει γνώση με την χρήση πραγματικών δεδομένων.

4.3. Ο ΜΗΧΑΝΙΣΜΟΣ ΑΝΑΖΗΤΗΣΗΣ ΚΑΙ ΕΞΟΡΥΞΗΣ

Ο μηχανισμός αναζήτησης και εξόρυξης της παρούσας μεταπτυχιακής εργασίας βασίζεται στο *2W Model* και αποτελείται από δύο διαφορετικούς μηχανισμούς: τον μηχανισμό αναζήτησης και τον μηχανισμό εξόρυξης. Οι μηχανισμοί αν και είναι δυο ξεχωριστές διεργασίες έχουν την δυνατότητα να συνδυάζονται μεταξύ τους με τέσσερις διαφορετικές διασυνδέσεις. Συγκεκριμένα, η πρώτη διασύνδεση είναι ο συνδυασμός ενός ερωτήματος που ανήκει στο μηχανισμό αναζήτησης και έχει ως είσοδο το αποτέλεσμα ενός προηγούμενου ερωτήματος του μηχανισμού αναζήτησης. Με τον ίδιο τρόπο, η δεύτερη διασύνδεση αφορά την εκτέλεση ενός αλγόριθμου από τον μηχανισμό εξόρυξης με είσοδο το αποτέλεσμα ενός προηγούμενου αλγόριθμου. Και οι δυο διασυνδέσεις εκτελούν ερωτήματα αναζήτησης ή εξόρυξης μέσα από ένα βρόγχο όπως φαίνεται στην παρακάτω εικόνα και υποδουλώνουν μια διαδικασία φιλτραρίσματος των αρχικών δεδομένων ή των μοντέλων. Επίσης, υπάρχει η δυνατότητα ο συνδυασμός ενός ερωτήματος από τον κόσμο εξόρυξης με είσοδο το αποτέλεσμα ενός ερωτήματος αναζήτησης. Αυτού του είδους οι συνδυασμοί είναι όταν ο χρήστης επιθυμεί να εκτελέσει έναν αλγόριθμο εξόρυξης γνώσης από την προεπεξεργασία των δεδομένων που έχει γίνει σε προηγούμενο στάδιο από τον μηχανισμό αναζήτησης έτσι ώστε ο αλγόριθμος να αναδείξει πιο ελκυστικά αποτελέσματα. Αντιθέτως, ένα ερώτημα αναζήτησης μπορεί να εκτελεστεί από το αποτέλεσμα ενός αλγόριθμου εξόρυξης γνώσης ως είσοδο. Αυτή η διαδικασία είναι χρήσιμη για μετά-επεξεργαστικούς λόγους όπως είναι η πράξη του φιλτραρίσματος ενδιαφερόντων προτύπων.



Εικόνα 4.1: Ο μηχανισμός αναζήτησης και εξόρυξης.

Στην παρούσα πλατφόρμα, ο μηχανισμός αναζήτησης περιέχει τριών ειδών απλών ερωτημάτων και είναι λειτουργίες που υποστηρίζει η βάση *HERMES* όπως οι ακόλουθες.

- ✓ *Range Queries* – επιλέγουμε μια περιοχή και τον αντίστοιχο χρονικό διάστημα και το επιστρεφόμενο αποτέλεσμα είναι ένα σύνολο τροχιών μέσα σε αυτήν την περιοχή που να ανήκει στο χρονικό διάστημα που του έχουμε ορίσει. Σίγουρα, όποιες τροχιές δεν τηρούν κανένα από τις παραπάνω παραμέτρους αγνοούνται εντελώς και σε όσες τροχιές, βρίσκεται ένα τμήμα μέσα στην δοσμένη περιοχή ή το χρονικό διάστημα, διατηρείται μόνο το συγκεκριμένο κομμάτι. Επίσης, μπορούμε να περιοριστούμε μόνο σε χωρικά ή χρονικά ερωτήματα, οι οποίες είναι οι υπό-περιπτώσεις των *Range Queries*.
- ✓ *Nearest Neighbor Queries* – περιλαμβάνει δύο τρόπους. Στον πρώτο, επιλέγουμε μια τροχιά και τον αριθμό των τροχιών που θα επιστρέψει όπου το αποτέλεσμα είναι ένα σύνολο τροχιών που τοποθετούνται πιο κοντά στην δοσμένη τροχιά. Στο δεύτερο, επιλέγουμε ένα σημείο και ένα χρονικό διάστημα και το αποτέλεσμα είναι ένα δοσμένο αριθμό τροχιών που βρίσκονται πιο κοντά στο δοσμένο χώρο χρονικό σημείο.
- ✓ *Topological Queries* – επιλέγουμε μια περιοχή και τον αντίστοιχο χρονικό διάστημα όπου το επιστρεφόμενο αποτέλεσμα είναι ένα σύνολο τροχιών που διέρχονται ή εξέρχονται ή και τα δύο από αυτήν την περιοχή μέσα στο χρονικό διάστημα που του έχουμε ορίσει.

Όπως και με το *D-World* του *2W Model*, ο μηχανισμός αναζήτησης είναι ο κόσμος των δεδομένων που αναπαριστά τα δεδομένα κίνησης που είναι να αναλυθούν καθώς επίσης τις ιδιότητές τους και τις αντίστοιχες σχέσεις τους. Με λίγα λόγια, ο κόσμος αυτός μπορεί να απεικονιστεί ως ένα σύνολο μιας

βάσης D = {T_1, T_2, …, T_v} όπου T ανήκει σε μια τροχιά. Κάθε $T_ι$ περιέχει ένα μοναδικό ID, το συνολικό αριθμό των σημείων και ένα σύνολο αριθμών $x_ι$, $y_ι$ και $t_ι$ όπου είναι οι χωρικές συντεταγμένες και το χρονικό διάστημα του συγκεκριμένου σημείου, αντίστοιχα.

Από την άλλη μεριά, ο μηχανισμός εξόρυξης είναι ο κόσμος των μοντέλων που απεικονίζει ένα σύνολο πολύπλοκων αντικειμένων M = {P_1, P_2, …, P_v} όπου M ανήκει σε ένα πρότυπο (π.χ. συστάδα ή μοντέλο του *T-Pattern*). Κάθε P είναι ένα αντικείμενο ενός αλγόριθμου από το μηχανισμό εξόρυξης απεικονισμένο σε διαφορετική μορφή ανάλογα με τον αλγόριθμο περιέχοντας ένα σύνολο τροχιών DT. Δηλαδή, κάθε πρότυπο P μοντελοποιείται ως μια συνάθροιση ή μια ομάδα που εμπεριέχει ένα σύνολο κινούμενων δεδομένων. Στην παρούσα πλατφόρμα, ο μηχανισμός εξόρυξης περιέχει τους αλγόριθμους εξόρυξης γνώσης οι οποίοι είναι οι εξής:

- ✓ Αλγόριθμος προτύπων όπως είναι ο *T-Pattern*
- ✓ Αλγόριθμοι συσταδοποίησης π.χ. *T-Optics, Traclus, KMedoids* και *Bisecting KMedoids*
- ✓ Αλγόριθμοι συσταδοποίησης για ανώνυμα δεδομένα π.χ. *TR-FCM, CenTR-I-FCM, CenTra* και *TX-CenTra*
- ✓ Αλγόριθμος δειγματοληψίας π.χ. *T-Sampling*
- ✓ Αλγόριθμοι ανωνυμοποίησης των δεδομένων π.χ. *Never Walk Alone* και *Wait 4 Me*

Όπως αναφέρθηκε παραπάνω, η διαδικασία προοδευτικής εξόρυξης και αναζήτησης γίνεται βήμα προς βήμα. Τεχνικά, αυτή η διαδικασία είναι απλή δηλαδή έστω ότι θέλουμε να βρούμε όλες τις τροχιές που εισήλθαν και εξήλθαν μέσα στο κέντρο του Μιλάνο στις 31/03/2008. Οι τροχιές που θα επιστρέψει η βάση αποθηκεύονται σε ένα πίνακα που του έχουμε δώσει ως παράμετρο. Στην συνέχεια, αν θέλουμε να εκτελέσουμε έναν αλγόριθμο συσταδοποίησης ή να συνεχίσουμε με ένα άλλο απλό ερώτημα πάνω στα προηγούμενα αποτελέσματα δίνουμε ως είσοδο το όνομα του αντίστοιχου πίνακα. Όλα τα ερωτήματα είτε ανήκουν στον μηχανισμό αναζήτησης είτε στον μηχανισμό εξόρυξης αποθηκεύονται σε πίνακες, οπότε η παραπάνω διαδικασία μπορεί να συνεχιστεί και με άλλα ερωτήματα αξιοποιώντας τους αντίστοιχους πίνακες. Παραδείγματος χάριν, ο χρήστης θα μπορεί να επιτύχει συνδυασμούς:

1. Ερώτημα αναζήτησης ως είσοδο και εκτέλεση ερωτήματος αναζήτησης
2. Ερώτημα αναζήτησης ως είσοδο και εκτέλεση ερωτήματος εξόρυξης
3. Ερώτημα εξόρυξης ως είσοδο και εκτέλεση ερωτήματος εξόρυξης
4. Ερώτημα εξόρυξης ως είσοδο και εκτέλεση ερωτήματος αναζήτησης

Οπότε ο χρήστης έχει την δυνατότητα π.χ. να εκτελεί τα ερωτήματα 1,1,1 δηλαδή τα ερωτήματα του *Hermes* ή να συνεχίζει την διαδικασία με ερωτήματα 2,2,3,2,4,3,4,4,1 κτλ έως ότου φθάσει στο επιθυμητό αποτέλεσμα. Στην επόμενη υποενότητα, θα περιγραφούν παραδείγματα που θα πετυχαίνουν τους παραπάνω συνδυασμούς μέσα από την εφαρμογή και θα αναδεικνύουν ενδιαφέροντα αποτελέσματα για έναν αναλυτή.

4.4. ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΕΩΝ

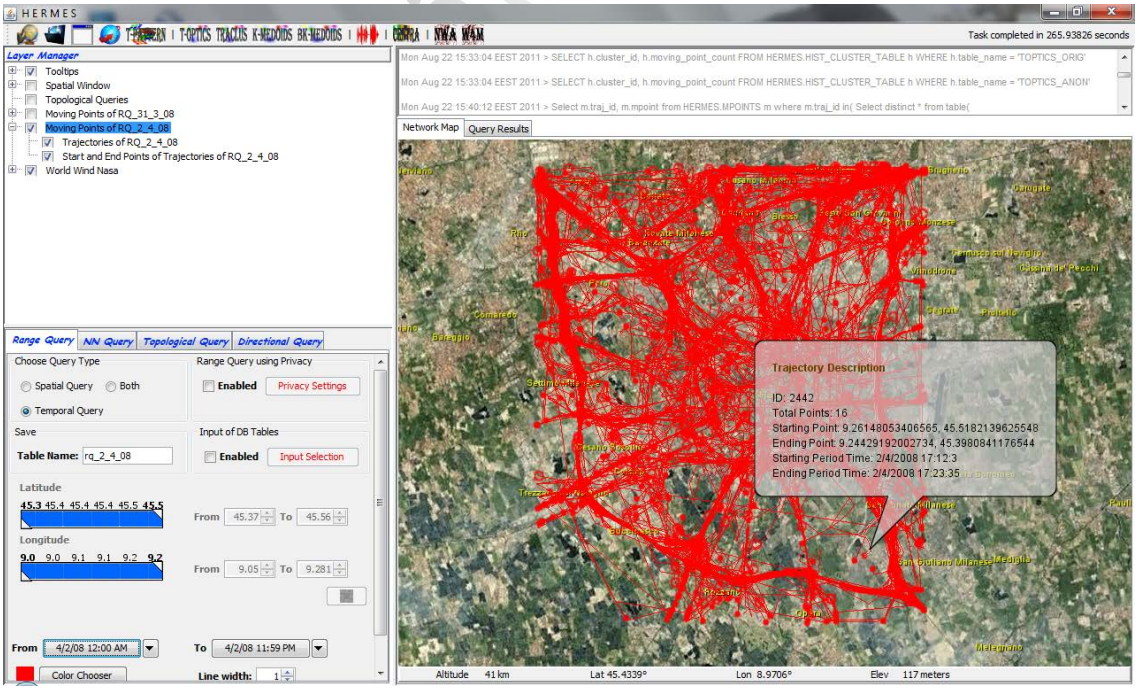
Σε αυτήν την ενότητα, θα περιγραφούν οι τέσσερις διαφορετικοί συνδυασμοί που υποστηρίζει η εφαρμογή μέσα από παραδείγματα. Στα παραδείγματα, θα επιδεικνύονται ο τρόπος που ένας χρήστης θα ακολουθεί για να εκτελέσει ένα ερώτημα και θα συνεχίζει την ανάλυση με την εφαρμογή ενός άλλου ερωτήματος πάνω στα προηγούμενα αποτελέσματα. Η διαδικασία αυτή θα συνεχίζεται σταδιακά έως ότου ο χρήστης φτάσει στο επιθυμητό αποτέλεσμα. Η βάση δεδομένων κίνησης που εφαρμόστηκε στην μελέτη περιπτώσεων είναι από την περιοχή του Μιλάνο και περιέχει τροχιές των οχημάτων που συλλέχτηκαν από συσκευές GPS την περίοδο 31/03/2008 έως 06/04/2008. Αναλυτικότερα, όπως αναφέρθηκαν στο προηγούμενο κεφάλαιο, οι τέσσερις συνδυασμοί είναι οι εξής:

- o Ερωτήματα αναζήτησης ως είσοδο και εκτέλεση ερωτημάτων αναζήτησης του *Hermes*

Η κατηγορία αυτή αφορά την εκτέλεση ερωτημάτων αναζήτησης από τον μηχανισμό του *Hermes* (π.χ. *range query, topological query*, κτλ) που ως είσοδο δέχεται τα αποτελέσματα ενός προηγούμενου ερωτήματος αναζήτησης. Συγκεκριμένα, θα παρουσιαστούν τρία διαφορετικά σενάρια, (α) θα εκτελείται ένα χρονικό ερώτημα και στην συνέχεια, το αποτέλεσμά του θα χρησιμοποιείται για την εφαρμογή ενός τοπολογικού ερωτήματος, (β) θα εκτελείται ένα τοπολογικό ερώτημα και στην συνέχεια, το αποτέλεσμά του θα χρησιμοποιείται για την εφαρμογή ενός χωρικού ερωτήματος και (γ) θα εκτελείται ένα χώρο χρο-

νικό ερώτημα και στην συνέχεια, το αποτέλεσμά του θα χρησιμοποιείται για την εφαρμογή ενός χωρικού ερωτήματος. Το τελευταίο σενάριο αναφέρεται στον συνδυασμό όχι μόνο ίδιου τύπου μηχανισμού ερωτήματος (ερώτημα αναζήτησης με ερώτημα αναζήτησης) αλλά και ίδιου τύπου ερωτήματος (*range query* με *range query*). Αυτό ισχύει για οποιοδήποτε ερώτημα του μηχανισμού *Hermes* κάνοντας με αυτόν τον τρόπο την εφαρμογή πιο ευέλικτη.

Όσο αφορά την πρώτη μελέτη περίπτωσης, ο χρήστης επιλέγει από το πλαίσιο '*Choose Query Type*', την επιλογή '*Temporal Query*' έτσι ώστε να τρέξει ένα χρονικό ερώτημα. Στην συνέχεια, συμπληρώνει το όνομα του πίνακα που θα αποθηκευτεί στην βάση στο πεδίο '*Table Name*'. Οτιδήποτε ερώτημα θέτει ο χρήστης, αυτό αποθηκεύεται στην βάση του *Hermes* ώστε να επαναχρησιμοποιείται αργότερα για περαιτέρω ανάλυση. Αφού ο χρήστης συμπληρώσει τα εισαγωγικά στοιχεία, θέτει τις παραμέτρους του χρονικού ερωτήματος της κατηγορίας '*Range Query*' δηλαδή τον αρχικό και τελικό χρόνο που θα περιλαμβάνονται μέσα στο χρονικό διάστημα τα κινούμενα δεδομένα. Αξίζει να σημειωθεί ότι δεν χρειάζεται να γράψουμε κάποια *SQL* γλώσσα για να εκτελέσουμε ένα ερώτημα, αλλά να επιλέξουμε τις παραμέτρους του κάθε ερωτήματος συμπληρώνοντας κάποιες παραμέτρους του ερωτήματος από μια εύχρηστη διεπαφή. Με αυτόν τον τρόπο, ένας χρήστης δεν χρειάζεται να είναι γνώστης κάποιας γλώσσας βάσης δεδομένων για να εκτελέσει κάποιο ερώτημα όπως έχουμε εξακολούθηση παρατηρήσει στην βιβλιογραφία αλλά μέσα από αλληλεπιδραστικά παράθυρα, θα μπορεί να εκτελεί ένα ερώτημα γρήγορα και εύκολα. Με λίγα λόγια, η πλατφόρμα δεν απευθύνεται μόνο σε εμπειρογνώμονες αλλά και σε απλούς χρήστες. Βέβαια, αν ο χρήστης επιθυμεί να εκτελέσει ένα ερώτημα γράφοντας *SQL* γλώσσα χωρίς να θέλει να ακολουθήσει τις αλληλεπιδραστικές διεπαφές, έχει την δυνατότητα να το κάνει από το εργαλείο '*SQL Plus*' (βλέπε επόμενες ενότητες). Επιπλέον, μπορούμε να αλλάξουμε τα προεπιλεγμένα χαρακτηριστικά της τροχιάς όπως είναι το χρώμα του και το πάχος του. Τέλος, αφού ολοκληρώσουμε τα στοιχεία αυτά, μπορούμε να εκτελέσουμε το ερώτημα πατώντας το κουμπί '*Submit*'. Τα αποτελέσματα απεικονίζονται πάνω στο χάρτη όπως φαίνεται στην παρακάτω εικόνα και στο παράδειγμα, παρατηρούμε όλες οι τροχιές που συλλέχτηκαν όλη την ημέρα στις 02/04/2008. Επίσης, αν μετακινήσουμε το ποντίκι πάνω σε μια τροχιά, θα εμφανιστεί μια επισημείωση (*tooltip*) που θα περιγράφει μερικά χαρακτηριστικά της τροχιάς όπως είναι το ID, τα συνολικά σημεία του, τα αρχικά και τελικά σημεία πάνω στο χάρτη καθώς και τους αρχικούς και τελικούς χρόνους. Ακόμα, ο χρήστης μπορεί να ενεργοποιήσει ή να απενεργοποιήσει τα οπτικά αποτελέσματα πάνω στο χάρτη από το πλαίσιο '*Layer Manager*' με την χρήση του εργαλείου 'CheckboxTree' που έχει εισαχθεί από τους Ακουμιανάκη Δ. και τους συνεργάτες του [2].



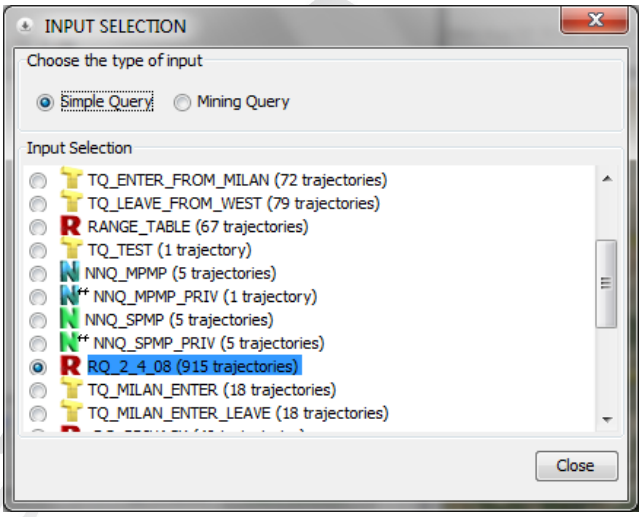
Εικόνα 4.2: Όλες οι τροχιές από την βάση του Μιλάνο για την μέρα 02/04/2008.

Κάθε ερώτημα που εκτελούμε αποτυπώνεται στο *History Panel* που βρίσκεται πάνω από τον χάρτη. Συγκεκριμένα, σε κάθε ερώτημα σημειώνεται η ώρα που εκτελέστηκε το ερώτημα, ακολουθείται ένα

βελάκι '>' και τέλος το ερώτημα στην μορφή *SQL*. Για το παράδειγμα μας, το *SQL* ερώτημα φαίνεται στο παρακάτω πίνακα και αναλυτικότερα, η βάση επιστρέφει όλα τα id (m.traj_id) και τα αντίστοιχα κινούμενα δεδομένα (m.mpoint) από τον πίνακα 'mpoints', το οποίο εκπληρώνει τις προϋποθέσεις μέσα στο 'WHERE'. Το αντικείμενο 'mpoint' είναι μια επέκταση της Oracle και είναι υλοποιημένο από τον N. Πελέκη [26] με σκοπό να υποστηρίζει χώρο χρονικά δεδομένα όπως είναι μια τροχιά ενός οχήματος. Επίσης, το ερώτημα που διατυπώνεται στον παρακάτω πίνακα ανήκει στον μηχανισμό του Hermes [28] όπως αναφέρθηκε στο 2ο κεφάλαιο.

```
SELECT m.traj_id, m.mpoint.at_period(tau_tll.d_period_sec(
tau_tll.D_Timepoint_Sec(2008,4,2,0,0,0),
tau_tll.D_Timepoint_Sec(2008,4,2,23,59,59))) FROM HERMES.MPOINTS m
WHERE m.mpoint.at_period(tau_tll.d_period_sec(
tau_tll.D_Timepoint_Sec(2008,4,2,0,0,0),
tau_tll.D_Timepoint_Sec(2008,4,2,23,59,59))) IS NOT NULL;
```

Όπως παρατηρούμε στην Εικόνα 4.2, ένας αναλυτής δεν μπορεί να εξάγει χρήσιμα συμπεράσματα λόγω της μεγάλης ποσότητας των δεδομένων που περιέχεται στον χάρτη. Για αυτόν τον λόγο, ίσως επιλέξει να διευρύνει βαθύτερα την ανάλυση με ένα επιπλέον ερώτημα ακολουθώντας μια προοδευτική αναζήτηση των δεδομένων κίνησης. Στο παράδειγμα, έχουμε επιλέξει την κατηγορία ερωτήματος '*Topological Query*' και στην συνέχεια, αφού ενεργοποιήσουμε την επιλογή '*Input Selection*', μπορούμε να επιλέξουμε ένα αποθηκευμένο πίνακα από την βάση για να συμπεριληφθεί ως είσοδος στο τοπολογικό ερώτημα. Όπως βλέπουμε στην παρακάτω εικόνα, έχουμε επιλέξει τον πίνακα 'RQ_2_4_08', το οποίο είναι το αποτέλεσμα του προηγούμενου ερωτήματος που θέσαμε στην βάση και περιέχει συνολικά 915 τροχιές (ο αριθμός μέσα στην παρένθεση).

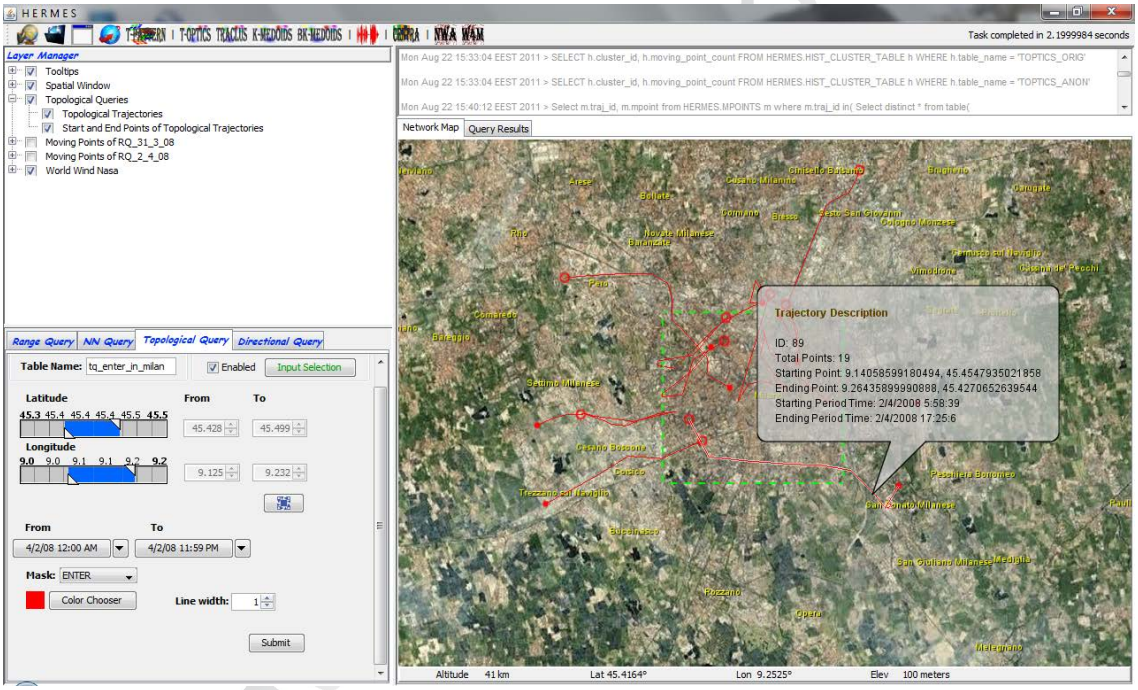


Εικόνα 4.3: Επιλογή του πίνακα 'RQ_2_4_08' για την εκτέλεση ενός τοπολογικού ερωτήματος.

Αξίζει να σημειωθεί ότι αν ενεργοποιήσουμε την επιλογή '*Input Selection*', θα τρέξουμε ένα υποσύνολο των κινούμενων δεδομένων από την βάση και πιο συγκεκριμένα, θα είναι οι τροχιές που αποθηκεύτηκαν από προηγούμενες ερωτήσεις από τον χρήστη διαφορετικά αν είναι απενεργοποιημένο, το ερώτημα εκτελείται παίρνοντας ως είσοδο ολόκληρη την βάση δεδομένων κίνησης.

Γενικά, η κατηγορία '*Topological Query*' εκπληρώνει ερωτήματα που σχετίζονται με την τοπολογία σε ένα χρονικό αντικείμενο. Δηλαδή, ένα τοπολογικό ερώτημα επιστρέφει όλες τις τροχιές που εισέρχονται ή εξέρχονται από ένα χρονικό παράθυρο. Το αντικείμενο αναφοράς είναι ένα Static Spatial αντικείμενο και επιστρέφει ως αντικείμενο δεδομένων κινούμενα δεδομένα. Έστω, λοιπόν ότι θέλουμε να βρούμε όλες τις τροχιές από το πίνακα που επιλέξαμε παραπάνω (RQ_2_4_08) που εισήλθαν μέσα στην πόλη του Μιλάνο για την μέρα 02/04/2008 έτσι ώστε να αναλύσουμε την κυκλοφοριακή συμφόρηση εκείνης της χρονικής περιόδου. Ξεκινώντας, μπορούμε να επιλέξουμε την περιοχή του Μιλάνο είτε με τους *bisliders* είτε με το εργαλείο  απευθείας πάνω στο χάρτη. Αξίζει να σημειωθεί ότι τα δυο αντικείμενα αυτά είναι συγχρονισμένα οπότε μπορούμε να χρησιμοποιήσουμε και τα δύο για την επιλογή μιας περιοχής

στο χάρτη. Παραδείγματος χάριν, ο χρήστης μπορεί να επιλέξει μια περιοχή πάνω στο χάρτη και μετά να κατευθυνθεί στους bisliders για να κάνει μερικές διορθώσεις στην αρχική του επιλογή. Στην συνέχεια, επιλέγουμε την χρονική διάρκεια της περιοχής ενδιαφέροντος από ένα πάνελ ώρας. Στο συγκεκριμένο παράδειγμα (βλέπε παρακάτω εικόνα), επιλέξαμε ως αρχικό χρόνο 02/04/2008 12:00πμ και ως τελικό χρόνο 02/04/2008 11:59μμ . Στην συνέχεια, επιλέγουμε από το πεδίο '*Mask*', το enter, το οποίο αφορά αν οι τροχιές θα εισέρχονται (στην προκειμένη περίπτωση, έχουμε επιλέξει αυτό) ή εξέρχονται ή και τα δυο από την περιοχή ενδιαφέροντος (δηλαδή το κέντρο του Μιλάνο). Αφού συμπληρώσουμε τα πεδία αυτά πιέζουμε το κουμπί '*Submit*' για να πραγματοποιηθεί το ερώτημα και τα αποτελέσματα παρουσιάζονται στην παρακάτω εικόνα. Όπως και με το προηγούμενο παράδειγμα, οτιδήποτε ερώτημα κι αν εκτελούμε, ένα νέο στρώμα προστίθεται στο '*Layer Manager*', ώστε να μπορούμε να το διαχειριστούμε. Δηλαδή, να μπορούμε να ενεργοποιούμε ή να απενεργοποιούμε τα στοιχεία του κάθε αποτελέσματος είτε είναι οι τροχιές ξεχωριστά είτε είναι τα αρχικά και τελικά σημεία των τροχιών. Εν συντομία, οι τροχιές που στο χάρτη αναπαριστώνται ως καμπυλωμένες γραμμές και στο διαχειριστή στρώματος ως 'Topological Trajectories' αλλά και τα αρχικά και τελικά σημεία που στο χάρτη απεικονίζονται ως κύκλοι (ανοικτοί και γεμάτοι, αντίστοιχα) και στο διαχειριστή στρώματος ως 'Start and End Points of Topological Trajectories' εμφανίζονται ή διαγράφονται από τον γεωγραφικό χάρτη ανάλογα με το αν είναι επιλεγμένο το αντίστοιχο *checkbox*.



Εικόνα 4.4: Οι τροχιές που εισέρχονται μέσα στο κέντρο του Μιλάνο.

Ένα SQL τοπολογικό ερώτημα είναι όπως στο παρακάτω πίνακα και ως παραμέτρους δέχεται ένα χωρικό παράθυρο (δεξιά κάτω και αριστερά πάνω συντεταγμένες), το χρονικό διάστημα (αρχικός και τελικός χρόνος) και το τύπο του '*Mask*' (enter, leave ή enter/leave).

```
SELECT m.traj_id, m.mpoint FROM HERMES.RQ_2_4_08 m WHERE m.traj_id
IN( SELECT DISTINCT * FROM TABLE( tbFunc-
tions.tb_topological_query( SDO_GEOMETRY(2003, 82087, NULL,
SDO_ELEM_INFO_ARRAY(1,1003,3),
SDO_ORDINATE_ARRAY(1509693.29691271, 5030301.22823545,
1518274.1996853, 5038566.1074933)),
tau_tll.D_period_sec(tau_tll.d_timepoint_sec(2008,4, 2, 0, 0,0),
tau_tll.d_timepoint_sec(2008, 4, 2, 23, 59, 59)), 'ENTER')));
```

Επιπλέον, μπορούμε να παρατηρήσουμε τα αποτελέσματα των ερωτημάτων με την μορφή κειμένου αν πατήσουμε την ετικέτα '*Query Results*' όπως φαίνεται στην παρακάτω εικόνα. Οι συνολικές τροχιές που έχει επιστρέψει είναι 9 και τα πεδία που περιέχει είναι ένας αύξοντα αριθμός (ROW), ο κωδικός της τρο-

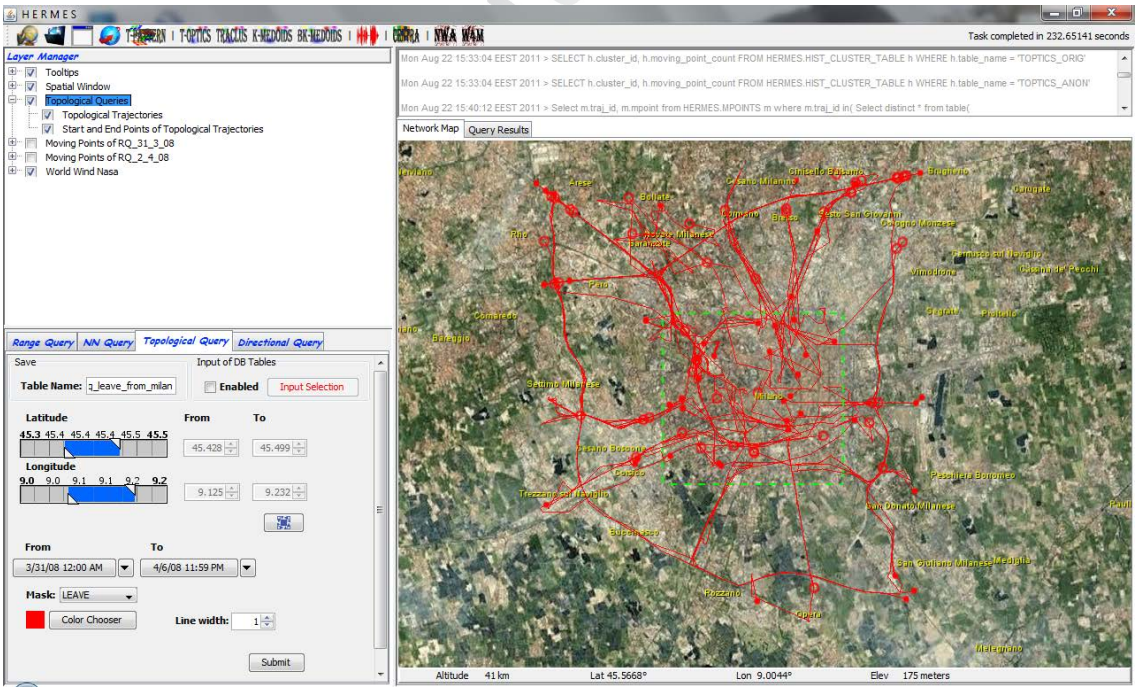
χιάς (Trajectory ID) και το αντικείμενο της τροχιάς (Moving_point) όπως αυτό αποθηκεύεται στην βάση. Γενικά, παρατηρούμε ότι ένας αναλυτής ακολουθώντας την διαδικασία της προοδευτικής ανάλυσης των δεδομένων, έχει την δυνατότητα να εξάγει πιο χρήσιμα συμπεράσματα από ότι θα είχε πετύχει κάνοντας ένα ερώτημα σε ολόκληρη την βάση. Ο λόγος είναι ότι σε κάθε στάδιο παρατηρεί την μορφή των δεδομένων και κινείται βαθύτερα σε σχέση με το τι θέλει να αναλύσει.

Network Map | Query Results

ROW	Trajectory ID	Moving Point
1	31	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2008
2	35	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2008
3	40	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2008
4	59	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2008
5	70	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2008
6	81	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2008
7	89	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2008
8	102	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2008
9	127	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2008

Εικόνα 4.5: Οι τροχιές παρουσιάζονται σε μορφή κειμένου από την ετικέτα 'Query Results'. Στο παρόν παράδειγμα, οι συνολικές τροχιές που έχει επιστρέψει το τοπολογικό ερώτημα είναι 9.

Σχετικά με το δεύτερο σενάριο, θα εκτελέσουμε πρώτα ένα τοπολογικό ερώτημα και στην συνέχεια, τα αποτελέσματα του θα χρησιμοποιηθούν για ένα χωρικό ερώτημα. Έστω, λοιπόν ότι θέλουμε να βρούμε όλες τις τροχιές από την βάση που εξήλθαν από το κέντρο του Μιλάνο. Ξεκινώντας, μπορούμε να επιλέξουμε την περιοχή του Μιλάνο όπως και παραπάνω είτε με τους *bisliders* είτε με το εργαλείο  απευθείας πάνω στο χάρτη. Στην συνέχεια, επιλέγουμε την χρονική διάρκεια της περιοχής ενδιαφέροντος από τα πάνελ ώρας. Στο συγκεκριμένο παράδειγμα, επιλέξαμε ως αρχικό χρόνο την μέρα 31/03/2008 12:00πμ και ως τελικό χρόνο την μέρα 02/04/2008 11:59μμ . Μετέπειτα, επιλέγουμε από το πεδίο '*Mask*', το leave, το οποίο αφορά αν οι τροχιές εξέρχονται από το κέντρο του Μιλάνο. Αφού συμπληρώσουμε τα παραπάνω πεδία, εκτελούμε το ερώτημα με το κουμπί '*Submit*' και τα αποτελέσματα απεικονίζονται στο χάρτη (βλέπε Εικόνα 4.6).



Εικόνα 4.6: Όλες οι τροχιές που εξέρχονται από την πόλη του Μιλάνο.

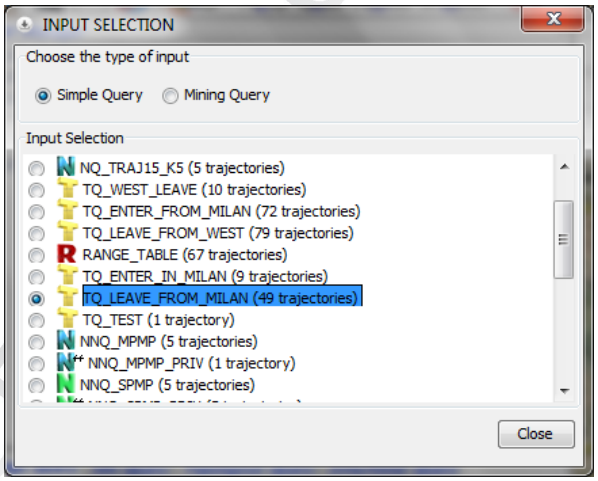
Όπως παρατηρούμε από την πάνω εικόνα, οι πολίτες του Μιλάνο κατευθύνονται κυρίως βορειοδυτικά ή βορειοανατολικά και υπάρχει μικρότερη κίνηση στην νότια πλευρά του Μιλάνο. Όσο αφορά την βορειοδυτική πλευρά, οι Μιλανέζοι χρησιμοποιούν δύο εναλλακτικές διαδρομές για να επιστρέψουν στα σπίτια τους από την δουλειά τους. Η πρώτη διαδρομή είναι η πιο κοντινή και περνάει μέσα από το κέντρο του Μιλάνο αλλά είναι πιο πυκνή από την δεύτερη διαδρομή που διέρχεται από την εξωτερική δυτική

πλευρά της πόλης και κατευθύνεται βόρεια (προς τα πάνω). Οι πιο πιθανοί λόγοι που μπορεί να συμβαίνει αυτό είναι ότι οι Μιλανέζοι μπορεί να θέλουν να αποφύγουν την κίνηση που υπάρχει στην πρώτη διαδρομή ή να θέλουν να περάσουν από κάποιο σημείο ενδιαφέροντος (π.χ. supermarket ή εστιατόριο) που τυχόν να υπάρχει στην δεύτερη διαδρομή όταν τελειώνουν από την δουλειά και πριν φθάσουν στα σπίτια τους. Βέβαια, τέτοια συμπεράσματα είναι πολύ χρήσιμα για ένα συγκοινωνιολόγο για να αναλύσει την κυκλοφοριακή κίνηση της πόλης ή για έναν επιχειρηματία για οικονομικούς λόγους.

Το SQL ερώτημα γι' αυτό το παράδειγμα ήταν το εξής:

```
SELECT m.traj_id, m.mpoint FROM HERMES.MPOINTS m WHERE m.traj_id
IN( SELECT DISTINCT * FROM TABLE( tbFunc-
tions.tb_topological_query( SDO_GEOMETRY(2003, 82087, NULL,
SDO_ELEM_INFO_ARRAY(1,1003,3),
SDO_ORDINATE_ARRAY(1509692.90610237, 5030555.673798,
1518274.1996853, 5038566.1074933)),
tau_tll.D_period_sec(tau_tll.d_timepoint_sec(2008,3, 31, 0, 0,0),
tau_tll.d_timepoint_sec(2008, 4, 6, 23, 59, 59)), 'LEAVE')));
```

Με βάση τα παραπάνω, ένας αναλυτής μπορεί να θέλει να περιοριστεί στα κινούμενα δεδομένα της δυτικής πλευράς του Μιλάνο, οπότε να συνεχίσει σε μια επόμενη ερώτηση πάνω στα αποτελέσματα του προηγούμενου ερωτήματος. Συγκεκριμένα, έχουμε επιλέξει την κατηγορία ερωτήματος '*Range Query*' και στην συνέχεια, αφού ενεργοποιήσουμε την επιλογή '*Input Selection*', μπορούμε να επιλέξουμε ένα αποθηκευμένο πίνακα από την βάση για να συμπεριληφθεί ως είσοδος στο χωρικό ερώτημα. Όπως βλέπουμε στην παρακάτω εικόνα, έχουμε επιλέξει τον πίνακα 'TQ_LEAVE_FROM_MILAN', το οποίο είναι το προηγούμενο ερώτημα που θέσαμε στην βάση και περιέχει συνολικά 49 τροχιές όπως φαίνεται μέσα στην παρένθεση.

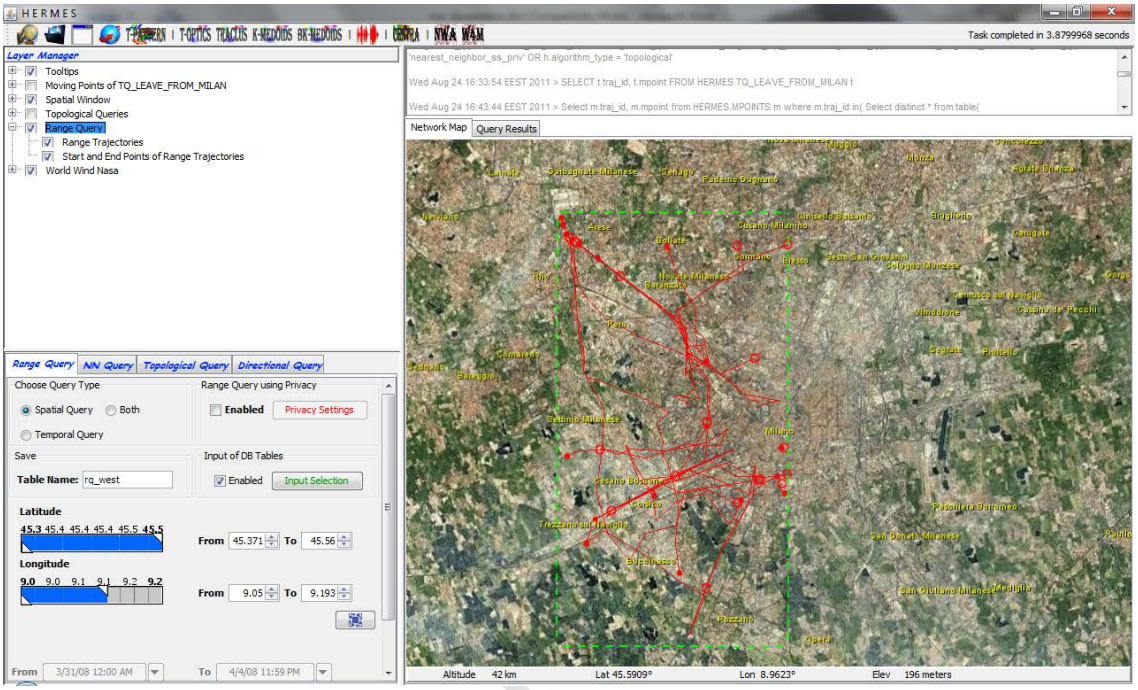


Εικόνα 4.7: Επιλογή του πίνακα 'TQ_LEAVE_FROM_MILAN' για την εκτέλεση ενός χωρικού ερωτήματος.

Έπειτα, επιλέγουμε από το πλαίσιο '*Choose Query Type*', την επιλογή '*Spatial Query*' έτσι ώστε να τρέξει ένα χωρικό ερώτημα και συμπληρώνουμε το όνομα του πίνακα (δηλαδή το rq_west) που θα αποθηκευτεί στην βάση στο πεδίο '*Table Name*'. Όπως και με το τοπολογικό ερώτημα, επιλέγουμε την δυτική περιοχή του Μιλάνο είτε με τους *bisliders* είτε με το εργαλείο  απευθείας πάνω στο χάρτη και τέλος, εκτελούμε το ερώτημα (βλέπε Εικόνα 4.8).

Το αντίστοιχο SQL ερώτημα γι' αυτό το παράδειγμα είναι το εξής:

```
SELECT m.traj_id, m.mpoint.f_intersection(SDO_GEOMETRY(2003, 82087,
NULL, SDO_ELEM_INFO_ARRAY(1,1003,3),
SDO_ORDINATE_ARRAY(1503924.18833113, 5024193.11778193,
1515072.38243052, 5045197.11178347)),0.001) FROM HER-
MES.TQ LEAVE FROM MILAN m;
```



Εικόνα 4.8: Οι τροχιές που βρίσκονται δυτικά από το κέντρο του Μιλάνο.

Οι συνολικές τροχιές που επέστρεψε το χωρικό ερώτημα ήταν 18 και φαίνονται με την μορφή κειμένου στην παρακάτω εικόνα.

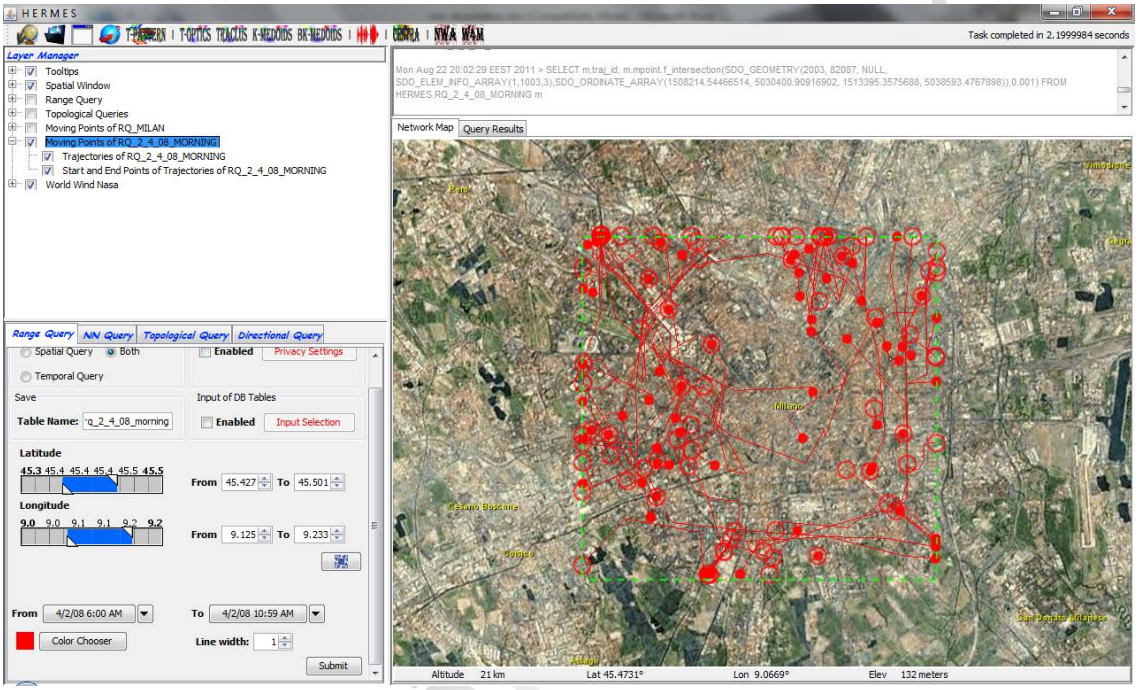
Network Map | Query Results

ROW	Trajectory ID	Moving Point
1	4775	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
2	4871	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
3	4825	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
4	4795	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
5	4807	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
6	4790	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
7	4855	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
8	4786	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
9	4859	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
10	4888	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
11	4797	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
12	4873	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
13	4891	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
14	4851	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
15	4852	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
16	4862	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
17	4796	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20
18	4783	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(20

Εικόνα 4.9: Οι τροχιές του πίνακα 'RQ_WEST' ως μορφή κειμένου.

Όσο αφορά το τρίτο σενάριο, θα εκτελέσουμε αρχικά ένα χώρο χρονικό ερώτημα και στην συνέχεια, θα ολοκληρώσουμε το παράδειγμα με ένα χωρικό ερώτημα από το αποτέλεσμα του προηγούμενου ερωτήματος. Αξίζει να σημειωθεί ότι τα δύο ερωτήματα ανήκουν στην κατηγορία '*Range Query*' οπότε ένας χρήστης έχει την δυνατότητα να ακολουθήσει την προοδευτική αναζήτηση των δεδομένων κίνησης μέσω από κοινού μηχανισμού ερωτημάτων δηλαδή '*Range Query*' με '*Range Query*', '*Topological Query*' με '*Topological Query*', κτλ. . Έστω, λοιπόν ότι θέλουμε να βρούμε όλες τις τροχιές από ολόκληρη την βάση που βρίσκονται μέσα στο κέντρο του Μιλάνο στις ώρες αιχμής (6:00π.μ. – 11:00π.μ.) για την μέρα

Τετάρτη 02/04/08. Όπως και παραπάνω, επιλέγουμε από το πλαίσιο '*Choose Query Type*', την επιλογή '*Both*' έτσι ώστε να τρέξει ένα χώρο χωρικό ερώτημα, συμπληρώνουμε το όνομα του πίνακα και επιλέγουμε την περιοχή του Μιλάνο όπως φαίνεται στην παρακάτω εικόνα, το πράσινο παράθυρο. Τέλος, επιλέγουμε την χρονική διάρκεια από τα πάνελ ώρας και εκτελούμε το ερώτημα πιέζοντας το κουμπί '*Submit*'. Τα αποτελέσματα απεικονίζονται στο χάρτη (βλέπε Εικόνα 4.10) και διακρίνουμε ότι υπάρχει κίνηση στην δυτική και βορειοανατολική πλευρά και λιγότερο στην νότια πλευρά του κέντρου του Μιλάνο. Λόγω του ότι έχουμε επιλέξει μια καθημερινή μέρα σε μια πρωινή ώρα αιχμής, η αυξημένη κίνηση στην δυτική και βορειοανατολική πλευρά του Μιλάνο μπορεί να οφείλεται στους χώρους εργασίας που τοποθετούνται σε αυτές τις περιοχές.



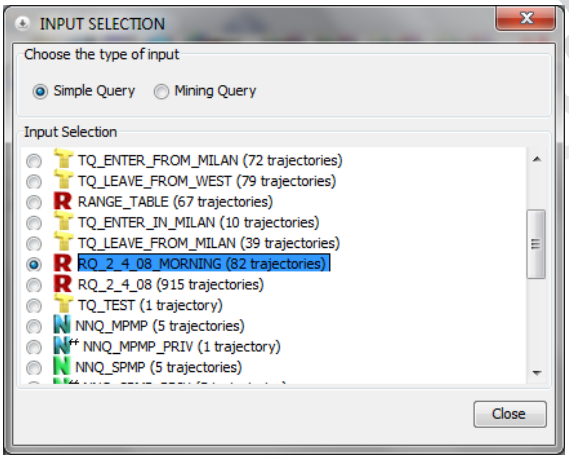
Εικόνα 4.10: Οι τροχιές που βρίσκονται μέσα στο κέντρο του Μιλάνο στις ώρες αιχμής (6:00π.μ. – 11:00π.μ.) την μέρα Τετάρτη 02/04/08.

Το SQL ερώτημα γι' αυτό το παράδειγμα είναι το εξής:

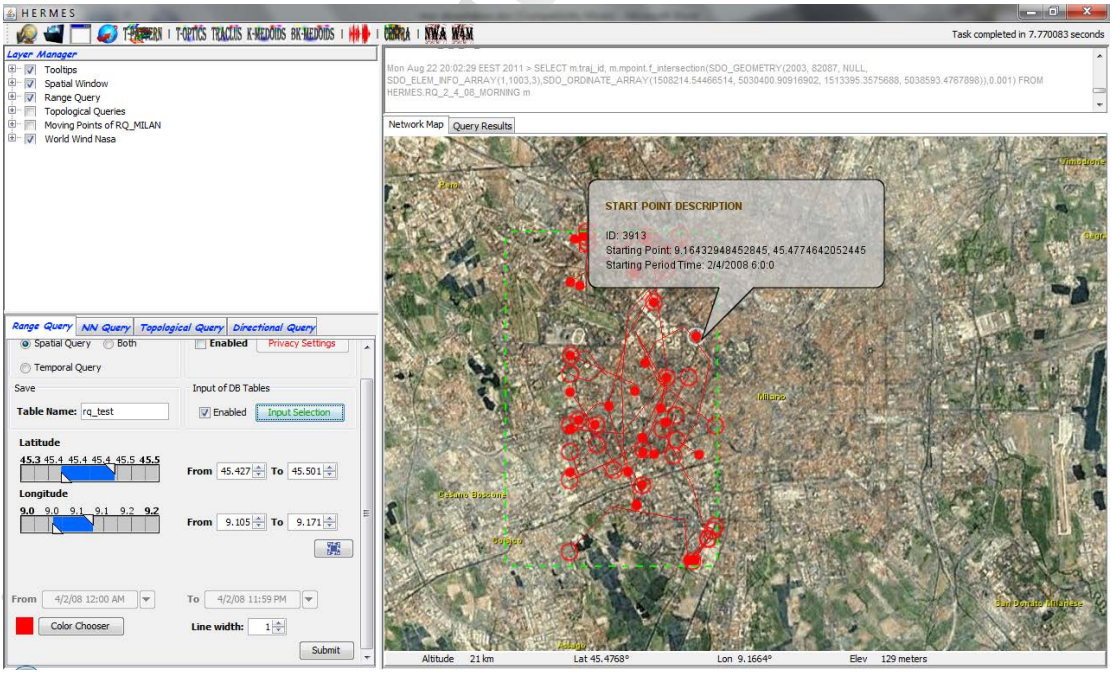
```
SELECT m.traj_id, m.mpoint.at_period(tau_tll.d_period_sec(
tau_tll.D_Timepoint_Sec(2008,4,2,6,0,0),
tau_tll.D_Timepoint_Sec(2008,4,2,10,59,59))).f_intersection2(mdsys
.sdo_geometry(2003, 82087, NULL,
mdsys.sdo_elem_info_array(1,1003,3),
mdsys.sdo_ordinate_array(1509796.41016974, 5030535.15987181,
1518248.24944689, 5038605.68870805)), 0.005) FROM MPOINTS m WHERE
hpv.minof_2(tau_tll.d_period_sec(tau_tll.D_Timepoint_Sec(2008,4,2,
6,0,0),
tau_tll.D_Timepoint_Sec(2008,4,2,10,59,59)).e.get_abs_date(),
m.mpoint.f_final_timepoint().get_abs_date()) >
hpv.maxof_2(tau_tll.d_period_sec(tau_tll.D_Timepoint_Sec(2008,4,2,
6,0,0),
tau_tll.D_Timepoint_Sec(2008,4,2,10,59,59)).b.get_abs_date(),
m.mpoint.f_initial_timepoint().get_abs_date()) AND
m.mpoint.at_period(tau_tll.d_period_sec(tau_tll.D_Timepoint_Sec(20
08,4,2,6,0,0),
tau_tll.D_Timepoint_Sec(2008,4,2,10,59,59))).f_intersection2(mdsys
.sdo_geometry(2003, 82087, NULL,
```

```
mdsys.sdo_elem_info_array(1,1003,3),
mdsys.sdo_ordinate_array(1509796.41016974, 5030535.15987181,
1518248.24944689, 5038605.68870805)), 0.005) IS NOT NULL;
```

Αφού θέλουμε να φιλτράρουμε τα αποτελέσματα του χώρο χρονικού ερωτήματος περιορίζοντας στα δεδομένα κίνησης που βρίσκονται δυτικά του κέντρου του Μιλάνου, ενεργοποιούμε την επιλογή '*Input Selection*' και επιλέγουμε τον πίνακα από το προηγούμενο αποτέλεσμα (βλέπε Εικόνα 4.11). Τέλος, ακολουθούμε ακριβώς την ίδια διαδικασία που εφαρμόσαμε παραπάνω για την εκτέλεση ενός χωρικούς ερωτήματος. Τα οπτικά αποτελέσματα αλλά και το αντίστοιχο SQL ερώτημα απεικονίζονται στις παρακάτω εικόνες.



Εικόνα 4.11: Επιλογή του πίνακα 'RQ_2_4_08_MORNING' για την εκτέλεση ενός χωρικού ερωτήματος.



Εικόνα 4.12: Οι τροχιές που βρίσκονται στην δυτική πλευρά του κέντρου του Μιλάνο.

Τελειώνοντας, όταν μετακινήσουμε το ποντίκι σε ένα αρχικό ή τελικό σημείο πάνω στο χάρτη, εμφανίζεται ένα tooltip, το οποίο περιγράφει μερικά στοιχεία του συγκεκριμένου σημείου (id, latitude και longitude).

```
SELECT m.traj_id, m.mpoint.f_intersection(SDO_GEOMETRY(2003, 82087,
NULL,
SDO_ELEM_INFO_ARRAY(1,1003,3),SDO_ORDINATE_ARRAY(1508214.54466514,
5030400.90916902, 1513395.3575688, 5038593.4767898)),0.001) FROM
HERMES.RQ 2 4 08 MORNING m;
```

Οι συνολικές τροχιές που επέστρεψε το χωρικό ερώτημα ήταν 36 και φαίνονται με την μορφή κειμένου στην Εικόνα 4.13.

Network Map | Query Results

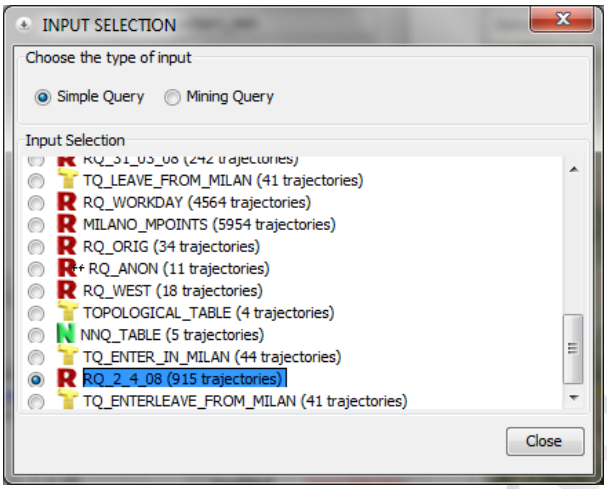
ROW	Trajectory ID	Moving Point
2	3913	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
3	2265	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
4	3834	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
5	2688	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
6	2771	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
7	5474	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
8	4171	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
9	1050	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
10	181	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
11	586	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
12	720	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
13	102	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
14	1844	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
15	510	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
16	1953	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
17	1432	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
18	698	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
19	4558	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
20	3056	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
21	2144	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
22	4519	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
23	2511	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
24	4279	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
25	208	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
26	1109	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
27	3242	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
28	4538	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
29	716	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
30	2277	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
31	2131	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
32	3015	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
33	5068	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
34	1161	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
35	3914	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(
36	855	HERMES.MOVING_POINT(HERMES.MOVING_POINT_TAB(HERMES.UNIT_MOVING_POINT(Tau_tll.d_period_sec(TAU_tll.D_timepoint_Sec(2(

Εικόνα 4.13: Τα δεδομένα του πίνακα 'RQ_TEST' ως μορφή κειμένου. Συνολικά, οι τροχιές είναι 36.

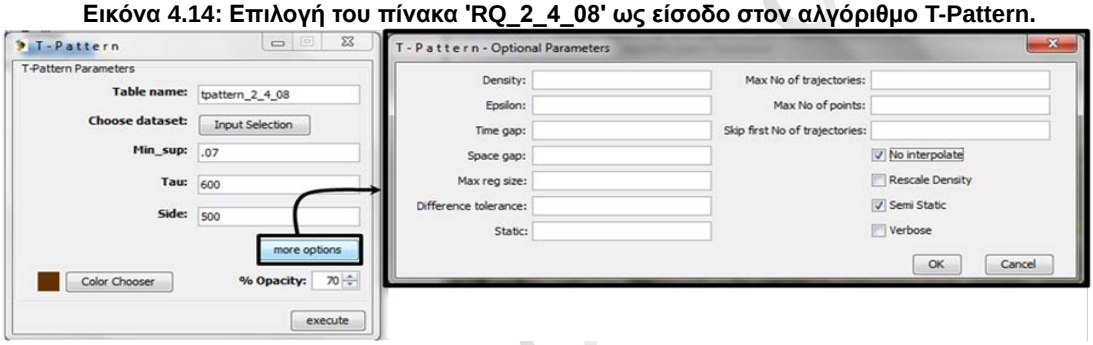
- Ερώτημα αναζήτησης ως είσοδο και εκτέλεση ερωτήματος εξόρυξης

Σε αυτήν την κατηγορία, η διαδικασία που έπεται είναι η εφαρμογή ενός ερωτήματος εξόρυξης γνώσης από τα αποτελέσματα ενός ερωτήματος αναζήτησης. Αναλυτικότερα, θα περιγραφούν τρία σενάρια χρησιμοποιώντας την ίδια πηγή δεδομένων για το καθένα. Η πηγή δεδομένων είναι ο πίνακας 'RQ_2_4_08' που δημιουργήθηκε από τα προηγούμενα ερωτήματα και περιλαμβάνει τις τροχιές που συλλέχτηκαν την μέρα 02/04/2008. Στο πρώτο σενάριο, θα εκτελεστεί ο αλγόριθμος *T-Pattern*, στο δεύτερο, ο αλγόριθμος συσταδοποίησης *T-Optics* και στο τρίτο, ο *Tr-FCM*.

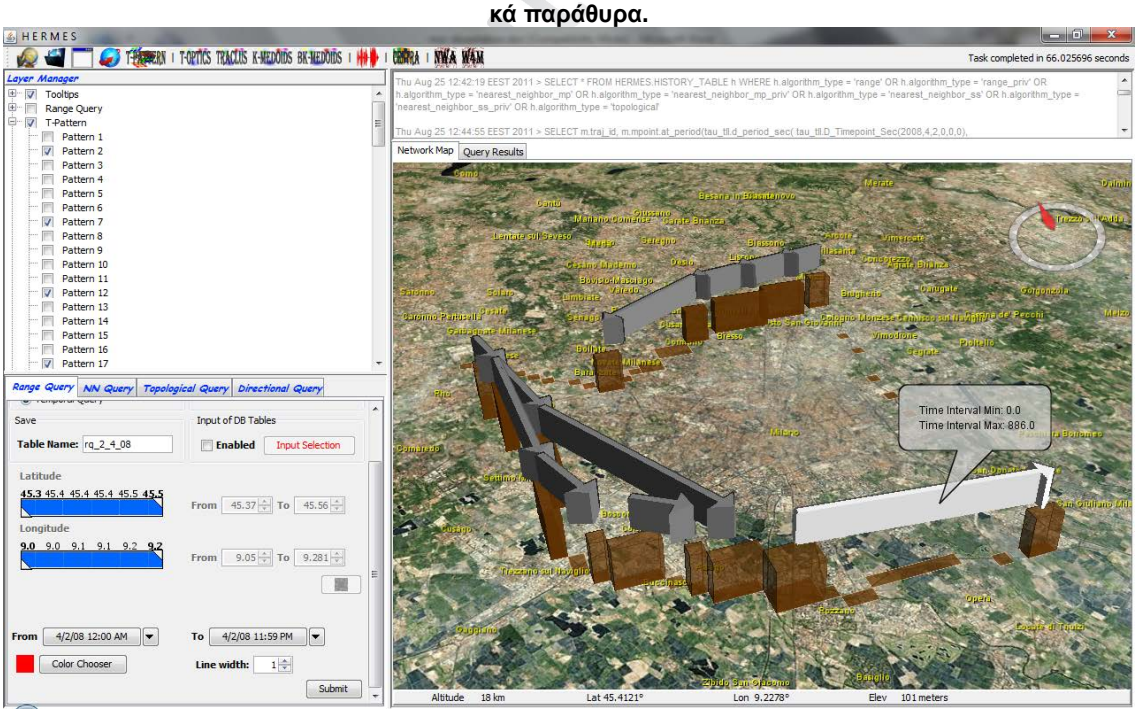
Ξεκινώντας, επιλέγουμε το εργαλείο *T-Pattern* από την μπάρα εργαλείων, διαλέγουμε το κουμπί '*Input Selection*' και από το σύνολο των πινάκων στην βάση δεδομένων κίνησης της κατηγορίας '*Simple Query*' (βλέπε Εικόνα 4.14), εισάγουμε το πίνακα 'RQ_2_4_08' ως η πηγή δεδομένων για τον αλγόριθμο. Όπως διακρίνεται στην εικόνα 4.15, ο χρήστης μπορεί να συμπληρώσει τις βασικές παραμέτρους του *T-Pattern* καθώς επίσης και τις προαιρετικές μέσα από διαδραστικά παράθυρα χωρίς να είναι απαραίτητο να γράψει κάποια γλώσσα ερωτημάτων (π.χ. *DMQL*) όπως έχει προταθεί στην βιβλιογραφία μέχρι σήμερα. Επίσης, η εφαρμογή δίνει την δυνατότητα στον τελικό χρήστη να αλλάξει τα προεπιλεγμένα χαρακτηριστικά των αποτελεσμάτων του *T-Pattern* (π.χ. χρώμα και διαφάνεια του οπτικού αντικειμένου).



Εικόνα 4.14: Επιλογή του πίνακα 'RQ_2_4_08' ως είσοδο στον αλγόριθμο T-Pattern.



Εικόνα 4.15: Ο χρήστης μπορεί να ρυθμίσει τις παραμέτρους του T-Pattern μέσα από διαδραστικά παράθυρα.

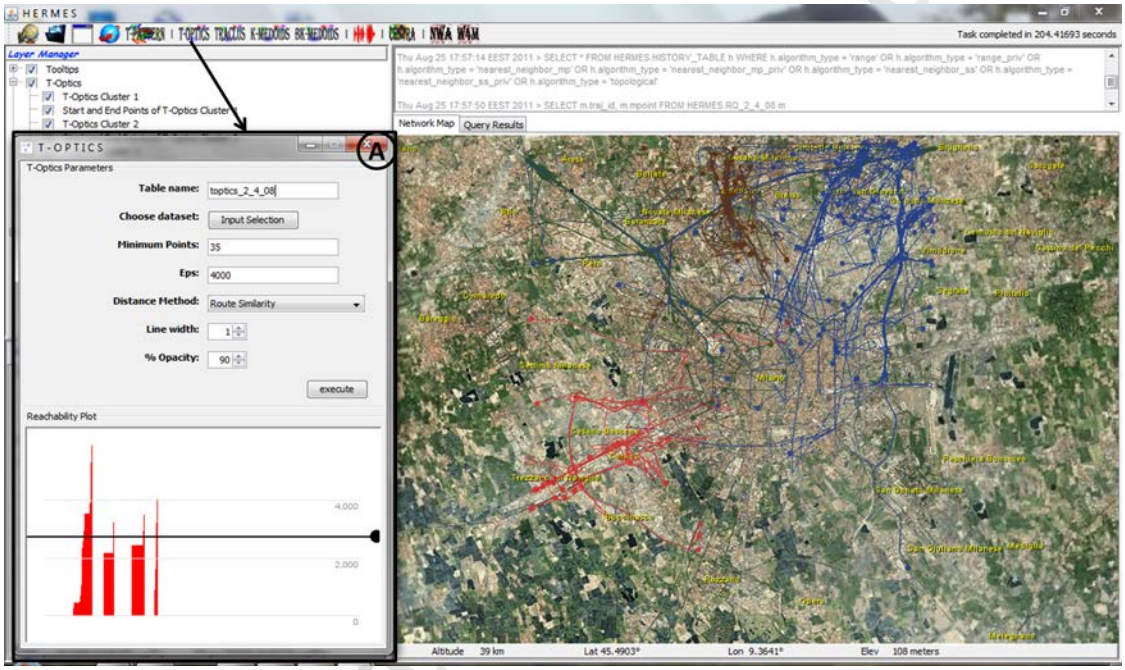


Εικόνα 4.16: Τα σημεία ενδιαφέροντος και τα μοντέλα του T-Pattern.

Ο *T-Pattern* επιστρέφει δύο αρχεία: το πρώτο αρχείο περιέχει τις περιοχές ενδιαφέροντος και απεικονίζονται στην εικόνα 4.16 ως τετραγωνάκια πάνω στον χάρτη του Μιλάνο ενώ το δεύτερο αρχείο περιλαμβάνει τις μαζικές κινήσεις των Μιλανέζων (αναπαρίστανται ως βέλη και κύβους). Ιδιαίτερο ενδιαφέρον υπάρχει στο δεύτερο αρχείο καθώς μπορούμε να παρατηρήσουμε τις τάσεις των ανθρώπων σε μια πόλη δηλαδή μετακινήσεις από μια περιοχή σε μία άλλη συμπεριλαμβανομένου και του αντίστοιχου χρονικού

διαστήματος. Παραδείγματος χάριν, η τάση του υποσύνολου της βάσης δεδομένων κίνησης του παραδείγματος είναι ότι η συχνότητα των τροχιών είναι αρκετά αυξημένη στην δυτική πλευρά του χάρτη από ότι στο υπόλοιπο μέρος της πόλης. Επιπλέον, η προέλευση των μετακινήσεων είναι πιο έντονη στην βορειοδυτική και βορειοανατολική πλευρά του Μιλάνο και λιγότερο στις υπόλοιπες περιοχές.

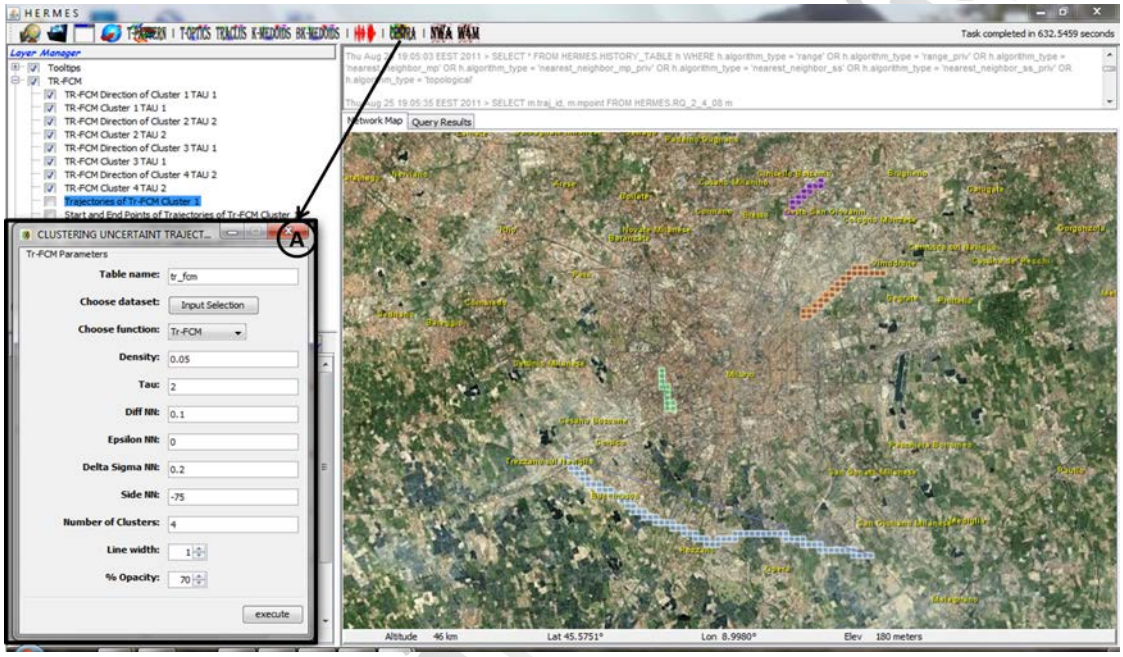
Κρατώντας το ίδιο υποσύνολο της βάσης ως είσοδο δεδομένων για τον αλγόριθμο συσταδοποίησης, επιλέγουμε τον *T-Optics* T-OPTICS από την μπάρα εργαλείων και συμπληρώνουμε τις παραμέτρους του όπως φαίνεται στην εικόνα 4.17. Συγκεκριμένα, οι βασικοί παράμετροι που πρέπει να συμπληρώσει ο χρήστης είναι ο ελάχιστος αριθμός σημείων '*Minimum Points*' και η παράμετρος *Eps* καθώς επίσης να επιλέξει και την κατάλληλη μέθοδο απόστασης. Στο συγκεκριμένο παράδειγμα, το *Minimum Points* είναι 35, το Eps: 4000 και έχουμε επιλέξει το *Route Similarity*. Τα αποτελέσματα του *T-Optics* παρουσιάζονται στην παρακάτω εικόνα και οι τροχιές απεικονίζονται στο χάρτη με 90% διαφάνεια και πάχος γραμμής 1 (ο βαθμός διαφάνειας μπορεί διαδραστικά να αλλάξει όπως και το πάχος της γραμμής).



Εικόνα 4.17: Ο αλγόριθμος T-Optics έχει ανιχνεύσει τέσσερις συστάδες. Στο παράθυρο Α, ο χρήστης μπορεί να συμπληρώσει τους παραμέτρους του T-Optics και να τον εκτελέσει καθώς και να παρατηρήσει ή να διαφοροποιήσει τις συστάδες από το διαδραστικό εργαλείο 'Reachability Plot'.

Η εικόνα 4.17 απεικονίζει τέσσερις συστάδες, 157 τροχιές με το χρώμα μπλε, 47 τροχιές με το καφέ, 41 με το σκούρο πράσινο και 40 τροχιές με το κόκκινο. Ξεκάθαρα, μπορούμε να ξεχωρίσουμε ότι όλες οι συστάδες κατευθύνονται προς στο κέντρο του Μιλάνο αλλά έχουν διαφορετικά σημεία προέλευσης. Ουσιαστικά, μπορούμε να εντοπίσουμε τις κυριότερες περιοχές όπου οι άνθρωποι συναντούν ή διέρχονται από αυτές μια καθημερινή μέρα. Ο αλγόριθμος *T-Optics* επιστρέφει ένα '*Reachability Plot*' σαν το διάγραμμα στην εικόνα 4.17 (παράθυρο Α) και αν τα αποτελέσματα δεν είναι ικανοποιητικά, ο χρήστης μπορεί να μετακινήσει την μαύρη μπάρα έτσι ώστε να πετύχει καλύτερα. Ένα από τα βασικά πλεονεκτήματα του *T-Optics* είναι ότι δεν συμπεριλαμβάνει κάθε τροχιά σε μια συστάδα αλλά όταν μια τροχιά δεν είναι τόσο κοινή με ένα αριθμό κινούμενων δεδομένων, εκλαμβάνεται ως 'θόρυβο', δηλαδή μένει εκτός από οποιαδήποτε συστάδα. Στην εφαρμογή, ο θόρυβος απεικονίζεται με γκρι χρώμα και στο παραπάνω παράδειγμα, έχει απενεργοποιηθεί από το Διαχειριστή Στρώματος. Από την άλλη μεριά, η εικόνα 4.18 δείχνει τέσσερις συστάδες, οι οποίες απεικονίζονται με την μορφή αλλεπάλληλων κελιών παρουσιάζοντας έναν εναλλακτικό τρόπο παρουσίασης των δεδομένων κίνησης. Ειδικότερα, ο *Tr-FCM* είναι ένας αλγόριθμος συσταδοποίησης για την ανάδειξη ομάδων σε ευμετάβλητα δεδομένα κίνησης και μπορούμε να το επιλέξουμε από το κουμπί CENTRA στην μπάρα εργαλείων. Όπως και με τους προηγούμενους αλγόριθμους, οι παράμετροι του *Tr-FCM* μπορούν να συμπληρωθούν μέσα από διαδραστικές διεπαφές διευκολύνοντας με αυτόν τον τρόπο τον χρήστη για την εκτέλεση του (βλέπε εικόνα 4.18 Α). Στο παραπάνω παράδειγμα, τα τελικά αποτελέσματα αναδείχθηκαν, αρχικά από τις τροχιές που συλλέχτηκαν την μέρα 02/04/2008 (χρονικό ερώτημα) και στην συνέχεια, χρησιμοποιήθηκαν ως είσοδο δεδομένων στον αλγό-

ριθμο *Tr-FCM*. Ο αλγόριθμος *Tr-FCM* παρουσιάζει την κάθε συστάδα με συνεχόμενα κελιά και με μια γραμμή δείχνοντας την κατεύθυνση της. Σημείωση ότι μπορούμε να χρησιμοποιήσουμε τους αλγόριθμους *CenTr-I-FCM, TX-CenTra* και *CenTra* εκτός από τον *Tr-FCM* επιλέγοντάς τους από το '*Choose function*' του παράθυρου (βλέπε Εικόνα 4.18). Όπως παρατηρούμε τις παραπάνω εικόνες, οι αλγόριθμοι συσταδοποίησης της προτεινόμενης εφαρμογής επιδεικνύουν τις συστάδες με διαφορετικό χρώμα, το οποίο το επιλέγουν τυχαία από μια λίστα χρωμάτων. Επίσης, ο διαχειριστής στρώματος '*Layer manager*' με την χρήση του εργαλείο checkboxtree, το οποίο είναι μια επέκταση του μοντέλου MVC του *SWING Java Toolkit* δίνει την δυνατότητα στον χρήστη να περιοριστεί στα μοντέλα των αλγόριθμων που των ενδιαφέρουν αποκλείοντας με αυτόν τον τρόπο, τα μη ικανοποιητικά πρότυπα χωρίς να είναι απαραίτητο η εκτέλεση εκ νέου ερωτήματος (βλέπε Εικόνα 4.16).



Εικόνα 4.18: Τα αποτελέσματα του Tr-FCM στον γεωγραφικό χάρτη. Οι παράμετροι του Tr-FCM συμπληρώνονται στο παράθυρο A όπου ο χρήστης μπορεί να αλλάξει τον αλγόριθμο Tr-FCM σε CenTR-I-FCM, TX-CenTra ή CenTra από την επιλογή Choose function.

Τελειώνοντας, η δομή του ερωτήματος κάθε αλγορίθμου εξόρυξης γνώσης έχει την παρακάτω μορφή:

```
SELECT t.object FROM MINE(HERMES.TOPTICS_2_4_08 ; CLUSTER ; 35 ;
4000 ; ROUTE_SIMILARITY) t;
```

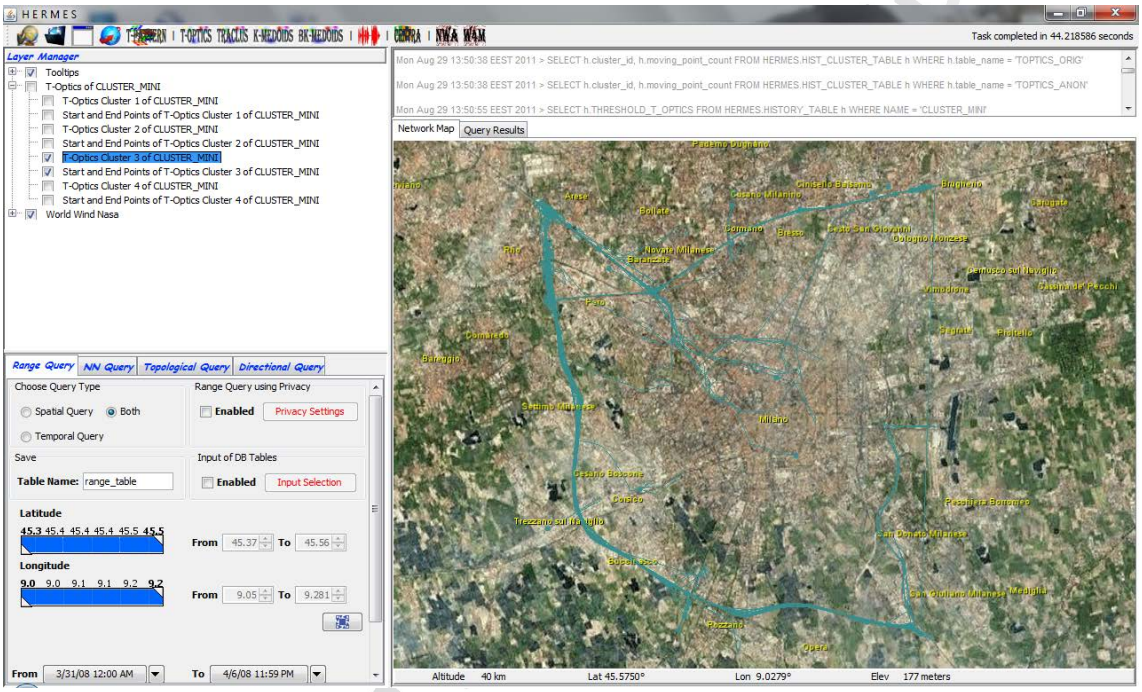
Μέσα στην παρένθεση του *MINE*, βρίσκεται πρώτα το όνομα του πίνακα που τα αποτελέσματα του αλγορίθμου θα αποθηκευτούν, στην συνέχεια, ακολουθείται το όνομα του αλγορίθμου και τέλος, οι παράμετροι του. Στο παραπάνω ερώτημα, εκτελείται ο αλγόριθμος *T-Optics* από το παράδειγμα της εικόνας 4.17.

- Ερώτημα εξόρυξης ως είσοδο και εκτέλεση ερωτήματος εξόρυξης

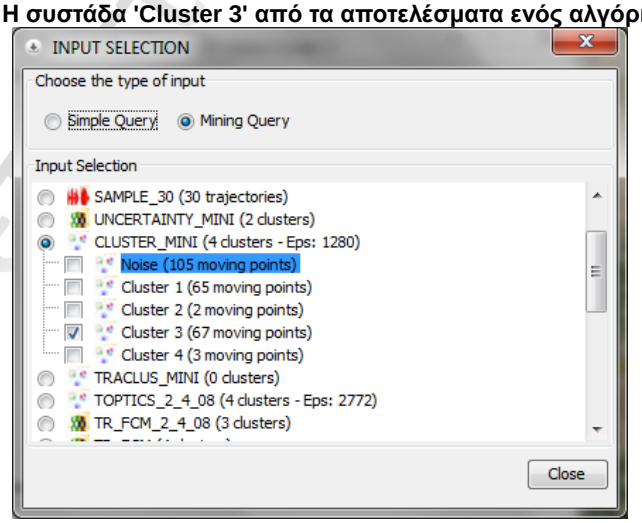
Στην τρίτη κατηγορία, δημιουργείται ο βρόγχος πάνω στο μηχανισμό εξόρυξης, δηλαδή ένας χρήστης εκτελεί ένα αλγόριθμο εξόρυξης από το μηχανισμό εξόρυξης και τα αποτελέσματα του χρησιμοποιούνται για την εκτέλεση ενός επόμενου αλγόριθμου ή του ίδιου από τον μηχανισμό αυτόν. Η διαδικασία αυτήν είναι πολύ αποδοτική και χρήσιμη σε έναν αναλυτή καθώς με αυτόν τον τρόπο μπορεί να φιλτράρει, να εμβαθύνει και να αναδείξει γνώση από τα αποτελέσματα ενός αλγόριθμου εξόρυξης. Παραδείγματος χάριν, όπως θα δούμε και στο πρώτο σενάριο, ένας χρήστης μπορεί να εκτελέσει ένα αλγόριθμο συσταδοποίησης και χρησιμοποιώντας ένα υποσύνολο των συστάδων να εκτελέσει ξανά τον αλγόριθμο αλλάζοντας τους παραμέτρους ή την μέθοδο απόστασης. Έτσι, η εικόνα των τελικών αποτελεσμάτων θα είναι πιο ξεκάθαρη στον χρήστη για να εξάγει τα συμπεράσματά του. Στο δεύτερο σενάριο, παίρνουμε μια συστάδα από τα αποτελέσματα του *T-Optics* και εκτελούμε το αλγόριθμο *T-Pattern* έτσι ώστε να παρατηρήσουμε τις τάσεις των κινήσεων των οχημάτων στο Μιλάνο. Στο τρίτο σενάριο, εκτελούμε τον αλγό-

ριθμο *Tr-FCM* και τα αποτελέσματά του αξιοποιούνται στην εκτέλεση του αλγόριθμου *TX-CenTra*. Το τελικό αποτέλεσμα είναι σαφώς πιο βελτιωμένο και παρουσιάζει περισσότερο ενδιαφέρον στον χρήστη. Στο τελευταίο σενάριο, χρησιμοποιούμε το *T-Sampling* για να πάρουμε τα πιο αντιπροσωπευτικά δείγματα από την βάση και στην συνέχεια εκτελούμε τον αλγόριθμο συσταδοποίησης *K-Medoids*. Αναφορικά με τα προηγούμενα σενάρια, κάναμε πρώτα προ-επεξεργασία των δεδομένων από την βάση με την δειγματοληψία και στην συνέχεια, ομαδοποιήσαμε τα δεδομένα.

Ξεκινώντας από το πρώτο παράδειγμα, επιλέξαμε την 3η συστάδα από τον πίνακα 'CLUSTER_MINI' (βλέπε Εικόνες 4.19, 4.20), τα αποτελέσματα του οποίου παράχθηκαν από τον αλγόριθμο *T-Optics* με μέθοδο απόστασης το '*Common Source*'. Η συστάδα αποτελείται από 67 τροχιές, οι οποίες αναχωρούν από το βορειοδυτικό τμήμα του χάρτη και καταλήγουν σε τρία διαφορετικά σημεία: βορειοανατολικά, στο κέντρο της πόλης και νότια.



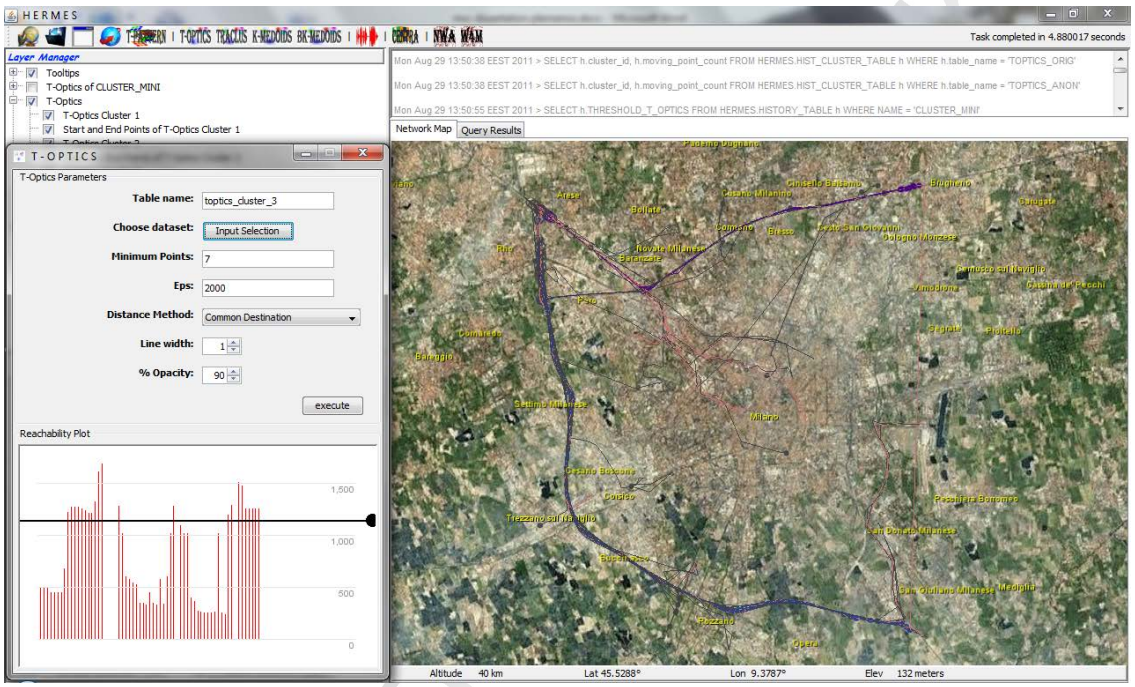
Εικόνα 4.19: Η συστάδα 'Cluster 3' από τα αποτελέσματα ενός αλγόριθμου T-Optics.



Εικόνα 4.20: Επιλογή της 3ης συστάδας από τις τέσσερις συνολικά του πίνακα 'CLUSTER_MINI'.

Στην συνέχεια, εκτελούμε ξανά τον αλγόριθμο *T-Optics* αλλάζοντας την μέθοδος απόστασης σε '*Common Destination*' και τις παραμέτρους του. Αξίζει να σημειωθεί ότι χρησιμοποιούμε χαμηλότερες τιμές στις παραμέτρους *minimum points* και *eps* καθώς προχωράμε σε περισσότερο βάθος στην ανάλυση των δεδομένων με τέτοιο τρόπο ώστε να αποκλείσουμε τις μη ενδιαφέροντες τροχιές στο χάρτη και να βελτιώσουμε το τελικό αποτέλεσματα. Σε τέτοιες περιπτώσεις, αναδεικνύεται επιπλέον θόρυβος στα δε-

δομένα. Μετά την εκτέλεση του αλγόριθμου, η συστάδα χωρίστηκε σε τρεις μικρότερες συστάδες, η πρώτη συστάδα με το χρώμα μοβ, η δεύτερη με το ροζ και η τρίτη με το μπλε (βλέπε Εικόνα 4.21). Η πρώτη συστάδα καταλήγει στην βορειοανατολική πλευρά του χάρτη, η δεύτερη στο κέντρο του Μιλάνο ενώ η τρίτη διέρχεται μέσα από την δυτική πλευρά και καταφθάνει στην νότια μεριά της πόλης. Οι υπόλοιπες τροχιές είναι θόρυβος και απεικονίζονται με σκούρο γκρι χρώμα. Γενικά, η παραπάνω διαδικασία αποκαλείται στην βιβλιογραφία ως προοδευτική συσταδοποίηση των δεδομένων και ο σκοπός της συγκεκριμένης μεθοδολογίας είναι η έκβαση περισσότερο ερμηνευτικών αποτελεσμάτων. Ασφαλώς, η υποστήριξη διαδραστικών εργαλείων μέσα από την προοδευτική συσταδοποίηση παίζει ένα σημαντικό ρόλο για την διευκόλυνση του χρήστη στην ανάλυση των δεδομένων. Έτσι, ο σκοπός της εφαρμογής είναι να αξιοποιήσει την προοδευτική συσταδοποίηση και γενικότερα, την προοδευτική ανάλυση των δεδομένων μέσα από αλληλεπιδραστικά εργαλεία όπως το '*Reachability Plot*' της παρακάτω εικόνας.

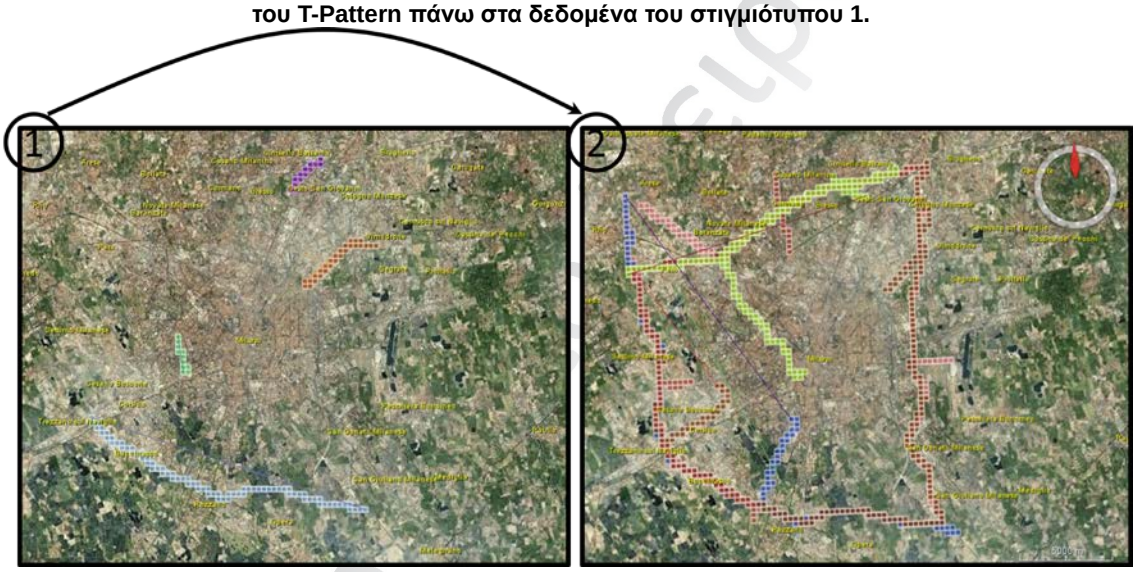


Εικόνα 4.4.21: Το 'Cluster 3' διασπάστηκε σε τρεις υπό συστάδες. Ο θόρυβος απεικονίζεται με σκούρο γκρι χρώμα.

Παίρνοντας τα αποτελέσματα της εικόνας 4.21, συνεχίσαμε την ανάλυση των δεδομένων σε τρίτο στάδιο και όπως εμφανίζεται στην παρακάτω εικόνα, εφαρμόσαμε τον αλγόριθμο *T-Pattern* για να αναλύσουμε τις τάσεις των οχημάτων στην δυτική πλευρά του Μιλάνο. Είναι σημαντικό να επισημάνω ότι η εφαρμογή δίνει την δυνατότητα στον χρήστη στην επιλογή ενός υποσύνολου από τα αποτελέσματα ενός ερωτήματος αντί ολόκληρου του συνόλου προσφέροντας με αυτόν τον τρόπο περισσότερο ευελιξία και 'ελευθερία' στον χρήστη. Παραδείγματος χάριν, όπως φαίνεται από το στιγμιότυπο 1 της εικόνας 4.22, έχουμε επιλέξει την 3η συστάδα του αποτελέσματος της εικόνας 4.21 έχοντας αποκλείσει τα υπόλοιπα δύο υποσύνολα και τον θόρυβο. Μετέπειτα, εκτελέσαμε τον αλγόριθμο *T-Pattern* όπου τα αποτελέσματα του παρουσιάζονται στο στιγμιότυπο 2 της εικόνας 4.22. Συνοψίζοντας από την προηγούμενη προοδευτική διαδικασία, πρώτα πραγματοποιήσαμε ένα αλγόριθμο συσταδοποίησης από τα αποτελέσματα ενός ίδιου αλγόριθμου με αποτέλεσμα να διασπάσουμε την συστάδα σε τρείς υπό συστάδες και αφού επικεντρωθήκαμε σε ένα υποσύνολο (την μια υπό συστάδα) από το συνολικό αποτέλεσμα της συσταδοποίησης, εφαρμόσαμε τον αλγόριθμο *T-Pattern* έτσι ώστε να προβάλουμε τις μαζικές μετακινήσεις των οχημάτων στο συγκεκριμένο υποσύνολο.

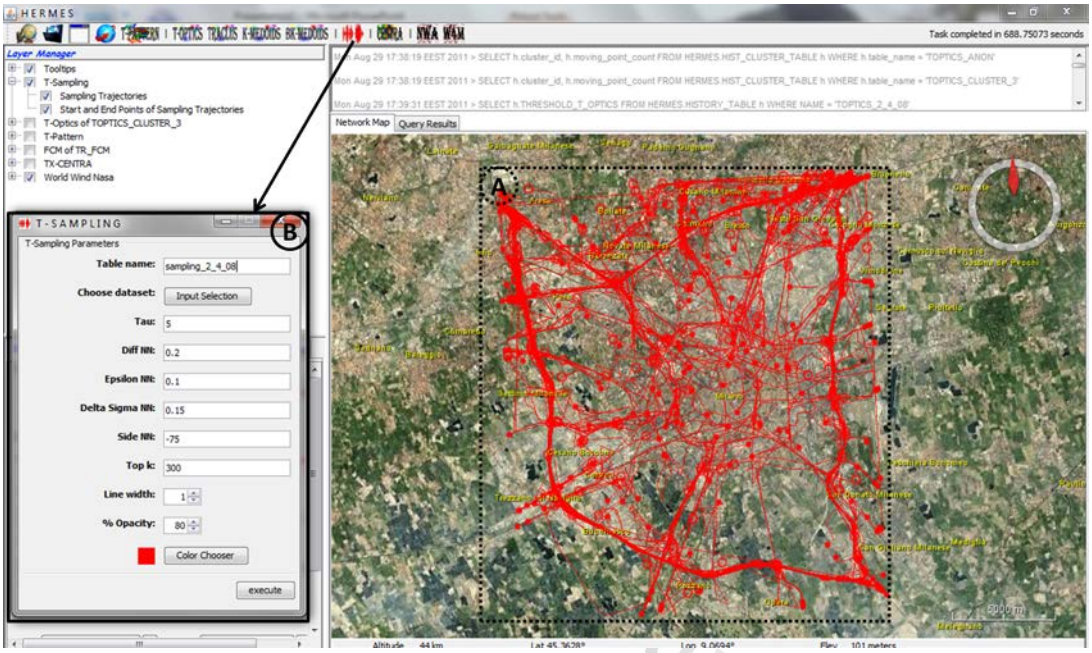
Με τον ίδιο τρόπο, μπορούμε να χρησιμοποιήσουμε τα αποτελέσματα του *Tr-FCM* για να εκτελέσουμε το *TX-CenTra*. Ειδικότερα, τα αποτελέσματα του *TX-CenTra* (βλέπε στιγμιότυπο 2 της εικόνας 4.23) είναι σαφώς πιο βελτιωμένα και ξεκάθαρα από αυτά του *Tr-FCM*. Με λίγα λόγια, ο *TX-CenTra* ή *CenTra* έχει νόημα να εφαρμόζεται πάνω στα παραγόμενα αντικείμενα του *Tr-FCM* ή *CenTr-I-FCM* [23] και αυτό πραγματοποιείται μόνο με την διαδικασία της προοδευτικής ανάλυσης των δεδομένων. Όμως, αυτό δεν σημαίνει ότι δεν είναι χρήσιμο ή ενδιαφέρον να χρησιμοποιούμε τους αλγορίθμους *Tr-FCM* και *CenTr-I-FCM* ή τους *TX-CenTra* ή *CenTra* σε άλλα σύνολα δεδομένων.

Εικόνα 4.22: 1) η μια υπό συστάδα του 'Cluster 3' από το προηγούμενο ερώτημα, 2) τα πρότυπα του T-Pattern πάνω στα δεδομένα του στιγμιότυπου 1.

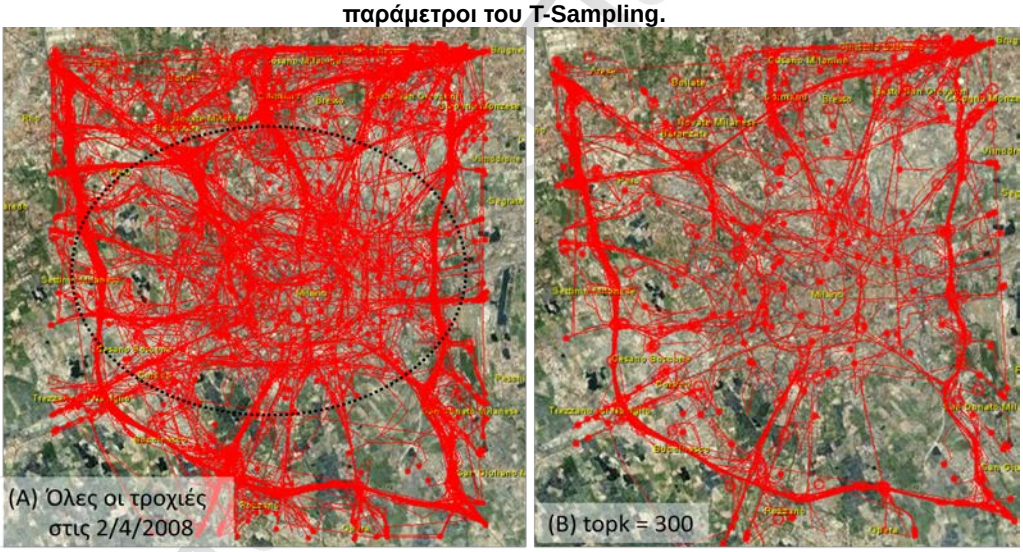


Εικόνα 4.23: 1) τα αποτελέσματα του Tr-FCM της εικόνας 3.18, 2) τα αποτελέσματα του TX-CenTra πάνω στα αποτελέσματα του Tr-FCM.

Στο τέταρτο παράδειγμα, επιλέξαμε το κουμπί για να τρέξουμε τον αλγόριθμο *T-Sampling* σε όλες τις τροχιές που έγιναν στις 2/4/08. Όπως διακρίνουμε μέσα στο διακεκομμένο κύκλο της εικόνας 4.25 Α, το σύνολο των δεδομένων είναι αρκετά μεγάλο και δημιουργεί ένα πυκνό στρώμα τροχιών σε πολλά σημεία του Μιλάνο. Από την μία μεριά, είναι αρκετά δύσκολο σε ένα αναλυτή να ερμηνεύσει τέτοιου είδους αποτελέσματα, από την άλλη μεριά, ένας αλγόριθμος εξόρυξης γνώσης θα χρειαστεί αρκετό χρόνο για να παράγει τα μοντέλα από ένα τόσο μεγάλο σύνολο δεδομένων. Έτσι, δημιουργείται το εξής ερώτημα, πως θα αποσπάσουμε ένα κατάλληλο υποσύνολο της βάσης δεδομένων κίνησης, το οποίο θα συλλαμβάνει τα ίδια πρότυπα σε μετέπειτα επεξεργασίες των αλγορίθμων εξόρυξης. Η απάντηση είναι ότι μπορούμε να επιτύχουμε τα ίδια αποτελέσματα επιταχύνοντας την ανάλυση και εξόρυξη των εργασιών στο γνωστικό πεδίο με τους κατάλληλους αλγόριθμους δειγματοληψίας. Συγκεκριμένα, ο *T-Sampling* επιστρέφει τα K πιο αντιπροσωπευτικά δείγματα από ένα δοσμένο σύνολο τροχιών. Όπου K είναι ο συνολικός αριθμός των δειγμάτων και δίνεται από τον χρήστη. Η εκτέλεση του *T-Sampling* και οι παράμετροί του ρυθμίζονται από το πλαίσιο που φαίνεται στο στιγμιότυπο Β της εικόνας 4.24. Επίσης, τα αποτελέσματα της δειγματοληψίας απεικονίζονται στο γεωγραφικό χάρτη όπου παρουσιάζονται τα 300 πιο αντιπροσωπευτικά δείγματα που επιλέχτηκαν από την βάση. Όπως ξεκάθαρα αντιλαμβανόμαστε από την εικόνας 4.25, το σχήμα της βάσης έχει διατηρηθεί μετά την εφαρμογή του αλγόριθμου δειγματοληψίας μειώνοντας σημαντικά το συνολικό αριθμό δεδομένων.

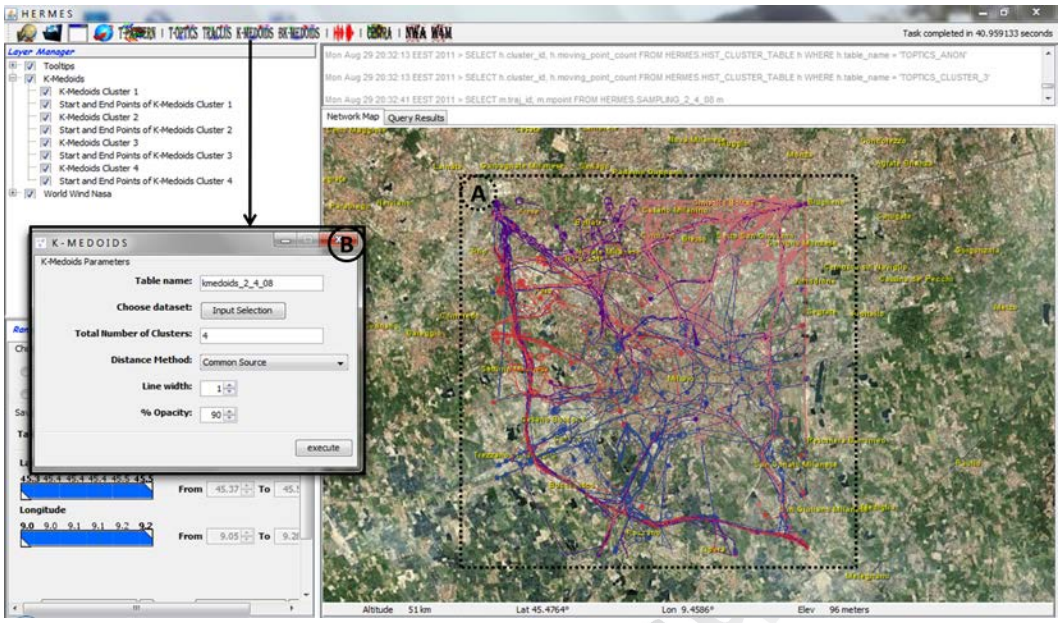


Εικόνα 4.24: Α) Δειγματοληψία από τις 300 πιο αντιπροσωπευτικές τροχιές του 'RQ_2_4_08'. Β) Οι παράμετροι του T-Sampling.



Εικόνα 4.25: Οπτικοποίηση των οχημάτων του Μιλάνο πριν την εκτέλεση του T-Sampling (A) και μετά (B) για την μέρα 2/4/2008.

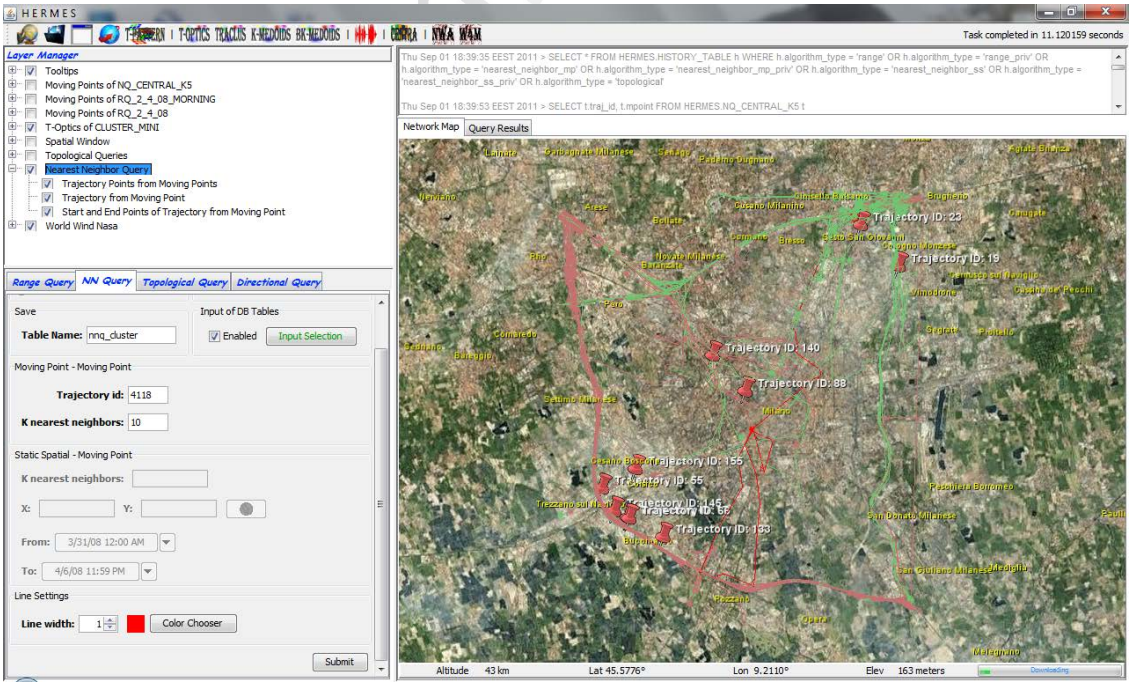
Στην συνέχεια, επιλέγουμε το K-MEDOIDS, έναν διαφορετικό αλγόριθμο ομαδοποίησης, ο οποίος διαλέγει K τροχιές ως '*medoids*' και συσχετίζει κάθε τροχιά στο κοντινότερο *medoid*. Σε κάθε συσχέτιση μιας τροχιάς με το *medoid*, υπολογίζεται ξανά το βέλτιστο *medoid* και η διαδικασία συνεχίζεται έως ότου εξεταστούν όλες οι τροχιές. Οι παράμετροι του K-Medoids είναι δύο, ο ορισμός των ομάδων που θα δημιουργήσει και η μέθοδος απόστασης (βλέπε Εικόνα 4.26 Β). Σε αντίθετη περίπτωση με τον *T-Optics*, ο χρήστης πρέπει να ορίσει τον αριθμό των συστάδων που θα υπολογίσει ο *K-Medoids*. Στην εικόνα 4.26 Α, παρουσιάζονται οι τέσσερις τροχιές του αλγόριθμου ομαδοποίησης πάνω στα δεδομένα της δειγματοληψίας. Αξίζει να σημειωθεί ότι ο χρόνος υπολογισμού για την εξαγωγή των συστάδων ήταν αρκετά μειωμένος από ότι θα ήταν αν εκτελούσαμε τον ίδιο αλγόριθμο σε ολόκληρη την βάση. Όσο αφορά τα αποτελέσματα του K-Medoids, ο αλγόριθμος έχει την τάση να σχηματίζει κυκλικά σχήματα λόγω του μηχανισμού του. Παραδείγματος χάριν, ο K-Medoids σχημάτισε τέσσερις συστάδες, η πρώτη τοποθετείται βορειοανατολικά, η δεύτερη βορειοδυτικά, η τρίτη στο κέντρο και η τελευταία δυτικά.



Εικόνα 4.26: Α) Οι τέσσερις ομάδες τροχιών του K-Medoids από τα αποτελέσματα του T-Sampling. Β) Οι παράμετροι του K-Medoids (ο αριθμός των συστάδων ορίζεται από τον χρήστη).

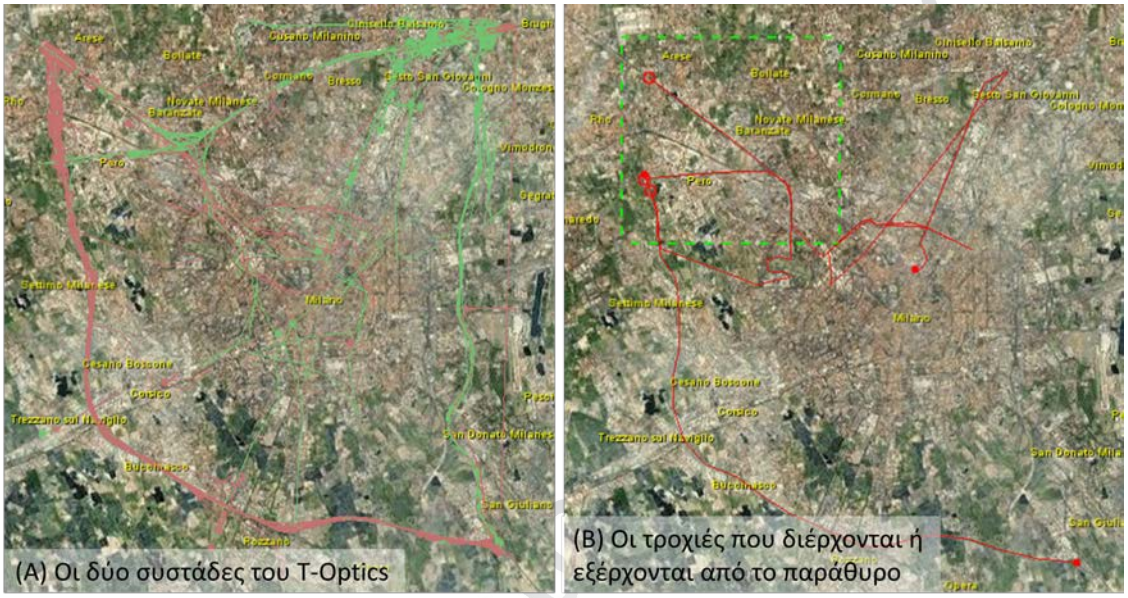
- Ερώτημα εξόρυξης ως είσοδο και εκτέλεση ερωτήματος αναζήτησης

Στην τελευταία κατηγορία, ένας χρήστης έχει την δυνατότητα να πραγματοποιήσει την εκτέλεση ενός ερωτήματος αναζήτησης πάνω στα αποτελέσματα των αλγόριθμων εξόρυξης γνώσης. Αναλυτικότερα, θα περιγραφούν δύο σενάρια χρήσης εξετάζοντας τα αποτελέσματα ενός αλγόριθμου ομαδοποίησης και στα δύο παραδείγματα. Κατ' αρχήν, επιλέξαμε την ετικέτα '*NN Query*' για να τρέξουμε ένα ερώτημα *Nearest Neighbor*. Το *Nearest Neighbor* χωρίζεται σε δύο κατηγορίες: σχετικά με την πρώτη κατηγορία, επιστρέφει τις K τροχιές που βρίσκονται κοντινότερα από μια δοσμένη τροχιά και όσο αφορά την δεύτερη, ανακαλύπτει τις K τροχιές που τοποθετούνται κοντινότερα σε ένα δοσμένο χώρο χρονικό σημείο.



Εικόνα 4.27: Αναπαράσταση του αντικειμένου αναφοράς (κόκκινη τροχιά) και του αντικείμενου δεδομένων (κόκκινες σημάνσεις) του Nearest neighbor query πάνω στις δύο συστάδες.

Έστω λοιπόν ότι θέλουμε να βρούμε τις 10 πρώτες τροχιές από τις συστάδες του *T-Optics* που βρίσκονται πιο κοντά στην τροχιά 4118, η οποία διέρχεται μέσα στο κέντρο του Μιλάνο. Αφού επιλέξουμε την πρώτη κατηγορία από την εφαρμογή, διαλέγουμε τις δυο συστάδες από τις τέσσερις συνολικά του αλγόριθμου *T-Optics* ενεργοποιώντας το κουμπί '*Input Selection*' (όπως ακολουθήσαμε παραπάνω). Στην συνέχεια, συμπληρώνουμε τις παραμέτρους του *Nearest Neighbor Query* που είναι το id της τροχιάς (αντικείμενο αναφοράς) με id = 4118 και το συνολικό αριθμό των κοντινότερων γειτόνων με k = 10 και εκτελούμε το ερώτημα. Τα αποτελέσματα του ερωτήματος απεικονίζονται στην εικόνα 4.27 όπου το αντικείμενο αναφοράς του *Nearest Neighbor Query* παριστάνεται στον χάρτη ως κόκκινη τροχιά και οι 10 κοντινότεροι γείτονες με κόκκινες σημάνσεις.



Εικόνα 4.28: Προοδευτική ανάλυση των τροχιών του αλγόριθμου ομαδοποίησης από ένα απλό ερώτημα αναζήτησης.

Διατηρώντας την ίδια είσοδο δεδομένων, πραγματοποιούμε ένα τοπολογικό ερώτημα για να αναδείξουμε τις τροχιές που εισήλθαν ή εξήλθαν στη βορειοδυτική περιοχή του Μιλάνο. Η εικόνα 4.28 απεικονίζει την εξέλιξη και ανάλυση των δεδομένων ενός απλού ερωτήματος (*Topological Query*) πάνω στα αποτελέσματα ενός αλγόριθμου συσταδοποίησης (*T-Optics*). Εν συντομία, ο μηχανισμός της πλατφόρμας δίνει την δυνατότητα στον χρήστη αφού αναλύσει τα δεδομένα κίνησης με διάφορους αλγόριθμους εξόρυξης γνώσης, να ξανά γυρίσει προς τα πίσω, φανερώνοντας τα δεδομένα εκείνα που συμπεριλήφθηκαν στον υπολογισμό των εξορυγμένων μοντέλων. Έτσι, ένας χρήστης μπορεί να επινοήσει τα δεδομένα κίνησης που συμμετείχαν στην δημιουργία των πιο σημαντικών μοντέλων (π.χ. συστάδες) της ανάλυσης.

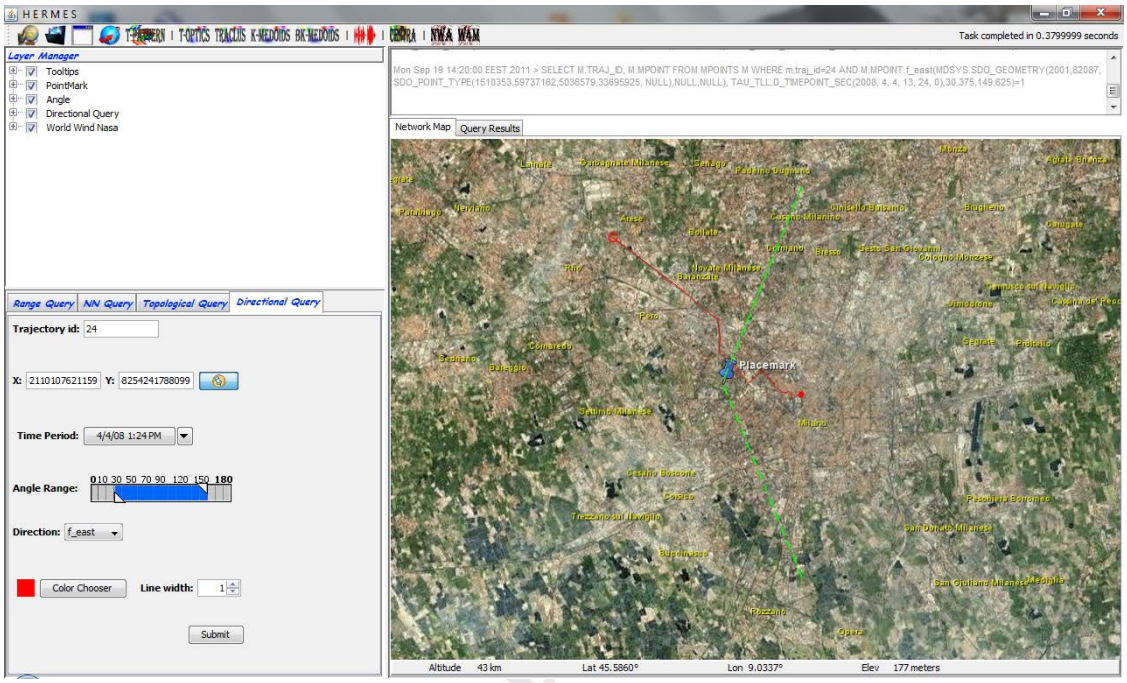
4.5. ΔΙΑΔΡΑΣΤΙΚΑ ΕΡΓΑΛΕΙΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Σε αυτό το υποκεφάλαιο, θα περιγραφούν τα επιπλέον διαδραστικά εργαλεία του συστήματος. Συγκεκριμένα, θα αναλυθεί ο μηχανισμός *Directional Query* του *HERMES*, ο αλγόριθμος συσταδοποίησης *BK-MEDOIDS* και τα εργαλεία *DB Connector*, *Open*, *SQL Plus* και *T-aggregator* από την μπάρα εργαλείων.

Directional Query

Το 'Directional Query' δίνει την δυνατότητα στον χρήστη να βρει τις τροχιές, οι οποίες οι θέσεις τους τοποθετούνται στην ανατολή, δύση, βορρά, νότο, μπροστά, πίσω, αριστερά ή δεξιά από ένα σημείο σε μια δεδομένη στιγμή του χρόνου (χρονικό σημείο). Παραδείγματος χάριν, έστω ότι ο χρήστης θέλει να βρει το κινούμενο αντικείμενο με id = 24 το οποίο να βρίσκεται στην ανατολή με βάση το χρονικό σημείο που του έχει δοθεί. Ξεκινώντας, θα συμπληρώσει το πεδίο 'trajectory id', θα επιλέξει ένα σημείο πάνω στον χάρτη με το εργαλείο  και θα σημειώσει το χρονικό σημείο από το πάνελ ώρας.

Αναλυτικότερα, το χωρικό σημείο που επιλέχτηκε, απεικονίζεται ως μπλε πινέζα στην εικόνα 4.29 και η αντίστοιχη χρονική στιγμή που επιλέχτηκε, είναι η μέρα 4/4/2008 1:24μμ. Επιπρόσθετα, ρυθμίστηκε το εύρος της γωνίας με το bislider (από 30 μέχρι 150 μοίρες) και επιλέχτηκε το *f_east* από το πεδίο 'Direction'. Αξίζει να σημειωθεί ότι ο χρήστης μπορεί εκτός από το *f_east*, να διαλέξει το *f_west*, *f_south*, *f_north*, *f_front*, *f_behind*, *f_right* ή *f_left* ανάλογα με την κατεύθυνση της τροχιάς από το δοσμένο σημείο. Τέλος, αφού πιέστηκε το κουμπί '*Submit*', πραγματοποιήθηκε το ερώτημα με τα αποτελέσματα να παρουσιάζονται οπτικά στο χάρτη (βλέπε παρακάτω εικόνα).



Εικόνα 4.29: Ο μηχανισμός Directional Query σε χρήση.

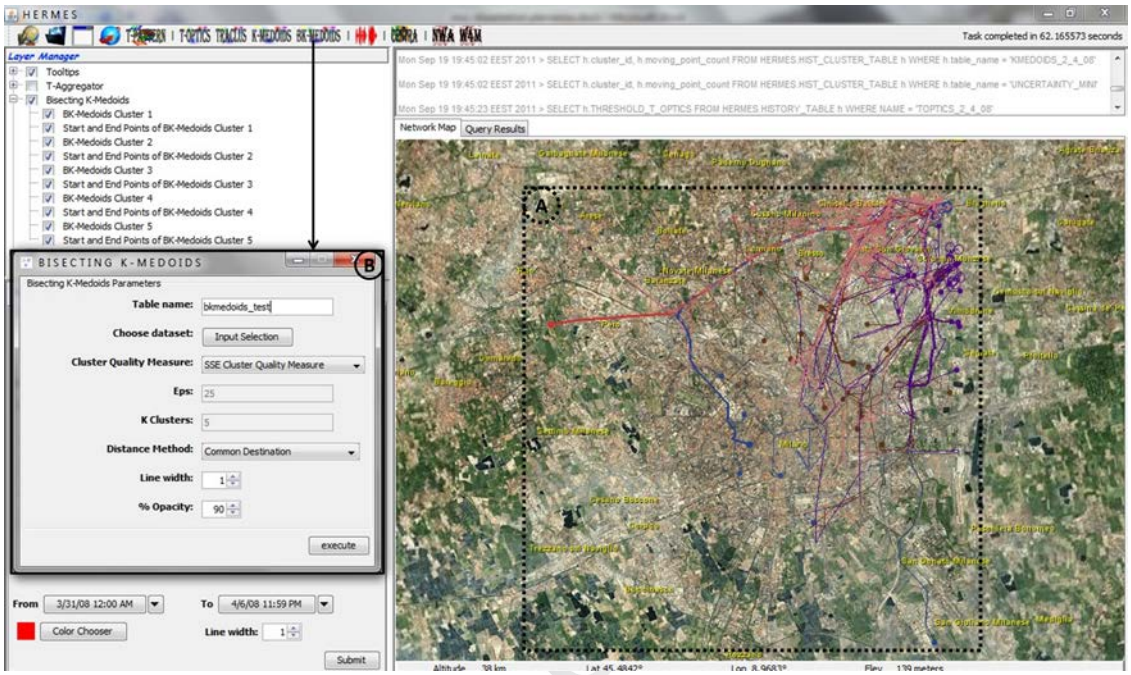
Το αντίστοιχο SQL ερώτημα γι' αυτό το παράδειγμα είναι το εξής:

```
SELECT M.TRAJ_ID, M.MPOINT FROM MPOINTS M WHERE m.traj_id=24 AND
M.MPOINT.f_east(MDSYS.SDO_GEOMETRY(2001,82087,
SDO_POINT_TYPE(5807981.23469962,1233116.8572235, NULL),NULL,NULL),
TAU TLL.D TIMEPOINT SEC(2008, 4, 4, 13, 24, 0),30.0,150.0)=1;
```

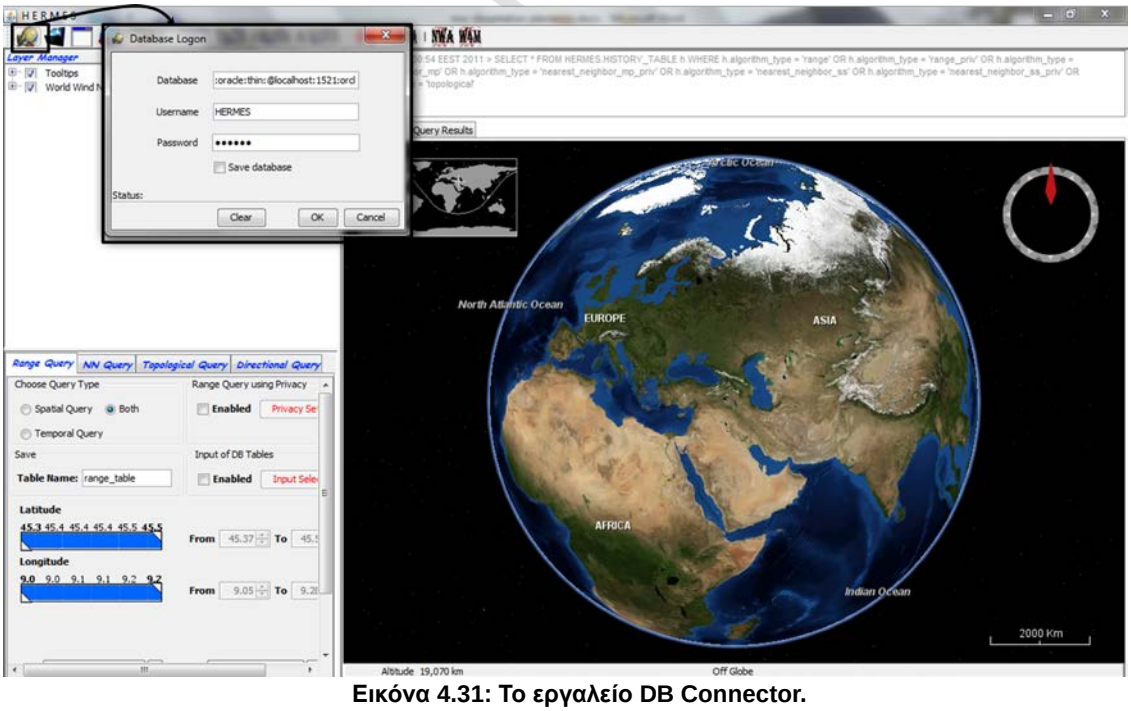
BK-Medoids

Έναν αλγόριθμο συσταδοποίησης, ο οποίος δεν παρουσιάστηκε στα προηγούμενα κεφάλαια είναι ο *BK-Medoids*. Ο *BK-Medoids* είναι βασισμένος στον αλγόριθμο *K-Medoids* δηλαδή θέτει μια τροχιά από κάθε συστάδα που θα συμπεριφέρεται ως medoid και θα είναι το κρίσιμο σημείο για τον υπολογισμό των συστάδων. Επιπλέον, παρέχει στον χρήστη την δυνατότητα να επιλέξει από μια ειδική μετρητική τιμή '*Cluster Quality Measure*', τρεις διαφορετικές μεθόδους. Η πρώτη μέθοδος '*SSE Cluster Quality Measure*', υπολογίζει τις συστάδες χωρίς να συμπληρώσει καμία παράμετρο ο χρήστης. Η δεύτερη μέθοδος '*Diameter Cluster Quality Measure*', ο χρήστης συμπληρώνει μόνο την τιμή *Eps* ενώ στην τρίτη μέθοδο '*Found K-Cluster Quality Measure*', ο χρήστης επιλέγει μόνο των αριθμό των συστάδων που θα υπολογίσει. Ξεκινώντας, επιλέχτηκε το BK-MEDOIDS από την μπάρα εργαλείων. Οι παράμετροι του *BK-Medoids* εξαρτώνται από την επιλογή του *Cluster Quality Measure* (όπως αναφέρθηκε προηγουμένως) αλλά σε κάθε περίπτωση, ο χρήστης επιλέγει την μέθοδο απόστασης που θα συμπεριληφθεί στον αλγόριθμο και στο συγκεκριμένο παράδειγμα, ορίστηκε το '*Common Destination*' (βλέπε εικόνα 4.30 Β). Η μέθοδος '*Cluster Quality Measure*' που επιλέχτηκε στο παράδειγμα ήταν ο *SSE* οπότε δεν χρειάστηκε να συμπληρωθεί κάποια επιπλέον τιμή. Αφού εκτελέστηκε ο αλγόριθμος συσταδοποίησης, ανίχνευσε πέντε συστάδες όπως φαίνεται στην εικόνα 4.30 Α. Αξίζει να σημειωθεί ότι ο *BK-Medoids* δεν έχει την ικανότητα να εξιχνιάσει τον θόρυβο από το σύνολο των δεδομένων όπως ο *T-Optics* με αποτέλεσμα να

συμπεριλαμβάνει υποχρεωτικά κάθε τροχιά σε μια συστάδα. Αυτό έχει ως αποτέλεσμα να δυσκολεύει τον αναλυτή να εξορύξει σημαντική πληροφορία από τα δεδομένα διότι ο θόρυβος μπορεί να επηρεάσει σημαντικά την ανάλυση τους. Βέβαια, αυτό δεν ισχύει μόνο στον συγκεκριμένο αλγόριθμο συσταδοποίησης αλλά σε οποιοδήποτε δεν έχει την δυνατότητα να ξεχωρίσει τον θόρυβο από τα αποτελέσματα.



Εικόνα 4.30: Α) Οπτική παρουσίαση των πέντε συστάδων του BK-Medoids. Β) Οι παράμετροι του BK-Medoids.



Εικόνα 4.31: Το εργαλείο DB Connector.

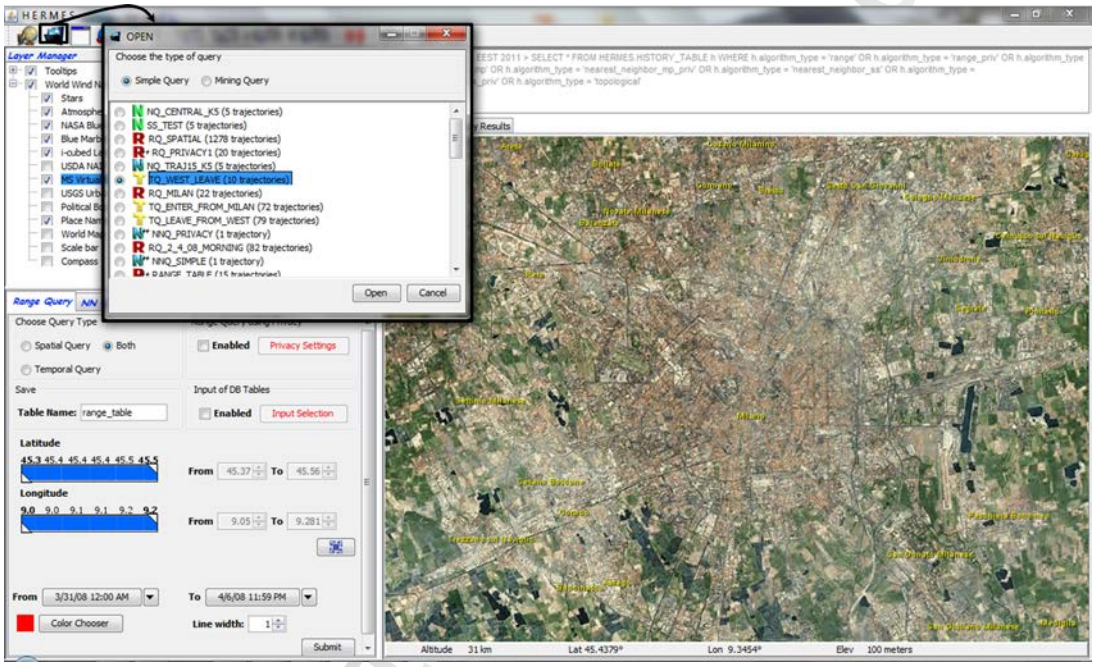
DB Connector

Όταν ανοίγουμε την εφαρμογή, αυτή συνδέεται απευθείας στην βάση δεδομένων κίνησης του HERMES παίρνοντας τα στοιχεία σύνδεσης από το αρχείο database.properties. Σε περιπτώσεις που ο χρήστης επιθυμεί να συνδεθεί σε άλλη βάση δεδομένων κίνησης ή οι κωδικοί πρόσβασης της

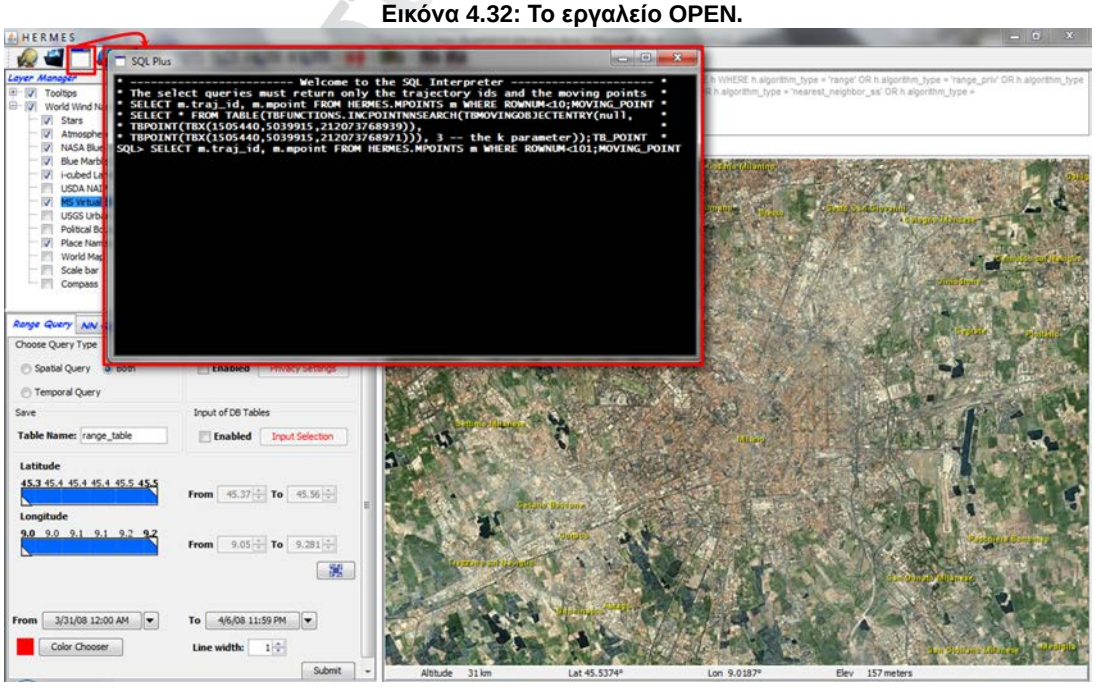
προεπιλεγμένης βάσης έχουν αλλάξει, θα πρέπει να χρησιμοποιήσει το εργαλείο DB Connector (βλέπε εικόνα 4.31). Με αυτό το εργαλείο, ο χρήστης αλλάζει τα στοιχεία του αρχείου ώστε να προσαρμοστούν στους νέους κωδικούς της βάσης δεδομένων ή στην πρόσβαση μιας άλλης βάσης.

OPEN Tool

Ένας χρήστης μπορεί να παρατηρήσει τα αποθηκευμένα αποτελέσματα ενός ερωτήματος είτε ανήκει στον μηχανισμό του *HERMES* και *HERMES++* ή είναι από έναν αλγόριθμο εξόρυξης γνώσης. Το εργαλείο *OPEN* ανοίγει τα αποτελέσματα των ερωτημάτων αναζήτησης και εξόρυξης γνώσης από τους πίνακες που έχουν δημιουργηθεί από τα εκτελεσμένα ερωτήματα στην βάση τροχιών (βλέπε Εικόνα 4.32). Συγκεκριμένα, το *OPEN* παρουσιάζει οπτικά τις αποθηκευμένες τροχιές ή πρότυπα πάνω στον χάρτη ή σε μορφή κειμένου στο *QUERY RESULTS*. Έτσι, ο χρήστης μπορεί εύκολα και γρήγορα να εξετάσει τα αποθηκευμένα αποτελέσματα και στην συνέχεια, να τα χρησιμοποιήσει για περαιτέρω ανάλυση.



Εικόνα 4.32: Το εργαλείο OPEN.



Εικόνα 4.33: Το εργαλείο SQL Plus.

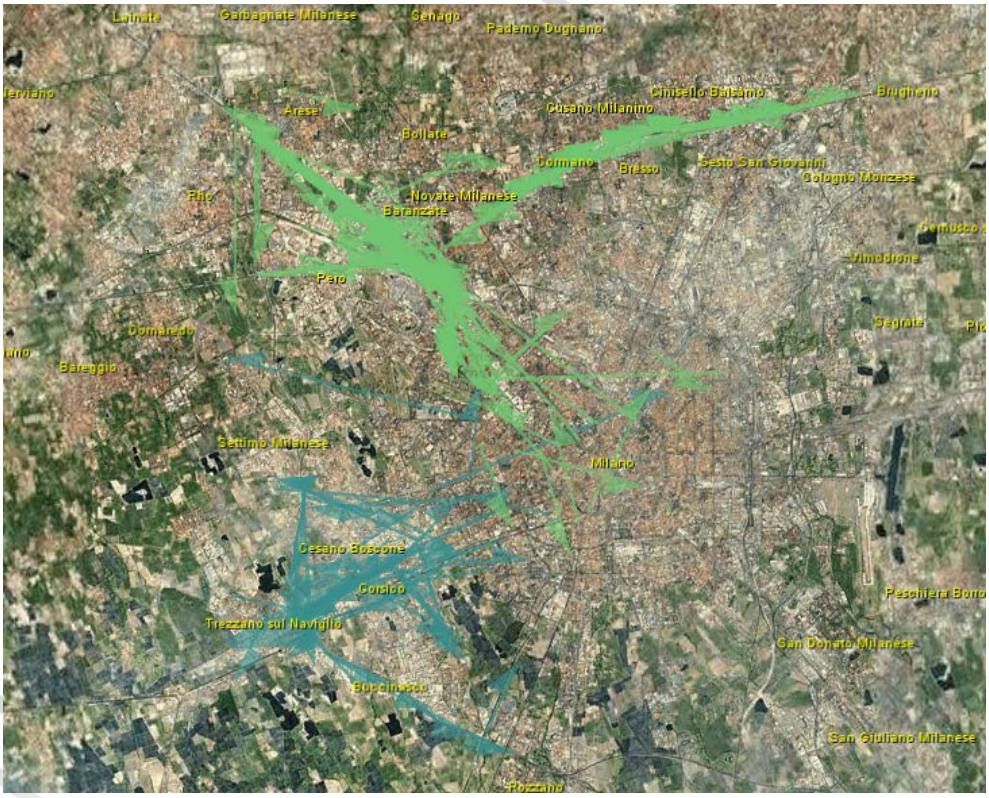
SQL Plus

Η εφαρμογή δίνει την δυνατότητα στο χρήστη να γράψει SQL ερωτήματα και αυτά να παρουσιαστούν οπτικά στον χάρτη με το εργαλείο *SQL Plus*. Συγκεκριμένα, μπορούμε να επιλέξουμε από την Μπάρα Εργαλείων το *SQL Plus* και από κει να εκτελέσουμε χώρο-χρονικά ερωτήματα όπως φαίνεται στην εικόνα 4.33. Αυτό που πρέπει να σημειωθεί είναι ότι θα πρέπει να του ορίσουμε στο τέλος και το είδος του αντικειμένου δεδομένων, δηλαδή αν είναι κινούμενο αντικείμενο (Moving_point) ή χωρικό σημείο (TB_Point). Παραδείγματος χάριν, αν θέλουμε να πάρουμε τις πρώτες 100 τροχιές από την βάση του HERMES, θα γράψουμε τα εξής:

```
SELECT m.traj_id, m.mpoint FROM HERMES.MPOINTS m WHERE
ROWNUM<101;MOVING_POINT <-Μετά το ερωτηματικό, ορίζουμε το αντικείμενο δεδο-
μένων (data object)
```

Ενώ αν θέλουμε να εκτελέσουμε ένα NN Query, τότε το ερώτημα θα ήταν το εξής:

```
SELECT * FROM TABLE( tbfunctions.incpointnnsearch( tbMovingObject-
Entry(NULL, TbPoint(tbX(5791356.86863182, 1239022.32055846,
2.120737248E11)), TbPoint(tbX(5791356.86863182, 1239022.32055846,
2.12074329599E11))), 5));TB_POINT
```

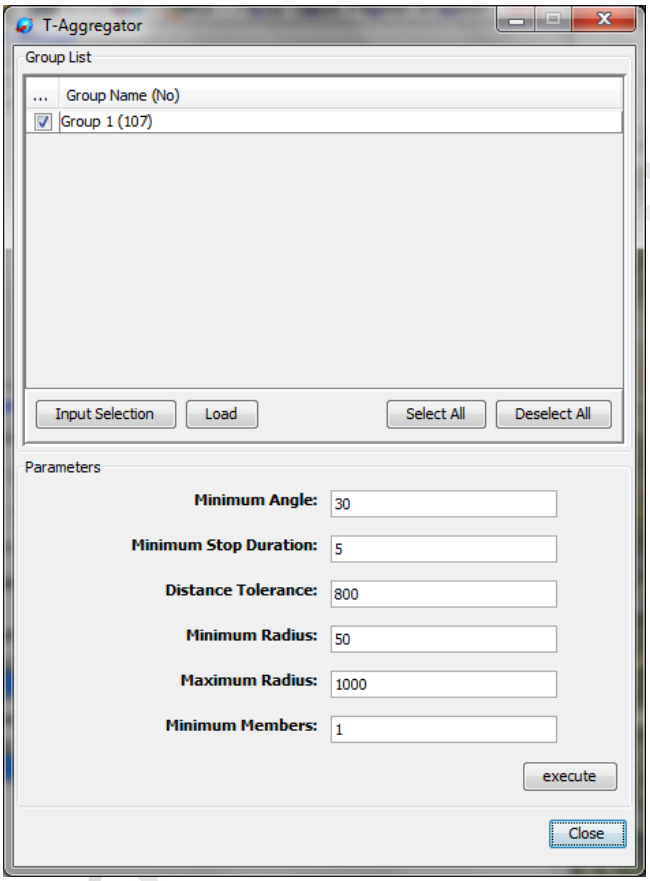


Εικόνα 4.34: Συγκεντρωτική αναπαράσταση των διαδρομών σε δύο συστάδες.

T-Aggregator

Η παρουσίαση των πολλαπλών τροχιών με γραμμές δεν επιτρέπουν στον αναλυτή να παρατηρήσει πόσες διαδρομές υπάρχουν και να διακρίνει συχνές πορείες από τις λιγότερο συχνότερες και περιστασιακές. Για τον λόγο αυτό, το εργαλείο *T-Aggregator* (από την μπάρα εργαλείων) έχει ενσωματώσει έναν αλγόριθμο συνάθροισης δεδομένων βασισμένο σε αυτόν των συγγραφέων G. Andrienko, N. Andrienko και S. Wrobel [10], το οποίο παρουσιάζει τις πολλαπλές διαδρομές με μια γενικευμένη και συγκεντρωτι-

κή διαδικασία όπου τα βέλη δείχνουν την κατεύθυνση των μετακινήσεων και το πάχος είναι ανάλογο με τον αριθμό των μετακινήσεων (βλέπε Εικόνα 4.34). Εάν οι τροχιές ανήκουν σε μια συστάδα ή γενικότερα σε ένα αντικείμενο δεδομένων, τότε τα βελάκια του *T-Aggregator* οπτικοποιούνται με το ίδιο χρώμα. Όπως εύκολα διακρίνουμε από την εικόνα 4.34, ο αλγόριθμος έχει υπολογίσει τις συγκεντρωτικές μετακινήσεις δύο συστάδων (με το ανοιχτό πράσινο και μπλε χρώμα) του αλγόριθμου συσταδοποίησης *T-Optics*.



Εικόνα 4.35: Οι παράμετροι του T-Aggregator.

Για να εκτελεστεί ο T-Aggregator, ο χρήστης επιλέγει ένα σύνολο ή υποσύνολο των αποθηκευμένων τροχιών από την βάση δεδομένων κίνησης ως είσοδο στο αλγόριθμο με την χρήση διαδραστικών παραθύρων καθώς επίσης συμπληρώνει τις παραμέτρους του αλγόριθμου των συγκεντρωτικών μετακινήσεων (βλέπε Εικόνα 4.35).

4.6. ΣΥΝΟΨΗ

Η προοδευτική αναζήτηση και εξόρυξη των δεδομένων είναι ακόμη μια πρόκληση για το γνωστικό πεδίο της εξόρυξης γνώσης και αποτελεί ένα από τους βασικούς τρόπους για την ανάλυση των δεδομένων. Η διαδικασία αυτή αν και κατανοητή και απλή, δεν έχει χρησιμοποιηθεί αρκετά στην ανάλυση. Σε αυτό το κεφάλαιο, ο κύριος στόχος του ήταν να επιδείξει τον μηχανισμό της προοδευτικής ανάλυσης μέσα από την χρήση διεπαφών που θα διευκολύνουν τον χρήστη στην εξόρυξη των δεδομένων. Ο μηχανισμός της προοδευτικής ανάλυσης περιέχει τέσσερις διαφορετικές λειτουργίες, (α) την εκτέλεση ενός απλού ερωτήματος από τα αποτελέσματα ενός προηγούμενου ερωτήματος, (β) την εκτέλεση ενός αλγόριθμου εξόρυξης από τα αποτελέσματα ενός απλού ερωτήματος, (γ) την εφαρμογή ενός αλγόριθμου εξόρυξης πάνω στα αποτελέσματα ενός αλγόριθμου εξόρυξης και (δ) την εφαρμογή ενός απλού ερωτήματος πάνω στα αποτελέσματα ενός αλγόριθμου εξόρυξης. Τελειώνοντας, είναι σημαντικό να τονιστεί ότι τα αποτελέσματα των ερωτημάτων δεν ήταν ο σκοπός για την ερμηνεία των δεδομένων αλλά η αξιοποίηση των οπτικών

διαδραστικών εργαλείων και οπτικών αντικειμένων πάνω στο χάρτη έπαιξαν σημαντικό ρόλο στην πλατφόρμα ώστε να βοηθήσουν τον αναλυτή να κερδίσει γνώση από μια μεγάλη βάση δεδομένων.

Παρ' όλα αυτά, υπάρχουν διάφορα ζητήματα που αξίζουν περισσότερη έρευνα. Η χρήση αλληλεπιδραστικών παραθύρων αν και διευκολύνει τον χρήστη να χρησιμοποιεί την προοδευτική διαδικασία χωρίς να απαιτείται να γράψει κάποια γλώσσα ερωτημάτων, τον περιορίζει στην εκτέλεση πιο περίπλοκων συνδυασμών. Για να υποστηριχθούν αυτοί οι συνδυασμοί, θα πρέπει να αναπτυχθούν δυναμικές και προσαρμοσμένες διεπαφές από αυτές που προτείνονται μέχρι τώρα. Επίσης, η ανάπτυξη στρατηγικών για την βελτιστοποίηση των επεξεργασιών των ερωτημάτων θα αυξήσει τις συνολικές επιδόσεις της προοδευτικής ανάλυσης.

5. ΟΠΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΤΟΥ HERMES++

5.1. ΕΙΣΑΓΩΓΗ

Λόγω της μεγάλης αύξησης των κινητών συσκευών και των γεωγραφικών συστημάτων, τα κινούμενα αντικείμενα συλλέγονται σε μεγάλη κλίμακα και γίνονται όλο και περισσότερο άφθονα, περίπλοκα και πανταχού παρών με αποτέλεσμα η ανάλυση τους να εξελίσσεται σε μια μεγάλη πρόκληση. Σήμερα, η δημοτικότητα των βάσεων δεδομένων κίνησης (TD – Trajectory Database) [6] έχει προκαλέσει μεγάλα ερευνητικά κίνητρα στο γνωστικό πεδίο της εξόρυξης γνώσης. Ένας από τους αντικειμενικούς στόχους της εξόρυξης γνώσης πάνω στα χώρο χρονικά δεδομένα είναι η ανάλυση των δεδομένων αυτών και η αποκάλυψη ενδιαφέρων και χρήσιμων προτύπων. Παρ' όλα αυτά, η συλλογή και η γνωστοποίηση των προσωπικών κινούμενων πληροφοριών αυξάνει το ρίσκο της παραβίασης της ιδιωτικής ζωής των ατόμων. Αν και διάφοροι αλγόριθμοι ανωνυμοποίησης δεδομένων έχουν πρόσφατα προταθεί, δεν υπάρχει κάποια αξιόλογη προσπάθεια μέχρι στιγμής που να εντάσσει αυτούς τους αλγόριθμους κάτω από μια κοινή πειραματική εφαρμογή.

Παραπλεύρως αυτής της κατεύθυνσης, η πλατφόρμα της παρούσας μεταπτυχιακής διατριβής δίνει την δυνατότητα στους χρήστες (α) να θέτουν απλά ερωτήματα και ερωτήματα προστασίας δεδομένων του *HERMES* και του *HERMES++* αντίστοιχα, (β) να εφαρμόζουν τους αλγόριθμους ανωνυμοποίησης όπως είναι ο *NWA* και ο *W4M* στα δεδομένα ενώ έχουν την ικανότητα να συγκρίνουν και να αξιολογούν τα αποτελέσματα μεταξύ των αρχικών και ανώνυμων δεδομένων μέσα από μια σειρά τεχνικών εξόρυξης γνώσης και (γ) να σχεδιάζουν και να εκτελούν πειράματα έτσι ώστε να αξιολογήσουν την απόδοση των αλγόριθμων ανωνυμοποίησης χρησιμοποιώντας διαφορετικούς φόρτους εργασίας πάνω στα ερωτήματα. Συγκεκριμένα, ένας χρήστης μπορεί να εκτελέσει απλές λειτουργίες του *HERMES* ή να πραγματοποιήσει αλγόριθμους εξόρυξης γνώσης με σκοπό να ανακτήσει εποικοδομητικά πρότυπα. Επιπλέον, ο χρήστης μπορεί να εκτελέσει ερωτήματα, τα οποία τα επιστρεφόμενα αποτελέσματα να διαφυλάσσουν την ιδιωτική ζωή των ατόμων που η κάθε κίνηση τους καταγράφεται στην βάση, μέσω της παραγωγής ψεύτικων αλλά καλά διατυπωμένων τροχιών. Αυτό επιτυγχάνεται χρησιμοποιώντας την λειτουργικότητα, η οποία παρέχεται από τον *HERMES++* , μια μηχανή ερωτημάτων που προστατεύει τους χρήστες από διάφορων ειδών επιθέσεων όταν θέτουν ένα ερώτημα. Επίσης, η πλατφόρμα δίνει την δυνατότητα στον χρήστη να σχεδιάζει και να πραγματοποιεί πειράματα περιέχοντας διαφορετικό αριθμό και ειδών ερωτημάτων. Τα εν λόγω πειράματα μπορούν να χρησιμοποιηθούν για τη μέτρηση της χρησιμότητας των ανώνυμων στοιχείων είτε με την εφαρμογή τεχνικών εξόρυξης γνώσης και τη σύγκριση των μοντέλων που προέρχονται από τα αρχικά και ανώνυμα δεδομένα, ή θέτοντας ερωτήματα στα αρχικά και ανώνυμα δεδομένα (ή πρότυπα). Τέλος, η πλατφόρμα υποστηρίζει ερωτήματα ελεγκτικών τεχνικών που μπορούν να χρησιμοποιηθούν για την κατασκευή του προφίλ του χρήστη με βάση των ερωτημάτων που θέτει στη βάση δεδομένων με σκοπό να εντοπιστούν οι τυχών ύποπτες συμπεριφορές σε κάθε χρήστη. Με λίγα λόγια, αυτή είναι η πρώτη προσπάθεια που παρουσιάζει ένα ολοκληρωμένο σύνολο των '*state-of-the-art*' αλγόριθμων ανωνυμοποίησης δεδομένων κίνησης καθώς και των τεχνικών εξόρυξης γνώσης, τα οποία έχουν ενοποιηθεί μαζί με την μηχανή απλών ερωτημάτων και την μηχανή προστασίας ευαίσθητων πληροφοριών.

Το υπόλοιπο αυτού του κεφαλαίου είναι οργανωμένο ως εξής. Το δεύτερο υποκεφάλαιο παρέχει μια περιγραφή των τεχνικών, οι οποίες έχουν ενσωματωθεί σε αυτήν την πλατφόρμα. Στο τρίτο υποκεφάλαιο, επισημαίνονται οι τρόποι ελέγχου των προφίλ των χρηστών για την εξακρίβωση των κακόβουλων ή όχι χρηστών. Στο τέταρτο υποκεφάλαιο, παρουσιάζονται παραδείγματα από την εφαρμογή χρησιμοποιώντας μια βάση πραγματικών δεδομένων κίνησης. Το τελευταίο υποκεφάλαιο ολοκληρώνεται με την διατύπωση χρήσιμων συμπερασμάτων καθώς και επιπλέων μελλοντικών κατευθύνσεων.

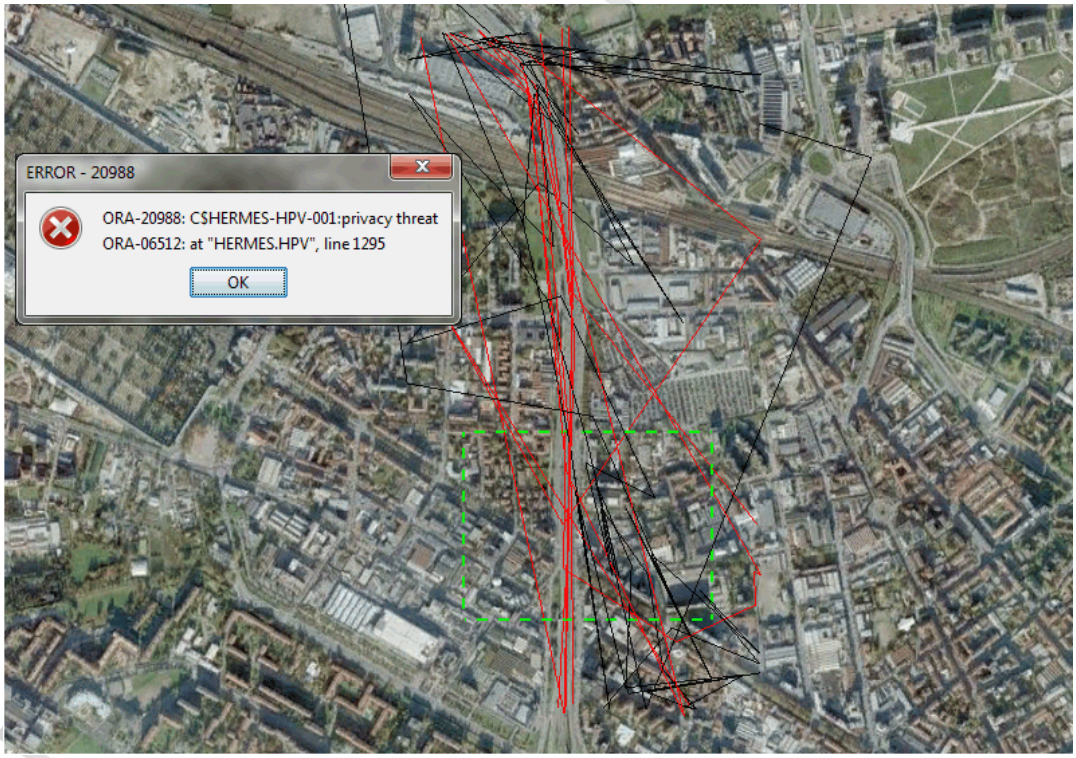
5.2. ΑΝΑΓΝΩΡΗΣΗ ΚΑΙ ΑΠΟΤΡΟΠΗ ΕΠΙΘΕΣΕΩΝ ΜΕΣΑ ΑΠΟ ΤΗΝ ΠΛΑΤΦΟΡΜΑ

Η πρώτη κατεύθυνση της μεταπτυχιακής εφαρμογής είναι η υποστήριξη των υπηρεσιών *Location-Based Services* (*LBS*) και του μηχανισμού ερωτημάτων προστασίας προσωπικών δεδομένων (*privacy-aware*

query engine). Ο *HERMES*, ο οποίος εμπεριέχεται μέσα στην πλατφόρμα της μεταπτυχιακής διατριβής, είναι μια μηχανή ερωτημάτων βασισμένη σε μια δυναμική γλώσσα για την ανάλυση βάσεων δεδομένων κίνησης που ενεργοποιεί την υποστήριξη των *LBS*. Παραδείγματος χάριν, ο *HERMES* υποστηρίζει διάφορα είδη ερωτημάτων όπως είναι τα ερωτήματα *range*, *nearest neighbor*, *topological* και *directional*. Από την άλλη πλευρά, ο *HERMES++*, ο οποίος περιλαμβάνεται στην πλατφόρμα και περιέχει ερωτήματα διαφύλαξης ευαίσθητης πληροφορίας, επιτρέπει στους χρήστες να έχουν περιορισμένη πρόσβαση στην βάση για να πραγματοποιήσουν εργασίες ανάλυσης. Συγκεκριμένα, η μηχανή του *HERMES++* (α) ελέγχει τα ερωτήματα με σκοπό να μπλοκάρει επιθέσεις παραβίασης προσωπικών δεδομένων των χρηστών, (β) υποστηρίζει χωρικά και χώρο χρονικά ερωτήματα στους τύπους: range, distance και nearest neighbor, και (γ) διατηρεί την ανωνυμία των χρηστών στις απαντήσεις των ερωτήσεων κατασκευάζοντας ψεύτικες τροχιές, οι οποίες είναι ρεαλιστικά δημιουργημένες.

Ο *HERMES++* διαθέτει την ικανότητα να διασφαλίζει την ιδιωτική ζωή των ατόμων αποκλείοντας τριών ειδών επιθέσεων όπου οι κακόβουλοι χρήστες προσπαθούν να επιδιώξουν σε μια πραγματική βάση δεδομένων κίνησης:

- *Επιθέσεις ταυτοποίησης χρήστη*: Αυτή η επίθεση οδηγεί στην αποκάλυψη της ταυτότητας του χρήστη. Ο κακόβουλος χρήστης προσπαθεί να συσχετίσει ένα άτομο από την τροχιά του κάνοντας στην βάση επαναλαμβανόμενα χώρο χρονικά ερωτήματα εύρους (*range*) όπου το κάθε ερώτημα επικαλύπτεται από το προηγούμενο. Στην εικόνα 5.1, ένας κακόβουλος χρήστης πραγματοποίησε ένα χώρο χρονικό ερώτημα εύρους σε μια μικρή περιοχή του Μιλάνο και στην συνέχεια, επιδίωξε την εκτέλεση ενός δεύτερου ερωτήματος πάνω στις τροχιές του προηγούμενου ερωτήματος σε ακόμα μικρότερη περιοχή. Ο μηχανισμός αναγνώρισε την επίθεση και δεν επέτρεψε στον χρήστη να παρατηρήσει τα αποτελέσματα του δεύτερου ερωτήματος εμφανίζοντας ένα μήνυμα στην οθόνη.



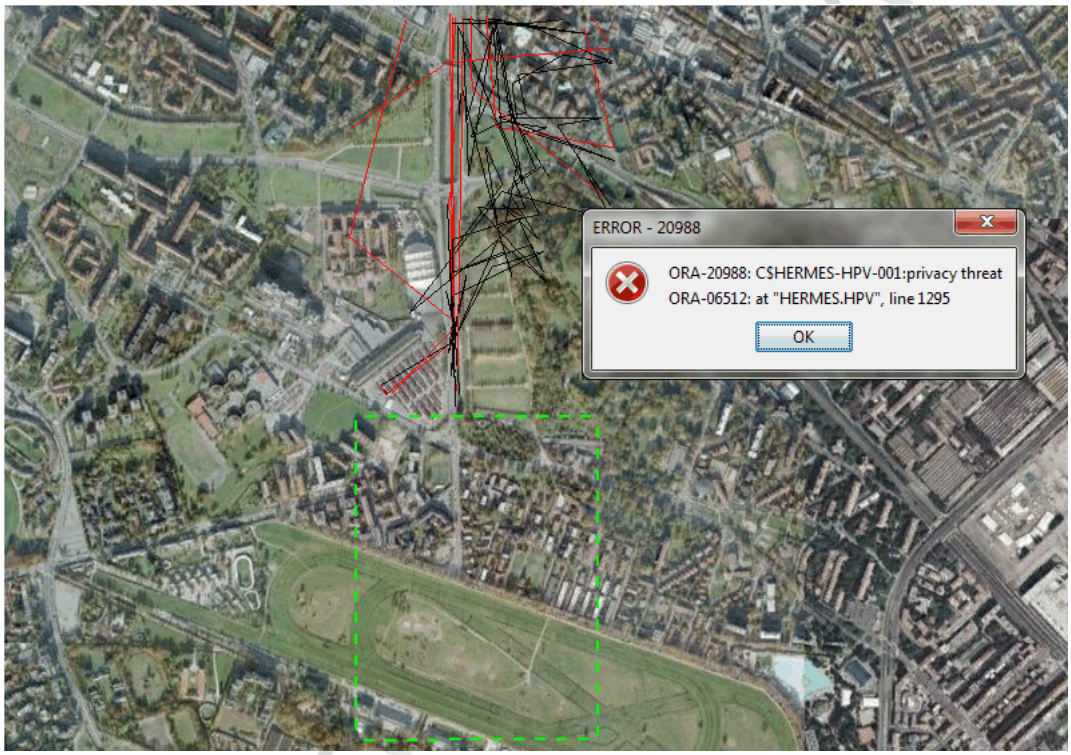
Εικόνα 5.1: Αποτρέποντας την επίθεση ταυτοποίησης χρήστη.

Αξίζει να σημειωθεί ότι οι μαύρες τροχιές είναι οι τροχιές που κατασκευάστηκαν από τον μηχανισμό με σκοπό να διατηρήσουν την ανωνυμία των δεδομένων. Στο συγκεκριμένο παράδειγμα, οι ψεύτικες τροχιές έχουν επιλεχτεί να εμφανίζονται με διαφορετικό χρώμα από τις πραγματικές αλλά υπάρχει η επιλογή στην εφαρμογή αυτές οι τροχιές να απεικονίζονται στο ίδιο χρώμα με τις πραγματικές (π.χ. κόκκινο).

- *Επιθέσεις εντοπισμού περιοχών ενδιαφέροντος*: Ο κακόβουλος χρήστης προσπαθεί να αναδείξει σημεία ενδιαφέροντος π.χ. αρχικά και τελικά σημεία των ατόμων που μπορεί να είναι η

διεύθυνση των σπιτιών τους ή της δουλειάς τους και προσπαθεί να ταιριάξει τα σημεία αυτά με πιθανόν ευαίσθητη τοποθεσία των χρηστών αυτών με σκοπό να αποκαλύψει την ταυτότητά τους. Τα σημεία αυτά αποκαλούνται 'ευαίσθητα' (προσωπικά) και για αυτό δεν πρέπει να εμφανίζεται στους επιτιθέμενους.

- *Επιθέσεις παρακολούθησης τροχιών*: Σε αυτήν την επίθεση, ο επιτιθέμενος προσπαθεί να ακολουθήσει την τροχιά του κάθε ατόμου πάνω στο χάρτη, εκπληρώνοντας μια ακολουθία από ερωτήματα σε περιοχές που είναι δίπλα η μία με την άλλη και το αντίστοιχο χρονικό διάστημα να είναι κοντά με το προηγούμενο. Ο σκοπός των κακόβουλων χρηστών είναι να μάθουν προσωπικά σημεία που επισκέφτηκαν οι χρήστες και για τον λόγο αυτό η γνωστοποίηση των συνήθειων των ατόμων οδηγεί στην παραβίαση της ιδιωτικής ζωής τους.



Εικόνα 5.2: Αποτρέποντας την επίθεση παρακολούθησης τροχιών.

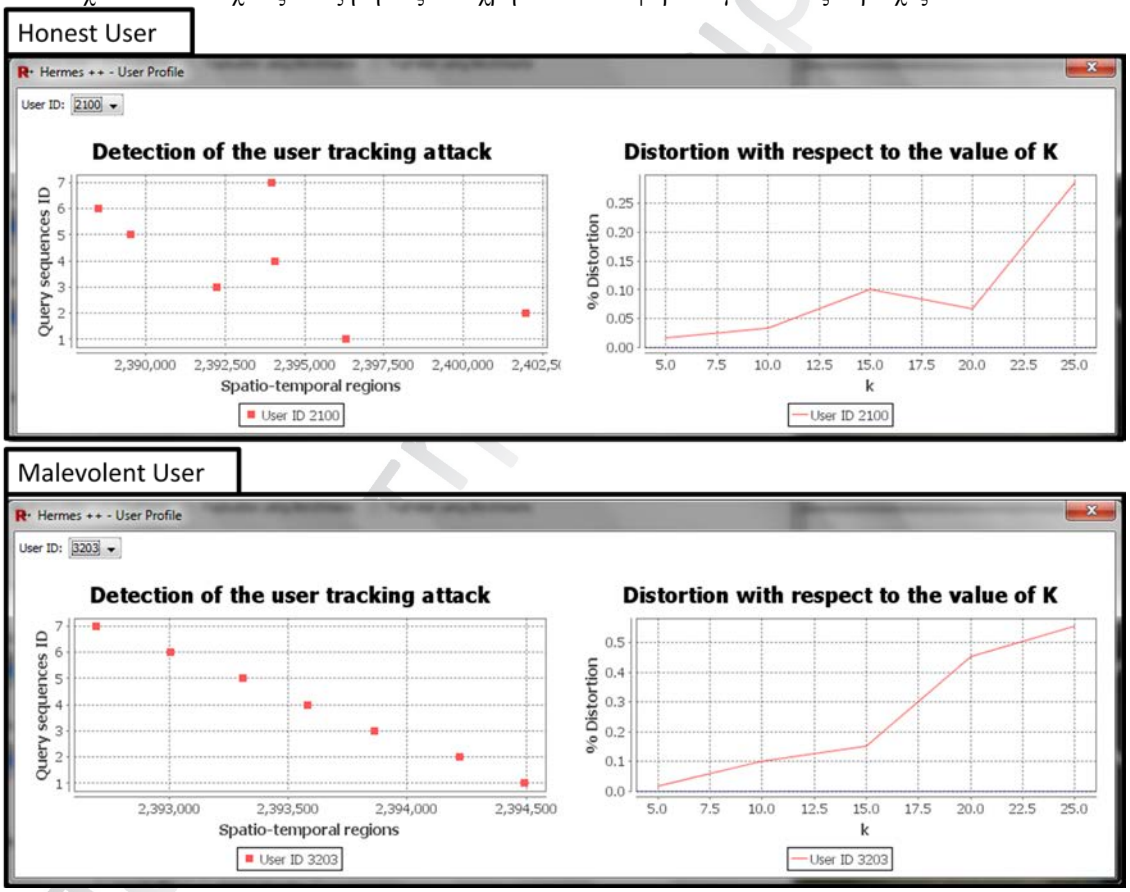
Στο παράδειγμα της εικόνας 5.2, ο επιτιθέμενος αυτήν την φορά επιδίωξε την εφαρμογή ενός χώρο χρονικού ερωτήματος κοντά στις προηγούμενες τροχιές (αλλά χωρίς να τις επικαλύπτει) ακολουθώντας την κατεύθυνση τους. Και αυτήν την φορά, ο *Hermes++* μπλόκαρε τον χρήστη στο να ανακαλύψει την υπόλοιπη διαδρομή των τροχιών.

Εκτός από το *LBS* και τις μηχανές αναζήτησης, ένας άλλος τομέας της προστασίας της ιδιωτικής ζωής των χρηστών, που υποστηρίζεται από αυτή την πλατφόρμα, περιλαμβάνει την διαχείριση των βάσεων δεδομένων κίνησης (*MODs*). Ειδικότερα, ένας από τους στόχους είναι η ανωνυμοποίηση των δεδομένων κίνησης, η οποία 'μεταβάλλει' ένα σύνολο δεδομένων έτσι ώστε ένας κακόβουλος χρήστης να μην μπορεί πλέον να ταιριάζει τα καταγεγραμμένα δεδομένα ενός αντικειμένου σε ένα συγκεκριμένο άτομο. Οι αλγόριθμοι που έχουν ενσωματωθεί σε αυτή την πλατφόρμα για να βοηθήσουν στην παραγωγή ανώνυμων τροχιών είναι ο *NWA* και ο *W4M*. Όσο αφορά τον πρώτο, ο *NWA* εισάγει την έννοια της (k, δ)-ανωνυμίας, όπου δ αντιπροσωπεύει την ανακριβής θέση. Η μέθοδος βασίζεται στην ομαδοποίηση των τροχιών και στους χωρικούς μετασχηματισμούς, προκειμένου να καταστεί μια τροχιά μέσα στον κύλινδρο που θα περιέχει τουλάχιστον K-1 άλλες τροχιές. Για την επίτευξη του χώρο χρονικού μετασχηματισμού, οι συγγραφείς πρότειναν τον *W4M*, ο οποίος χρησιμοποιεί ένα διαφορετικό μέτρο απόστασης. Και οι δύο αλγόριθμοι παίρνουν ως είσοδο ένα σύνολο τροχιών και δημοσιεύουν ανώνυμες τροχιές κατάλληλες για εφαρμογές εξόρυξης γνώσης. Πάνω σε αυτήν την κατεύθυνση, η πλατφόρμα δίνει την δυνατότητα στους χρήστες να χρησιμοποιούν διάφορων ειδών αλγόριθμων εξόρυξης γνώσης (όπως αναφέρθηκαν σε προηγούμενα κεφάλαια) πάνω στα αποτελέσματα των αλγόριθμων ανωνυμίας με σκοπό είτε να εξάγουν

γνώση ή να συγκριθούν την απόδοση των αλγόριθμων εξόρυξης. Εν συντομία, οι αλγόριθμοι εξόρυξης που έχουν ενσωματωθεί στην παρούσα εφαρμογή επικεντρώνονται κυρίως σε τεχνικές συσταδοποίησης.

5.3. ΤΕΧΝΙΚΕΣ ΕΛΕΓΧΟΥ ΠΡΟΦΙΛ ΧΡΗΣΤΩΝ

Ο *HERMES++* έχει την δυνατότητα να αποθηκεύει το ιστορικό του χρήστη (π.χ. γεωμετρική θέση, είδος ερωτήματος, το συνολικό αριθμό των ψεύτικων τροχιών ανά ερώτημα, κτλ) στην βάση έτσι ώστε να αξιοποιηθεί μετέπειτα στην αποκάλυψη των ύποπτων συμπεριφορών. Ο Γκουλαλάς-Ντιβάνης Α. και ο Βερύκιος Β. [1] εισήγαγαν τρόπους που ένας διαχειριστής συστήματος θα έχει την δυνατότητα να επιβλέπει και να μπλοκάρει πολλές από τις προσπάθειες των επιτιθέμενων στην βάση δεδομένων κίνησης. Παραδείγματος χάριν, η εικόνα 5.3 απεικονίζει δύο μηχανισμούς για να ξεχωρίσει τους κακόβουλους χρήστες από τους αξιόπιστους. Συγκεκριμένα, η πρώτη κατηγορία διαγραμμάτων (‘*Detection of user tracking attack*’) αναπαριστά την γεωμετρική θέση των ερωτημάτων που έχει πραγματοποιήσει ένας χρήστης. Δηλαδή, για κάθε χρήστη του συστήματος δημιουργείται ένα διάγραμμα συλλαμβάνοντας τις χώρο χρονικές περιοχές με βάση το ιστορικό των ερωτημάτων, ταξινομημένες ως προς την τοπική τους σχέση (άξονας x) έναντι της σειριακής ακολουθίας των ερωτημάτων που πραγματοποίησε (άξονας y). Οπότε, τα σημεία που βρίσκονται κοντά το ένα με το άλλο στο επίπεδο αντιστοιχούν σε διαδοχικές αναζητήσεις των χρηστών και αφορούν γειτονικές περιοχές.



Εικόνα 5.3: Σύγκριση συμπεριφορών δυο χρηστών με την βοήθεια των ιστορικών στοιχείων από τα ερωτήματα τους.

Μια τέτοια συμπεριφορά είναι τυπική για ένα κακόβουλο χρήστη ενώ δεν εμφανίζεται στους αξιόπιστους. Στην παρακάτω εικόνα, ο χρήστης 3203 παρουσιάζει ύποπτη συμπεριφορά διότι πραγματοποιεί ερωτήματα σε μια μικρή χώρο χρονική περιοχή και η σειριακή ακολουθία των ερωτημάτων είναι κοντά η μια με την άλλη. Αντιθέτως, ο χρήστης 2100 θέτει ερωτήματα στην βάση σε μια τυχαία σειρά και οι αντίστοιχες χώρο χρονικές περιοχές είναι σε μεγαλύτερο εύρος.

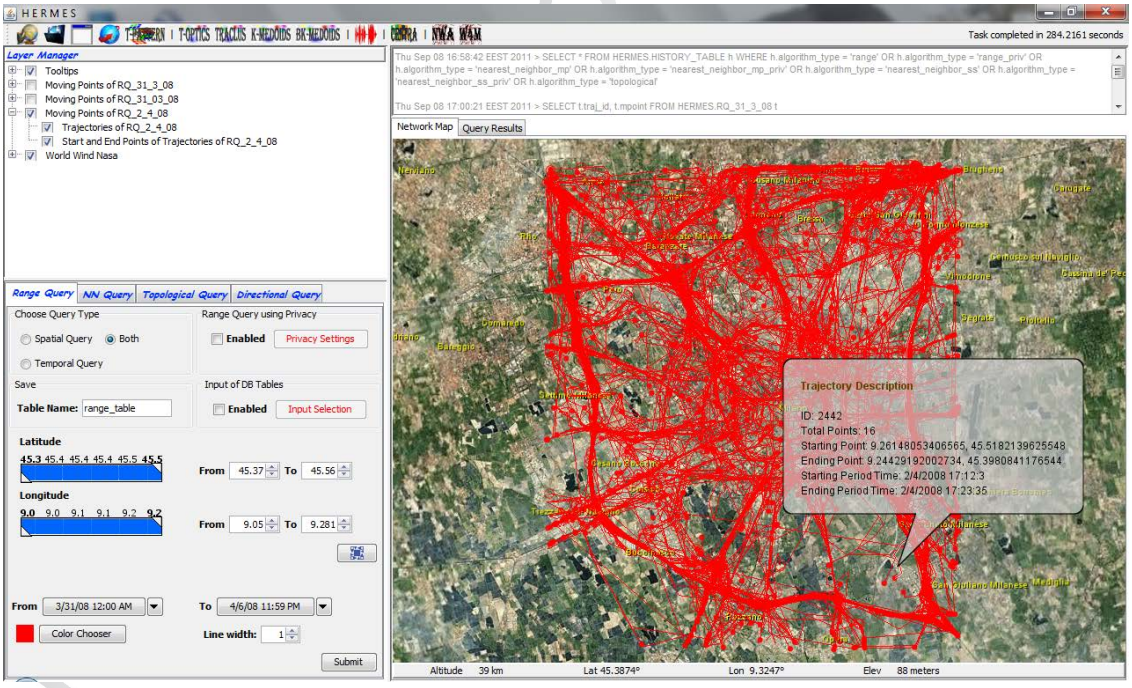
Ένας εναλλακτικός τρόπος για να ξεσκεπάσουμε ύποπτες συμπεριφορές είναι από την διαστρέβλωση των δεδομένων στην βάση, το οποίο προκαλείται από την δημιουργία των ψεύτικων τροχιών. Η διαστρέβλωση μετριέται από το ποσοστό των ψεύτικων εγγραφών στην βάση δεδομένων κίνησης για οποιαδήπο-

τε χρονική στιγμή. Για παράδειγμα, η δεύτερη κατηγορία διαγραμμάτων (‘*Distortion with respect to the value of k*’) της παραπάνω εικόνας απεικονίζει το ποσοστό διαστρέβλωσης της βάσης δεδομένων σε σχέση με την τιμή k. Ως αναμενόμενο, χαμηλές τιμές του k προκαλούν μικρή διαστρέβλωση στην βάση λόγω του ότι λίγες ψεύτικες τροχιές απαιτούνται να δημιουργηθούν καθώς και λίγες τροχιές χρειάζονται να συναντήσουν τις απαιτήσεις του *k-anonymity*. Επιπλέον, μεγάλες σειριακές ακολουθίες από ερωτήματα εστιαζόμενα σε μια μικρή περιοχή οδηγούν σε επιπλέον καταγραφή ψεύτικων εγγραφών. Μια τέτοια συμπεριφορά παρατηρείται από τον χρήστη 3203 όπου έχει δημιουργήσει μεγαλύτερη διαστρέβλωση (συνολικά 0.5%) σε σχέση με τον χρήστη 2100 (συνολικά 0.25%) για τις αντίστοιχες τιμές του k. Χρησιμοποιώντας τους παραπάνω γράφους, η πλατφόρμα της μεταπτυχιακής διατριβής δίνει την δυνατότητα σε έναν διαχειριστή συστήματος (*administrator*) να παρατηρεί τις συμπεριφορές των χρηστών με σκοπό να μπορεί να μπλοκάρει τυχόν ύποπτες κινήσεις.

5.4. ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΕΩΝ

Μέσω των παραδειγμάτων χρήσεων, οι χρήστες μπορούν να δοκιμάσουν το σύστημα χρησιμοποιώντας μια πραγματική βάση δεδομένων κίνησης περιέχοντας τα ίχνη των GPS από τις μετακινήσεις των αυτοκινήτων στην πόλη του Μιλάνο. Εν συντομία, θα παρουσιαστούν παραδείγματα για το πως επηρεάζονται τα δεδομένα με την χρήση αλγόριθμων κ-ανωνυμίας και πως τα αποτελέσματα των αλγόριθμων αυτών επιδρούν πάνω στην εκτέλεση των αλγόριθμων εξόρυξης γνώσης. Στην συνέχεια, θα εκτελεστούν τα ερωτήματα του *HERMES++* και θα συγκριθούν με τα αντίστοιχα ερωτήματα του *HERMES*. Τέλος, θα επιδειχτούν τα πειράματα αξιολόγησης από τα ερωτήματα του *HERMES++*.

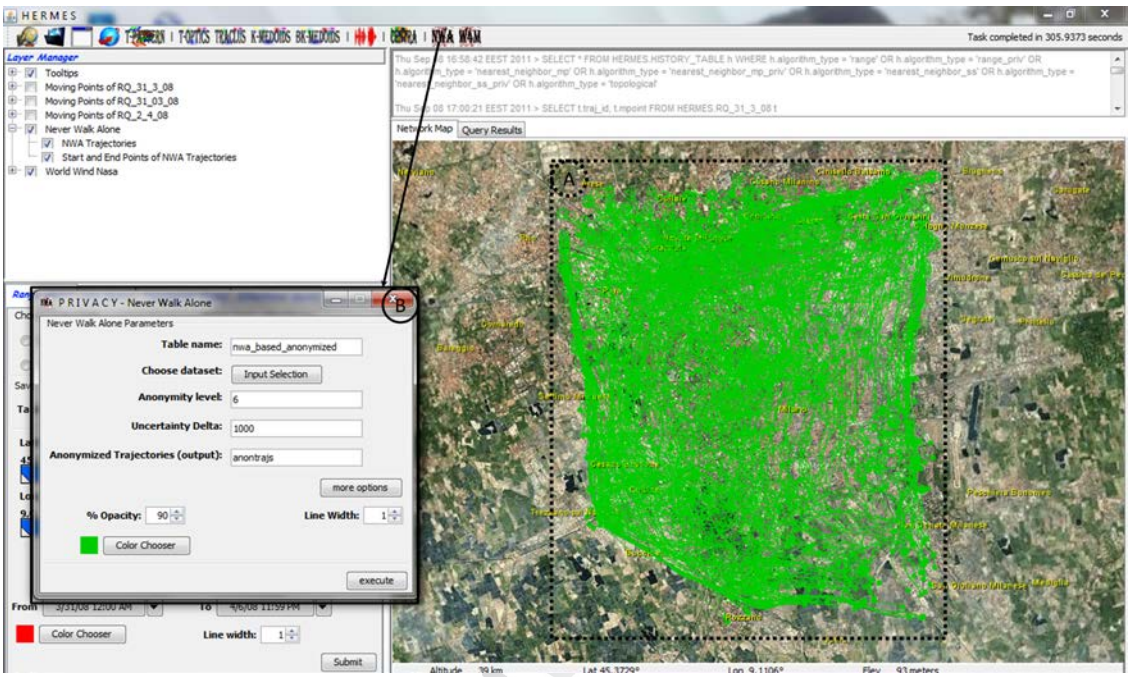
Καταρχήν, οι επόμενες εικόνες αναπαριστούν στιγμιότυπα από το γραφικό περιβάλλον της εφαρμογής παρουσιάζοντας την κάθε λειτουργία τους. Στην εικόνα 5.4, απεικονίζονται τα αρχικά δεδομένα κίνησης που συλλέχτηκαν την μέρα 2/4/08 χρησιμοποιώντας ένα *Range Query* (όπως περιγράφτηκε στο προηγούμενο κεφάλαιο).



Εικόνα 5.4: Απεικόνιση πραγματικών δεδομένων από την εφαρμογή ενός Range Query.

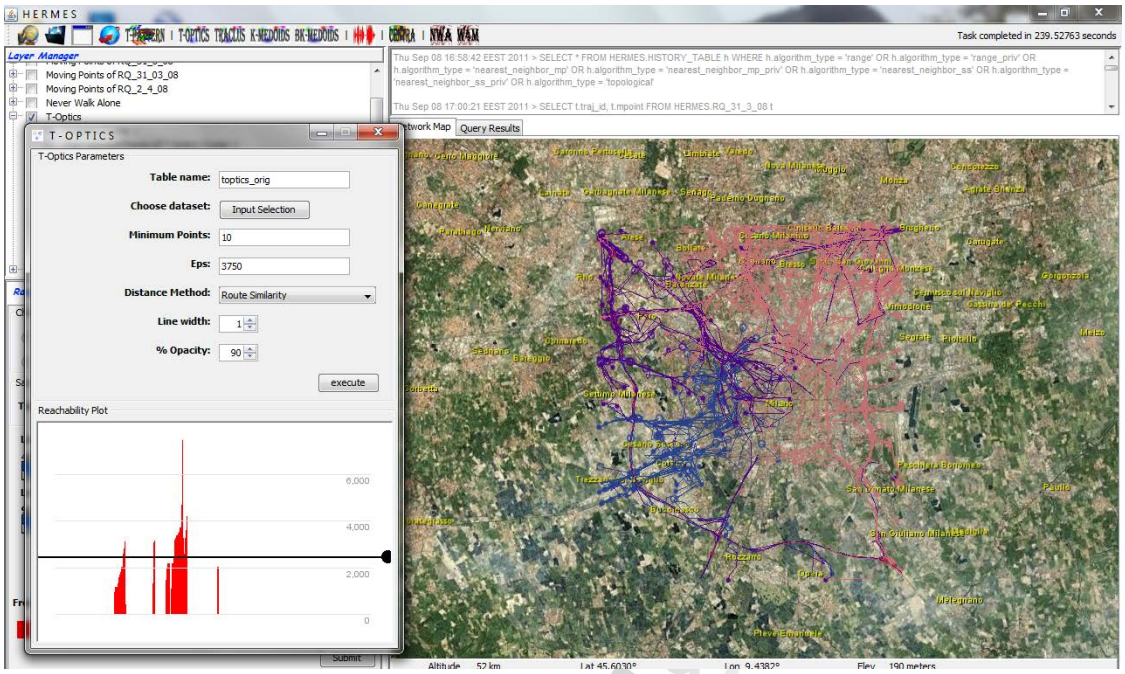
Στην εικόνα 5.5, χρησιμοποιήθηκε το σύνολο των αποτελεσμάτων του *Range Query* ως είσοδο δεδομένων για την πραγματοποίηση του αλγόριθμου *Never Walk Alone* (*NWA*). Αναλυτικότερα, επιλέχτηκε ο *NWA* από την μπάρα εργαλείων και συμπληρώθηκαν οι παράμετροι όπως φαίνεται στην εικόνα 5.5 Β. Συγκεκριμένα, οι βασικοί παράμετροι που πρέπει να συμπληρώσει ο χρήστης είναι το επίπεδο της ανωνυμίας ‘*Anonymity Level*’ και η παράμετρος *Uncertainty Delta* καθώς επίσης μπορεί να συμπληρώσει και μερικές προαιρετικές μεταβλητές από το ‘*more options*’ (*π.χ. pi, delta max* και *trash max*). Στο συγκεκριμένο παράδειγμα, το *Anonymity Level*’ είναι 6 και το *Uncertainty Delta*: 1000. Τα αποτελέσματα του

NWA παρουσιάζονται στην παρακάτω εικόνα (στιγμιότυπο Α) και οι τροχιές απεικονίζονται στο χάρτη με πράσινο χρώμα και με 90% διαφάνεια καθώς επίσης, το πάχος γραμμής είναι 1 (το χρώμα της τροχιάς, ο βαθμός διαφάνειας και το πάχος της γραμμής μπορούν διαδραστικά να αλλάξουν). Από τις εικόνες 5.4 και 5.5, ο χρήστης μπορεί να συγκρίνει την παραποίηση που έχει υποστεί η βάση δεδομένων κίνησης μετά την εφαρμογή του αλγόριθμου *NWA*. Όπως ήταν αναμενόμενο, δημιουργήθηκαν κ-ανώνυμα δεδομένα δίπλα από τα αρχικά με σκοπό να προστατεύσει την ευαίσθητη πληροφορία χωρίς να αλλάξει δραματικά το σχήμα σε σχέση με το σχήμα των πραγματικών τροχιών.

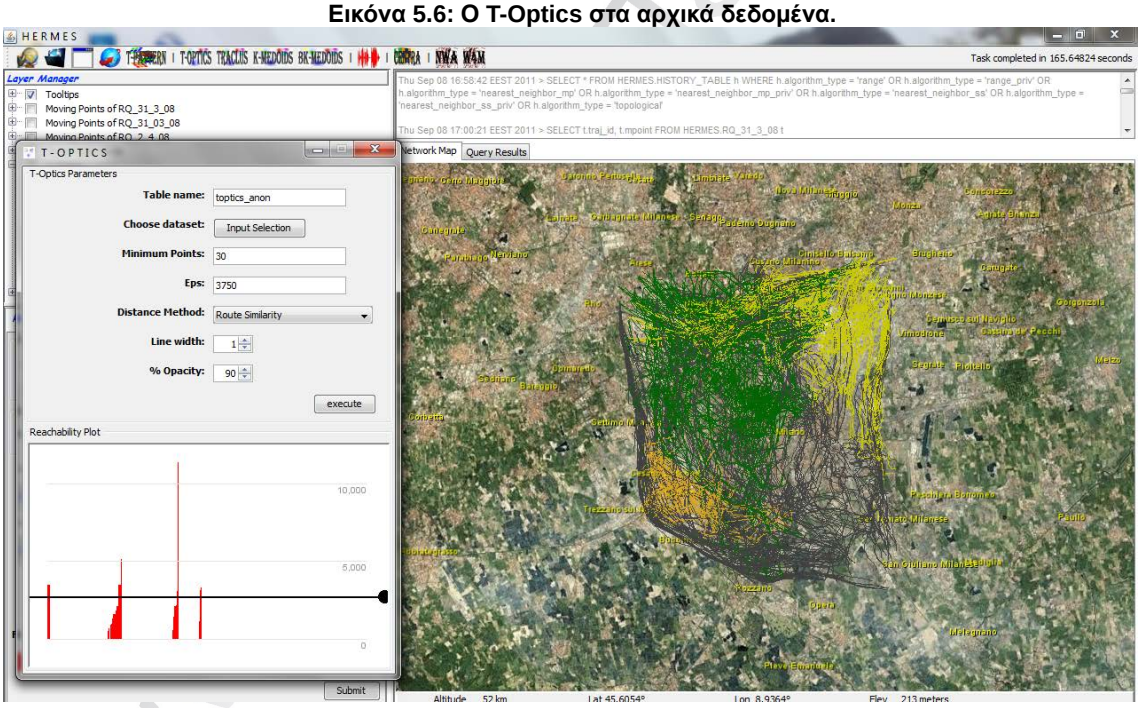


Εικόνα 5.5: Απεικόνιση ανώνυμων δεδομένων μετά την εκτέλεση του NWA.

Στις εικόνες 5.6 και 5.7, παρουσιάζονται τα αποτελέσματα από την εφαρμογή του *T-Optics* στα αρχικά και ανώνυμα δεδομένα, αντίστοιχα. Όσο αφορά την πρώτη εικόνα, ο *T-Optics* ανακάλυψε τέσσερις συστάδες, η πρώτη με το ανοικτό ροζ χρώμα, η δεύτερη με το μωβ, η τρίτη με το κόκκινο και η τέταρτη με το μπλε (ο θόρυβος έχει απενεργοποιηθεί από το *Layer Manager*) ενώ σχετικά με την δεύτερη εικόνα, ο αλγόριθμος διέκρινε τρεις συστάδες, η πρώτη με το κίτρινο χρώμα, η δεύτερη με το πράσινο και η τρίτη με το πορτοκαλί (ο θόρυβος απεικονίζεται με σκούρο γκρι χρώμα). Όπως είναι φανερό, οι συστάδες των ανώνυμων δεδομένων είναι πιο πυκνές από αυτές των πραγματικών δεδομένων.

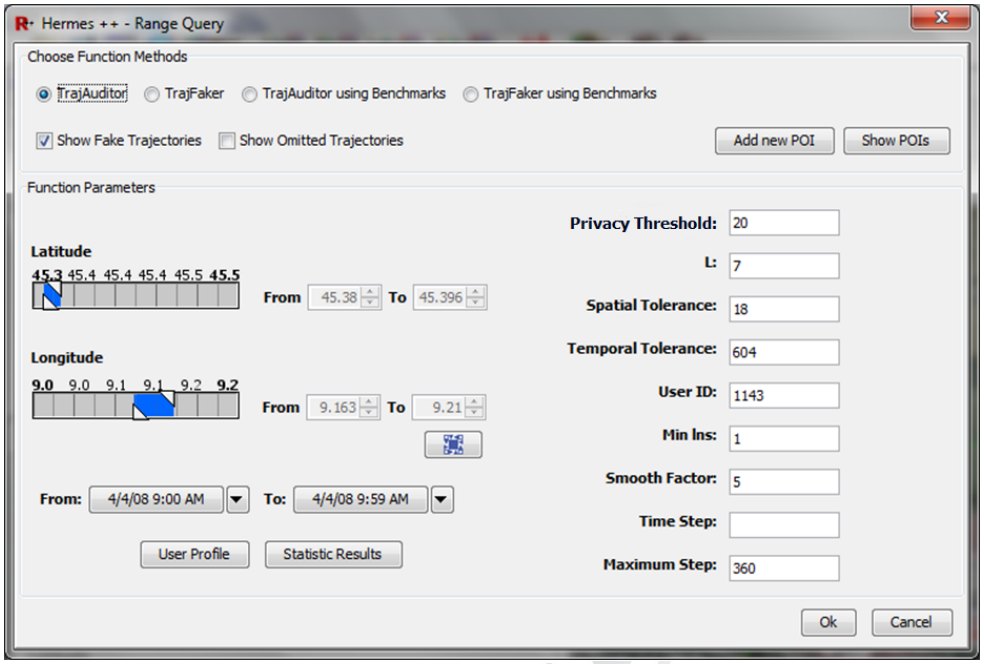


Εικόνα 5.6: Ο T-Optics στα αρχικά δεδομένα.



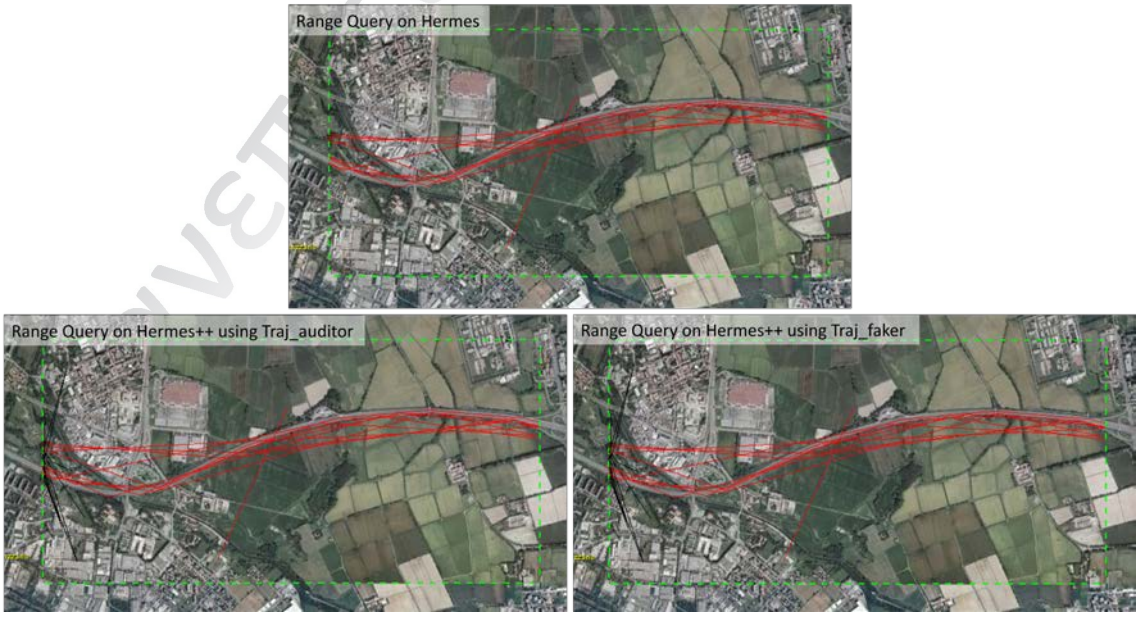
Εικόνα 5.7: Ο T-Optics στα ανώνυμα δεδομένα.

Ο μηχανισμός του *HERMES++* υποστηρίζει range, k-nearest neighbor και distance ερωτήματα πάνω σε χωρικά και χώρο χρονικά δεδομένα. Σε κάθε είδος ερωτημάτων, δύο αλγόριθμοι, ο Trajectory Auditor και ο Trajectory Faker εμπεριέχονται στην λειτουργικότητα του *HERMES++*. Ο Trajectory Auditor είναι η διαδικασία που ελέγχει τα ερωτήματα των χρηστών και διατηρεί την ανωνυμία των δεδομένων στις απαντήσεις των χρηστών ενώ ο Trajectory Faker είναι ο αλγόριθμος που παράγει πλασματικές τροχιές, οι οποίες ακολουθούν την τάση του συνόλου των πραγματικών τροχιών. Ο χρήστης μπορεί να εκτελέσει ένα ερώτημα του *HERMES++* πηγαίνοντας στην αντίστοιχη ετικέτα του ερωτήματος (π.χ. *Range Query*) και ενεργοποιώντας την επιλογή '*Privacy Settings*'. Οι παράμετροι του *Range Query++* παρουσιάζονται στην εικόνα 5.8 και εκτός από τις χώρο χρονικές μεταβλητές (όπως είναι οι αντίστοιχες του *Range Query*), ο χρήστης πρέπει να συμπληρώσει και κάποιες επιπλέον τιμές που αφορούν την προστασία των δεδομένων.

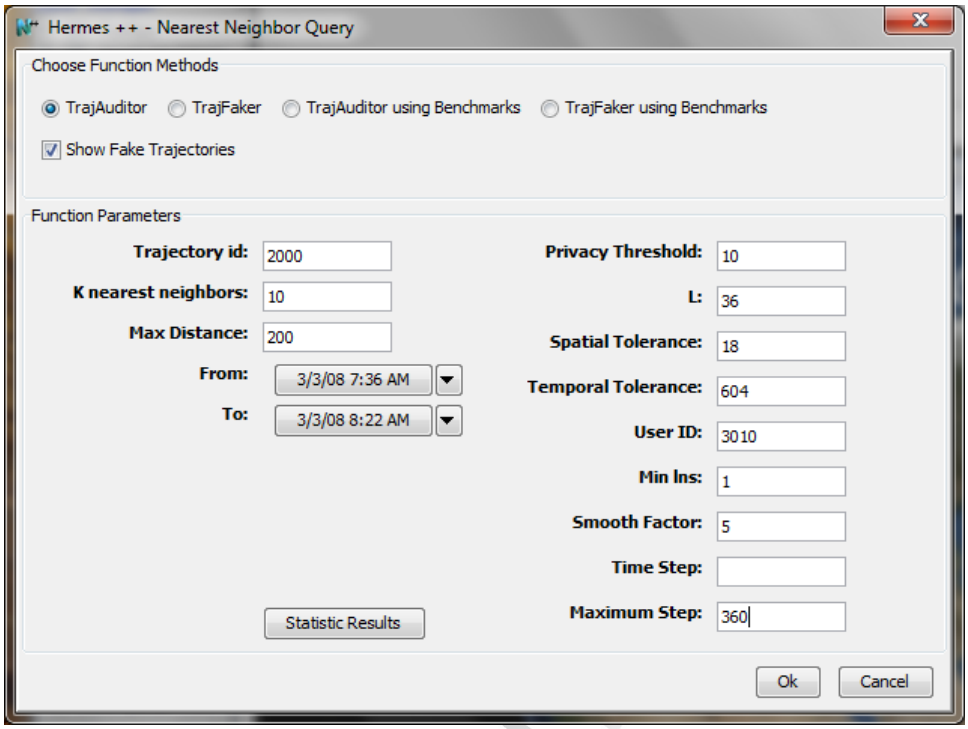


Εικόνα 5.8: Οι παράμετροι του Range Query++.

Για να γίνει καλύτερα κατανοητό, εκτελέστηκαν τρία παραδείγματα από τρεις διαφορετικούς μηχανισμούς με την χρήση ίδιων παραμέτρων με σκοπό να συγκριθούν τα αποτελέσματα μεταξύ τους. Συγκεκριμένα, (α) εφαρμόστηκε ένα *Range Query* σε μια πολύ μικρή περιοχή του Μιλάνο, (β) πραγματοποιήθηκε ένα *Range Query++* χρησιμοποιώντας τον αλγόριθμο *Trajectory Auditor* στην ίδια περιοχή και (γ) εκτελέστηκε ένα *Range Query++* χρησιμοποιώντας τον αλγόριθμο *Trajectory Faker*. Τα αντίστοιχα αποτελέσματα παρουσιάζονται στην εικόνα 5.9. Ο μηχανισμός του *HERMES++* τόσο στον αλγόριθμο *Trajectory Auditor* όσο και στον *Trajectory Faker* δημιούργησε οκτώ πλασματικές τροχιές (με το μαύρο χρώμα). Η εφαρμογή δίνει την επιλογή στον χρήστη να παρατηρήσει τις πλασματικές τροχιές που παράχθηκαν απεικονίζοντας τις τροχιές αυτές με διαφορετικό χρώμα, αλλιώς αν η επιλογή δεν έχει ενεργοποιηθεί αναπαριστώνται στο χάρτη με το ίδιο χρώμα των αληθινών τροχιών. Επίσης, αξίζει να σημειωθεί ότι ο *Trajectory Auditor* πραγματοποίησε μεγαλύτερες χρονικές καθυστερήσεις από τον *Trajectory Faker* λόγω του ελεγκτικού μηχανισμού που περιέχει.

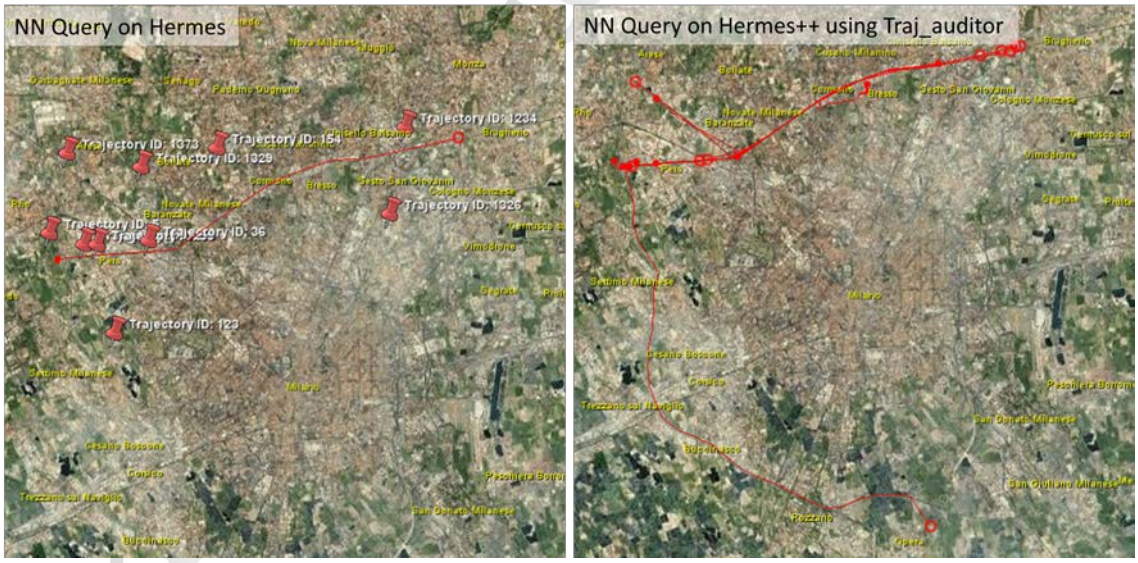


Εικόνα 5.9: Αναπαράσταση των αποτελεσμάτων σε τρεις διαφορετικές διεργασίες του Range Query.



Εικόνα 5.10: Οι παράμετροι του Nearest Neighbor Query++.

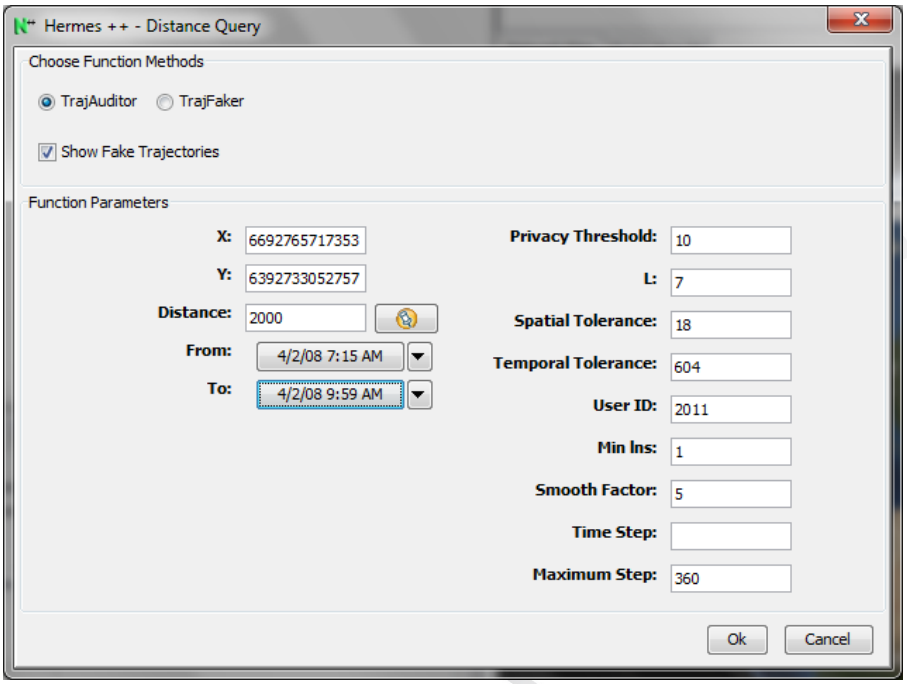
Ακολουθώντας την ίδια διαδικασία, εκτελέστηκε ένα *Nearest Neighbor Query* τόσο στο μηχανισμό του *HERMES* όσο και στον *HERMES++*. Τα αποτελέσματα φαίνονται στην παρακάτω εικόνα. Επίσης, στην εικόνα 5.10, παρουσιάζονται οι παράμετροι του *Nearest Neighbor Query* για τον *HERMES++*.



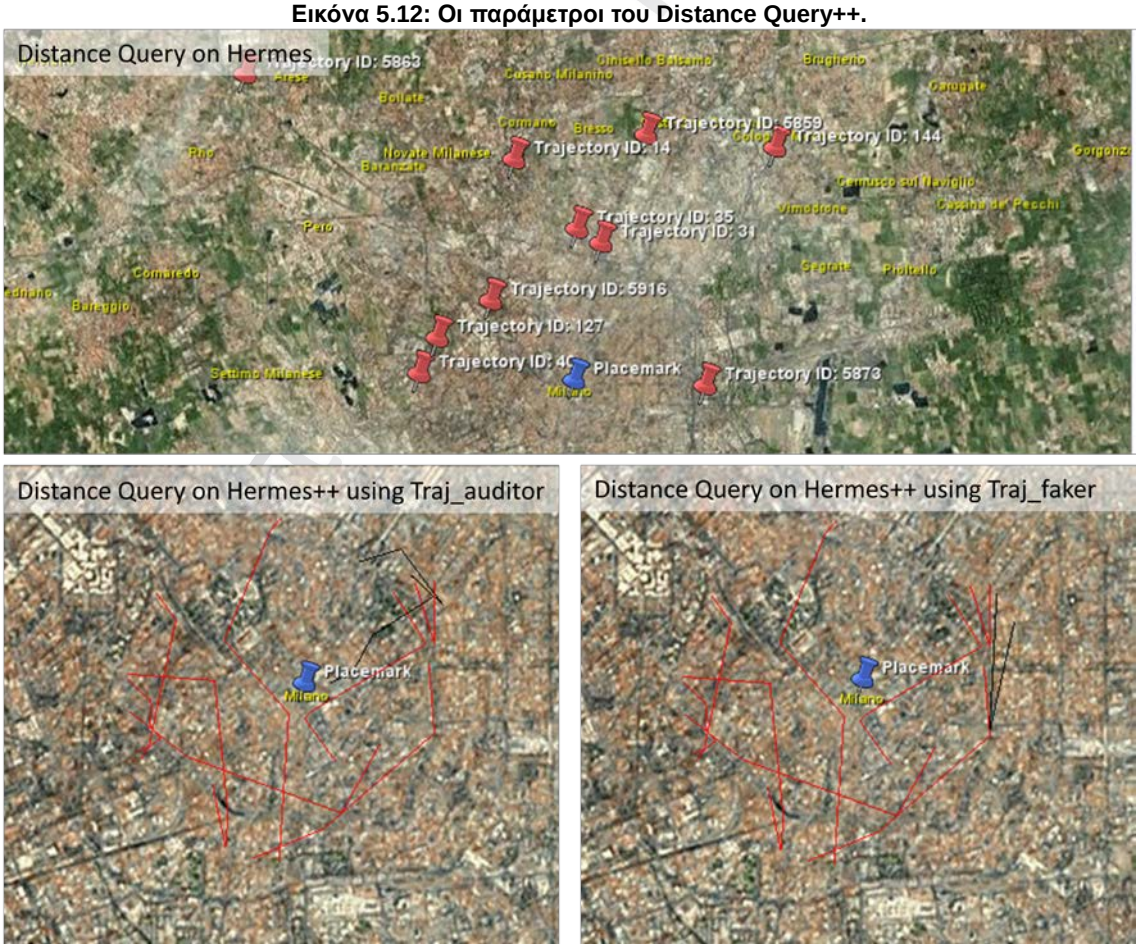
Εικόνα 5.11: Αναπαράσταση των αποτελεσμάτων του NN Query και του NN Query++.

Με τον ίδιο τρόπο, πραγματοποιήθηκε ένα *Distance Query* με τρεις διαφορετικούς τρόπους. Πρώτον, επιλέχτηκε το '*Static Spatial – Moving Point*' από το '*Choose Query*' της ετικέτας *NN Query* και συμπληρώθηκαν τα πεδία του *Distance Query* του *HERMES*. Συγκεκριμένα, συμπληρώθηκε η τιμή 10 στο πεδίο '*k nearest neighbor*' και με το εργαλείο  , επιλέχτηκε πάνω στο χάρτη το κέντρο της πόλης. Επίσης, η χρονική διάρκεια που ορίστηκε ήταν από τις 2/4/2008 7:15πμ μέχρι τις 2/4/2008 9:59πμ. Μετά την πραγματοποίηση του ερωτήματος, η βάση δεδομένων κίνησης επέστρεψε τις 10 κοντινότερες τροχιές που υπήρχαν γύρω από το κέντρο του Μιλάνο στο συγκεκριμένο χρονικό διάστημα. Δεύτερον, εκτελέστηκε με τις ίδιες παραμέτρους το *Distance Query++* για τον αλγόριθμο *Trajectory Auditor*. Στην εικόνα 5.12, ο χρήστης συμπληρώνει επιπλέων στοιχεία που αφορούν την προστασία της προσωπικής πληροφο-

ρία έτσι ώστε να πραγματοποιηθεί το ερώτημα. Τρίτον, ακολουθώντας την ίδια διαδικασία, εφαρμόστηκε ο *Distance Query++* για τον αλγόριθμο *Trajectory Faker*.

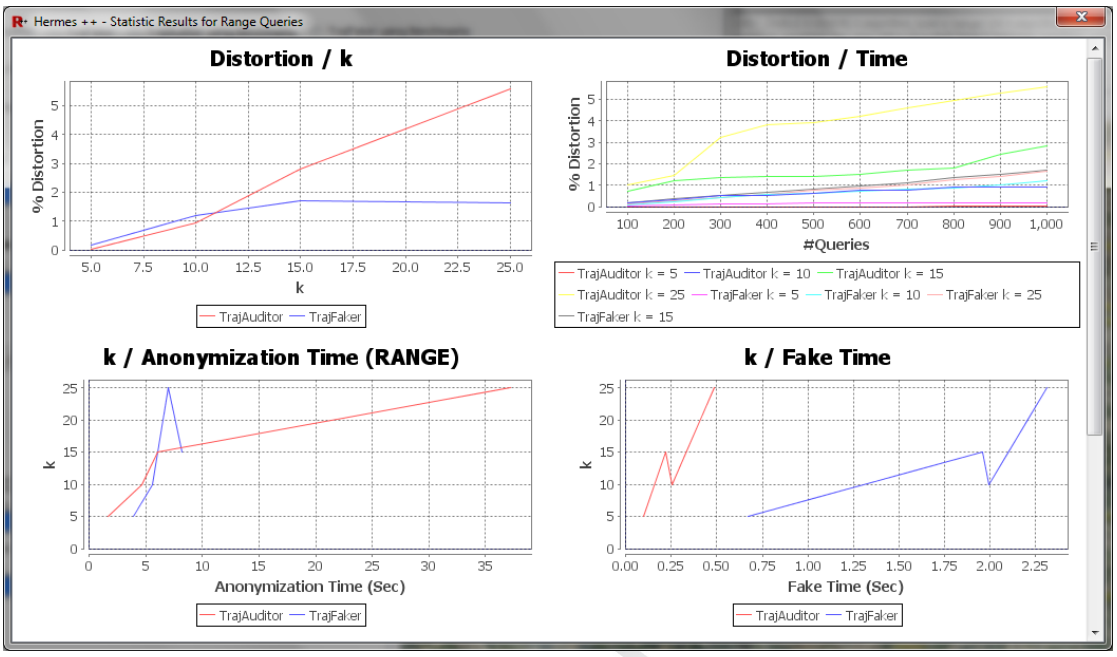


Εικόνα 5.12: Οι παράμετροι του Distance Query++.

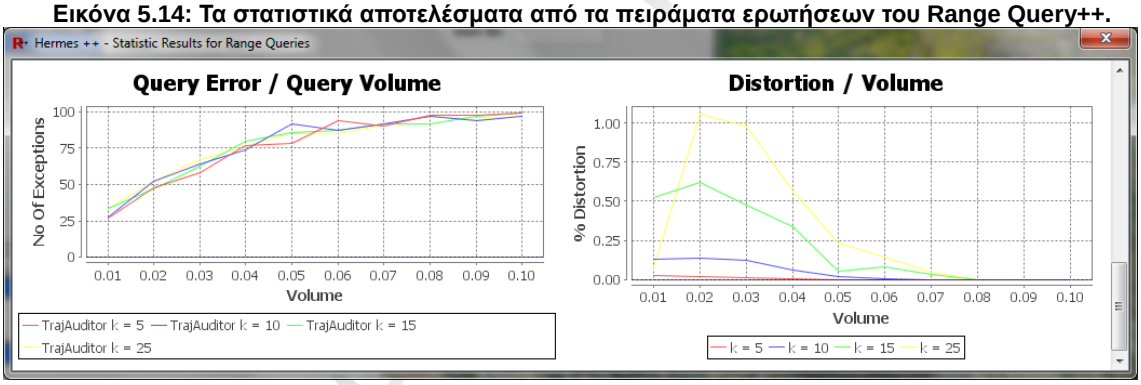


Εικόνα 5.13: Αναπαράσταση των αποτελεσμάτων σε τρεις διαφορετικές διεργασίες του Distance Query.

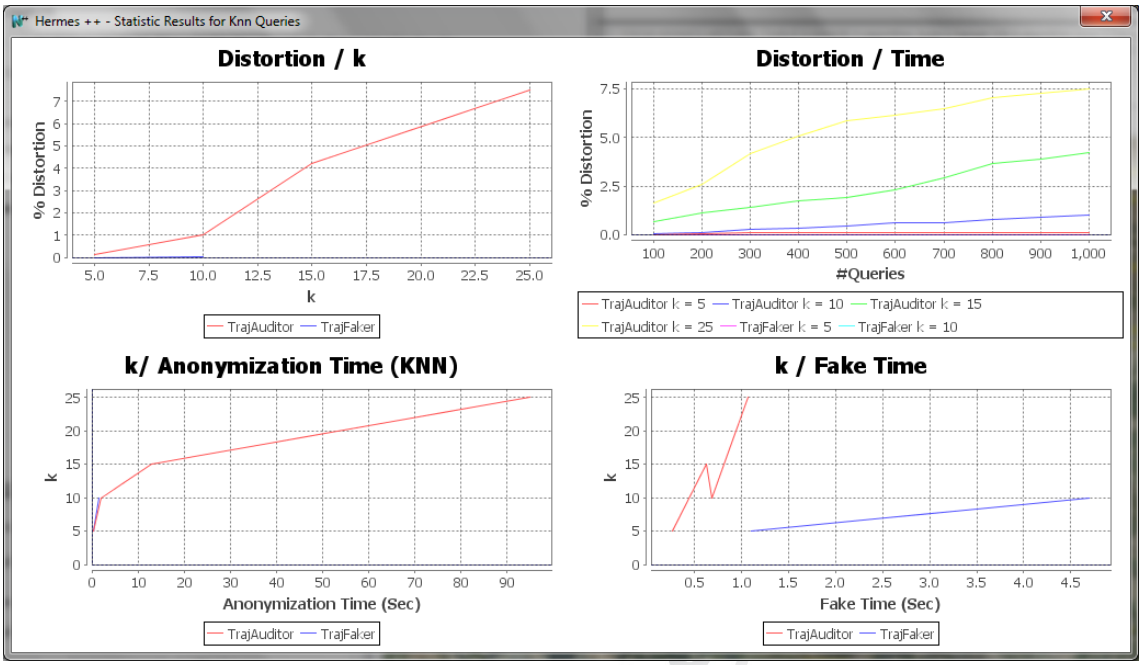
Τα οπτικά αποτελέσματα παρουσιάζονται στην εικόνα 5.13 και αν και ο μηχανισμός του *Trajectory Auditor* και του *Trajectory Faker* του HERMES++ δημιούργησαν μια πλασματική τροχιά (με το μαύρο χρώμα), η τροχιά αυτή ακολούθησε διαφορετική πορεία σε κάθε περίπτωση.



Εικόνα 5.14: Τα στατιστικά αποτελέσματα από τα πειράματα ερωτήσεων του Range Query++.



Εικόνα 5.15: Τα στατιστικά αποτελέσματα από τα πειράματα ερωτήσεων του Range Query++.



Εικόνα 5.16: Τα στατιστικά αποτελέσματα από τα πειράματα ερωτήσεων του NN Query++.

Τελειώνοντας, ο *HERMES++* δίνει την δυνατότητα να τρέχει ομάδες ερωτήσεων είτε από τον μηχανισμό Trajectory Auditor ή από τον μηχανισμό Trajectory Faker. Οι ομάδες ερωτήσεων είναι πολύτιμες για πειραματικές αξιολογήσεις πάνω στους τύπους των ερωτήσεων (π.χ. το ποσοστό της παραμόρφωσης των δεδομένων σε σχέση με το k-anonymity, ο εκτιμώμενος χρόνος ανάλογα με το k-anonymity, το ποσοστό λάθους ως προς το συνολικό αριθμό των ερωτήσεων, κτλ). Η εφαρμογή της μεταπτυχιακής εργασίας δίνει την δυνατότητα στον χρήστη να παρατηρήσει και να εκτιμήσει τα στατιστικά αποτελέσματα των πειραμάτων μέσω των γραφικών αναπαραστάσεων (π.χ. ιστογράμματα) για κάθε τύπο των ερωτήσεων. Δηλαδή, στην εικόνα 5.14, 5.15 και 5.16, παρουσιάζονται τα ιστογράμματα των πειραμάτων για τις ερωτήσεις *Range Query++* και *Nearest Neighbor Query++*, αντίστοιχα.

5.5. ΣΥΝΟΨΗ

Η εξέλιξη των γεωγραφικών πληροφοριακών συστημάτων έχει δώσει την ικανότητα στην αύξηση της συλλογής και αποθήκευσης των κινούμενων δεδομένων σε ειδικές βάσεις δεδομένων. Από την μία μεριά, η εξόρυξη των δεδομένων αυτών έχει προσφέρει σημαντικό κίνητρο την τελευταία δεκαετία για την ανάδειξη προτύπων με σκοπό να βελτιώσουν τις υπηρεσίες βασισμένες στην συμπεριφορά των χρηστών ή των πελατών, από την άλλη μεριά, δημοσιεύοντας την γεωγραφική πληροφορία στους τελικούς χρήστες είτε αυτή προέρχεται από αλγόριθμους εξόρυξης ή από απλές τροχιές μπορεί να παραβιάσει την προστασία των δεδομένων. Ο σκοπός του κεφαλαίου ήταν να παρουσιάσει μια πλατφόρμα που θα εφαρμόζει τεχνικές εξόρυξης γνώσης σε δεδομένα κίνησης διατηρώντας την ανωνυμία των δεδομένων. Συγκεκριμένα, ο στόχος της πλατφόρμας είναι διττός. Πρώτον, παρέχει μηχανισμούς ερωτημάτων τόσο στην διαχείριση των δεδομένων κίνησης όσο και στην προστασία των αποτελεσμάτων από τους κακόβουλους χρήστες. Δεύτερον, ενσωματώνει αλγόριθμους ανωνυμίας που αξιολογούνται από διάφορες τεχνικές εξόρυξης γνώσης με σκοπό τα πρότυπα να μην παρουσιάζουν προσωπική πληροφορία στους χρήστες. Μέσω της επίδοσης των πειραματικών αξιολογήσεων από τις γραφικές παραστάσεις, ο χρήστης μπορεί να εκτιμήσει την αποτελεσματικότητα των μηχανισμών προστασίας δεδομένων και την παραμόρφωση των δεδομένων από τις πλασματικές τροχιές που δημιουργούνται.

6. ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

6.1. ΕΙΣΑΓΩΓΗ

Το πρόγραμμα είναι μια γραφική διεπαφή χρήστη με δυνατότητες τρισδιάστατων απεικονίσεων, το οποίο υλοποιήθηκε στην γλώσσα προγραμματισμού *Java* και βασίστηκε στην βιβλιοθήκη του *SWING* για την ανάπτυξη των διαδραστικών παραθύρων. Τα αποτελέσματα από τις εκτελεσμένες διεργασίες και η βασική υποστήριξη του προγράμματος οπτικοποιήθηκαν σε ένα τρισδιάστατο παγκόσμιο χάρτη το οποίο προβάλλεται από την βιβλιοθήκη της *NASA World Wind* [3]. Το *World Wind* είναι μια 'open source' εφαρμογή που απεικονίζει μια εικονική γη υλοποιημένη εξ ολοκλήρου σε Java έτσι ώστε να ενσωματώνεται σε πολλαπλές πλατφόρμες και να είναι εύκολη στην χρήση. Εξαιτίας αυτού, το API του *World Wind* μπορεί να επεκταθεί και να γίνει μια δυναμική πλατφόρμα που θα προσφέρει σε οποιαδήποτε εφαρμογή την έννοια της διαχείρισης και ανάλυσης γεωγραφικών δεδομένων. Μερικά παραδείγματα χρήσεων πάνω στο *World Wind* είναι η παρακολούθηση καιρικών φαινομένων, η αναπαράσταση πόλεων ή εδαφών, οι κινήσεις των αεροπλάνων, πλοίων ή οχημάτων, η μόρφωση μαθητών για την γη και άλλα πολλά [17][16] (*http://goworldwind.org/demos/*). Για την παρουσίαση των στατιστικών γράφων επιδεικνύοντας την απόδοση των αποτελεσμάτων των αλγόριθμων, χρησιμοποιήθηκε η βιβλιοθήκη *JFreechart*. Η βιβλιοθήκη *JFreechart* είναι κατάλληλη στην ανάπτυξη μιας μεγάλης ποικιλίας γραφικών παραστάσεων όπως πίτες, μπάρες, ιστογράμματα αλλά και πιο πολύπλοκων στατιστικών αναπαραστάσεων. Το *RadioCkeckTree*, το οποίο αποτελεί μέρος της επαύξησης των συστατικών του *SWING* [2], χρησιμοποιήθηκε για την κατασκευή του γραφικού περιβάλλοντος του χρήστη στην αναζήτηση και εξαγωγή των δεδομένων. Συγκεκριμένα, αυτό το εργαλείο είναι ένα στοιχείο το οποίο υποστηρίζει μονή ή / και πολλαπλές επιλογές με την ανάμειξη των αντικειμένων *JRadioButton* και *JCheckBox*. Το επαυξημένο συστατικό επιτρέπει επίσης την αυτόματη αποδέσμευση επιλογή ενός γονέα του δέντρου όταν όλες οι επιλογές των παιδιών του είναι απενεργοποιημένες και το αντίστροφο (δηλαδή, αυτόματης αποδέσμευσης επιλογών των παιδιών όταν η επιλογή του γονέα είναι απενεργοποιημένη). Για την υλοποίηση του επαυξημένου εργαλείου, διάφορες επεκτάσεις αντικειμένων και κλάσεων έχουν προστεθεί στην βασική βιβλιοθήκη του *SWING*.

Επιπλέον, η διεπαφή συνδέεται στην βάση δεδομένων της *Oracle 11g release 2* (*http://www.oracle.com/technetwork/database/enterprise-edition/overview/index.html*) χρησιμοποιώντας το JDBC έτσι ώστε να έχει πρόσβαση στην διαχείριση των δεδομένων. Η *Oracle 11g release 2* είναι μια δυναμική βάση δεδομένων που υποστηρίζει πολλούς τύπους δεδομένων καθώς επίσης δίνει την δυνατότητα στον προγραμματιστή να επαυξήσει τα αντικείμενά της σε νέα και πιο πολύπλοκα χρησιμοποιώντας την γλώσσα *PL/SQL*. Επίσης, έχει πλήρης συνδεσιμότητα με πολλές γλώσσες υψηλού επιπέδου και κυρίως με την *Java*. Η *Oracle 11g release 2* είναι ικανή στην αποθήκευση μεγάλων όγκων δεδομένων και στην διαχείριση των δεδομένων αυτών έχοντας υψηλή απόδοση. Για την υποστήριξη μηχανισμών ανάλυσης πάνω στα κινούμενα δεδομένα, ενσωματώθηκε ο *HERMES*. Ο *HERMES* έχει αναπτυχθεί ως μια επέκταση στην Oracle10g για να υποστηρίζει είτε χρονικές είτε χωρικές λειτουργίες αλλά η κυριότερη λειτουργία του είναι η μοντελοποίηση, διαχείριση και ανάλυση βάσεων δεδομένων από αντικείμενα που κινούνται και μεταβάλλουν σχήμα ή θέση, στο χώρο και στο χρόνο, σε συνεχόμενα ή σε διακριτά βήματα. Το κύριο συστατικό του είναι το *Hermes Moving Data Cartridge* (*Hermes-MDC*) και διαθέτει ένα σύνολο από γεωμετρικά σχήματα που περιέχουν μέσα το χρόνο και δίνει την δυνατότητα στον τελικό χρήστη μέσα από διάφορες μεθόδους και συναρτήσεις να αναλύει και να επεξεργάζεται εύκολα και γρήγορα βάσεις δεδομένων κίνησης. Πάνω στο *HERMES*, έχει εισαχθεί και ο *HERMES++*, το οποίο είναι μια επαυξημένη έκδοση του *HERMES* με σκοπό να υποστηρίζει την λειτουργικότητα αποθήκευσης δεδομένων κίνησης σε μια βάση δεδομένων και το μηχανισμό χώρο χρονικών ερωτημάτων του *Hermes* διατηρώντας όμως την προσωπική πληροφορία των ατόμων. Συγκεκριμένα, ο *Hermes++* χρησιμοποιεί τις λειτουργίες του *HERMES* για να αποθηκεύει πραγματικές ή εικονικές τροχιές (οι οποίες δημιουργούνται από το μηχανισμό του *HERMES++*) καθώς επίσης, διατηρεί οποιαδήποτε ιστορική πληροφορία των ερωτημάτων που θέτουν οι χρήστες ώστε να χρησιμοποιηθούν στην αποτροπή διάφορων ειδών επιθέσεων από τους κακόβουλους χρήστες.

Όσο αφορά τις λειτουργίες της πλατφόρμας της μεταπτυχιακής διατριβής, είναι υλοποιημένες με τρεις διαφορετικούς τρόπους, (α) σε εκτελέσιμα αρχεία, (β) σε SQL συναρτήσεις και (γ) σε κλάσεις της Java. Στην πρώτη περίπτωση, το πρόγραμμα τρέχει τα εκτελέσιμα αρχεία με την εφαρμογή κλήσης του συστήματος. Δηλαδή, η εφαρμογή κάνει μια εξωτερική κλήση στα εκτελέσιμα αρχεία, τα οποία είναι υλοποιη-

μένα στην συμβατική γλώσσα C, τα αρχεία αυτά εκτελούνται και τέλος, αποθηκεύουν το αποτέλεσμα σε ειδικά αρχεία κειμένου. Κατόπιν, διαβάζονται από το πρόγραμμα και παρουσιάζονται στον τρισδιάστατο χάρτη της *World Wind*. Στην δεύτερη περίπτωση, το πρόγραμμα καλεί συναρτήσεις SQL μέσω του JDBC για την εκτέλεση των λειτουργιών που είναι υλοποιημένες σε PL/SQL και είναι ενσωματωμένες πάνω στην *Oracle*. Στην τρίτη περίπτωση, η εφαρμογή εκτελεί τις κλάσεις Java των αλγόριθμων άμεσα από το Virtual Machine της Java.

6.2. ΤΕΧΝΙΚΕΣ ΛΕΠΤΟΜΕΡΕΙΕΣ

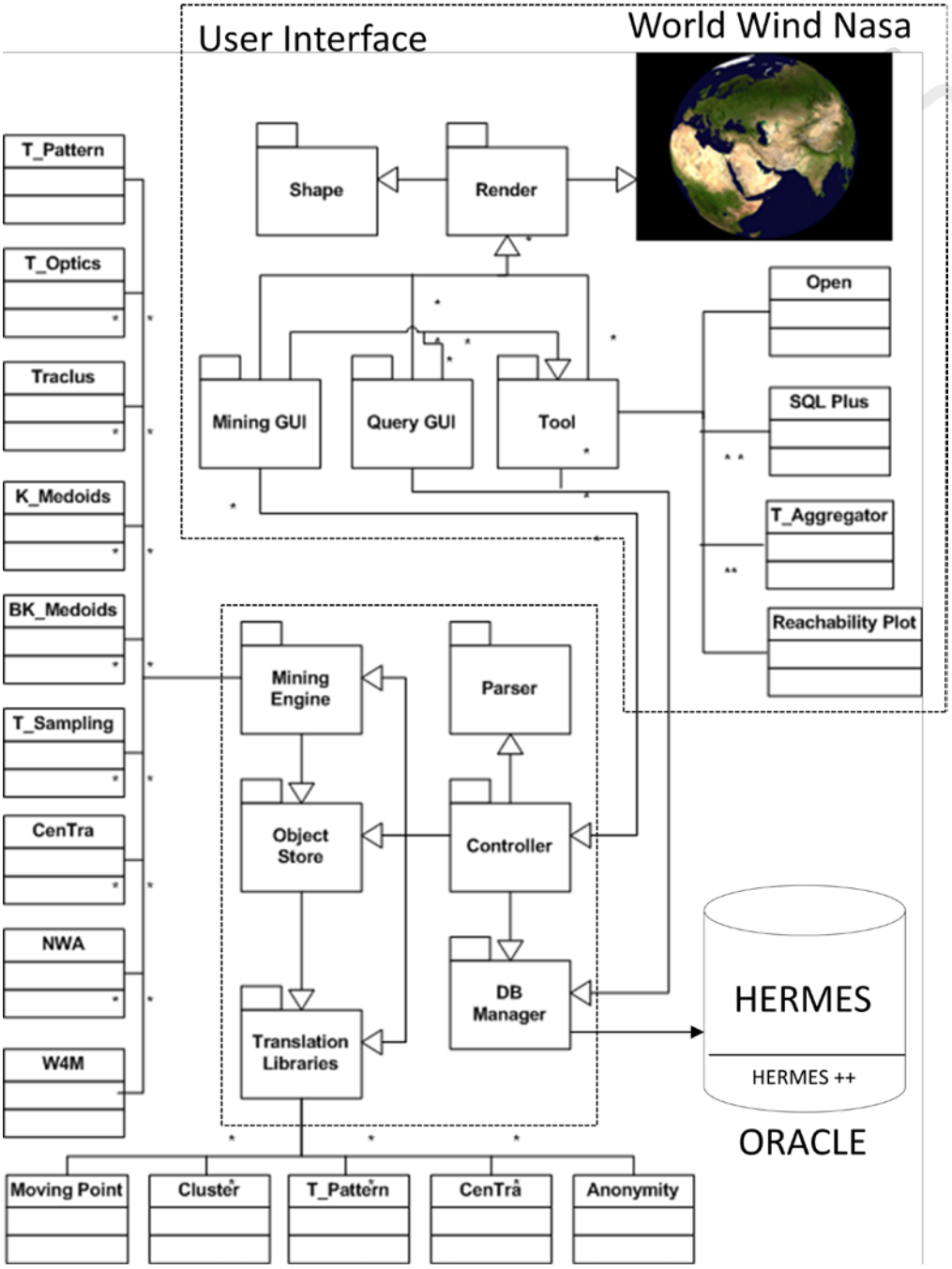
Σε αυτό το υποκεφάλαιο, θα περιγραφεί τεχνικά το διάγραμμα των βασικών κλάσεων που υλοποιήθηκαν για την ανάπτυξη του συστήματος. Πρώτον, θα αναλυθεί με περισσότερη λεπτομέρεια η αρχιτεκτονική του *DAEDALUS* και πώς βασίστηκε το σύστημα σε σχέση με αυτό καθώς επίσης, θα συζητηθούν οι επεκτάσεις της πλατφόρμας πάνω στο *DAEDALUS*. Δεύτερον, θα αναφερθεί η διαδικασία κλήσης των κλάσεων του συστήματος από την στιγμή που ο χρήστης εκτελεί ένα ερώτημα μέχρι το αποτέλεσμα να οπτικοποιηθεί στον χάρτη. Τρίτον, θα περιγραφεί η διαδικασία εισαγωγής ενός νέου αλγόριθμου στον παρόν σύστημα δηλαδή ποια είναι τα τεχνικά βήματα που πρέπει να ακολουθήσει ο προγραμματιστής για να εντάξει ένα νέο αλγόριθμο εξόρυξης γνώσης στην πλατφόρμα έτσι ώστε να μπορεί να χρησιμοποιείται αργότερα από τον τελικό χρήστη.

- ***Περιγραφή των διαγραμμάτων κλάσεων του συστήματος***

Όπως είχε αναφερθεί στα πρώτα κεφάλαια, το σύστημα χτίστηκε πάνω την αρχιτεκτονική του *DAEDALUS*, κάνοντας χρήση την γλώσσα *MO-DMQL** αλλά επιπλέον ενσωμάτωσε νέους αλγόριθμους εξόρυξης γνώσης και ανέπτυξε μια δικιά της διεπαφή χρήστη υπολογιστή. Καταρχήν, ο βασικός μηχανισμός είναι ο *Controller*. Ουσιαστικά, ο *Controller* είναι η κεντρική μονάδα για την εκτέλεση των αλγορίθμων εξόρυξης γνώσης και είναι υπεύθυνο για την διεκπεραίωση των ερωτημάτων από τους αλγόριθμους αυτούς, τον συντονισμό για τα καθήκοντα που εκτελούνται από τα άλλα συστατικά που αφορούν το κομμάτι της εξόρυξης γνώσης και την διαχείριση της επικοινωνίας με την βάση δεδομένων κίνησης. Η δήλωση των παραμέτρων των αλγόριθμων επικυρώνεται από τον *Parser*, που στη συνέχεια τις μετατρέπει σε μια Αφηρημένη Γλώσσα Ερωτημάτων *Object Query Language* (*OQL*), η οποία είναι χτισμένη για την απάντηση της δήλωσης. Ο *DB Manager* παρέχει κεντρική πρόσβαση στο στρώμα των δεδομένων. Βασικά, αλληλεπιδρά με τις αποθηκευμένες διεργασίες του *HERMES*, προκειμένου να εκτελεί χώροχρονικά ερωτήματα αλλά και άλλα πιο σύνθετα. Με την εκτέλεση αυτών των ερωτημάτων πραγματοποιούνται πολύπλοκοι μετασχηματισμοί από σχεσιακά αντικείμενα του *HERMES* σε αντικείμενα του επιπέδου της εφαρμογής και το αντίστροφο. Αυτό μπορεί να γίνει με την εκμετάλλευση του πακέτου *Translation Libraries* που μετατρέπει μια αναπαράσταση σε μια άλλη (π.χ. από *Oracle* σε *Java* και από *Java* σε *Oracle*). Συγκεκριμένα, το πακέτο *Translation Libraries* περιέχει τις κλάσεις *Moving_point* (μετατροπή κινούμενου δεδομένου-τροχιά από αντικείμενο της *Java* σε αντικείμενο της *Oracle* και αντίστροφα), *Cluster* (μετατροπή μιας συστάδας από αντικείμενο της *Java* σε αντικείμενο της *Oracle* και αντίστροφα), *T_Pattern* (μετατροπή ενός προτύπου από αντικείμενο της Java σε αντικείμενο της Oracle και αντίστροφα), *CenTra* (μετατροπή ενός αντικειμένου του *Tr-FCM, Centr-I-FCM, CenTra* ή *TX-CenTra* από αντικείμενο της *Java* σε αντικείμενο της *Oracle* και αντίστροφα) και *Anonymity* (μετατροπή ενός αντικειμένου των αλγορίθμων *NWA* ή *W4M* από αντικείμενο της *Java* σε αντικείμενο της *Oracle* και αντίστροφα). Το *Object Store* είναι ένα αντικείμενο που ενεργεί στο να αυξήσει την απόδοση των αλγορίθμων εξόρυξης γνώσης παράγοντας αρχεία .obj για την διαχείριση των δεδομένων εισαγωγής και εξαγωγής του κάθε αλγόριθμου. Επίσης, ο *Mining Engine* ενεργοποιείται από τον *Controller* για την εκτέλεση των αλγορίθμων εξόρυξης γνώσης. Αναλυτικότερα, ο *Mining Engine* περιέχει τις κλάσεις *T_Pattern, T_Optics, Traclus, K_Medoids, BK_Medoids, T_Sampling, CenTra, NWA* και *W4M* για την εκτέλεση των αντίστοιχων αλγορίθμων. Αξίζει να σημειωθεί ότι για την κλήση των αλγορίθμων, χρησιμοποιεί έναν από τους τρεις παρακάτω τρόπους, (α) σε εκτελέσιμα αρχεία, (β) σε *SQL* συναρτήσεις και (γ) σε κλάσεις της *Java*. Τα πρότυπα που δημιουργούνται από τον κάθε αλγόριθμο περνάνε από τον *Object Store* και στην

* Η έκδοση του προγράμματος του *DAEDALUS* που ενσωματώθηκε στον σύστημα ήταν σε στάδιο εξέλιξης. Η γλώσσα MO-DMQL της παρούσας πλατφόρμας δεν είναι τόσο εξελιγμένη στο να υποστηρίζει πολύπλοκες ερωτήσεις όπως έχει προταθεί στην βιβλιογραφία [36][37].

συνέχεια μετατρέπονται σε μια αντικείμενο-σχεσιακή παρουσίαση για να αποθηκευτούν στην βάση. Βέβαια, αυτό μπορεί να γίνει από τον *Controller* αξιοποιώντας την βιβλιοθήκη *Translation Libraries* και αποθηκεύοντας τα αποτελέσματα στην βάση δεδομένων του *HERMES* μέσω του *DB Manager*. Το πακέτο *Mining GUI* περιέχει τα διαδραστικά παράθυρα των κλάσεων των αλγορίθμων εξόρυξης γνώσης που εμφανίζονται στον τελικό χρήστη για την συμπλήρωση των παραμέτρων και την εκτέλεση των αλγορίθμων. Παρομοίως, το *πακέτο Query GUI* περιέχει τα διαδραστικά παράθυρα των κλάσεων των απλών ε



Εικόνα 6.1: Το διάγραμμα κλάσεων του συστήματος.

ρωτημάτων του *HERMES* όπου ο τελικός χρήστης συμπληρώνει τις παραμέτρους και στην συνέχεια, εκτελεί τα ερωτήματα. Το πακέτο *Tool* περιλαμβάνει διαδραστικά εργαλεία που συμβάλουν στην καλύτερη ανάλυση των δεδομένων κίνησης. Συγκεκριμένα, το *Tool* παρέχει τις κλάσεις *Open* (ανοίγει 'οπτικοποιεί' τα αποθηκευμένα αποτελέσματα των ερωτημάτων από τον *HERMES* στον τρισδιάστατο χάρτη), *SQL Plus* (τρέχει *SQL* ερωτήματα για την εκτέλεση των ερωτημάτων), *T_Aggregator* (αλγόριθμος συνάθροισης των κινούμενων δεδομένων ως βελάκια) και *Reachability Plot* (διαδραστικό ιστόγραμμα που

χρησιμοποιείται στον *T-Optics*). Το πακέτο *Mining GUI* συνδέεται με τον *Controller* για να εκτελεστεί κάποιος αλγόριθμος εξόρυξης γνώσης, αντιθέτως το *Query GUI* και το *Tool* επικοινωνούν απευθείας με τον *DB Manager* για την πραγματοποίηση των διεργασιών τους. Το πακέτο *Render* είναι ο μηχανισμός που αναλαμβάνει την οπτικοποίηση των αποτελεσμάτων των ερωτημάτων αναζήτησης και εξόρυξης στον χάρτη. Μερικές από τις κλάσεις που περιέχει είναι ο *TrajectoryRenderer* (μηχανισμός οπτικής αναπαράστασης κινούμενου δεδομένου), *AnonymityRenderer* (μηχανισμός οπτικής αναπαράστασης των ανώνυμων δεδομένων του *NWA* ή *W4M*), *PatternRenderer* (μηχανισμός οπτικής αναπαράστασης προτύπου του *T-Pattern*), κτλ. Το πακέτο *Shape* καλείται από το *Render* σε περιπτώσεις που ένα οπτικό σχήμα δεν είναι 'native' (δηλαδή να μην υπάρχει στην βιβλιοθήκη της *World Wind Nasa*) αλλά να είναι 'custom' (υλοποιημένο από τον προγραμματιστή). Ένα τέτοιο αντικείμενο περιέχεται στην κλάση Trajectory, το οποίο εκτός από ένα σχήμα 'polyline', σε κάθε σημείο διαθέτει ένα χρονικό στίγμα. Το πακέτο *Render* καλείται από τα πακέτα *Mining GUI*, *Query GUI* και *Tool* και μετέπειτα, οπτικοποιεί τα αποτελέσματα στον τρισδιάστατο χάρτη της *World Wind Nasa*. Συνοψίζοντας, τα πακέτα *Mining GUI*, *Query GUI*, *Tool*, *Render*, *Shape* και *World Wind Nasa* ανήκουν στο μηχανισμό *User Interface* (βλέπε εικόνα 6.1).

- ***Διαδικασία κλήσεων των κλάσεων του συστήματος***

Σε αυτήν την παράγραφο, θα περιγραφούν αναλυτικά τα βήματα των κλήσεων του συστήματος με την εκτέλεση ενός αλγόριθμου εξόρυξης γνώσης. Ξεκινώντας, έστω ότι θέλουμε να εκτελέσουμε ένα αλγόριθμο *T-Pattern*. Αφού η διαδικασία ξεκινήσει, ο *T-Pattern* επικοινωνεί με τον *Controller* μέσω της εντολής:

```
controller.executeCommand("SELECT t.object FROM MINE(HERMES." +
jTextFieldTable.getText() + ";TAS ; " + jTextFieldMinSup.getText() +
" ; " + jTextFieldTau.getText() + " ; " + jTextFieldSide.getText() +
") t ", mpoints);
```

Όπως παρατηρούμε από τον παραπάνω κώδικα, η μέθοδος 'executeCommand' λαμβάνει δύο παραμέτρους. Η πρώτη παράμετρος είναι ένα *STRING*, το οποίο είναι ένα ερώτημα *MO-DMQL*. Το ερώτημα *MO-DMQL* περιέχει μέσα στην *SELECT*, το αντικείμενο που θα επιστραφεί και μέσα στην παρένθεση, ο αλγόριθμος που θα εκτελεστεί (*TAS*) και οι αντίστοιχοι παράμετροί του. Η δεύτερη παράμετρος είναι ένας δυναμικός πίνακας (*mpoints*) που διαθέτει τα κινούμενα δεδομένα που θα συμπεριληφθούν ως είσοδο στον αλγόριθμο. Στην συνέχεια, ο *Controller* καλεί τον Parser μέσω την εντολής:

```
parser.parse(query, input_data);
```

Ο Parser επικυρώνει την δήλωση των παραμέτρων αποθηκεύοντας τους παραμέτρους του αλγόριθμου σε ένα πίνακα και αφού ολοκληρωθεί η διαδικασία αυτή, ο *Controller* επικοινωνεί με τον *Mining Engine* όπου με την σειρά του καλεί την μέθοδο:

```
mOp = factory.createMiningOperator(Params.get(1).toString());
```

Αυτή η μέθοδος αναλαμβάνει να αναγνωρίσει τον αλγόριθμο που θα πραγματοποιηθεί και δημιουργεί ένα instance για τον αλγόριθμο αυτό με σκοπό να προετοιμάσει την διαδικασία της εκτέλεσης. Συγκεκριμένα, η μέθοδος περιέχει τον εξής πηγαίο κώδικα:

```
1. public GenericMiningOperator createMiningOperator(String type)
2.  throws Exception {
3.        // instantiate the mining operator
4.        try {
5.         if (type.trim().compareTo("TAS") == 0) {
6.             return new MiningOperator_TAS();
7.         } else if (type.trim().compareTo("CLUSTER") == 0) {
8.             return new MiningOperator_Cluster();
9.         } else if (type.trim().compareTo("TRACLUS") == 0) {
10.            return new MiningOperator_TRACLUS();
```

```
11.       }else if (type.trim().compareTo("KMEDOIDS") == 0) {
12.           return new MiningOperator_KMedoids();
13.       }else if(type.trim().compareTo("BISECTINGKMEDOIDS") == 0){
14.           return new MiningOperator_BisectingKMedoids();
15.       } else if (type.trim().compareTo("UNCERTAINTY") == 0) {
16.           return new MiningOperator_Uncertainty();
17.       } else if (type.trim().compareTo("TSAMPLING") == 0) {
18.        return new MiningOperator_TSampling();
19.        } else if (type.trim().compareTo("NWA") == 0) {
20.           return new MiningOperator_NWA();
21.        } else if (type.trim().compareTo("W4M") == 0) {
22.           return new MiningOperator_W4M();
23.        } else {
24.           return null;
25.        }
26.     } catch (Exception e) {
27.         throw new Exception(ErrorStatic.ErrorMiningEngineClass);
28.     }
29.   }
```

Μετέπειτα, ο *Mining Engine* ρυθμίζει τις παραμέτρους του *T-Pattern* και επικοινωνεί με τον *MiningOperator_TAS* για να εκτελεστεί ο αλγόριθμος. Σημείωση ότι ο *MiningOperator_TAS* καλεί με εξωτερική κλήση τον αλγόριθμο T-Pattern αφού πρώτα έχει περάσει τα κινούμενα δεδομένα σε ένα αρχείο μέσω της κλάσης *ObjectStore*, το οποίο είναι το input του αλγόριθμου. Η εξωτερική κλήση του T-Pattern γίνεται με την εξής μέθοδο:

```
1. public void exec() {
2.    try {
3.      Process p = null;
4.      if(System.getProperty("os.name").startsWith("Windows")) {
5.         String[] cmd = new String[3];
6.         cmd[0] = "cmd.exe";
7.         cmd[1] = "/C";
8.         cmd[2] = commandline;
9.         System.out.println("Data Mining Algorithm is running...");
10.        p = Runtime.getRuntime().exec(cmd);
11.    }
12.    if (System.getProperty("os.name").startsWith("Mac")) {
13.        p = Runtime.getRuntime().exec(commandmac);
14.    }
15.    StreamGobbler errorGobbler = new
16.                        StreamGobbler(p.getErrorStream(), "ERROR");
17.    StreamGobbler outputGobbler = new
18.                        StreamGobbler(p.getInputStream(), "OUTPUT");
19.    errorGobbler.start();
20.    outputGobbler.start();
21.    p.waitFor();
22.    System.out.println("Data Mining Algorithm executed...");
23.   } catch (Exception err) {
24.     err.printStackTrace();
25.   }
26. }
```

Ο αλγόριθμος δημιουργεί το αρχείο ‘MiSTA_table.txt’ και αφού διαβάσει τα εσωτερικά δεδομένα και

τα περάσει σε ένα αρχείο .obj μεσω της *ObjectStore*, αναλαμβάνει ο *Controller* όπου αποθηκεύει το αποτέλεσμα του αρχείου .obj στην βάση δεδομένων κίνησης. Αξίζει να σημειωθεί ότι για την ανάγνωση των δεδομένων που χρησιμοποιούνται ως είσοδο στον αλγόριθμο αλλά και για την αποθήκευση των αποτελεσμάτων πίσω στην βάση, αναλαμβάνει το πακέτο *Translation Library*. Το πακέτο *Translation Library* περιέχει τις κατάλληλες κλάσεις που μετατρέπουν τα αντικείμενα της *Java* σε *Oracle* και αντίστροφα. Συγκεκριμένα, η μετατροπή από *Oracle* σε *Java* γίνεται από την μέθοδο *public TranslationLibrary* ***buildUp****(OracleResultSet rs, int column throws Exception;* ενώ η μετατροπή από Java σε Oracle διαμορφώνεται με δύο τρόπους, (α) με την μέθοδο *public String* ***getSQLValue****();* ή (β) με την μέθοδο *public boolean* ***Materialize****(Statement st, String name);*. Τέλος, η κλάση *Render* αναλαμβάνει να οπτικοποιήσει τα αποτελέσματα στον παγκόσμιο χάρτη, καλώντας την μέθοδο *public void* ***visualizeObjectOnEarth****(Object obj);* μέσω της διασύνδεσης *Renderer*.

- ***Διαδικασία εισαγωγής ενός νέου αλγόριθμου στο σύστημα***

Η αρχιτεκτονική του συστήματος έχει αναπτυχθεί με τέτοιο τρόπο ώστε να γίνεται εύκολη η διαδικασία εισαγωγής ενός νέου αλγόριθμου στην πλατφόρμα. Σε αυτήν την παράγραφο, θα περιγραφούν τα τεχνικά βήματα που χρειάζεται να εφαρμόσει ένας μηχανικός λογισμικού για να εντάξει ένα νέο αλγόριθμο στην παρούσα εφαρμογή. Γενικά, τα συνολικά βήματα που πρέπει να ακολουθήσει ένας προγραμματιστής είναι τέσσερα όπου υπό προϋποθέσεις γίνονται δυο. Συγκεκριμένα, τα βήματα είναι τα ακόλουθα:

1. *Mining Engine*

   Ένας προγραμματιστής πάει στο πακέτο 'miningEngine' και δημιουργεί μια νέα κλάση με το όνομα 'MiningOperator_**όνομα_αλγόριθμου**.java'. Μέσα στην κλάση, κάνει *implements* το *interface* 'GenericMiningOperator', ώστε να κληρονομήσει τις λειτουργίες της. Οι βασικές λειτουργίες του 'GenericMiningOperator' φαίνονται στον παρακάτω πίνακα και ο προγραμματιστής αυτό που πρέπει να κάνει είναι να τις υλοποιήσει ανάλογα με τις ιδιότητες του αλγόριθμου. Συγκεκριμένα, η μέθοδος ***setLinks*** δέχεται δύο παραμέτρους, (α) την *ObjectStore* και (β) το *ODBC*. Η *ObjectStore* χρησιμοποιείται για την δημιουργία των αρχείων .obj και το *ODBC* για την επικοινωνία με την βάση δεδομένων. Η μέθοδος ***setParams*** δέχεται μία παράμετρος, η οποία είναι ένας δυναμικός πίνακας και περιέχει τις παραμέτρους του αλγόριθμου για τον υπολογισμό. Εάν ο αλγόριθμος λαμβάνει ένα ερώτημα ως παράμετρο τότε η παράμετρος αυτή θα εκπροσωπηθεί από ένα STRING που ονομάζεται *data key* και χρησιμοποιείται για να ανακτήσει το αποτέλεσμα από μια εσωτερική μνήμη *cache* του συστήματος που ονομάζεται *ObjectStore*. Η μέθοδος ***execute*** αναλαμβάνει το ρόλο της εκτέλεσης του αλγόριθμου. Όπως είχε αναφερθεί σε προηγούμενα κεφάλαια, ο προγραμματιστής μπορεί να καλέσει τον αλγόριθμο ανάλογα με το αν είναι ενσωματωμένος μέσα στην βάση ή στο πρόγραμμα ή να είναι ένα εξωτερικό εκτελέσιμο αρχείο. Οπότε, σε αυτό το σημείο, θα υλοποιήσει την μέθοδο με έναν από τους τρεις τρόπους, υλοποίηση κώδικα για (α) εξωτερική κλήση, (β) για κλήση στην βάση (*SQL procedure/function*) ή (γ) για άμεση κλήση μέσα στο πρόγραμμα *Java*. Η μέθοδος ***storeResult*** δημιουργεί ένα αρχείο .obj και αποθηκεύει τα αποτελέσματα του αλγόριθμου στο αρχείο αυτό χρησιμοποιώντας ένα datakey ως αναγνωριστικό. Η μέθοδος ***getTranslation*** είναι απαραίτητη για τον *Controller* να γνωρίζει ποια παράμετρος (το *input*) πρέπει να μεταφράσει και ποια βιβλιοθήκη 'Translation Library' πρέπει να χρησιμοποιηθεί. Θεωρείστε ότι όλα τα αντικείμενα μέσα στο *Object Store* μετασχηματίζονται σε αντικείμενα *Java* χρησιμοποιώντας τις πληροφορίες που επιστρέφονται από αυτή την μέθοδο.

```
1.  package stdmql.miningEngine;
2.  import java.util.ArrayList;
3.  import stdmql.objectStore.ObjectStore;
4.  import stdmql.translation_libraries.Pair;
5.  import stdmql.translation_libraries.TranslationLibrary;
6.  import hermes.db_manager.ODBC;
7.
8.  public interface GenericMiningOperator {
9.  public void setLinks(ObjectStore objectStore, ODBC odbc);
10. public void setParams(ArrayList<Object> params) throws Exception;
```

```
11.     public void execute() throws Exception;
12.     public void storeResult(String dataKey) throws Exception;
13.     public TranslationLibrary getTranslation(int n);
14.     public TranslationLibrary getOutput();
15. }
```

Παρόμοια με την προηγούμενη μέθοδος, η ***getOutput*** επιστρέφει το είδος του αποτελέσματος του αλγορίθμου. Για την υλοποίηση αυτής της κλάσης, έχουμε πρόσβαση σε όλες τις βιβλιοθήκες του συστήματος 'Translation Library' (για να διαχειριστούμε τα αντικείμενα που λαμβάνονται από το Object Store), που αντιπροσωπεύουν το είδος των δεδομένων που μπορεί να διαχειριστεί. Για να καταλάβουμε πως λειτουργεί η κλάση, πρέπει να γνωρίζουμε την σειρά που θα καλέσουμε τις μεθόδους της όπως φαίνεται στον παρακάτω πίνακα:

```
[…]
GenericMiningOperator mOp = factory.createMiningOperator
(Params.get(1));
mOp.setLinks(objectStore, dbmanager);
mOp.setParams(Params);
mOp.execute();
mOp.storeResult(Params.get(1).toString() + "_result");
[…]
```

2. *Translation Library*

Η αρχιτεκτονική του συστήματος έχει κατασκευαστεί για να διαχειριστεί διαφορετικά είδη δεδομένων με πολύ γενικό τρόπο. Για να γίνει αυτό, τα περισσότερα τμήματα των συστατικών 'Components' αγνοούν το είδος των δεδομένων που διαχειρίζονται. Τα μόνα 2 εξαρτήματα που πρέπει να γνωρίζουν τον ορισμό των δεδομένων είναι τα εξής:

- Ο *ObjectStore*: το οποία λαμβάνει το αντικείμενο της *Oracle* από τον *Controller* και πρέπει να το αποθηκεύσει μέσα σε ένα αρχείο .obj.
- Η λειτουργία *Mining*: η οποία πρέπει να χρησιμοποιήσει τα δεδομένα για τον αλγόριθμο.

Για την υλοποίηση των εν λόγω βιβλιοθηκών, χτίζουμε κάποιο *pluggable* συστατικό που ονομάζεται *TranslationLibrary*. Όλες οι βιβλιοθήκες αυτές έχουν ένα αντίστοιχο τύπο δεδομένων στην βάση και αυτό που κάνουν είναι να μετατρέπουν τα *ORACLE* αντικείμενα σε *Java* αντικείμενα και αντίστροφα. Η διασύνδεση αυτών των κλάσεων παρουσιάζεται στον παρακάτω πίνακα. Ουσιαστικά, αυτή η διασύνδεση είναι ο κατασκευαστής (*factory*) των νέων τύπων των δεδομένων και ως εκ τούτου μια *TranslationLibrary* είναι επίσης το αντικείμενο *Java* που αντιπροσωπεύει τα δεδομένα.

```
1.    package stdmql.translation_libraries;
2.
3.    import java.io.Serializable;
4.    import java.sql.Statement;
5.    import oracle.jdbc.OracleResultSet;
6.
7.    public interface TranslationLibrary extends Serializable {
8.
9.    public TranslationLibrary buildUp(Object obj) throws Exception;
10.   public TranslationLibrary buildUp(OracleResultSet rs, int
11.   columnId)    throws Exception;
12.   public String getBindDbType();
13.   public String getSQLValue();
```

```
14.   public String toString();
15.   public boolean Materialize(Statement st, String name);
16.   public boolean InitializeTable(Statement st, String name);
17. }
```

Η μέθοδος ***buildUp****(OracleResultSet rs, int columnId)* παίρνει το αντικείμενο της *ORACLE* που αντιπροσωπεύει τη γραμμή που περιέχει τα δεδομένα στη συγκεκριμένη στήλη και το αποτέλεσμα είναι μια νέα *Translation Library* που είναι η αναπαράσταση του αντικειμένου της *Java*. Η μέθοδος ***buildUp****(Object obj)* παίρνει το αντικείμενο της *Java* και επιστρέφει μια νέα *Translation Library* το οποίο περιέχει τα δεδομένα. Η μέθοδος ***getBindDbType*** επιστρέφει σε μορφή *STRING*, την θέση του ορισμένου τύπου μέσα στην *ORACLE*. Η μέθοδος ***getSQLValue*** επιστρέφει την *SQL* αναπαράσταση του αντικειμένου, το οποίο χρησιμοποιείται για την κατασκευή του ερωτήματος για να εισάγουμε τα δεδομένα μέσα στην ORACLE. Η ***toString*** επιστρέφει το όνομα του αντικειμένου. Για το πέρασμα των δεδομένων στην βάση του συστήματος υπάρχουν δύο τρόποι για να υλοποιηθούν:

- Απλός τρόπος: το σύστημα υποθέτει ότι το αποτέλεσμα είναι ένας πίνακας με δύο στήλες, (α) *ID*: η ταυτότητα του αντικειμένου και (β) *OBJECT*: το είδος του αντικειμένου της *ORACLE*. Για να αναγνωρίσουμε τον είδος του αντικειμένου, καλούμε την μέθοδο ***getBindDbType***. Στην περίπτωση αυτή, το σύστημα για κάθε αντικείμενο (μπορεί να είναι μόνο ένα σε ορισμένες περιπτώσεις) εκτελεί μια *insert* με ένα αύξοντα αριθμό για το *id* και καλεί την ***getSQLValue*** για την αναπαράσταση *SQL* του τρέχοντος αντικειμένου.
- Σύνθετος τρόπος: Μερικές φορές ο συνηθισμένος τρόπος δεν είναι αρκετός για να υλοποιήσει το αντικείμενο *ORACLE*, ή το αποτέλεσμα δεν είναι μια απλή εισαγωγή. Σε αυτή την περίπτωση, έχουμε τη δυνατότητα να παρακάμψουμε την κανονική διαδικασία χρησιμοποιώντας τις μεθόδους: ***initializeTable*** και ***Materialize***. Αυτοί οι δύο μέθοδοι που καλούνται από το σύστημα πριν εκτελεστεί η τυπική διαδικασία και στην περίπτωση που επιστρέψουν *false* σημαίνει ότι δεν χρησιμοποιούνται (έτσι το σύστημα εκτελεί τον κλασικό τρόπο), διαφορετικά αλλάζουμε τον τρόπο που το σύστημα πρέπει να δημιουργήσει τον πίνακα και τον τρόπο που πρέπει μετασχηματίζουμε τα δεδομένα στη βάση δεδομένων χρησιμοποιώντας άμεσα ένα statement και το όνομα του πίνακα. Προφανώς η μέθοδος ***initializeTable*** καλείται μόνο μια φορά στην έναρξη της διαδικασίας, ενώ η μέθοδος ***Materialize*** καλείται μία φορά για κάθε αντικείμενο που πρέπει να αποθηκευτεί στην βάση δεδομένων.

Αξίζει να σημειωθεί ότι σε περιπτώσεις που ένα είδος αντικειμένου από *ORACLE* σε *JAVA* και το αντίστροφο είναι υλοποιημένο στην βιβλιοθήκη *TranslationLibrary* (π.χ. Moving_point), ο προγραμματιστής παραβλέπει την διαδικασία αυτή και προχωράει στην επόμενη.

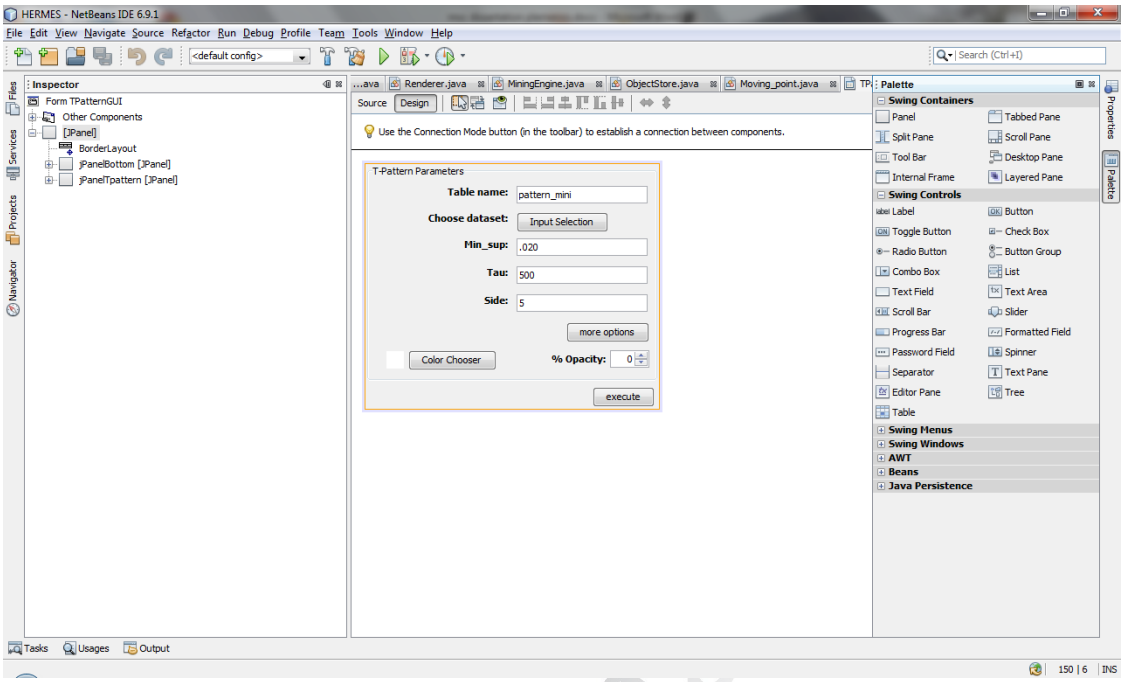
3. *Mining GUI*

Σε αυτήν την διαδικασία, ο προγραμματιστής/σχεδιαστής θα αναπτύξει την γραφική φόρμα του

αλγόριθμου που θα περιέχει τους αντίστοιχους παραμέτρους που ο χρήστης θα συμπληρώνει πριν εκτελέσει τον αλγόριθμο. Οι γραφικές φόρμες της πλατφόρμας έχουν υλοποιηθεί μέσα από το *Netbeans*. Το *Netbeans* είναι ένα ολοκληρωμένο πλαίσιο που παρέχει στον προγραμματιστή να υλοποιεί εφαρμογές σε desktop ή web αλλά το σημαντικότερο είναι ότι δίνει την δυνατότητα στον σχεδιαστή να σχεδιάζει γραφικές φόρμες μέσα από διαδραστικά παράθυρα και με την εισαγωγή λίγου κώδικα. Στην εικόνα 6.2, παρατηρούμε το γραφικό πλαίσιο του αλγόριθμου *T-Pattern*. Στο κέντρο του *Netbeans*, τοποθετείται το γραφικό πλαίσιο που θα αναπτυχθεί όπου ο χρήστης μπορεί να παρατηρήσει είτε το γραφικό πλαίσιο που σχεδιάζει με το *tab design* είτε τον κώδικα που δημιουργείται με το *tab source*. Στην δεξιά πλευρά, ο χρήστης διαλέγει ένα γραφικό συστατικό (π.χ. *JPanel*, *JLabel*, *JButton*, *JTextField*, κτλ) και το περνάει πάνω στο γραφικό πλαίσιο. Στην αριστερή πλευρά, παρουσιάζεται η δεντρική μορφή του γραφικού παραθύρου που αναπτύσσεται. Αφού ο προγραμματιστής υλοποιήσει την φόρμα του αλγόριθμου, θα πρέπει να αρχικοποιήσει και να εγκαταστήσει τους μηχανισμούς *Controller*, *Mining Engine*, *Object Store* και *Parser* όπως φαίνεται στο παρακάτω πίνακα.

```
stdmql.parser.Factory factoryParser =
                                stdmql.parser.Factory.getInstance();
stdmql.miningEngine.Factory factoryEngine =
                            stdmql.miningEngine.Factory.getInstance();
stdmql.controller.Factory factoryController =
                              stdmql.controller.Factory.getInstance();
stdmql.objectStore.Factory factoryObjStore =
```

```
                           stdmql.objectStore.Factory.getInstance();
controller = factoryController.createController();
Parser parser = factoryParser.createParser();
MiningEngine engine = factoryEngine.createMiningEngine();
try {
  ObjectStore objStore = factoryObjStore.createObjectStore();
  parser.setController(controller);
  engine.setLinks(objStore, odbc);
  controller.setODBC(odbc);
  controller.setMiningEngine(engine);
  controller.setParser(parser);
  controller.setObjectStore(objStore);
} catch (IOException ioe) {
   System.out.println(ioe.getMessage());
} catch (Exception e) {
   System.err.println("Error creating architecture: " +
   e.getMessage());
   e.printStackTrace();
}
```



Εικόνα 6.2: Τα κύρια μέρη του Netbeans.

4. *Renderer*

Ο Renderer αναλαμβάνει να οπτικοποιήσει τα αποτελέσματα του αλγόριθμου πάνω στον τρισδιάστατο χάρτη. Για να υλοποιήσει ο προγραμματιστής μια νέα οπτική αναπαράσταση που θα τοποθετείται στην υδρόγειο, θα πρέπει να δημιουργήσει μια νέα κλάση μέσα στο πακέτο Render με το όνομα **ΌνομαΑντικειμένου**Renderer.java. Μέσα στην κλάση, κάνει *extends* την αφηρημένη κλάση *ControlListener* και *implements* το *interface Renderer* ώστε να κληρονομήσει τις λειτουργίες τους. Όσο αφορά την διασύνδεση *Renderer*, η μέθοδος **visualizeObjectOnEarth** δέχεται ως παράμετρο το αντικείμενο που θα οπτικοποιήσει και ο προγραμματιστής με την βοήθεια της βιβλιοθήκης *World Wind Nasa* ή από το πακέτο *Shape*, ενεργοποιεί τα κατάλληλα σχήματα και ρυθμίζει την αντίστοιχη θέση τους στον χάρτη. Σημείωση ότι η βιβλιοθήκη *World Wind Nasa* περιλαμβάνει πολλά και διαφορετικά σχήματα αλλά σε περιπτώσεις που θέλουμε να υλοποιήσουμε ένα πολύπλοκο σχήμα όπως είναι το κινούμενο δεδομένο, θα πρέπει να φτιάξουμε το δικό μας σχήμα και να το τοποθετήσουμε μέσα στο πακέτο *Shape*. Η μέθοδος ***clearObjects*** αναλαμβάνει να καθαρίζει (διαγράφει) τα οπτικά αντικείμενα που έχουν δημιουργηθεί. Η μέθοδος ***addLayer*** ή ***removeLayer*** προσθέτει ή διαγράφει ένα στρώμα (κάθε στρώμα περιέχει ένα ή περισσότερα οπτικά αντικείμενα) στο διαχειριστή στρώματος (Layer Manager) και στον χάρτη, αντίστοιχα. Η μέθοδος ***setupDefaultMaterial*** ρυθμίζει το προεπιλεγμένο υλικό (π.χ. χρώμα, διαφάνεια και πλάτος αντικειμένου). Ενώ οι μέθοδοι ***getOpacity***, ***getColor***, ***getWidth***, ***setOpacity***, ***setColor*** και ***setWidth*** επιστρέφουν ή προσδιορίζουν τις τιμές της διαφάνειας, χρώματος και πλάτος του οπτικού αντικειμένου, αντίστοιχα.

```
1.  package hermes.render;
2.
3.  import java.awt.Color;
4.
5.  public interface Renderer {
6.      public void visualizeObjectOnEarth(Object obj);
7.      public void clearObjects();
8.      public void addLayer(String layer_name);
9.      public void removeLayer(String layer_name);
10.     public void setupDefaultMaterial(Object o, Color color);
11.     public void getOpacity();
12.     public void setOpacity(double opacity);
13.     public void getColor();
14.     public void setColor(Color color);
15.     public void getWidth();
16.     public void setWidth(double width);
17. }
```

```
1. package hermes.user_interface.render;
2.
3. import gov.nasa.worldwind.awt.WorldWindowGLCanvas;
4. import gov.nasa.worldwind.event.SelectEvent;
5. import gov.nasa.worldwind.event.SelectListener;
6. import gov.nasa.worldwind.pick.PickedObjectList;
7. import gov.nasa.worldwind.render.Annotation;
8.
9. public abstract class ControlListener {
10.   public WorldWindowGLCanvas wwd;
11.   public void initializeSelectionMonitoring(WorldWindowGLCanvas
12.                                             wwd) {...}
13.   public void highlight(Object o) {}
14.   public void showToolTip(Object o, SelectEvent e) {}
15.   public Annotation createToolTip(Object o, SelectEvent e) {
16.       return null;
17.   }
18.}
```

Από την άλλη μεριά, η αφηρημένη κλάση ***ControlListener*** αναλαμβάνει τον ρόλο της αλληλεπίδρασης του χρήστη με το αντικείμενο πάνω στον χάρτη. Δηλαδή, πως το οπτικό αντικείμενο θα συμπεριφερθεί όταν ο χρήστης τοποθετήσει το ποντίκι πάνω σε αυτό. Συγκεκριμένα, η μέθοδος ***initializeSelectionMonitoring*** είναι υλοποιημένη από το σύστημα και αρχικοποιεί τους χειριστές ελέγχου (π.χ. click, hover, drag, κτλ). Η μέθοδος ***highlight*** αλλάζει το χρώμα του αντικειμένου ανάλογα αν το ποντίκι είναι μέσα στο αντικείμενο ή όχι. Η μέθοδος ***showToolTip*** εμφανίζει στον χάρτη μια σημείωση που αναφέρει μερικά γενικά χαρακτηριστικά του αντικειμένου ενώ η μέθοδος ***createToolTip*** δημιουργεί γραφικά αυτήν την σημείωση πάνω στην υδρόγειο. Τέλος, όταν το αντικείμενο που θα οπτικοποιηθεί πάνω στην γη είναι ήδη υλοποιημένο (π.χ. *TrajectoryRenderer*) τότε ο προγραμματιστής παρακάμπτει αυτήν την διαδικασία και χρησιμοποιεί από το πακέτο Render την αντίστοιχη κλάση. Η κλήση της *Renderer* γίνεται από την κλάση που βρίσκεται μέσα στο πακέτο *MiningGUI* (βλέπε 3ο βήμα) και η σειρά κλήσης των μεθόδων του *Renderer* διατυπώνονται στον παρακάτω πίνακα.

```
[…]
TrajectoryRenderer tr = new TrajectoryRenderer(frame, odbc);
tr.clearObjects();
tr.removeLayer(“Trajectories of Milano”);
// moving_point is a Translation Library
tr.visualizeObjectOnEarth(moving_point);
tr.addLayer(“Trajectories of Milano”);
[…]
```

7. ΣΥΜΠΕΡΑΣΜΑΤΑ

Με την ολοκλήρωση της συγκεκριμένης μεταπτυχιακής εργασίας, έχει προκύψει ένα ολοκληρωμένο οπτικό αναλυτικό εργαλείο για την ανάλυση σε κινούμενα δεδομένα. Ο κυριότερος σκοπός της πλατφόρμας είναι η ανάλυση και διαχείριση των κινούμενων δεδομένων μέσα από διαδραστικά εργαλεία και η παρουσίαση τους σε έναν αλληλεπιδραστικό τρισδιάστατο χάρτη. Με λίγα λόγια, η πλατφόρμα της μεταπτυχιακής εργασίας διαθέτει μια πειραματική λογική, η οποία περιέχει τους πιο δημοφιλείς αλγόριθμους εξόρυξης γνώσης και αλγόριθμους ανωνυμίας καθώς επίσης, χρησιμοποιεί τους μηχανισμούς ερωτημάτων αναζήτησης του *HERMES* και *HERMES++*. Ο σημαντικότερος ρόλος της πειραματικής λογικής είναι να χρησιμοποιεί τα αποτελέσματά των ερωτημάτων αναζήτησης και εξόρυξης σε ένα κοινό περιβάλλον (τρισδιάστατη υδρόγειος) έτσι ώστε να διευκολύνει έναν αναλυτή στην διαχείριση και εξόρυξη γνώσης αλλά και στην σύγκριση της απόδοσης των αλγορίθμων μέσω στατιστικών αποτελεσμάτων. Η βάση για την πραγματοποίηση της διατριβής αποτέλεσε ο *HERMES*, οποίος είναι μια επέκταση της *ORACLE SPATIAL 10g* και εστιάζει στην παραγωγή, αποθήκευση, ανάκτηση και αναζήτηση χώρο χρονικών δεδομένων. Πάνω στον *HERMES*, ενσωματώθηκε ο *HERMES++*, ο οποίος έχει την δυνατότητα να υποστηρίζει τα ερωτήματα του *HERMES* αλλά επιπλέον να διατηρεί την προστασία των δεδομένων κίνησης και να μην αναδεικνύει τυχόν ευαίσθητη πληροφορία στο τελικό χρήστη.

Η κατεύθυνση που ακολούθησε η παρούσα μεταπτυχιακή διατριβή είναι διττή. Όσο αφορά την πρώτη κατεύθυνση, η εφαρμογή υποστηρίζει την προοδευτική αναζήτηση και εξόρυξη γνώσης μέσα από τεχνικές αλληλεπίδρασης χρήστη-υπολογιστή. Η διαδικασία της προοδευτικής αναζήτησης και εξόρυξης αναλύει τα δεδομένα κίνησης μέσα από μια σειριακή ακολουθία βημάτων χρησιμοποιώντας διαφορετικά είδη αποτελεσμάτων σε κάθε βήμα. Συγκεκριμένα, ένας χρήστης μπορεί να εκτελέσει ένα ερώτημα που ανήκει στο μηχανισμό αναζήτησης και ως είσοδο να περιέχει το αποτέλεσμα ενός προηγούμενου ερωτήματος του μηχανισμού αναζήτησης. Με τον ίδιο τρόπο, ένας χρήστης μπορεί να πραγματοποιήσει την εκτέλεση ενός αλγόριθμου από τον μηχανισμό εξόρυξης με είσοδο το αποτέλεσμα ενός προηγούμενου αλγόριθμου εξόρυξης. Επίσης, ο χρήστης έχει την δυνατότητα να εφαρμόσει ένα ερώτημα από τον κόσμο εξόρυξης με είσοδο το αποτέλεσμα ενός ερωτήματος αναζήτησης. Αυτού του είδους η διαδικασία είναι όταν ο χρήστης επιθυμεί να εκτελέσει έναν αλγόριθμο εξόρυξης γνώσης από την προ-επεξεργασία των δεδομένων που έχει γίνει σε προηγούμενο στάδιο από τον μηχανισμό αναζήτησης έτσι ώστε ο αλγόριθμος να αναδείξει τα επιθυμητά αποτελέσματα σε υψηλότερη απόδοση. Αντιθέτως, ένας χρήστης μπορεί να ακολουθήσει την αντίστροφη διαδικασία εκτελώντας ένα ερώτημα αναζήτησης από το αποτέλεσμα ενός αλγόριθμου εξόρυξης γνώσης ως είσοδο. Αυτή η διαδικασία είναι χρήσιμη για μετά-επεξεργαστικούς λόγους όπως είναι η πράξη του φιλτραρίσματος ενδιαφερόντων προτύπων. Σχετικά με την δεύτερη κατεύθυνση, η πλατφόρμα μέσα από μια πειραματική εφαρμογή αναλύει τα δεδομένα κίνησης χωρίς να παραβιάζει την προσωπική πληροφορία των ατόμων που έχει αποθηκευτεί στην βάση τροχιών. Εν συντομία, δίνει την δυνατότητα στους χρήστες (α) να θέτουν απλά ερωτήματα και ερωτήματα προστασίας δεδομένων του *HERMES* και του *HERMES++* αντίστοιχα, (β) να εφαρμόζουν τους αλγόριθμους ανωνυμοποίησης όπως είναι ο *NWA* και ο *W4M* στα δεδομένα ενώ να έχουν την ικανότητα να συγκρίνουν και να αξιολογούν τα αποτελέσματα μεταξύ των αρχικών και ανώνυμων δεδομένων μέσα από μια σειρά τεχνικών εξόρυξης γνώσης και (γ) να σχεδιάζουν και να εκτελούν πειράματα έτσι ώστε να αξιολογήσουν την απόδοση των αλγόριθμων ανωνυμοποίησης μέσω στατιστικών γραφικών παραστάσεων χρησιμοποιώντας διαφορετικούς φόρτους εργασίας πάνω στα ερωτήματα. Τέλος, η πλατφόρμα υποστηρίζει ερωτήματα ελεγκτικών τεχνικών που μπορούν να χρησιμοποιηθούν για την κατασκευή του προφίλ του χρήστη με βάση τα ερωτήματα που έχει θέσει στη βάση δεδομένων με σκοπό να εντοπιστούν οι τυχών ύποπτες συμπεριφορές του χρήστη. Ουσιαστικά, αυτή είναι η πρώτη προσπάθεια που παρουσιάζει μια ολοκληρωμένη πλατφόρμα που φιλοξενεί τους '*state-of-the-art*' αλγόριθμους εξόρυξης γνώσης και ανωνυμοποίησης δεδομένων κίνησης, τα οποία συνεργάζονται με τους μηχανισμούς των απλών ερωτημάτων και των ερωτημάτων προστασίας προσωπικών δεδομένων. Τα αλληλεπιδραστικά εργαλεία που εμπεριέχονται στην εφαρμογή έχουν παίξει σημαντικό ρόλο στην διευκόλυνση του χρήστη να αναλύσει και να διαχειριστεί ένα μεγάλο όγκο πληροφοριών. Πάνω σε αυτό, ένα από τα μεγαλύτερα πλεονεκτήματα της εφαρμογής είναι ότι δεν απευθύνεται μόνο σε έμπειρους χρήστες γράφοντας μια γλώσσα *SQL* για την εκτέλεση των ερωτημάτων αλλά και σε απλούς, οι οποίοι έχουν την δυνατότητα να πραγματοποιήσουν οποιοδήποτε ερώτημα μέσω των διαδραστικών διεπαφών.

7.1. ΑΝΟΙΚΤΑ ΘΕΜΑΤΑ

Σε γενικές γραμμές, η πλατφόρμα είναι ένα ολοκληρωμένο σύστημα που παρέχει στον χρήστης την ικανότητα να χρησιμοποιήσει τους πιο δημοφιλής αλγόριθμους εξόρυξης γνώσης και αλγόριθμους ανωνυμίας καθώς επίσης και τα ερωτήματα αναζήτησης του *HERMES* και *HERMES++* μέσω μιας πειραματικής εφαρμογής. Λόγω του ότι πιάνει μια τόσο μεγάλη γκάμα περιοχών, υπάρχουν πολλές μελλοντικές κατευθύνσεις που μπορεί να ακολουθήσει. Πρώτον, στην έκτη ενότητα είχε επισημανθεί ο *T-Aggregator,* ένας μηχανισμός γενίκευσης και συνάθροισης των τροχιών με την μορφή βέλους όπου η μύτη του βέλους επιδεικνύει την κατεύθυνση των μαζικών μετακινήσεων και το πάχος του το πλήθος των διαδρομών. Η οπτική παρουσίαση του αποτελέσματος είναι σε δυσδιάστατη μορφή με αποτέλεσμα να απεικονίζει απλές διαδρομές χωρίς επιστροφές και διακλαδώσεις και να υπολείπεται στην διαύγεια πιο περίπλοκων περιπτώσεων. Μια πιθανή λύση θα μπορούσε να είναι η αναπαράσταση των βελών σε τρισδιάστατη μορφή όπου η τρίτη διάσταση (κάθετος άξονας) θα αντιστοιχεί στους σχετικούς χρόνους των αντίστοιχων μετακινήσεων. Με αυτόν τον τρόπο μειώνονται οι διακλαδώσεις και οι επικαλύψεις των βελών μεταξύ τους. Παραπλεύρως αυτής της κατεύθυνσης, υπάρχουν ανοικτά ζητήματα στην ανάπτυξη αλγορίθμων για την κατασκευή πολύπλοκων γενικευμένων και συγκεντρωτικών μοντέλων. Παραδείγματος χάριν, ο *T-Pattern* παράγει δυσδιάστατα μοντέλα, τα οποία παρουσιάζουν μαζικές μετακινήσεις από την μια περιοχή σε μία άλλη εμπεριέχοντας το χρονικό διάστημα των μετακινήσεων αυτών. Παρ' όλα αυτά, αναπτύσσοντας ένα τρισδιάστατο μοντέλο όπου το ύψος θα έδινε το μέγεθος των τροχιών που υπολογίστηκαν στο συγκεκριμένο πρότυπο και το αντίστοιχο χρώμα, την διάρκεια της μετακίνησης, θα έδινε στον χρήστη περισσότερη πληροφορία άμεσα. Δηλαδή, όσο πιο ψηλά βρίσκεται ένα πρότυπο του *T-Pattern,* τόσο πιο μεγάλο πλήθος τροχιών περιέχει και όσο το χρώμα του είναι πιο σκούρο (ή κόκκινο ανάλογα με τις διαβαθμίσεις που θα χρησιμοποιηθούν), τόσο πιο μεγάλη διάρκεια είχε η μετακίνηση. Επίσης, ο χρήστης έχει την δυνατότητα να χρησιμοποιήσει τα αποτελέσματα του αλγόριθμου και να πραγματοποιήσει ένα ερώτημα αναζήτησης ή εξόρυξης. Τρίτον, η εφαρμογή έχει αναπτυχθεί μέσα σε ένα πειραματικό πλαίσιο δηλαδή ένας αναλυτής μπορεί να εκτελέσει ένα απλό ερώτημα ή ένα αλγόριθμο, να παρατηρήσει τα αποτελέσματα στον χάρτη και στην συνέχεια, να εκτελέσει ένα διαφορετικό ερώτημα και να συγκρίνει τα αποτελέσματά τους. Επίσης, ο χρήστης μπορεί να εκτελέσει ερωτήσεις πειραμάτων και να παρατηρήσει την επίδοση των ερωτημάτων αυτών μέσω στατιστικών γραφικών παραστάσεων. Μια μελλοντική κατεύθυνση είναι η ανάπτυξη νέων και καινοτόμων αλγορίθμων και η ενσωμάτωση τους στην πλατφόρμα. Αυτό δίνει την δυνατότητα στον χρήστη να παρατηρήσει την συμπεριφορά των αλγόριθμων στον χάρτη και να συγκρίνει τα αποτελέσματα με τα αντίστοιχα των αλγόριθμων που ήδη περιέχονται στην πλατφόρμα. Τέλος, η ανάπτυξη στρατηγικών και τεχνικών που θα βελτιστοποιούν την απόδοση των ερωτημάτων αναζήτησης και εξόρυξης είναι ακόμη ένα ζήτημα για τους επιστήμονες. Η αναπαράσταση των τροχιών ή των μοντέλων σε τρισδιάστατους χάρτες απαιτεί υψηλές απαιτήσεις γραφικών, μνήμης και επεξεργαστή του υπολογιστή για την καλύτερη λειτουργία της εφαρμογής. Επιπρόσθετα, οι αλγόριθμοι εξόρυξης γνώσης απαιτούν και αυτοί υψηλές απαιτήσεις λόγω των πολύπλοκων μαθηματικών πράξεων που εκτελούν. Οπότε η εισαγωγή νέων τεχνικών που θα ανακτούν, θα επεξεργάζονται, θα αποθηκεύουν και θα οπτικοποιούν τα αποτελέσματα σε ένα τρισδιάστατο περιβάλλον με υψηλή απόδοση είναι ακόμη και σήμερα μια μεγάλη πρόκληση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. A.Gkoulalas-Divanis and V.S. Verykios. "A Privacy Aware Trajectory Tracking Query Engine". *ACM SIGKDD Explorations*, 10(1): 40-49, July 2008.
2. D. Akoumianakis, G. Milolidakis, G. Kotsalis and D. Vellis. "Generic strategies for manipulating graphical interaction objects: Augmenting, expanding, and integrating components". In Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS 2008), Barcelona, Spain. June 2008.
3. D. G. Bell, F. Kuehnel, C. Maxwell, R. Kim, K. Kasraie, T. Gaskins, P. Hogan and J. Coughlan. "NASA World Wind: Opensource GIS for Mission Operations". In Proceedings of the 2007 IEEE Aerospace Conference, Big Sky, Montana, March 2007.
4. E. Frentzos, K. Gratsias, N. Pelekis and Y. Theodoridis. "Algorithms for Nearest Neighbor Search on Moving Object Trajectories". *Geoinformatica*, 11:159–193, January 2007.
5. F. Giannotti and R. Trasarti. "Mobility, Data Mining and Privacy: The GeoPKDD Paradigm". In the Proceedings of the 9th SIAM International Conference on Data Mining (SDM 2009), Sparks, Nevada, April 2009.
6. F. Giannotti, M. Nanni, D. Pedreschi and F. Pinelli. "Trajectory pattern mining". In the Proceedings of the 13th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2007), San Jose, California, August 2007.
7. F. Pinelli, A. Monreale, R. Trasarti and F. Giannotti. "Location Prediction within the Mobility Data Analysis Environment Daedalus". In the Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous 2008), Dublin, Ireland, July 2008.
8. G. Andrienko and N. Andrienko. "A general framework for using aggregation in visual exploration of movement data". *The cartographic journal*, 47(1):22-40, January 2010.
9. G. Andrienko and N. Andrienko. "Spatio-Temporal Aggregation for Visual Analysis of Movements". In the Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST 2008), pp. 51-58, Columbus Ohio, USA October 2008.
10. G. Andrienko, N. Andrienko and S. Wrobel. "Visual Analytics Tools for Analysis of Movement Data". *ACM SIGKDD Explorations*, 9(2): 38-46, December 2007.
11. G. Andrienko, N. Andrienko, P. Jankowski, D. Keim, M.-J. Kraak, A. MacEachren and S. Wrobel. "Geovisual analytics for spatial decision support: Setting the research agenda". Special issue of the *International Journal of Geographical Information Science*, 21(8):839–857, July 2007.
12. G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi and F. Giannotti. "Interactive Visual Clustering of Large Collections of Trajectories". In the Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST 2009). Atlantic City, New Jersey, USA, October 2009.
13. Jae G. Lee, J. Han and Kyu Y. Whang. "Trajectory clustering: a partition-and-group framework". In the Proceedings of the ACM International Conference on Management of Data (SIGMOD 2007), Beijing, China, June 2007.
14. KDD LAB. "Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases". Available at: http://www-kdd.isti.cnr.it/NWA/. 2008.
15. KDD LAB. "Wait 4 Me: Time-tolerant Anonymization of Moving Objects Databases". Available at: http://kdd.isti.cnr.it/W4M/. 2010.
16. L. Boschetti, D. Roy, P. Barbosa, R. Boca and C. Justice. "A MODIS assessment of the summer 2007 extent burned in Greece". *International Journal of Remote Sensing*, 29(8):2433-2436, April 2008.
17. L. Boschetti, D. Roy and C. Justice. "Using NASA's world wind virtual globe for interactive visualization of the global MODIS burned area product". *International Journal of Remote Sensing*, 29(11):3067−3072, June 2008.
18. M. Ester, H. P. Kriegel, J. Sander and X. Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise". In the Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD 1996), AAAI Press, pp. 226–231, 1996.
19. M. Nanni and D. Pedreschi. "Time-focused clustering of trajectories of moving objects". *Journal Intelligent Information Systems*, 27:267-289, November 2006.
20. M. Nanni, R. Trasarti, C. Renso, F. Giannotti and D. Pedreschi. "Advanced Knowledge Discovery on Movement Data with the GeoPKDD system". In the Proceedings of the 13th International Conference on Extending Database Technology (EDBT 2010), Lausanne, Switzerland, March 2010.
21. N. Pelekis, A. Gkoulalas-Divanis, M. Vodas, D. Kopanaki and Y. Theodoridis. "Privacy-Aware Querying over Sensitive Trajectory Data". In the Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM 2011), Glasgow, Scotland, UK, October 2011.
22. N. Pelekis, E. Frentzos, N. Giatrakos and Y. Theodoridis. "HERMES: Aggregative LBS via a Trajectory DB Engine". In the Proceedings of the ACM SIGMOD Conference, Vancouver, 2008.
23. N. Pelekis, E. Frentzos, N. Giatrakos, and Y. Theodoridis. "HERMES: A Trajectory DB Engine for Mobility-Centric Applications". *International Journal of Knowledge-based Organizations*, 2011, *in press*.
24. N. Pelekis, E. Frentzos, N. Giatrakos, and Y. Theodoridis. "On the Support of Mobility in ORDBMS". *International Journal of Knowledge-based Organizations*, 2011, *in press*.

25. N. Pelekis, I. Kopanakis, C. Panagiotakis and Y. Theodoridis. "Unsupervised Trajectory Sampling". In the Proceedings of the ECML PKDD 2010 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD 2010), LNAI 6323, pp. 17-33. Springer, Heidelberg, Barcelona, Spain, September 2010.
26. N. Pelekis, I. Kopanakis, E. Kotsifakos, E. Frentzos and Y. Theodoridis. "Clustering Uncertain Trajectories". *Knowledge and Information Systems* (KAIS), 28(1):117-147, July 2011.
27. N. Pelekis, I. Kopanakis, E. Kotsifakos, E. Frentzos and Y. Theodoridis. "Clustering Trajectories of Moving Objects in an Uncertain World". In the Proceedings of the IEEE International Conference on Data Mining (ICDM'09), Miami, U.S.A., December 2009. Best application paper award.
28. N. Pelekis, I. Kopanakis, I. Ntoutsi, G. Marketos, G. Andrienko and Y. Theodoridis. "Similarity Search in Trajectory Databases". In the Proceedings of the 14th IEEE International Symposium on Temporal Representation and Reasoning (TIME 2007), Alicante, Spain, 2007.
29. N. Pelekis. "STAU: A spatio-temporal extension to ORACLE DBMS", PhD Thesis, UMIST, 2002.
30. N.Pelekis and Y.Theodoridis."Boosting Location-Based Services with a Moving Object Database Engine". In the Proceedings of the 5th International ACM SIGMOD Workshop on Data Engineering for Wireless and Mobile Access (MobiDE 2006), Chicago, USA, June 2006.
31. N.Pelekis, Y.Theodoridis, S.Vosinakis, and T.Panayiotopoulos. "HERMES - A Framework for Location- Based Data Management". In the Proceedings of the 10th International Conference on Extending Database Technology (EDBT 2006), Munich, Germany, March 2006.
32. O. Abul, F. Bonchi and M. Nanni. "Anonymization of moving objects databases by clustering and perturbation". *Information Systems Journal*, 35(8): 884-910, December 2010.
33. O. Abul, F. Bonchi and M. Nanni. "Never Walk Alone: uncertainty for anonymity in moving objects databases". In Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE 2008), Cancun, Mexico, April 2008.
34. Oracle Corp. *Oracle®*. "Oracle Database Documentation Library", 10g Release 1 (10.1), http://otn.oracle.com/pls/db10g/.
35. P. Samarati and L. Sweeney. "Generalizing data to provide anonymity when disclosing information (abstract)". In Proceedings of the 17th ACM Symposium on Principles of Database Systems (PODS 1998), Seattle, Washington, June 1998.
36. R. Ortale, E. Ritacco, N. Pelekis, R. Trasarti, G. Costa, F. Giannotti, G. Manco, C. Renso and Y. Theodoridis. "The DAEDALUS framework: progressive querying and mining of movement data". In the Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS 2008), Irvine, California, November 2008.
37. R. Ortale, E. Ritacco, N. Pelekis, R. Trasarti, G. Costa, F. Giannotti, G.Manco, C. Renso and Y. Theodoridis. "Towards Progressively Querying and Mining Movement Data". ICAR-CNR technical Report CS-ICAR-02-2008. Available at http://150.145.63.4/biblio.
38. S. Rinzivillo, D. Pedreschi, M. Nanni, F. Giannotti, N. Andrienko and G. Andrienko. "Visually–driven analysis of movement data by progressive clustering". *Information Visualization*, 7(3/4):225-239, October 2008.
39. Wikipedia. "Data Mining". Available at: http://en.wikipedia.org/wiki/Data_mining. July 2011.
40. Wikipedia. "OPTICS algorithm". Available at: http://en.wikipedia.org/wiki/OPTICS_algorithm. July 2011.